Research Article

A Network Traffic Prediction Model Based on Quantum-Behaved Particle Swarm Optimization Algorithm and Fuzzy Wavelet Neural Network

Kun Zhang, Zhao Hu, Xiao-Ting Gan, and Jian-Bo Fang

School of Mathematics and Statistics, Chuxiong Normal University, Chuxiong, Yunnan 675000, China

Correspondence should be addressed to Kun Zhang; zhangkunpost@qq.com

Received 2 November 2015; Revised 14 January 2016; Accepted 3 February 2016

Academic Editor: Ahmed Kattan

Copyright © 2016 Kun Zhang et al. 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.

Due to the fact that the fluctuation of network traffic is affected by various factors, accurate prediction of network traffic is regarded as a challenging task of the time series prediction process. For this purpose, a novel prediction method of network traffic based on QPSO algorithm and fuzzy wavelet neural network is proposed in this paper. Firstly, quantum-behaved particle swarm optimization (QPSO) was introduced. Then, the structure and operation algorithms of WFNN are presented. The parameters of fuzzy wavelet neural network were optimized by QPSO algorithm. Finally, the QPSO-FWNN could be used in prediction of network traffic simulation successfully and evaluate the performance of different prediction models such as BP neural network, RBF neural network, fuzzy neural network, and FWNN-GA neural network. Simulation results show that QPSO-FWNN has a better precision and stability in calculation. At the same time, the QPSO-FWNN also has better generalization ability, and it has a broad prospect on application.

1. Introduction

With the rapid development of computer network technology, network applications have infiltrated every corner of human society and play an important role in various industries and situations. Since the network topology structure is gradually complicated, the problem of network's emergencies and congestion are more and more serious. Through monitoring and accuracy prediction of network traffic, it can prevent network congestion and can effectively improve the utilization rate of the network [1].

In general, the network traffic data is a kind of time series data and the problem of network traffic prediction is to forecast future network traffic rate variations as precisely as possible based on the measured history. The traditional prediction model, such as Markov model [2], ARMA (Autoregressive Moving Average) model [3], ARIMA (Autoregressive Integrated Moving Average) model [4], and FARIMA (Fractional Autoregressive Integrated Moving Average) [5] model, has been proposed. As the network traffic is affected by many factors, the network traffic time series show quite obvious

multiscale, long-range dependence, and nonlinear characteristic. The methods mentioned above have the weakness of low-level efficiency [6].

An artificial neural network (ANN) is an analysis paradigm that is roughly modeled after the massively parallel structure of the brain. Artificial neural networks can be thought of as “black box” devices that accept inputs and produce outputs and are able to give better performance in dealing with the nonlinear relationships between the output and the input theoretically [7]. Although artificial neural networks have been successfully used for modeling complex nonlinear systems and predicting signals for a wide range of engineering applications, artificial neural networks (ANNs) have limited ability to characterize local features, such as discontinuities in curvature, jumps in value or other edges [8]. These local features, which are located in time and/or frequency, typically embody important process-critical information such as aberrant process modes or faults.

The fuzzy neural networks (FNN) are the hybrid systems which combine both advantages of the fuzzy systems and artificial neural networks. The FNN possesses the merits of

the low-level learning and computational power of neural networks, and the high-level human knowledge representation and thinking of fuzzy theory [9]. A fuzzy wavelet neural network (FWNN) is a new network structure that combines wavelet theory with fuzzy logic and NNs. The synthesis of a fuzzy wavelet neural inference system includes the determination of the optimal definitions of the premise and the consequent part of fuzzy IF-THEN rules [10]. However, many fuzzy neural network models, including FWNN, have common problems derived from their fundamental algorithm [11]. For example, the design process for FNN and FWNN combined tapped delays with the backpropagation (BP) algorithm to solve the dynamic mapping problems [12]. Unfortunately, the BP training algorithm has some inherent defects [13, 14], such as low learning speed, existence of local minima, and difficulty in choosing the proper size of network to suit a given problem. Thus the systems which employ basic fuzzy inference theory make the degree of each rule extremely small and often make it underflow when the dimension of the task is large. In such a situation, the learning and inference cannot be carried out correctly.

