

Research Article

Short-Term Load Forecasting Model Based on the Fusion of PSRT and QCNN

Zhisheng Zhang

School of Automation and Electrical Engineering, Qingdao University, Qingdao 266071, China

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

Received 11 April 2017; Revised 12 July 2017; Accepted 29 August 2017; Published 3 October 2017

Academic Editor: Yaguo Lei

Copyright © 2017 Zhisheng Zhang. 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 (STLF) model based on the fusion of Phase Space Reconstruction Theory (PSRT) and Quantum Chaotic Neural Networks (QCNN) was proposed. The quantum computation and chaotic mechanism were integrated into QCNN, which was composed of quantum neurons and chaotic neurons. QCNN has four layers, and they are the input layer, the first hidden layer of quantum hidden nodes, the second hidden layer of chaotic hidden nodes, and the output layer. The theoretical basis of constructing QCNN is Phase Space Reconstruction Theory (PSRT). Through the actual example simulation, the simulation results show that proposed model has good forecasting precision and stability.

1. Introduction

Short-term load forecasting (STLF) is a key basic research project. It has an important role in the economy, reliability, and operation management of power system. Smart grids will become the trend of the future development of the electric power system. Because of the higher economy and reliability of the smart grids, it will put forward higher forecasting accuracy for STLF model [1–3].

In the early stage of STLF, many forecasting methods based on mathematical statistics theory are put forward [4, 5]. But these methods are not suitable for the prediction of dynamic load time series. With the development of artificial intelligence technology, many forecasting models based on intelligent theory are proposed and applied [6, 7]. As an important branch of intelligent theory, neural network has been widely used in the STLF field because of its unique advantages.

The research results show that there are quantum phenomena and quantum mechanisms in the process of human brain information processing [8]. The quantum theory is introduced into the modeling of artificial neural network, which can improve the generalization ability and information processing efficiency of artificial neural network. The fusion

of quantum theory and artificial neural network has been applied in many fields [9–11]. The quantum neural network has been applied to STLF, and satisfactory prediction results have been obtained [12].

Some studies had shown that the load time series is nonlinear and chaotic [13]. Chaotic characteristics had been found in the biological neurons. Chaotic neural network is the neural network in which chaotic characteristics are integrated. It has a strong sensitivity to initial values and has the ability to process complicated information. A number of literatures have discussed the application of chaotic neural network in STLF, and the prediction effect is satisfactory [14, 15].

The STLF model based on the fusion of Phase Space Reconstruction Theory (PSRT) and Quantum Chaotic Neural Networks (QCNN) is constructed in this paper. QCNN is composed of quantum neurons and modified chaotic neurons, and quantum computation and chaotic mechanism are integrated into the neural network. PSRT is the theoretical basis of constructing QCNN. The K -nearest neighbor approach (NNA) is used to form the training samples.

Through the example simulation, the simulation results show that the STLF model based on the fusion of PSRT and QCNN has good forecasting performance.

2. Quantum Chaotic Neural Networks

QCNN is composed of quantum neurons and chaotic neurons, and quantum computation and chaotic mechanism are integrated into QCNN.

2.1. Quantum Neuron Model. In quantum neurons, the basic storage unit of quantum information is the qubit. The state of a qubit is a vector in two-dimensional complex number space. The qubit differs from the classical bit in that a quantum state can exist continuously and randomly on any superposition state of two polarized states. Quantum gates are the basis for the physical implementation of quantum computation, and 1-bit qubit rotation gate and 2-bit controlled NOT gate are the most basic universal gates.

2.1.1. Qubit. In quantum computation, a qubit state $|\varphi\rangle$ can be expressed as

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

where $|0\rangle$ and $|1\rangle$ are the basic states of microscopic particles. α and β are the probability amplitudes, and they should satisfy the following equation:

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

Quantum logic gates are the basis of quantum computation. The 1-bit qubit rotation gate and the 2-bit controlled NOT gate are the basic universal quantum gates.

