

Fuzzy wavelet neural network based on fuzzy clustering and gradient techniques for time series prediction

Rahib H. Abiyev

Received: 25 August 2009 / Accepted: 7 June 2010 / Published online: 26 June 2010
© Springer-Verlag London Limited 2010

Abstract This paper presents the development of fuzzy wavelet neural network system for time series prediction that combines the advantages of fuzzy systems and wavelet neural network. The structure of fuzzy wavelet neural network (FWNN) is proposed, and its learning algorithm is derived. The proposed network is constructed on the base of a set of TSK fuzzy rules that includes a wavelet function in the consequent part of each rule. A fuzzy c -means clustering algorithm is implemented to generate the rules, that is the structure of FWNN prediction model, automatically, and the gradient-learning algorithm is used for parameter identification. The use of fuzzy c -means clustering algorithm with the gradient algorithm allows to improve convergence of learning algorithm. FWNN is used for modeling and prediction of complex time series and prediction of foreign-exchange rates. Exchange rates are dynamic process that changes every day and have high-order nonlinearity. The statistical data for the last 2 years are used for the development of FWNN prediction model. Effectiveness of the proposed system is evaluated with the results obtained from the simulation of FWNN-based systems and with the comparative simulation results of previous related models.

Keywords Time series prediction · Fuzzy wavelet neural network · Prediction of exchange rates · Neuro-fuzzy modeling

1 Introduction

Time series prediction is one of the important research and application areas. The prediction results of time series can be applied to different areas, such as business, engineering, economics, weather and stock market forecasting, inventory and production control, signal processing, and many other fields. Exchange rates are among the most important economic indices in the international monetary markets and play an important role in controlling dynamics of exchange market. The large business companies, which are engaged to the trading, use foreign investment and make currency transfers in the course of business. Exchange rates are affected by many economic, political, and even psychological factors. Exchange rate series is characterized by complexity, volatility, and unpredictability. Many times exchange rate series is high-order nonlinear. Accurate forecasting of exchange rate movements can result in substantial improvement in the firm's overall profitability and very important for the success of many business and fund managers. Numerous techniques have been developed to explore this nonlinearity and improve the accuracy of prediction of exchange rate. These are well-known Box–Jenkins method [1], autoregressive random variance (ARV) model [2], autoregressive conditional heteroscedasticity (ARCH) [3], and general autoregressive conditional heteroscedasticity (GARCH) [4]. While these models may be good for a particular situation, they does not give satisfactory results for the nonlinear time series [5]. The applied traditional methods used for prediction are based on technical analysis of time series, such as looking for trends, stationarity, seasonality, random noise variation, moving average. Most of them are linear approaches that have shortcomings.

Hence, the idea of applying nonlinear models, like soft-computing technologies, such as neural networks, fuzzy

R. H. Abiyev (✉)
Near East University, Lefkosa,
North Cyprus, Mersin-10, Turkey
e-mail: rahib@neu.edu.tr

systems, genetic algorithms, has become important for time series prediction. These methods have shown clear advantages over the traditional statistical ones [6, 7]. Recently, neural networks [8–11], radial-based networks [12–14], and neuro-fuzzy networks [15, 16] are widely used for time series forecasting. In [17], it was found that neural networks are better than random walk models in predicting exchange rates. In [18, 19], a multilayer perceptron network is applied to predict the exchange rate between USD and DEM. In [20–22], different neural network structures are used to forecast exchange rates.

In this paper, to increase prediction accuracy and reduce search space and time for achieving the optimal solution, the combination of soft-computing technologies, such as wavelet neural networks with a fuzzy knowledge base is used for time series prediction, in particular, for the prediction of exchange rate between USA Dollars and Turkish Lira (USA/TL).