As a variant of PSO, quantum-behaved particle swarm optimization (QPSO) is a novel optimization algorithm inspired by the fundamental theory of particle swarm and features of quantum mechanics such as the use of Schrödinger equation and potential field distribution [15]. As a global optimization algorithm, the QPSO can seek many local minima and thus increase the likelihood of finding the global minimum. This advantage of the QPSO can be applied to neural networks to optimize the topology and/or weight parameters [16].

In order to predict the network traffic more accurately, a prediction model of network traffic based on QPSO algorithm and fuzzy wavelet neural network is proposed in this paper. The network traffic data is trained by QPSO and fuzzy wavelet neural network and weights are progressively updated until the convergence criterion is satisfied. The objective function to be minimized by the QPSO algorithm is the predicted error function.

The rest of this paper is arranged as follows. Section 2 gives a brief introduction to classical PSO algorithm and quantum-behaved particle swarm optimization (QPSO) algorithm. In Section 3, the fuzzy wavelet neural network is introduced and the fuzzy wavelet neural network based on QPSO (QPSO-FWNN) algorithm is presented in detail. In Section 4, simulation results are presented. Performance metrics of the several prediction methods are analyzed and compared in Section 5. Finally, some conclusions are given in Section 6.

2. Quantum-Behaved Particle Swarm Optimization

2.1. Classical Particle Swarm Optimization. Particle swarm optimization (PSO) is an evolutionary computation technique that is proposed by Kennedy and Eberhart in 1995 [17]. Similarly to other genetic algorithms (GA), PSO is initialized with a population of random solutions. However, it is unlike GA, PSO does not have operators, such as

crossover and mutation. In the PSO algorithm, each potential solution, called “particles,” moves around in a multidimensional search space with a velocity constantly updated by the particle’s own experience and the experience of the particle’s neighbors or the experience of the whole swarm [18].

In the PSO, each particle keeps track of its coordinates in the search space which are associated with the best solution it has achieved so far and this value is called *pbest*. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population [19]. This location is called *gbest*.

The process for implementing the global version of PSO is given by the following steps.

Step 1. Initialize a population (array) of particles with random positions and velocities in the D -dimensional problem space. For a D -dimensional problem with N number of particles, the position vector X_i^t and velocity vector V_i^t are represented as

$$\begin{aligned} X_i^t &= [X_{i,1}^t, X_{i,2}^t, \dots, X_{i,D}^t], \\ V_i^t &= [V_{i,1}^t, V_{i,2}^t, \dots, V_{i,D}^t], \end{aligned} \quad (1)$$

where $i = 1, 2, \dots, N$.

Step 2. For each particle, evaluate the desired optimization fitness function in D variables.

Step 3. Compare each particle’s fitness evaluation with the particle’s *pbest*. If the current value is better than *pbest*, then set the *pbest* value equal to the current value and the *pbest* location equal to the current location in D -dimensional space.

Step 4. Compare the fitness evaluation with the population’s overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle’s array index and value.

Step 5. Update the velocity and position of the particle according to (2) and (3), respectively. One has

$$\begin{aligned} V_{i,d}^{t+1} &= V_{i,d}^t + c_1 r_1 (pbest_{i,d}^t - X_{i,d}^t) \\ &\quad + c_2 r_2 (gbest_d^t - X_{i,d}^t), \end{aligned} \quad (2)$$

$$X_{i,d}^{t+1} = X_{i,d}^t + V_{i,d}^{t+1}, \quad (3)$$

where c_1 and c_2 are two positive constants, known as the cognitive and social coefficients, which control the relative proportion of cognition and social interaction, respectively, and the values of c_1 and c_2 were decreased with each iteration [20]. r_1 and r_2 are two random values in the range $[0, 1]$. $V_{i,d}^t$, $X_{i,d}^t$, and $pbest_{i,d}^t$ are the velocity, position, and the personal best of i th particle in d th dimension for the t th iteration, respectively. The $gbest_d^t$ is the d th dimension of best particle in the swarm for the t th iteration.

Step 6. Loop to Step 2 until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iteration generations.

2.2. Quantum-Behaved Particle Swarm Optimization. Motivated by concepts in quantum mechanics and particle swarm optimization, Sun et al. proposed a new version of PSO, quantum-behaved particle swarm optimization (QPSO) [21]. In the QPSO, the state of a particle is depicted by a wave function $\psi(x, t)$, instead of position and velocity. The probability density function of the particle's position is $|\psi(X, t)|^2$ in position X_i [22].

