

## Research Article

# Short-Term Load Forecasting Model Based on Quantum Elman Neural Networks

Zhisheng Zhang<sup>1</sup> and Wenjie Gong<sup>2</sup>

<sup>1</sup>College of Automation and Electrical Engineering, Qingdao University, Qingdao 266071, China

<sup>2</sup>Qingdao Electric Power Company, Qingdao 266002, China

Correspondence should be addressed to Zhisheng Zhang; slnzszs@126.com

Received 27 March 2016; Revised 18 June 2016; Accepted 11 July 2016

Academic Editor: Antonino Laudani

Copyright © 2016 Z. Zhang and W. Gong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Short-term load forecasting model based on quantum Elman neural networks was constructed in this paper. The quantum computation and Elman feedback mechanism were integrated into quantum Elman neural networks. Quantum computation can effectively improve the approximation capability and the information processing ability of the neural networks. Quantum Elman neural networks have not only the feedforward connection but also the feedback connection. The feedback connection between the hidden nodes and the context nodes belongs to the state feedback in the internal system, which has formed specific dynamic memory performance. Phase space reconstruction theory is the theoretical basis of constructing the forecasting model. The training samples are formed by means of  $K$ -nearest neighbor approach. Through the example simulation, the testing results show that the model based on quantum Elman neural networks is better than the model based on the quantum feedforward neural network, the model based on the conventional Elman neural network, and the model based on the conventional feedforward neural network. So the proposed model can effectively improve the prediction accuracy. The research in the paper makes a theoretical foundation for the practical engineering application of the short-term load forecasting model based on quantum Elman neural networks.

## 1. Introduction

Short-term load forecasting (STLF) is the basis for the normal and safe operation of power system. Since the introduction of competition mechanism in power market, the power company has put forward higher requirements on the accuracy and rapidity for load forecasting [1–3]. Many classical methods were applied to the study of STLF model, such as ARIMA model and ARMAX model [4, 5]. However, the ability of these classical methods to deal with the nonlinear time series is poor. With the introduction of artificial intelligence technology to the study of STLF model, it has opened up many new research methods such as neural network and expert system [6, 7]. Because of the strong nonlinear time series processing ability, neural network is widely used in the study of STLF.

The process of human brain information processing may be related to the quantum phenomenon, and quantum mechanical effects may exist in the brain, which had been

shown by some research results [8]. The quantum systems have the similar dynamic characteristics with the biological neural networks. The fusion of artificial neural network and quantum theory can better simulate the process of human brain information processing. Quantum neural network can improve the approximation ability and information processing efficiency of artificial neural network, which had been used for many applications [9–13].

The STLF model based on quantum Elman neural networks (QENN) is constructed in this paper. QENN is composed of the quantum neurons. The quantum neuron model consists of the weighting part, the aggregating part, the activating part, and the excitation part. Different from the traditional multilayer feedforward neural network, QENN has not only the feedforward connection but also the Elman feedback connection. The Elman feedback connection between the hidden nodes and the context nodes belongs to the state feedback in the internal system, which has formed specific dynamic memory performance [14].

Some studies had shown that the load time series is nonlinear and chaotic [15]. Phase space reconstruction theory (PSRT) is the important analysis method for research on chaotic time series. The training samples are formed by means of  $K$ -nearest neighbor approach (KNNA) based on PSRT.

Through the actual example simulation, it is proved that the proposed model can effectively improve the prediction accuracy and has adaptability to different load time series.

## 2. Quantum Elman Neural Networks

QENN is composed of quantum neurons, and quantum neurons are the basic elements of building QENN.

### 2.1. Quantum Neuron Model

**2.1.1. Qubit.** In the classical computation, the binary numbers “0” and “1” are used to represent the information, and they are usually called bits. In the quantum computation,  $|0\rangle$  and  $|1\rangle$  are used to represent the two basic states of microscopic particles, and they are usually called qubits. The difference between qubits and bits is that, in addition to the state of  $|0\rangle$  and  $|1\rangle$ , the state of qubits can be a linear combination of states, which is usually called the superposition.

In quantum computation, a qubit state  $|\phi\rangle$  can be expressed as

$$|\phi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (1)$$

where  $|0\rangle$  and  $|1\rangle$  express the basic states of microscopic particles.  $\alpha$  and  $\beta$  are the probability amplitudes, and they should meet the following relations:

$$|\alpha|^2 + |\beta|^2 = 1. \quad (2)$$

In order to further understand the superposition of the qubits, an electron is used to illustrate. An electron can be in the ground state  $|0\rangle$  of low energy level, and it can also be in the excitation state  $|1\rangle$  of high energy level. According to the superposition principle, the electron can be in the state of linear combination of these two states, that is, the superposition state  $\alpha|0\rangle + \beta|1\rangle$ . It is shown in Figure 1.