2.1.2. 1-Bit Qubit Rotation Gate. The 1-bit qubit rotation gate is expressed as

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (3)$$

The quantum state $|\varphi\rangle$ is expressed as

$$|\varphi\rangle = [\cos \theta_0 \quad \sin \theta_0]^T. \quad (4)$$

By the 1-bit qubit rotation gate, $|\varphi\rangle$ can be transformed to

$$\begin{aligned} R(\theta)|\varphi\rangle &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} \cos \theta_0 \\ \sin \theta_0 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta + \theta_0) \\ \sin(\theta + \theta_0) \end{bmatrix}. \end{aligned} \quad (5)$$

2.1.3. 2-Bit Controlled NOT Gate. The 2-bit controlled NOT gate is expressed as

$$C(\gamma) = \begin{bmatrix} \cos\left(\frac{\gamma\pi}{2} - 2\theta_0\right) & -\sin\left(\frac{\gamma\pi}{2} - 2\theta_0\right) \\ \sin\left(\frac{\gamma\pi}{2} - 2\theta_0\right) & \cos\left(\frac{\gamma\pi}{2} - 2\theta_0\right) \end{bmatrix}. \quad (6)$$

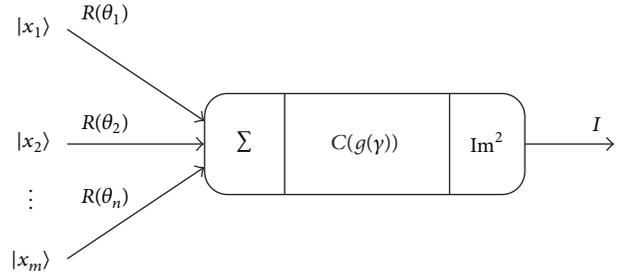


FIGURE 1: The quantum neuron model.

By the 2-bit controlled NOT gate, $|\varphi\rangle$ can be transformed to

$$\begin{aligned} C(\gamma)|\varphi\rangle &= \begin{bmatrix} \cos\left(\frac{\gamma\pi}{2} - 2\theta_0\right) & -\sin\left(\frac{\gamma\pi}{2} - 2\theta_0\right) \\ \sin\left(\frac{\gamma\pi}{2} - 2\theta_0\right) & \cos\left(\frac{\gamma\pi}{2} - 2\theta_0\right) \end{bmatrix} \\ &\times \begin{bmatrix} \cos \theta_0 \\ \sin \theta_0 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\gamma\pi}{2} - \theta_0\right) \\ \sin\left(\frac{\gamma\pi}{2} - \theta_0\right) \end{bmatrix}. \end{aligned} \quad (7)$$

When $\gamma = 1$, $C(\gamma)|\varphi\rangle = \begin{bmatrix} \sin(\theta_0) \\ \cos(\theta_0) \end{bmatrix}$, and it represents the reversal rotation.

2.1.4. Quantum Neuron Model. The quantum neuron model is shown in Figure 1. It consists of the input part, the rotation part, the aggregation part, the reversal rotation part, and the output part.

In order to solve the practical problems, the input variable x_i needs to be transformed to $|x_i\rangle$:

$$\begin{aligned} |x_i\rangle &= \cos\left(\frac{2\pi}{1 + \exp(-x_i)}\right)|0\rangle \\ &\quad + \sin\left(\frac{2\pi}{1 + \exp(-x_i)}\right)|1\rangle \\ &= \left[\cos\left(\frac{2\pi}{1 + \exp(-x_i)}\right) \quad \sin\left(\frac{2\pi}{1 + \exp(-x_i)}\right) \right]^T. \end{aligned} \quad (8)$$

In the rotation part, the input qubit $|x_i\rangle = [\cos \phi_i \quad \sin \phi_i]^T$ can be transformed by the 1-bit qubit rotation gate, as follows:

$$\begin{aligned} R(\theta_i)|x_i\rangle &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \times \begin{bmatrix} \cos \phi_i \\ \sin \phi_i \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta_i + \phi_i) \\ \sin(\theta_i + \phi_i) \end{bmatrix}. \end{aligned} \quad (9)$$

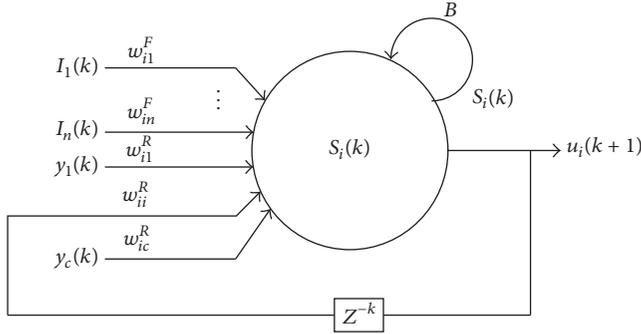


FIGURE 2: The chaotic neuron model.