Assume that, at iteration t , particle i moves in D -dimensional space with a δ potential well centered at $pbest_{i,j}^t$ on the d th dimension. The wave function at iteration $t + 1$ is given by the following equation:

$$\psi(X_{i,d}^{t+1}) = \frac{1}{\sqrt{L_{ij}^t}} \exp\left(-\frac{|X_{i,d}^t - pbest_{i,d}^t|}{L_{ij}^t}\right), \quad (4)$$

where $L_{i,d}^t$ is the standard deviation of the double exponential distribution, varying with iteration number t . Hence the probability density function Q is defined as

$$\begin{aligned} Q(X_{i,d}^{t+1}) &= |\psi(X_{i,d}^{t+1})|^2 \\ &= \frac{1}{L_{i,d}^t} \exp\left(-\frac{2|X_{i,d}^{t+1} - pbest_{i,d}^t|}{L_{i,d}^t}\right) \end{aligned} \quad (5)$$

and the probability distribution function F is given by the following equation:

$$F(X_{i,d}^{t+1}) = 1 - \exp\left(-\frac{2|X_{i,d}^{t+1} - pbest_{i,d}^t|}{L_{i,d}^t}\right). \quad (6)$$

By using Monte-Carlo method, the j th component of position X_i at iteration $t + 1$ can obtain by the following equation:

$$X_{i,d}^{t+1} = pbest_{i,d}^t \pm \frac{1}{2} L_{i,d}^t \ln\left(\frac{1}{u_{i,d}^{t+1}}\right), \quad (7)$$

where $u_{i,d}^{t+1}$ is a uniform random number in the interval $[0, 1]$. The value of $L_{i,d}^{t+1}$ is calculated as

$$L_{i,d}^t = 2\alpha |C_d^t - X_{i,d}^t|, \quad (8)$$

where parameter α is known as the contraction-expansion (CE) coefficient, which can be tuned to control the convergence speed of the algorithms [23]. C^t is the mean best position ($mbest$) and is defined as

$$\begin{aligned} C^t &= (C_1^t, C_2^t, \dots, C_D^t) \\ &= \left(\frac{1}{M} \sum_{i=1}^M pbest_{i,1}^t, \frac{1}{M} \sum_{i=1}^M pbest_{i,2}^t, \dots, \frac{1}{M} \sum_{i=1}^M pbest_{i,D}^t \right), \end{aligned} \quad (9)$$

where M is the size of the population. Hence the position of the particle is updated according to the following equation:

$$X_{i,d}^{t+1} = pbest_{i,d}^t \pm \alpha |C_d^t - X_{i,d}^t| \ln\left(\frac{1}{u_{i,d}^{t+1}}\right). \quad (10)$$

From (4) and (10), the new position of the particle is calculated as

$$\begin{aligned} X_{i,d}^{t+1} &= \begin{cases} pbest_{i,d}^t + \alpha |C_d^t - X_{i,d}^t| \ln\left(\frac{1}{u_{i,d}^{t+1}}\right) & k \in (0.5, 1], \\ pbest_{i,d}^t - \alpha |C_d^t - X_{i,d}^t| \ln\left(\frac{1}{u_{i,d}^{t+1}}\right) & k \in (0, 0.5], \end{cases} \end{aligned} \quad (11)$$

where k is a random number in the range $[0, 1]$. α is linearly decreasing factor from 1.0 to 0.3 with iteration as

$$\alpha^t = \alpha_{\max} - \frac{\alpha_{\max} - \alpha_{\min}}{t_{\max}} \cdot t, \quad (12)$$

where $t_{\max}$ is the maximum number iteration used in algorithm.

3. Fuzzy Wavelet Neural Network Based on QPSO

3.1. The Wavelet Base Function. In $L^2(R)$, a wavelet dictionary is constructed by dilating and translating from a wavelet base function $\psi(t)$ of zero average [24]:

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (13)$$

which is dilated with a scale parameter a and translated by b

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a > 0. \quad (14)$$

3.2. Fuzzy Wavelet Neural Network. The basic architecture of fuzzy wavelet neural network could be described as a set of Takagi-Sugeno models. Assume that there are r rules in the rule base and the Takagi-Sugeno fuzzy if-then rules are usually in the following form:

$$\begin{aligned} R^i: & \text{IF } x_1 \text{ is } F_1^i, \dots, x_r \text{ is } F_r^i, \\ & \text{THEN } y^i = \alpha_0^i + \alpha_1^i x_1 + \dots + \alpha_r^i x_r, \end{aligned} \quad (15)$$

where $x_1, x_2, \dots, x_r$ are input of T-S rule, F_j^i is the i th linguistic variable value of the j th input, which is a fuzzy set characterized by wavelet function. α_j^i is constant coefficients which are usually referred to as consequent parameters determined during the training process.