If the electron is in the superposition state  $\alpha|0\rangle + \beta|1\rangle$ , the electron can select the ground state  $|0\rangle$  according to the probability  $|\alpha|^2$  and select the excitation state  $|1\rangle$  according to the probability  $|\beta|^2$ . It is shown in Figure 2.

**2.1.2. Quantum Neuron Model.** The quantum neuron model adopted in the paper consists of the weighting part, the aggregating part, the activating part, and the excitation part. The weighting part is the simulation of the bond strength between synapses of nerve cells. The aggregating part is the simulation of the spatial and temporal integration of stimuli received by multiple dendrites. The activating part is the simulation of the interaction between the current activation values and membrane potential change of nerve cells. The excitation part is the simulation of the nonlinear characteristics of nerve cells, such as excitation, inhibition, fatigue, refractory period, and threshold. According to [16], the quantum behavior of neural cells is mainly reflected in the neurotransmitter

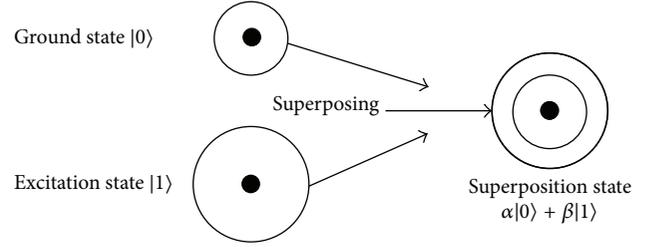


FIGURE 1: Schematic diagram of superposition state of ground State  $|0\rangle$  and excitation state  $|1\rangle$ .

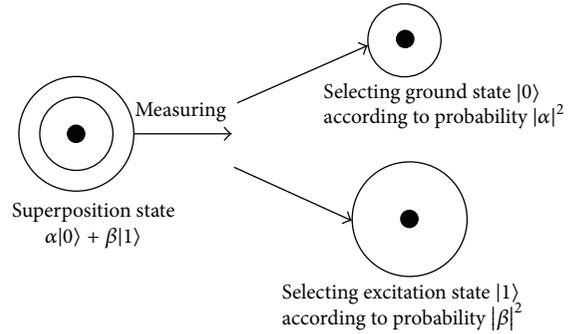


FIGURE 2: Schematic diagram of the ground state and excitation state after the superposition state is collapsing.

transmission of the cells. Transmitter release from nerve terminals in the form of small packages, with multiple molecules for biological neurons, and the small package is called quantum. A small package or quantum describes the smallest unit of transmitter secretion. In general, the quantum content of a synaptic response may be different, but the number of molecules in a quantum is relatively certain; that is, the quantum size is relatively determined. Since 1990s, some research results show that the process of human brain information processing may be related to the quantum phenomenon, and quantum mechanical effects may exist in the brain, and the quantum systems have the similar dynamic characteristics with the biological neural networks. So combining artificial neural network with quantum theory can better simulate the process of human brain information processing.

The quantum neuron model is shown in Figure 3. Considering the actual prediction problem, the input vector  $X = (x_1, x_2, \dots, x_n)^T$  is real-valued vector, and the output  $y$  is the real number.

The weight and activation values are qubits.  $|\phi\rangle = (|\phi_1\rangle, |\phi_2\rangle, \dots, |\phi_n\rangle)^T$  is the weight vector, and  $|\phi_i\rangle$  is the weight between the input  $x_i$  and the quantum neuron.  $|\phi\rangle$  can be described as

$$\begin{aligned} |\phi\rangle &= (|\phi_1\rangle, \dots, |\phi_n\rangle)^T \\ &= \left[ \begin{pmatrix} \cos \theta_1 \\ \sin \theta_1 \end{pmatrix}, \dots, \begin{pmatrix} \cos \theta_n \\ \sin \theta_n \end{pmatrix} \right]^T. \end{aligned} \quad (3)$$

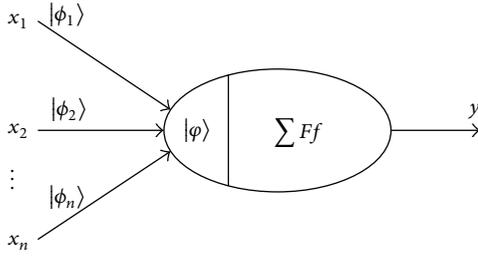


FIGURE 3: Quantum neuron model.