The aggregation part can be described as

$$\sum_{i=1}^m R(\theta_i) |x_i\rangle = [\cos \delta \quad \sin \delta]^T, \quad (10)$$

where

$$\begin{aligned} \delta &= \arg \left(\sum_{i=1}^m R(\theta_i) |x_i\rangle \right) \\ &= \arctan \left(\frac{\sum_{i=1}^m \sin(\theta_i + \phi_i)}{\sum_{i=1}^m \cos(\theta_i + \phi_i)} \right). \end{aligned} \quad (11)$$

The reversal part can be described as

$$\begin{aligned} C(g(\gamma)) \sum_{i=1}^m R(\theta_i) |x_i\rangle \\ = \left[\cos \left(\frac{\pi}{2} g(\gamma) - \delta \right) \quad \sin \left(\frac{\pi}{2} g(\gamma) - \delta \right) \right]^T, \end{aligned} \quad (12)$$

where $g(\cdot)$ is the activation function.

The output of the quantum neuron is the probability amplitudes of $|1\rangle$. So the relationship between input and output can be expressed as

$$\begin{aligned} I &= \sin \left(\frac{\pi}{2} g(\gamma) - \delta \right) \\ &= \sin \left(\frac{\pi}{2} g(\gamma) - \arg \left(\sum_{i=1}^m R(\theta_i) |x_i\rangle \right) \right). \end{aligned} \quad (13)$$

2.2. Chaotic Neuron Model. The modified Aihara chaotic neuron was presented in [16], which is shown in Figure 2. The modified Aihara chaotic neuron is used to construct QCNN model in this paper.

$S_i(k+1)$ is the internal state of the i th chaotic neuron at time $k+1$, which can be expressed as

$$S_i(k+1) = BS_i(k) + \sum_{j=1}^n w_{ij}^F I_j(k) + \sum_{l=1}^c w_{il}^R u_l(k), \quad (14)$$

where $S_i(k)$ is the internal state of the i th chaotic neuron at time k . w_{ij}^F is the weight between the j th external input

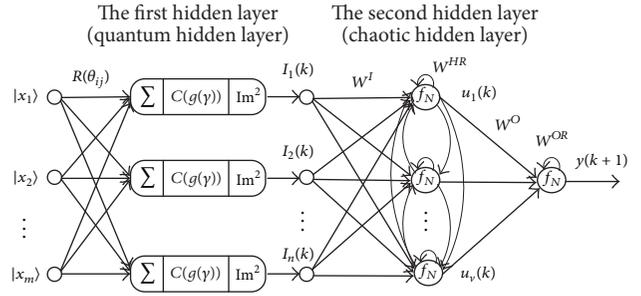


FIGURE 3: The Quantum Chaotic Neural Network.

neuron and the i th chaotic neuron. $I_j(k)$ is the input value of the j th external input neuron at time k . The number of external input neurons is n . w_{il}^R is the weight between the l th internal feedback neuron and the i th chaotic neuron. $u_l(k)$ is the output of the l th chaotic neuron at time k . The number of internal feedback input neurons applied to the chaotic neuron is c . The damping factors of the external, feedback, and refractoriness are equal to B .

$u_i(k+1)$ is the output of the i th chaotic neuron at time $k+1$, which can be expressed as

$$u_i(k+1) = f_N(S_i(k+1)), \quad (15)$$

where f_N can be expressed as

$$f_N(S_i(k+1)) = \frac{1}{1 + \exp(-S_i(k+1))}. \quad (16)$$

2.3. Quantum Chaotic Neural Networks Model. The QCNN model is constructed by quantum neurons and chaotic neurons, which is shown in Figure 3.