Figure 1 shows the architecture of the proposed FWNN modeling. The FWNN is a 4-layer feedforward network and detailed descriptions and equations for each layer are given here.

FIGURE 1: The architecture of FWNN.

Layer 1 (input variables layer). This layer is the input signals of the FWNN and each node of this layer, respectively, represents an input linguistic variable. The node output and the node input are related by

$$\begin{aligned} I_i^{(1)} &= x_i \\ O_i^{(1)} &= I_i^{(1)} \quad i = 1, 2, \dots, n, \end{aligned} \quad (16)$$

where $I_i^{(1)}$ and $O_i^{(1)}$ are, respectively, the input and output of i th node in Layer 1.

Layer 2 (membership functions layer). In this layer, nodes represent fuzzy sets in the antecedents of fuzzy rules. The outputs of this layer are the values of the membership functions. The membership function is wavelet function and is often taken as

$$\begin{aligned} u_i^j &= \varphi\left(\frac{x_i - b_i^j}{a_i^j}\right) \\ &= \cos\left(\frac{x_i - b_i^j}{2a_i^j}\right) \exp\left(-\frac{1}{2}\left(\frac{x_i - b_i^j}{a_i^j}\right)^2\right), \end{aligned} \quad (17)$$

where u_i^j is the j th membership function of x_i and $j = 1, 2, \dots, p$. a_i^j, b_i^j are the dilation and translation parameter of wavelet function.

In this layer, the relation between the output and input is represented as

$$\begin{aligned} I_{ij}^{(2)} &= O_i^{(1)}, \\ O_{ij}^{(2)} &= u_{ij}^j(I_{ij}^{(2)}) \\ &= \cos\left(\frac{O_i^{(1)} - b_i^j}{2a_i^j}\right) \exp\left(-\frac{1}{2}\left(\frac{O_i^{(1)} - b_i^j}{a_i^j}\right)^2\right), \end{aligned} \quad (18)$$

where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, p$.

Layer 3 (rule layer). In this layer, the number of rules is equal to the number of nodes. The output can be calculated as follows according to the AND (min) operation [12]:

$$O_{ij}^{(3)} = I_{ij}^{(3)} = \min(O_{1j}^{(2)}, O_{2j}^{(2)}, \dots, O_{nj}^{(2)}). \quad (19)$$

Layer 4 (output layer). This layer consists of output nodes. The output are given by

$$I_k^{(4)} = \sum_{i=1}^m \sum_{j=1}^p O_{ij}^{(3)} \omega_{ij}^{(3)}, \quad (20)$$

$$y_k = O_k^{(4)} = \frac{I_k^{(4)}}{\sum_{i=1}^m \sum_{j=1}^p O_{ij}^{(3)}}. \quad (21)$$

To train the parameters of FWNN, backpropagation (BP) training algorithm is extensively used as a powerful training method which can be applied to the forward network architecture [25]. For this purpose, mean square error (MSE) is selected as performance index which is given by

$$J = \frac{1}{2} (D - Y)^T (D - Y), \quad (22)$$

where D and Y are current and desired output values of network, correspondingly.

The all adjustable parameters of FWNN can be calculated by the following formulas:

$$\begin{aligned} \omega_{ij}^{(3)}(t+1) &= \omega_{ij}^{(3)}(t) - \eta \cdot \frac{\partial J}{\partial \omega_{ij}^{(3)}(t)} + \alpha \cdot \Delta \omega_{ij}^{(3)}(t), \\ b_{ij}^{(2)}(t+1) &= b_{ij}^{(2)}(t) - \eta \cdot \frac{\partial J}{\partial b_{ij}^{(2)}(t)} + \alpha \cdot \Delta b_{ij}^{(2)}(t), \\ a_{ij}^{(2)}(t+1) &= a_{ij}^{(2)}(t) - \eta \cdot \frac{\partial J}{\partial a_{ij}^{(2)}(t)} + \alpha \cdot \Delta a_{ij}^{(2)}(t), \end{aligned} \quad (23)$$

where t represents the backward step number and η and α are the learning and the momentum constants, differing in the ranges 0.01 to 0.1 and 0.1 to 0.9, respectively.