$|\varphi\rangle$  is the activation value, which can be described as

$$|\varphi\rangle = [\cos \xi, \sin \xi]^T. \quad (4)$$

The activation values are the important parameters for the activating part. The simulation of the interaction between the current activation values and membrane potential change of nerve cells is the core of the activating part. The activation values affect the accuracy of prediction. The parameters can be optimized by genetic algorithm.

In Figure 1,  $\sum$  represents the aggregation operator, and  $F$  represents the activation function.  $f$  is the excitation function, which adopts the *Sigmoid* function, and it can be described as

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (5)$$

The relationship between input and output can be expressed as

$$\begin{aligned} y &= f(F(X^T |\phi\rangle, |\varphi\rangle) - \tau) = f\left(\sum_{i=1}^n x_i \langle \phi_i | \varphi \rangle - \tau\right) \\ &= f\left(\sum_{i=1}^n x_i \cos(\theta_i - \xi) - \tau\right), \end{aligned} \quad (6)$$

where  $\tau$  is the threshold.

**2.2. Quantum Elman Neural Networks Model.** Considering Figure 4, QENN has four layers which are the input layer, the hidden layer, the context layer, and the output layer. The hidden layer and the output layer are constructed by quantum neurons.

For QENN, each node in the hidden layer has a corresponding node connected to it in the context layer.

The input nodes and the hidden nodes, the hidden nodes and the output nodes, and the context nodes and the hidden nodes are connected by adjustable weights.

Compared with the feedforward network structure, QENN has the context layer. The role of the nodes in the context layer is to store the internal state of neural network. They are used to store the state of the hidden nodes at the current time, and the state will be passed to the hidden nodes at the next time. The connection between the hidden nodes and the context nodes belongs to the state feedback in the internal system, which has formed specific dynamic memory performance.

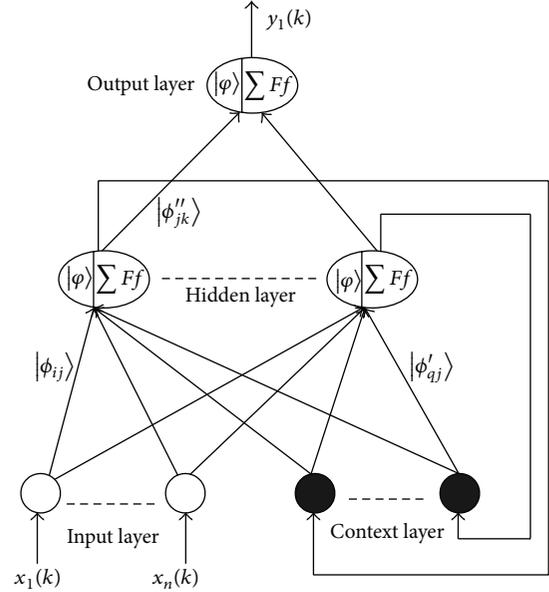


FIGURE 4: Quantum Elman neural networks.

QENN can not only store the information in the current input data but also store some historical information. In the training process, it can dynamically backtrack to the input at the past time related to the expected output at the current time.

$n, m$  are the number of the input nodes and the hidden nodes, respectively. The number of the context nodes is equal to the number of the hidden nodes. There is one output node. The number of iterations is denoted as  $k$  ( $k = 1, 2, \dots, T$ ).  $|\phi_{ij}\rangle$  is the weight between  $i$ th input node and  $j$ th hidden node, and it can be described as

$$|\phi_{ij}\rangle = \begin{pmatrix} \cos \theta_{ij} \\ \sin \theta_{ij} \end{pmatrix}. \quad (7)$$

$|\phi'_{qj}\rangle$  is the weight between  $q$ th context node and  $j$ th hidden node, and it can be described as

$$|\phi'_{qj}\rangle = \begin{pmatrix} \cos \theta'_{qj} \\ \sin \theta'_{qj} \end{pmatrix}. \quad (8)$$

$|\phi''_{jk}\rangle$  is the weight between  $j$ th hidden node and  $k$ th output node, and it can be described as

$$|\phi''_{jk}\rangle = \begin{pmatrix} \cos \theta''_{jk} \\ \sin \theta''_{jk} \end{pmatrix}. \quad (9)$$

$x_i(k)$  is the input value of  $i$ th external input neuron at time  $k$ .