Considering Figure 3, QCNN has four layers, and they are the input layer, the first hidden layer of quantum hidden nodes, the second hidden layer of chaotic hidden nodes, and the output layer.

m, n, v are the number of the input nodes, the quantum hidden nodes, and the chaotic hidden nodes. There is one output node.

There are three different types of connection weights in QCNN. The first type $R(\theta_{ij})$ is the weights for forward direction between the input layer and the first hidden layer. The second type is the connection weights from the first hidden layer to the second hidden layer and from the second hidden layer to the output layer. The third type is the connection weights among the neurons of interlayer, which includes self-feedback in the second hidden layer and in the output layer.

In the input layer, $|x_i\rangle$ is the i th input qubit, which has been transformed from the input variable x_i according to (8). $\{|x_1\rangle, |x_2\rangle, \dots, |x_m\rangle\}$ is the input vector of QCNN.

In the first hidden layer, $I_i(k)$ is the output of the i th quantum hidden neuron at time k , and it can be expressed as

$$I_i(k) = \sin \left(\frac{\pi}{2} g(\gamma) - \arg \left(\sum_{j=1}^m R(\theta_{ij}) |x_j\rangle \right) \right). \quad (17)$$

In the second hidden layer, $S_j^H(k)$ is the internal state of the j th chaotic neuron at time k , which can be described as

$$S_j^H(k) = \sum_{i=1}^n w_{ij}^I I_i(k) + \sum_{q=1}^v w_{jq}^{HR} u_q(k-1) + B \cdot S_j^H(k-1), \quad (18)$$

where w_{ij}^I is the weight between the i th quantum hidden neuron in the first hidden layer and the j th chaotic neuron in the second hidden layer. w_{jq}^{HR} is the interconnecting weight between the j th chaotic neuron and the q th chaotic neuron in the second hidden layer. $u_q(k-1)$ is the output of q th chaotic neuron in the second hidden layer. B is refractory parameter of chaotic neuron.

$u_j(k)$ are the output of j th chaotic neuron in the second hidden layer, which can be expressed as

$$u_j(k) = f_N[S_j^H(k)], \quad (19)$$

where f_N is a sigmoid function.

In the output layer, $S^O(k)$ is the internal state of the chaotic neuron at time k , which can be expressed as

$$S^O(k) = \sum_{j=1}^v w_j^O u_j(k) + w^{OR} y(k-1) + B \cdot S^O(k-1), \quad (20)$$

where w_j^O is the weight between the j th chaotic neuron in the second hidden and the only chaotic neuron in the output layer. w^{OR} is the interconnecting weight among neurons in the output layer. $y(k)$ is the output of the chaotic neuron at time k , which can be described as

$$y(k) = f_N[S^O(k)]. \quad (21)$$

3. STLF Model Based on the Fusion of PSRT and QCNN

The STLF model fuses PSRT and QCNN. QCNN is the core of the forecasting model. PSRT provides a theoretical basis for the construction of the model. For the QCNN model, the number of input nodes is decided by the embedding dimension m according to PSRT. The k -nearest neighbor phase points are obtained by K -nearest neighbor approach [17], which can form the training samples of QCNN.

3.1. Phase Space Reconstruction Theory. PSRT is the important theoretical basis for the analysis of chaotic time series. The chaotic time series can be reconstructed by PSRT. To chaotic time series $x_1, x_2, \dots, x_{n-1}, x_n, \dots$, the reconstructed phase space can be expressed as

$$X(t_i) = [x(t_i), x(t_i + \tau), \dots, x(t_i + (m-1)\tau)], \quad (22)$$

where m is the embedding dimension and τ is the delay time. According to [18], the G-P method is used to acquire

the embedding dimension. According to [19], the multiple-autocorrelation method is used to acquire the delay time τ .

The main steps of the G-P method are as follows:

- (1) For the time series $x_1, x_2, \dots, x_{n-1}, x_n, \dots$, a smaller embedding dimension value m_0 is used to reconstruct the phase space. For all the phase points, an arbitrary small number r is given, and the Euclidean distance between the corresponding phase points in the phase space is computed.
- (2) Computing correlation function: it can be calculated by the following formula:

$$C(r) = \lim_{N \rightarrow \infty} \frac{1}{s^2} \sum_{i,j=1}^s h(r - |Y(t_i) - Y(t_j)|), \quad (23)$$

where s is the number of phase points; $|Y(t_i) - Y(t_j)|$ is the distance between the phase point $Y(t_i)$ and the phase point $Y(t_j)$. $h(z)$ is the Heaviside function, and it can be expressed as

$$h(z) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0. \end{cases} \quad (24)$$

$C(r)$ is the cumulative distribution function, which represents the probability that the distance between two phase points is less than r in phase space.