3.3. Fuzzy Wavelet Neural Network Trained by QPSO Algorithm. Computational intelligence has gained popularity in training of neural networks because of their ability to find a global solution in a multidimensional search space. The QPSO algorithm is a global algorithm, which has a strong ability to find global optimistic results and QPSO algorithm has proven to have advantages than the classical PSO due its less control parameters [26]. Therefore, by combining the QPSO with the fuzzy wavelet neural network, a new algorithm referred to as QPSO-FWNN algorithm is formulated in this paper.

When QPSO algorithm is used to train the FWNN model, a decision vector represents a particular group of network

parameters including the connection weight, the dilation and translation parameter. It is further denoted as

$$X_i = (a_{1,1}, \dots, a_{1,p}, \dots, a_{n,p}, b_{1,1}, \dots, b_{1,p}, \dots, b_{n,p}, \omega_{1,1}, \dots, \omega_{1,p}, \dots, \omega_{m,p}), \quad (24)$$

where a_{ij} and b_{ij} are the dilation and translation parameter of wavelet function in Layer 2. $\omega_{l,k}$ are the connection weight in (20). Since a component of the position corresponds to a network parameter, FWNN is structured according to the particle's position vector. Training the corresponding network by inputting the training samples, we can obtain an error value computed by (22). In a word, the mean square error is adopted as the objective function to be minimized in FWNN based on QPSO.

The specific procedure for the QPSO-FWNN algorithm can be summarized as follows.

Step 1. Define the structure of the FWNN according to the input and output sample.

Step 2. Treat the position vector of each particle as a group of network parameter by (24).

Step 3. Initialize the population by randomly generating the position vector X_i of each particle and set $pbest_i = X_i$.

Step 4. Evaluate m and α of QPSO algorithm using (9) and (12), respectively.

Step 5. Conclude the objective function of each particle by (22).

Step 6 (update $pbest$). Each particle's current fitness value is compared with previous best value $pbest$. If the current value is better than the $pbest$ value, then set the $pbest$ value to the current value.

Step 7 (update $gbest$). Determine the swarm best $gbest$ as minimum of all the particles $pbest$.

Step 8. Judge the stopping criteria, if the maximal iterative times are met, stop the iteration, and the positions of particles are the optimal solution. Otherwise, the procedure is repeated from Step 4.

4. Simulation Results

Experimental data set consists of 10 hours observations which comes from monitoring the traffic between clients in our campus network and servers. The minimal time interval in network traffic time series is 10 seconds. Figure 2 shows the normalization of network traffic time series.

In this paper, the design of a discrete filter predictor consists in finding the relation between the future data $x(n)$ and the past observations $x(n-1), x(n-2), \dots, x(n-M)$, where M is the number of considered input elements. The

FIGURE 2: The normalization of network traffic time series.

predictor relationship can be described by the following convolution sum [27]:

$$\hat{x}(n) = \sum_{k=1}^M h_o(k) x(n-k), \quad (25)$$

where $h_o(k)$ ($k = 1, 2, \dots, M$) is the filter coefficient vector. $x(n-k)$ is an k -step backward sample. $\hat{x}(n)$ is the desired output.

In order to test the performance of the prediction model, the front 1000 is the training data, and the latter 2100 is the prediction data. The number of time series windows was set as 3, which meant that the forth of measurement data would be predicted from the past three of measurement data. In the established prediction model base on QPSO-FWNN algorithm, the number of membership functions is five, the number of input variables layer nodes is three, and the number of output layer nodes is one. The population size of QPSO algorithm is 120 particles and D -dimensional search space of particle is 45. $t_{\max}$ is 30. $\alpha_{\max}$ is 1 and $\alpha_{\min}$ is 0.5. The CE coefficient decreases linearly from 1.0 to 0.3 during the search process according to (12) and After 50 times iterations, the cost function J of QPSO-FWNN neural network was 2.1887. The connection weight between input variables layer and output layer is given by