$S_q^C(k)$  is the internal state of  $q$ th context neuron at time  $k$ , which is the output of  $q$ th hidden neuron at time  $k - 1$ , and it can be expressed as

$$S_q^C(k) = y_q^H(k - 1), \quad (10)$$

where  $S_q^C(k)$  is the output of  $q$ th hidden neuron at time  $k - 1$ .

$y_q^C(k)$  is the output of  $q$ th context neuron at time  $k$ , and it can be expressed as

$$y_q^C(k) = S_q^C(k). \quad (11)$$

$S_j^H(k)$  is the internal state of  $j$ th hidden neuron at time  $k$ , which can be expressed as

$$\begin{aligned} S_j^H(k) &= \sum_{i=1}^n x_i(k) \langle \phi_{ij} | \varphi_j \rangle + \sum_{q=1}^m y_q^C(k) \langle \phi'_{qj} | \varphi_j \rangle \\ &\quad - \tau_j \\ &= \sum_{i=1}^n x_i(k) \cos(\theta_{ij} - \xi_j) \\ &\quad + \sum_{q=1}^m y_q^C(k) \cos(\theta'_{qj} - \xi_j) - \tau_j, \end{aligned} \quad (12)$$

where  $|\varphi_j\rangle$  is the activation value of  $j$ th hidden neuron, which can be described as

$$|\varphi_j\rangle = [\cos \xi_j, \sin \xi_j]^T. \quad (13)$$

$y_j^H(k)$  is the output of  $j$ th hidden neuron at time  $k$ , which can be expressed as

$$y_j^H(k) = f(S_j^H(k)). \quad (14)$$

$S^O(k)$  is the internal state of the output neuron at time  $k$ , and it can be expressed as

$$\begin{aligned} S^O(k) &= \sum_{j=1}^m y_j^H(k) \langle \phi''_{jk} | \varphi'_k \rangle - \tau'_k \\ &= \sum_{j=1}^m y_j^H(k) \cos(\theta''_{jk} - \xi'_k) - \tau'_k, \end{aligned} \quad (15)$$

where  $|\varphi'_k\rangle$  is the activation value of  $k$ th output neuron, which can be described as

$$|\varphi'_k\rangle = [\cos \xi'_k, \sin \xi'_k]^T. \quad (16)$$

$y^O(k)$  is the output of the output neuron at time  $k$ , which can be expressed as

$$y^O(k) = f(S^O(k)). \quad (17)$$

**2.3. Multiplicity of Convergent Solution for QENN.** According to (10)–(17),  $y^O(k)$  can also be expressed as

$$\begin{aligned} y^O(k) &= f \left( \sum_{j=1}^m f \left( \sum_{i=1}^n x_i(k) \cos(\theta_{ij} - \xi_j) \right. \right. \\ &\quad \left. \left. + \sum_{q=1}^m y_q^H(k-1) \cdot \cos(\theta'_{qj} - \xi_j) - \tau_j \right) \right. \\ &\quad \left. \cdot \cos(\theta''_{jk} - \xi'_k) - \tau'_k \right). \end{aligned} \quad (18)$$

If  $\theta^*$  and  $\xi^*$  are the convergent solution of the sequence  $\{\theta\}$  and  $\{\xi\}$ ,  $\hat{\theta}^*$  and  $\hat{\xi}^*$  which satisfy  $\hat{\theta}^* - \hat{\xi}^* = \pm(\theta^* - \xi^*)$  are the convergent solution of the sequence  $\{\theta\}$  and  $\{\xi\}$ .

This shows that many groups of  $|\phi\rangle$  and  $|\varphi\rangle$  are the convergent solution of the sequence  $\{|\phi\rangle\}$  and  $\{|\varphi\rangle\}$ . It only requires the argument of  $|\phi\rangle$  and  $|\varphi\rangle$  to satisfy a certain relation, and the specific value of the argument is not required. This ensures the multiplicity of convergent solution for QENN, which is the basis of improving the approximation capability of the neural networks and making its performance enhanced in the application to a short-term load prediction of QENN.