- (3) For an appropriate range of r , the dimension D of the attractor and the cumulative distribution function $C(r)$ should approach the logarithmic linear relationship. It can be expressed as

$$D(m) = \lim_{r \rightarrow 0} \frac{\ln c(r)}{\ln(r)}. \quad (25)$$

Thus, the correlation dimension $D(m_0)$ corresponding to m_0 can be obtained by fitting method.

- (4) Increasing embedding dimension m ($m > m_0$), the calculation steps (1) and (2) are repeated until the corresponding correlation dimension $D(m)$ does not change with the increase of m . It achieves convergence. m is the embedding dimension.

PSRT has been applied to the field of STLF, which opens up new research ideas for STLF [13, 20].

3.2. The Architectural Structure of STLF Model. The architectural structure of STLF model based on the fusion of PSRT and QCNN is shown in Figure 4.

In Figure 4, the function of "Historical Load Database" and "Storage Area for Forecasting Data" is to store the historical load data and forecasting data. The function of "Module for Calculating m and τ by PSRT" is to provide the basis for model construction. The function of "Module for Forming the Training Samples by KNN" is to form the training samples. The function of "Module for Data Normalization" is to normalize the sample data.

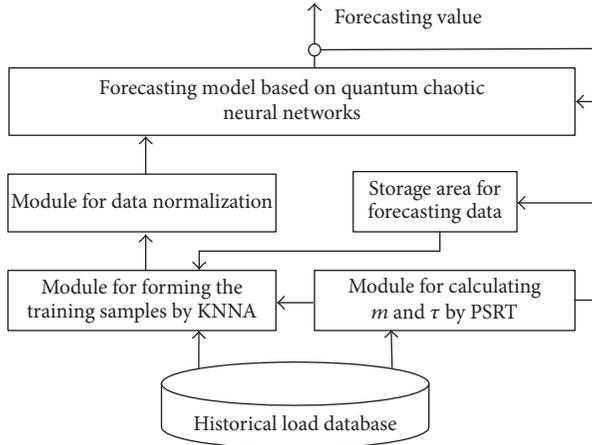


FIGURE 4: The architectural structure of STLF model based on the fusion of PSRT and QCNN.

4. Actual Example Simulation

4.1. Four STLF Models and Simulation Parameters. In order to verify the validity of the STLF model based on the fusion of PSRT and QCNN (Model I), three other models are also constructed to compare with Model I. They are the STLF model based on the fusion of PSRT and the quantum neural networks (Model II), the STLF model based on the fusion of PSRT and the chaotic neural networks (Model III), and the STLF model based on the fusion of PSRT and the conventional BP-NN (Model IV).

The time series in the simulation is the actual load in a regional power grid of Shandong Province in China in 2012. Twenty-four load points are collected each day, and the collected load values form the load time series.

The embedding dimension m is 7 according to the G-P method. The delay time τ is 1 according to the multiple-autocorrelation method. The number of the neighbor phase points is 10 by K -nearest neighbor approach.

For Model I, the number of the input nodes is 7, the number of the quantum hidden nodes is 9, the number of the chaotic hidden nodes is 7, and the number of the output points is 1.

For Model II, the number of the input nodes is 7, the number of the quantum hidden nodes is 9, and the number of the output points is 1.

For Model III, the number of the input nodes is 7, the number of the chaotic hidden nodes is 11, and the number of the output points is 1.