$$a = \begin{bmatrix} 1.018 & -0.196 & -0.582 & 0.419 & 0.743 \\ -1.299 & -0.885 & 0.676 & -0.170 & 0.134 \\ 1.082 & 0.206 & 0.557 & -0.408 & 0.453 \end{bmatrix},$$

$$b = \begin{bmatrix} 0.495 & 1.510 & 1.893 & 0.869 & 2.665 \\ 1.171 & -0.627 & 2.369 & 2.304 & 0.786 \\ -0.082 & 1.297 & 2.043 & 1.300 & 1.850 \end{bmatrix},$$

$$\omega = \begin{bmatrix} -0.927 \\ 0.408 \\ 0.053 \\ 2.128 \\ 2.195 \\ 1.652 \\ 1.875 \\ 1.436 \\ -1.537 \\ 2.416 \\ 1.595 \\ 1.146 \\ 2.072 \\ 0.965 \\ 1.884 \end{bmatrix}, \quad (26)$$

where a is the dilation parameter of the wavelet function. b is the translation parameter of wavelet function. ω is the connection weight of the output layer. Figure 3 shows the membership functions of input variable $x(t)$, $x(t-1)$, and $x(t-2)$ in FWNN units. Figure 4 shows the QPSO-FWNN convergence curves. These prediction results show that the QPSO-FWNN model is an effective, high-accuracy prediction model of network traffic in Figure 5.

5. Performance Metrics

In order to evaluate the prediction model more comprehensively, the following performance metrics are used to estimate the prediction accuracy.

(1) MSE (mean square error) is a scale-dependent metric which quantifies the difference between the predicted values and the actual values of the quantity being predicted by computing the average sum of squared errors [28]:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (27)$$

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

(2) NMSE (Normalized Mean Square Error) is defined as

$$\text{NMSE} = \frac{1}{\sigma^2} \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (28)$$

where σ^2 denotes the variance of the actual values during the prediction interval and is given as follows:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2, \quad (29)$$

where $\bar{y}$ is the mean value and is given by

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i. \quad (30)$$

It can be seen that, if $\text{NMSE} = 0$, the prediction performance is perfect, and if $\text{NMSE} = 1$, the prediction is a trivial predictor which statistically predicts the mean of the actual value. If $\text{NMSE} > 1$, it means that the prediction performance is worse than that of trivial predictor [29].

(3) MAPE (Mean Absolute Percentage Error) can calculate the prediction error as a percentage of the actual value. MAPE is defined as

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \times 100\%. \quad (31)$$

(4) Coefficient correlation is the covariance of the two variables divided by the product of their individual standard deviations. It is a normalized measurement of how the two variables are linearly related. The coefficient of correlation (r) is given as follows:

$$r = \frac{\text{COV}(Y, \hat{Y})}{\sigma_Y \sigma_{\hat{Y}}}, \quad (32)$$

where σ_Y and $\sigma_{\hat{Y}}$ indicate the standard deviation of the actual and the predicted values are given by (33)

$$\sigma_Y = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}. \quad (33)$$

$\text{COV}(X, Y)$ is the covariance between X and Y . It is obtained as follows:

$$\text{COV} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}). \quad (34)$$

Values for the correlation coefficient range are $[-1, 1]$. If $r = 1$, there is a perfect positive correlation between the actual and the predicted values, whereas $r = -1$ indicates a perfect negative correlation. If $r = 0$, we have a complete lack of correlation among the datasets.

(5) Coefficient of efficiency (CE) is defined as

$$\text{CE} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (35)$$

where the domain of the efficiency coefficient is $(-\infty, 1]$. If $\text{CE} = 1$, there is a perfect fit between the observed and the predicted data. When the prediction corresponds to estimating the mean of the actual values, $\text{CE} = 0$. If $\text{CE} \in (-\infty, 0]$, it indicates that the average of the actual values is a better predictor than the analyzed prediction method. The closer CE is to 1, the more accurate the prediction is.

In order to test the QPSO-FWNN method's validity and accuracy, we carried out the experiment which is compared with the other methods. The prediction model based on BP

FIGURE 3: The membership functions of input variable $x(t)$, $x(t-1)$, and $x(t-2)$ in FWNN units.

FIGURE 4: The QPSO-FWNN convergence curves.

TABLE 1: Performance comparison of the six prediction methods.