**2.4. Parameters Optimization of QENN by Genetic Algorithm.** Genetic algorithm is used to adjust the parameters of the QENN, which include

$$\begin{aligned} &\{\theta_{ij} \ (i = 1, \dots, n; \ j = 1, \dots, m); \\ &\theta'_{qj} \ (q = 1, \dots, m; \ j = 1, \dots, m); \\ &\theta''_{jk} \ (j = 1, \dots, m; \ k = 1); \ \xi_j \ (j = 1, \dots, m); \\ &\xi'_k \ (k = 1); \ \tau_j \ (j = 1, \dots, m); \ \tau'_k \ (k = 1)\}. \end{aligned} \quad (19)$$

The chromosome can be constituted by these parameters, which can be expressed as  $Z = [\theta_{11}, \theta_{12}, \dots, \theta_{nm}, \theta'_{11}, \theta'_{12}, \dots, \theta'_{mm}, \theta''_{11}, \theta''_{21}, \dots, \theta''_{m1}, \xi_1, \xi_2, \dots, \xi_m, \xi'_1, \tau_1, \tau_2, \dots, \tau_m, \tau'_1]$ .

Genetic algorithm uses the roulette method as selecting operator. The crossover operator adopts the nonuniform arithmetic crossover method, and the mutation operator uses the noncoherence mutation method. The fitness function mainly considers the accuracy of the output of the forecasting model; that is,  $f_{\text{fitness}} = 1/f_1(t)$ .  $f_1(t)$  is the mean square deviation between the desired output and the actual output of the forecasting model.

The process is shown in Figure 5.

**2.4.1. Nonuniform Arithmetic Crossover Operator.** The operation object of arithmetic crossover operator is chromosome  $Z$ , which is encoded by floating point numbers.

If two chromosome individuals are  $Z_A^t$  and  $Z_B^t$ , the two new individuals generated by the crossover operation can be expressed as

$$\begin{aligned} Z_A^{t+1} &= \alpha Z_B^t + (1 - \alpha) Z_A^t, \\ Z_B^{t+1} &= \alpha Z_A^t + (1 - \alpha) Z_B^t. \end{aligned} \quad (20)$$

For the nonuniform arithmetic crossover operator,  $\alpha$  is the random number, and its value range is from  $-0.5$  to  $1.5$ .

**2.4.2. Noncoherence Mutation Operator.** Supposing the chromosome individual  $Z = (z_1, z_2, \dots, z_n)$  is a parent solution, the component  $z_k$  is selected to mutate, and its value range is  $[a_k, b_k]$ . The mutated chromosome individual is  $Z' = (z_1, z_2, \dots, z_{k-1}, z'_k, \dots, z_n)$ .

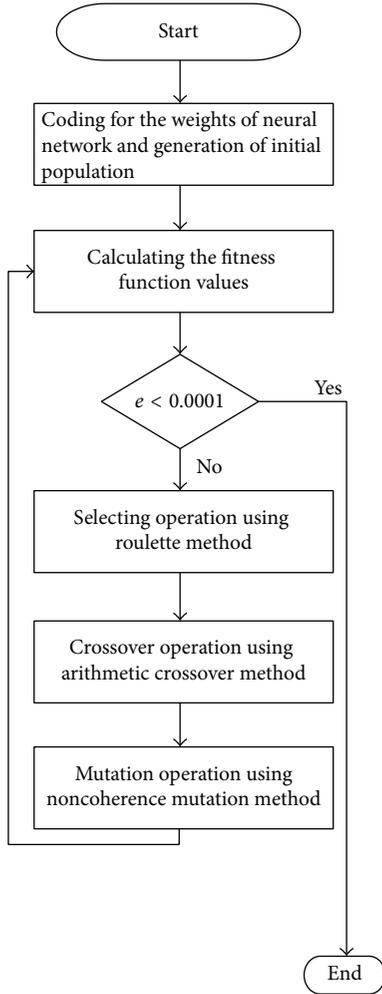


FIGURE 5: The process of parameters optimization of QENN by genetic algorithm.

$z'_k$  can be expressed as

$$z'_k = \begin{cases} z_k + \Delta(t, b_k - z_k) & \text{if } \text{rnd}(2) = 0 \\ z_k - \Delta(t, z_k - a_k) & \text{if } \text{rnd}(2) = 1, \end{cases} \quad (21)$$

where  $t$  is the current evolution generation, and  $\Delta(t, s)$  can be expressed as

$$\Delta(t, s) = s \times \left(1 - r^{(1-t/T)^\lambda}\right), \quad (22)$$

where  $r$  is a random number and its value range is from 0 to 1.  $T$  is the maximal generation.  $\lambda$  is a parameter for the nonuniform arithmetic crossover operator, and its value is 2.5.

### 3. PSRT and KNN

**3.1. Phase Space Reconstruction Theory.** The chaotic time series is usually analyzed by PSRT. The embedding theorem of Takens and Saner et al. is the basis of the prediction theory to chaotic time series.