For 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 STLF models is $\varepsilon = 10^{-4}$. For Model I, Model II, and Model III, genetic algorithm was used to adjust the parameters of the QCNN, the quantum neural networks, and the chaotic neural networks. 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

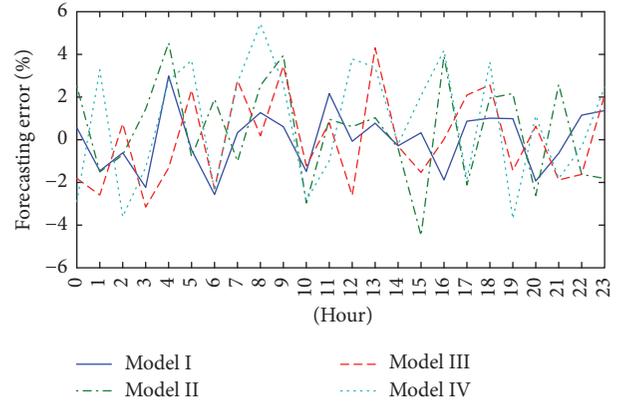


FIGURE 5: The daily load forecasting errors of four STLF models for spring season (2012-4-23).

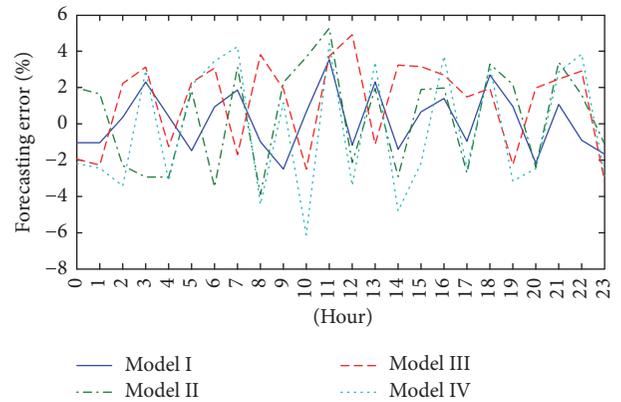


FIGURE 6: The daily load forecasting errors of four STLF models for summer season (2012-8-6).

and the actual output of the forecasting model. The crossover probability is 0.90, and the variation probability is 0.10. For Model I and Model III, the value of B is 0.1. For Model IV, the learning rate is 0.10, and the momentum factor is 0.90.

4.2. Real Example Simulation. Four models are used to forecast the actual load for different seasons including spring, summer, autumn, and winter.

The daily load forecasting errors of four models for spring, summer, autumn, and winter are shown in Figures 5–8.

The mean absolute percentage error (E_{MAPE}) and the maximum relative error (E_{MAX}) of four STLF models for spring, summer, autumn, and winter are described in Table 1.

According to Table 1, the mean absolute percentage error and the maximum relative error of Model IV are largest, because the conventional BP-NN is a static neural network, which cannot effectively characterize the chaotic dynamic behavior of the load system.

Compared with the Model IV, the mean absolute percentage error of Model II is reduced by 0.538%–0.648% and the maximum relative error of Model II is reduced by 0.848%–1.06%. This shows that quantum computation can

TABLE 1: The error comparison of four STLF models for four seasons.

Season	Model I		Model II		Model III		Model IV	
	E_{MAPE}	E_{MAX}	E_{MAPE}	E_{MAX}	E_{MAPE}	E_{MAX}	E_{MAPE}	E_{MAX}
Spring	1.166	2.976	2.091	4.529	1.827	4.295	2.629	5.429
Summer	1.437	3.550	2.622	5.277	2.549	4.912	3.270	6.125
Autumn	1.082	2.889	1.957	4.476	1.816	4.209	2.603	5.388
Winter	1.299	3.270	2.360	4.829	2.277	4.515	2.910	5.889

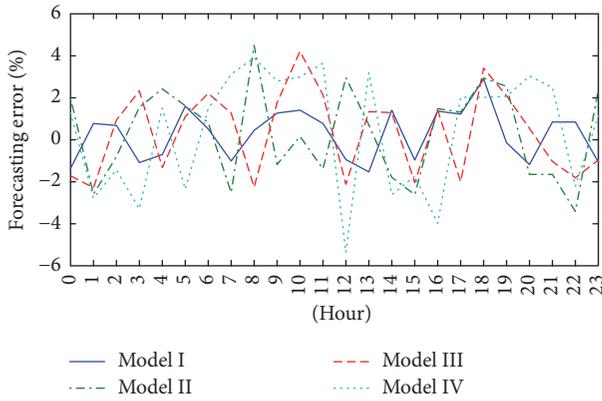


FIGURE 7: The daily load forecasting errors of four STLF models for autumn season (2012-10-22).