Method	MSE	NMSE	MAPE	r	CE
QPSO-FWNN	3.02×10^{-4}	0.0348	0.0181	0.9833	0.9667
BP	6.30×10^{-3}	0.3022	0.0986	0.8602	0.3091
RBF	4.20×10^{-3}	0.2182	0.0786	0.9115	0.5424
FNN	2.69×10^{-1}	1.0999	0.2041	-0.1692	-28.6128
FWNN-GA	1.01×10^{-3}	0.1475	0.0434	0.9560	0.8884
ARIMA	4.50×10^{-3}	0.5974	0.7326	0.7326	0.5011

neural network is 3 layers, in which the number of input layer nodes is 3, the number of hidden layer nodes is 7, the number of output layer nodes is 1, and the number of iterations is 1000. In the prediction model based on RBF neural network, the radial based distribute function “spread” is 1.5. The prediction model based on FNN neural network is 4 layers and the number of membership functions is five. The number of iterations is 1000. The architecture of prediction model based on fuzzy wavelet neural network and genetic algorithm (FWNN-GA) is the same with the QPSO-FWNN method. The population size of GA is 100. The crossover type

is one-point crossover, and crossover rate is 0.6. Mutation rate is 0.01 and the number of iterations is 500 [30]. The prediction model based on ARIMA model is built. The estimation of the model parameters is done using Maximum Likelihood Estimation and the best model is chosen as ARIMA (5, 1, 3) [31]. Figures 6, 7, 8, 9, and 10 show that prediction results with BP, RBF, FNN, and FWNN-GA neural network and ARIMA model, respectively. The performance comparison of the four prediction methods is shown in Table 1. From Table 1, one can look further into the prediction performance among the four prediction models. By comparing the value of MSE, NMSE, MAPE, coefficient correlation, and CE, QPSO-FWNN demonstrates better prediction accuracy than the other three methods. Therefore, the experimental results in this section show that the prediction method based on QPSO-FWNN is much more effective than BP, RBF, FNN, FWNN-GA, and ARIMA. It can be seen that the prediction method based on QPSO-FWNN is a better method to predict the time series of network traffic.

In order to test the prediction stability of each mode, the five prediction methods were predicted 10 times, respectively. Figures 11, 12, and 13 show that prediction results with QPSO-FWNN models are much more stable than BP, FNN, and FWNN-GA.

6. Conclusion

Predicting the direction of movements of network traffic is important as they enable us to detect potential network traffic jam spots. Since the network traffic is affected by many factors, the data of network traffic have the volatility and self-similarity features and the network traffic prediction becomes a challenge task. In this paper, the QPSO-FWNN method has been presented to predict the network traffic. The QPSO-FWNN combines the QPSO, which has the merit of powerful global exploration capability, with FWNN which can extract the mapping relations between the input and output data. The parameters of FWNN neural network are obtained by quantum-behaved particle swarm optimization (QPSO) and the time series of network traffic data was modeled by QPSO-FWNN. Finally, experiments showed that QPSO-FWNN model has faster and better performance in

FIGURE 5: Prediction with QPSO-FWNN model.

FIGURE 6: Prediction results with BP neural network.

FIGURE 7: Prediction results with RBF neural network.

FIGURE 8: Prediction results with FNN neural network.

FIGURE 9: Prediction results with FWNN-GA method.

FIGURE 10: Prediction results with ARIMA method.

FIGURE 11: MAPE and NMSE histogram of four prediction models.

FIGURE 12: MSE and CE histogram of four prediction models.

FIGURE 13: Coefficient correlation histogram of four prediction models.

prediction of nonlinear and nonstationary time series than many pure neural networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was partially supported by the National Natural Science Foundation of China (no. 11261001).