To chaotic time series  $x_1, x_2, \dots, x_{n-1}, x_n, \dots$ , the phase space can be reconstructed by PSRT, which can be described as

$$X(t_i) = [x(t_i), x(t_i + \tau), \dots, x(t_i + (m-1)\tau)], \quad (23)$$

where  $m$  is the embedding dimension and it can be acquired by the G-P method [17].  $\tau$  is the delay time, and it can be acquired by multiple-autocorrelation method [18]. In the  $m$ -dimension phase space, the evolution of the chaotic system can be described by the evolution track of phase points.

According to the G-P method, the embedding dimension  $m$  is 7. By multiple-autocorrelation method, the delay time  $\tau$  is 1 for the real example simulation in this paper.

PSRT provides a theoretical basis for the construction of the QENN model. For the model, the number of input nodes is decided by the embedding dimension  $m$  according to PSRT.

**3.2. K-Nearest Neighbor Approach.** The  $K$ -nearest neighbor approach is usually used in the forecasting research of chaotic time series [19], which is based on the Euclid distance. The similarity of two phase points is computed by the Euclid distance.

In the reconstructed phase space with  $m$ -dimension, the forecasting phase point is recorded as  $X_r(t)$ , and  $X_{ri}(t)$  denotes  $i$ th neighbor phase point.  $d_{ri}(t)$  is the Euclid distance between the forecasting phase points and  $i$ th neighbor phase point, which is expressed as

$$d_{ri}(t) = \|X_r(t) - X_{ri}(t)\|. \quad (24)$$

According to (24), the nearest neighbor phase points set can be formed, which is the training samples set of forecasting phase points for the QENN model.

## 4. Real Example Simulation

**4.1. Four STLF Models for Real Example Simulation.** Four STLF models are constructed, which are model-I based on the QENN, model-II based on the quantum feedforward neural network, model-III based on the conventional Elman neural network, and model-IV based on the conventional feedforward neural network.

The input nodes number of four STLF models for actual simulation load system is 7, which is acquired by the G-P method. The delay time  $\tau$  is 1, which is acquired by multiple-autocorrelation method.

The number of the neighbor phase points affects the prediction accuracy and calculation of the forecasting model. If the number of the neighbor phase points is too much, it will not only increase the amount of computation but also increase the number of false neighbor phase points, which has a great impact on the prediction performance. If the number of the neighbor phase points is too small, the effective information is not fully utilized, which also affects the prediction performance. In general, the relationship between the number of the neighbor phase points  $k$  and the embedding dimension  $m$  satisfies the following conditions:

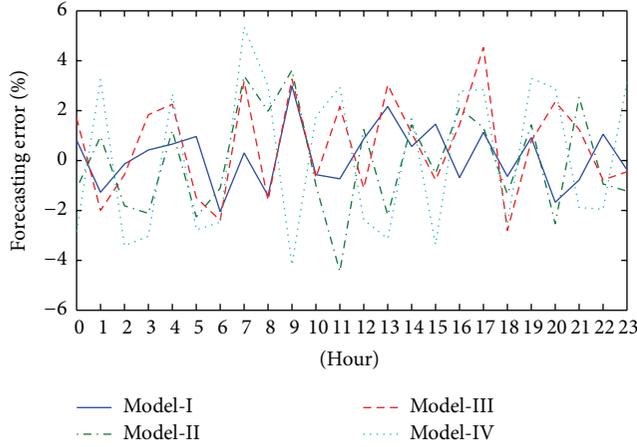


FIGURE 6: The daily load forecasting errors of four STL models for workday.

$k \geq m + 1$ . Based on multiple validation tests, the number of the neighbor phase points is 10 for the actual simulation load system in this paper.

For model-I and model-III, the number of the input nodes is 7, the number of the hidden nodes is 10, the number of the context nodes is 10, and the number of the output points is 1.

For model-II and model-IV, the number of the input nodes is 7, the number of the hidden nodes is 10, and the number of the output points is 1.

The maximum permissible error of the four STL models is  $\varepsilon = 10^{-4}$ . For model-I, model-II, model-III, and model-IV, genetic algorithm was used to adjusted the parameters of QENN, the quantum feedforward neural network, and the conventional Elman neural network. The fitness function mainly considers the accuracy of the output of the forecasting model. The crossover probability is 0.90, and the variation probability is 0.10.

**4.2. Simulation Results.** Four models are used to predict the actual load. The daily load forecasting errors and comparison of four models for workday and weekend are shown in Figures 6 and 7, respectively.