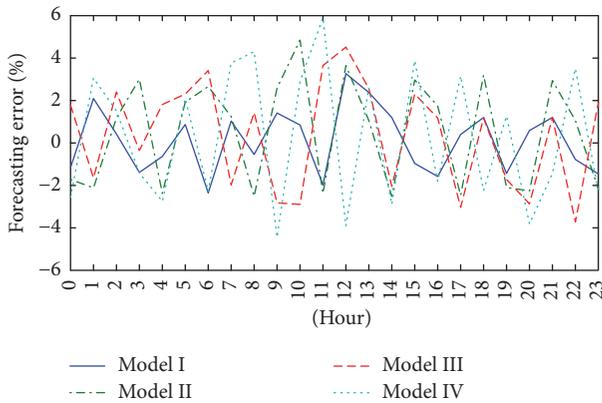


FIGURE 8: The daily load forecasting errors of four STLF models for winter season (2012-12-17).

enhance information processing capability and generalization ability of neural networks.

The error performance of Model III is better than that of Model II and Model IV, because chaotic neural networks can effectively characterize the chaotic dynamic behavior of the load system. The forecasting performance of Model I is the best, because quantum computing and chaotic mechanism are integrated into neural networks.

From Table 1 it can be seen that the forecasting error is larger in summer and winter, and the summer forecasting error is the biggest. In summer and winter, the weather factors such as temperature have a greater impact on the load, and the load fluctuation is relatively large, which leads to the

TABLE 2: The error comparison of four STLF models for four seasons.

Season	Model I		Model-SVM	
	E_{MAPE}	E_{MAX}	E_{MAPE}	E_{MAX}
Spring	1.166	2.976	2.252	4.640
Summer	1.437	3.550	2.800	5.335
Autumn	1.082	2.889	2.042	4.612
Winter	1.299	3.270	2.411	5.010

TABLE 3: The daily load forecasting errors in a week in spring.

Temperature °C	Error%	Model I	
		E_{MAPE}	E_{MAX}
Monday	11~17	1.166	2.976
Tuesday	11~15	1.097	2.885
Wednesday	12~16	1.138	2.903
Thursday	10~18	1.085	2.870
Friday	12~18	1.127	2.951
Saturday	11~17	1.205	3.016
Sunday	12~16	1.219	3.209

forecasting error being slightly larger than that of spring and autumn. The weather factors such as temperature have a little impact on the load in spring and autumn, and the load fluctuation is relatively small, which leads to the forecasting error being slightly small.

In order to further verify the prediction effect of the model proposed in this paper, the STLF model based on the fusion of PSRT and SVM is constructed and compared with the STLF model based on the fusion of PSRT and QCNN (Model I). The number of the input nodes is 7, and the number of the output points is 1. SVM uses the Gauss Kernel function, and σ is 0.15. Loss function parameter ϵ is 0.025, and penalty factor C is 3.

From Table 2 it can be seen that Model I is clearly superior to the STLF model based on the fusion of PSRT and SVM.

In order to further verify the precision performance and predictive stability, the real loads are used to test in a week in spring (2012-4-23~2012-4-29, a calm temperature week) and in summer (2012-8-6~2012-8-12, a high temperature week). The mean absolute percentage error and the maximum relative error in a calm temperature week are described in Table 3. The mean absolute percentage error and the maximum relative error in a high temperature week are described in Table 4. The mean absolute percentage error and

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

Temperature °C		Error% Model I	
		E_{MAPE}	E_{MAX}
Monday	24~36	1.437	3.550
Tuesday	26~34	1.362	3.439
Wednesday	25~37	1.466	3.510
Thursday	25~35	1.420	3.452
Friday	26~37	1.319	3.377
Saturday	24~34	1.492	3.980
Sunday	25~33	1.483	3.901

the maximum relative error of proposed Model I can meet the requirements of actual prediction.