References

- [1] H. Mahmassani, "Dynamic network traffic assignment and simulation methodology for advanced system management applications," *Networks and Spatial Economics*, vol. 1, no. 3, pp. 267–292, 2001.
- [2] A. Nogueira, P. Salvador, R. Valadas et al., "Markovian modelling of internet traffic," in *Network Performance Engineering*, vol. 5233 of *Lecture Notes in Computer Science*, pp. 98–124, Springer, Berlin, Germany, 2011.
- [3] S. Y. Chang and H.-C. Wu, "Novel fast computation algorithm of the second-order statistics for autoregressive moving-average processes," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 526–535, 2009.
- [4] M.-C. Tan, S. C. Wong, J.-M. Xu, Z.-R. Guan, and P. Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 60–69, 2009.
- [5] Y. Shu, Z. Jin, L. Zhang, L. Wang, and O. W. W. Yang, "Traffic prediction using FARIMA models," in *Proceedings of the IEEE International Conference on Communications (ICC '99)*, vol. 2, pp. 891–895, IEEE, Vancouver, Canada, June 1999.
- [6] R. Li, J.-Y. Chen, Y.-J. Liu, and Z.-K. Wang, "WPANFIS: combine fuzzy neural network with multiresolution for network traffic prediction," *Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 4, pp. 88–93, 2010.
- [7] K. Zhao, "Global robust exponential synchronization of BAM recurrent FNNs with infinite distributed delays and diffusion terms on time scales," *Advances in Difference Equations*, vol. 2014, article 317, 2014.
- [8] E. A. Rying, G. L. Bilbro, and J.-C. Lu, "Focused local learning with wavelet neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 304–319, 2002.
- [9] D. Lin, X. Wang, F. Nian, and Y. Zhang, "Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems," *Neurocomputing*, vol. 73, no. 16–18, pp. 2873–2881, 2010.
- [10] R. H. Abiyev and O. Kaynak, "Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 8, pp. 3133–3140, 2008.
- [11] H. Iyatomi and M. Hagiwara, "Adaptive fuzzy inference neural network," *Pattern Recognition*, vol. 37, no. 10, pp. 2049–2057, 2004.
- [12] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000.
- [13] W. Li and Y. Hori, "An algorithm for extracting fuzzy rules based on RBF neural network," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1269–1276, 2006.
- [14] G. K. Venayagamoorthy, "Online design of an echo state network based wide area monitor for a multimachine power system," *Neural Networks*, vol. 20, no. 3, pp. 404–413, 2007.
- [15] W. Fang, J. Sun, and W. Xu, "A new mutated quantum-behaved particle swarm optimizer for digital IIR filter design," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 367465, 7 pages, 2009.
- [16] L. dos Santos Coelho and V. C. Mariani, "Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects," *Energy Conversion and Management*, vol. 49, no. 11, pp. 3080–3085, 2008.
- [17] L. D. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1676–1683, 2010.
- [18] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 685175, 10 pages, 2008.
- [19] R. C. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementations*, Elsevier, Philadelphia, Pa, USA, 2009.
- [20] S. L. Sabat, L. dos Santos Coelho, and A. Abraham, "MES-FET DC model parameter extraction using Quantum Particle Swarm Optimization," *Microelectronics Reliability*, vol. 49, no. 6, pp. 660–666, 2009.
- [21] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '04)*, pp. 325–331, Portland, Ore, USA, June 2004.
- [22] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [23] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu, "Quantum-behaved particle swarm optimization with Gaussian distributed

- local attractor point,” *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3763–3775, 2011.
- [24] S. Mallate, *A Wavelet Tour of Signal Processing: The Sparse Way*, Elsevier, Philadelphia, Pa, USA, 2009.
- [25] F.-J. Lin, C.-H. Lin, and P.-H. Shen, “Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive,” *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 751–759, 2001.
- [26] S. M. Mikki and A. A. Kishk, “Quantum particle swarm optimization for electromagnetics,” *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 10, pp. 2764–2775, 2006.
- [27] F. H. T. Vieira, G. R. Bianchi, and L. L. Lee, “A network traffic prediction approach based on multifractal modeling,” *Journal of High Speed Networks*, vol. 17, no. 2, pp. 83–96, 2010.
- [28] M. Barabas, G. Boanea, A. B. Rus, V. Dobrota, and J. Domingo-Pascual, “Evaluation of network traffic prediction based on neural networks with multi-task learning and multiresolution decomposition,” in *Proceedings of the IEEE 7th International Conference on Intelligent Computer Communication and Processing (ICCP '11)*, pp. 95–102, Cluj-Napoca, Romania, August 2011.
- [29] Y. Liang, “Real-time VBR video traffic prediction for dynamic bandwidth allocation,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 34, no. 1, pp. 32–47, 2004.
- [30] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, “Hybrid taguchi-genetic algorithm for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 365–377, 2004.
- [31] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, “Long-term forecasting of Internet backbone traffic,” *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1110–1124, 2005.