The mean absolute percentage error and the maximum relative error of four STL models for workday and weekend are described in Table 1. The mean absolute percentage error can be calculated as

$$E_{MAPE} = \frac{1}{M} \sum_{i=1}^M \frac{|\hat{Y}_i - Y_i|}{Y_i} \times 100\%, \quad (25)$$

where  $M$  is the number of the forecasting points,  $\hat{Y}_i$  is the forecasting load, and  $Y_i$  is the actual load.

The maximum relative error is the maximum value of relative error of all predicted points. The relative error can be calculated as

$$E_{RE} = \frac{|\hat{Y}_i - Y_i|}{Y_i} \times 100\%. \quad (26)$$

TABLE 1: The error comparison of four STL models for two day types.

Day type	Model-I		Model-II		Model-III		Model-IV	
	$E_{MAPE}$	$E_{MAX}$	$E_{MAPE}$	$E_{MAX}$	$E_{MAPE}$	$E_{MAX}$	$E_{MAPE}$	$E_{MAX}$
Workday	1.025	2.977	1.826	4.450	1.813	4.521	2.903	5.327
Weekend	1.210	3.225	2.153	4.705	2.275	4.860	3.152	5.515

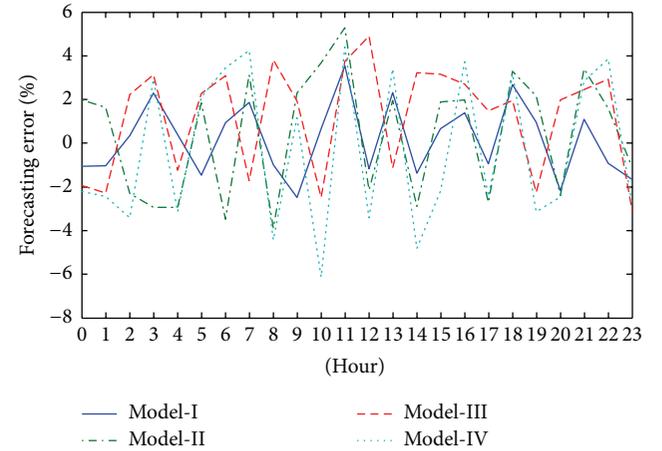


FIGURE 7: The daily load forecasting errors of four STL models for weekend.

The error performance of the model-IV is the worst in the all models, because the conventional feedforward neural network cannot effectively characterize the chaotic dynamic behavior of the load system. The error performance of model-II is better than that of model-IV, because quantum computation can enhance information processing capability and generalization ability of quantum feedforward neural networks. The error performance of model-III is better than that of model-IV, because the conventional Elman neural network has formed specific dynamic memory performance. The forecasting performance of model-I is the best, which is the fusion of quantum computation and feedback mechanism.

From Table 1 it can be seen that the forecasting error is larger in weekend. In weekend, the load mode is more diverse and complex, which leads to the forecasting error is slightly larger than that of workday.

In order to verify how sensitive are the simulation results on  $m$  and  $\tau$ , the forecasting errors of model-I with different  $m$  are described in Table 2, and the forecasting errors of model-I with different  $\tau$  are described in Table 3.

From Tables 2 and 3, it can be seen that the forecasting error is related to  $m$  and  $\tau$ . For the actual simulation load system in this paper,  $m = 7$  and  $\tau = 1$  are appropriate.

In order to further verify the performance of model-I, the real example are used to test in a week. The mean absolute percentage error and the maximum relative error in the week are described in Table 4. It is shown that the proposed model-I has good precision performance and steadiness.

TABLE 2: The forecasting error of model-I with different  $m$  ( $\tau = 1$ ).

	$m$									
	5		6		7		8		9	
	$E_{MAPE}$	$E_{MAX}$								
Monday	1.629	4.512	1.466	4.011	1.025	2.977	1.358	3.965	1.540	4.250

TABLE 3: The forecasting error of model-I with different  $\tau$  ( $m = 7$ ).

	$\tau$							
	1		2		3		4	
	$E_{MAPE}$	$E_{MAX}$	$E_{MAPE}$	$E_{MAX}$	$E_{MAPE}$	$E_{MAX}$	$E_{MAPE}$	$E_{MAX}$
Monday	1.025	2.977	1.376	3.852	1.405	4.009	1.466	4.542

TABLE 4: The daily load forecasting errors in a week.