5. Conclusions

A STLF model based on the fusion of PSRT and QCNN was proposed in this paper. QCNN was composed of quantum neurons and chaotic neurons. Quantum computation can effectively improve information processing ability and approximation capability of the QCNN. QCNN is sensitive to the initial load value and to the whole chaotic track because of chaotic mechanism. PSRT is the important analytical method for the analysis of chaotic time series, and it is the theoretical basis of constructing QCNN. Through the actual simulation verification, the testing results show that proposed model has good prediction accuracy and stability.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (51477078).

References

- [1] B. A. Hoverstad, A. Tidemann, H. Langseth, and P. Ozturk, "Short-term load forecasting with seasonal decomposition using evolution for parameter tuning," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1904–1913, 2015.
- [2] A. Ghasemi, H. Shayeghi, M. Moradzadeh, and M. Nooshyar, "A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management," *Applied Energy*, vol. 177, pp. 40–59, 2016.
- [3] S. A. Villalba and C. Á. Bel, "Hybrid demand model for load estimation and short term load forecasting in distribution electric systems," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 764–769, 2000.
- [4] A. El Desouky and M. Elkateb, "Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA," *IEE Proceeding—Generation, Transmission and Distribution*, vol. 147, no. 4, pp. 213–217, 2000.
- [5] C.-M. Huang, C.-J. Huang, and M.-L. Wang, "A particle swarm optimization to identifying the ARMAX model for short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1126–1133, 2005.
- [6] C. Xia, J. Wang, and K. McMenemy, "Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks," *International Journal of Electrical Power and Energy Systems*, vol. 32, no. 7, pp. 743–750, 2010.
- [7] D.-X. Niu, H.-F. Shi, and D. D. Wu, "Short-term load forecasting using bayesian neural networks learned by Hybrid Monte Carlo algorithm," *Applied Soft Computing Journal*, vol. 12, no. 6, pp. 1822–1827, 2012.
- [8] S. Kak, "On quantum neural computing," *Information Sciences*, vol. 83, no. 3-4, pp. 143–160, 1995.
- [9] K. Takahashi, M. Kurokawa, and M. Hashimoto, "Multi-layer quantum neural network controller trained by real-coded genetic algorithm," *Neurocomputing*, vol. 134, pp. 159–164, 2014.
- [10] C. Liu, Z. Y. He, and J. W. Yang, "A quantum neural network based fault diagnosis algorithm for power grid," *Power System Technology*, vol. 32, no. 9, pp. 56–60, 2008.
- [11] J. Li, "Quantum-inspired neural networks with application," *Open Journal of Applied Sciences*, vol. 5, no. 6, pp. 233–239, 2015.
- [12] P. Li, Y. Yan, W.-J. Zheng, J.-F. Jia, and B. Bai, "Short-term load forecasting based on quantum neural network by evidential theory," *Power System Protection and Control*, vol. 38, no. 16, pp. 49–53, 2010.
- [13] S. Kouhi, F. Keynia, and S. N. Ravadanegh, "A new short-term load forecast method based on neuro-evolutionary algorithm and chaotic feature selection," *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 862–867, 2014.
- [14] D. Niu, Y. Lu, X. Xu, and B. Li, "Short-term power load point prediction based on the sharp degree and chaotic RBF neural network," *Mathematical Problems in Engineering*, vol. 2015, Article ID 231765, 8 pages, 2015.
- [15] Y. He, Q. Xu, J. Wan, and S. Yang, "Electrical load forecasting based on self-adaptive chaotic neural network using Chebyshev map," *Neural Computing and Applications*, pp. 1–10, 2016.
- [16] S.-H. Kim, S.-D. Hong, and W.-W. Park, "An adaptive neurocontroller with modified chaotic neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '01)*, pp. 509–514, Washington, DC, USA, July 2001.
- [17] N. Zhang, A. Lin, and P. Shang, "Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting," *Physica A: Statistical Mechanics and its Applications*, vol. 477, pp. 161–173, 2017.
- [18] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D: Nonlinear Phenomena*, vol. 9, no. 1-2, pp. 189–208, 1983.
- [19] 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.
- [20] H. P. Yang, C. F. Wang, and K. C. Zhu, "Short-term load forecasting based on phase space reconstruction and Chebyshev orthogonal basis neural network," *Power System Protection and Control*, vol. 40, no. 24, pp. 95–99, 2012.



Hindawi

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