Day type	Model-I	
	$E_{MAPE}$	$E_{MAX}$
Monday	1.025	2.977
Tuesday	1.109	3.012
Wednesday	1.153	2.985
Thursday	1.128	2.944
Friday	1.090	2.870
Saturday	1.210	3.225
Sunday	1.227	3.412

### 5. Conclusions

This paper constructs a STLF model based on QENN. The quantum computation and Elman feedback mechanism were integrated into QENN. Quantum computation can effectively improve the approximation capability and the information processing ability of the neural networks. QENN has not only the feedforward connection but also the feedback connection. Phase space reconstruction theory is the theoretical basis of constructing the forecasting model. The training samples are formed by means of  $K$ -nearest neighbor approach. Through the example simulation, the testing results show that proposed model can effectively improve the prediction accuracy. The research in the paper makes a theoretical foundation for the practical engineering application of the STLF model based on QENN. The model based on QENN can also be used in the power prediction for new energy, such as wind power prediction and photovoltaic power prediction.

### Competing Interests

The authors declare that they have no competing interests.

### Acknowledgments

This work was supported by the project Supported by National Nature Science Foundation of China (51477078).

### References

- [1] E. Paparoditis and T. Sapatinas, "Short-term load forecasting: the similar shape functional time-series predictor," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3818–3825, 2013.
- [2] S. Skolthanasarat, U. Lewlomphaisarl, and K. Tungpimolrut, "Short-term load forecasting algorithm and optimization in smart grid operations and planning," in *Proceedings of the 2nd Annual IEEE Conference on Technologies for Sustainability (SusTech '14)*, pp. 165–171, IEEE, Portland, Ore, USA, July 2014.
- [3] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4356–4364, 2013.
- [4] A. K. Fard and M.-R. Akbari-Zadeh, "A hybrid method based on wavelet, ANN and ARIMA model for short-term load forecasting," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 26, no. 2, pp. 167–182, 2014.
- [5] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.
- [6] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Systems with Applications*, vol. 36, no. 1, pp. 273–279, 2009.
- [7] K.-H. Kim, J.-K. Park, K.-J. Hwang, S.-H. Kim, H.-G. Han, and S.-H. Kang, "Implementation of short-term load forecasting expert systems in a real environment," *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 8, no. 3, pp. 139–144, 2000.
- [8] S. C. Kak, "On quantum neural computing," *Information Sciences*, vol. 83, no. 3-4, pp. 143–160, 1995.
- [9] V. Gandhi, G. Prasad, D. Coyle, L. Behera, and T. M. McGinnity, "Quantum neural network-based EEG filtering for a brain-computer interface," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 278–288, 2014.
- [10] K. Takahashi, Y. Shiotani, and M. Hashimoto, "Remarks on model reference self-tuning PID controller using quantum neural network with qubit neurons," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR '13)*, pp. 253–257, Hanoi, Vietnam, December 2013.
- [11] P. Li, Y. Chai, and Q. Xiong, "Quantized neural modeling: hybrid quantized architecture in Elman networks," *Neural Processing Letters*, vol. 37, no. 2, pp. 163–187, 2013.

- [12] M. F. Ab Aziz and S. M. Hj Shamsuddin, "Quantum particle swarm optimization for Elman recurrent network," in *Proceedings of the 4th International Conference on Mathematical Modelling and Computer Simulation (AMS '10)*, pp. 133–137, Kota Kinabalu, Malaysia, May 2010.
- [13] P.-H. Li, Y. Chai, and Q.-Y. Xiong, "Quantum gate Elman neural network and its quantized extended gradient back-propagation training algorithm," *Acta Automatica Sinica*, vol. 39, no. 9, pp. 1511–1522, 2013.
- [14] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [15] A. C. Tsakoumis, P. Fessas, and V. M. Mladenov, "Application of chaotic time series for short-term load prediction," *WSEAS Transactions on Systems*, vol. 2, no. 3, pp. 517–523, 2003.
- [16] Z. Xu, G. Wu, and H. Wang, "Forecasting method for enhancing oil recovery in molecule deposition oil-displacement based on quantum neural network," *Acta Petrolei Sinica*, vol. 29, no. 1, pp. 84–88, 2008.
- [17] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D: Nonlinear Phenomena*, vol. 9, no. 1-2, pp. 189–208, 1983.
- [18] J. Y. Lin, Y. K. Wang, and Z. P. Huang, "Selection of proper time-delay in phase space reconstruction of speech signals," *Signal Processing*, vol. 15, no. 3, pp. 220–225, 1999.
- [19] I. Drezga and S. Rahman, "Input variable selection for ANN-based short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1238–1244, 1998.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