Fuzzy technology is an effective tool for dealing with complex, nonlinear processes characterized with ill-defined and uncertain factors. Traditionally, to develop a fuzzy system, human experts often carry out the generation of IF–THEN rules by expressing their knowledge. In the case of complicated processes, it is difficult for human experts to test all the input–output data and find necessary rules for the fuzzy system. To solve this problem and simplify the generating of IF–THEN rules, several approaches have been applied [23, 24]. Nowadays, the use of neural networks (NNs) takes more importance for this purpose. In this paper, the integration of NN and wavelet function is considered. Wavelet function is a waveform that has limited duration and an average value of zero. The integration of the localization properties of wavelets and the learning abilities of NN shows advantages of wavelet neural networks (WNNs) over NN in complex nonlinear system modeling. A WNN that uses wavelet functions has been proposed by researchers for solving approximation and classification problems [25–28]. Wavelet neural networks are used for the prediction of chaotic time series [29, 30], for short term and long-term prediction of electricity load [31, 32]. The wavelet analysis approximates the decomposed time series at different levels of resolution. Fuzzy wavelet neural network (FWNN) combines wavelet theory, fuzzy logic, and neural networks. The synthesis of fuzzy wavelet neural inference system includes the finding of the optimal definitions of the premise and consequent part of fuzzy IF–THEN rules through the training capability of wavelet neural networks, evaluating the error response of the system. A combination of fuzzy technology and WNN has been considered for solving signal-processing and control problems [33–37]. Wavelet network model of fuzzy inference system [33, 34], fuzzy systems with linear combination of the basis function [35, 36] are proposed.

Thuillard [33] proposed to choose the membership functions from the family of scaling functions and to construct the fuzzy system using wavelet techniques. Fuzzy wavelet network that includes combinations of three subnets: pattern recognition subnet, fuzzy reasoning subnet, and control synthesis subnet is introduced [36]. The FWNN structure that is constructed on the base of a set of fuzzy rules is proposed and used for approximating nonlinear functions [38]. The FWNN-based controller is developed for the control of dynamic plants [39, 40] and prediction of electricity consumption [41, 42]. The combination of wavelet network and fuzzy logic allows us to develop a system that has fast training speed and to describe nonlinear objects that are characterized with uncertainty. Wavelet transform has the ability to analyze nonstationary signals to discover their local details. Fuzzy logic allows us to reduce the complexity of the data and to deal with uncertainty. Neural network has a self-learning characteristic that increases the accuracy of the prediction. In this paper, these methodologies are used to construct fuzzy wavelet neural inference system to solve exchange rates prediction problem.

During design of FWNN system, one of the important problems is its learning and convergence. Recently, a number of different approaches have been used for the design of fuzzy neural network systems. These are the clustering techniques [43–47], the table look-up scheme [48, 49], the least-squares method (LSM) [24, 47], gradient algorithms [24, 39–42], and genetic algorithms [24, 42, 50]. Abiyev [42] uses combination of gradient and genetic algorithms for FWNN design. Genetic algorithms can find global optimal solution of the problem but needs more time for parameter updating. In this paper, to decrease learning time, the fuzzy clustering with the gradient algorithm is applied to design the FWNN prediction system, which will be demonstrated in Sect. 3.

The paper is organized as follows: Sect. 2 presents the structure of FWNN prediction model. Section 3 presents parameter update rules of the FWNN system. The descriptions of the fuzzy *c*-means clustering algorithm and gradient descent algorithm for learning of FWNN are given. Section 4 contains simulation results of the FWNN used for prediction of chaotic time series. The application of developed structure is used for prediction of exchange rates. Comparative results of different models for time series prediction are given. Finally, a brief conclusion is presented in Sect. 5.

2 Fuzzy wavelet neural network

The kernel of a fuzzy inference system is its fuzzy knowledge base. In a fuzzy knowledge base, the information that

consists of input–output data points of the system is interpreted into linguistically interpretable fuzzy rules that have IF–THEN form. Fuzzy systems are generally designed using either Mamdani or Takagi–Sugeno–Kanag (TSK) type IF–THEN rules. In the former type, both the antecedent and the consequent parts utilize fuzzy values. The TSK type fuzzy rules utilize fuzzy values in the antecedent part, crisp values or often linear functions in the consequent part. In many research works, it has been shown that TSK type fuzzy neural systems can achieve a better performance than the Mamdani type fuzzy neural systems in learning accuracy [24, 51]. The theory and design methodologies of Mamdani and TSK fuzzy systems are presented in [23, 24, 50–52]. This paper presents fuzzy wavelet neural network that integrates wavelet functions with the TSK fuzzy model. The consequent parts of TSK type fuzzy IF–THEN rules are represented by either a constant or a function. As a function, most of the fuzzy and neuro-fuzzy models use linear functions. Neuro-fuzzy systems can describe the considered problem by means of combination of linear functions. Sometimes these systems need more rules for modeling complex nonlinear processes in order to obtain the desired accuracy. Increasing the number of the rules leads to increasing number of neurons in the hidden layer of the network. To improve the computational power of the neuro-fuzzy system, we use wavelets in the consequent part of each rule. In this paper, the fuzzy rules that are constructed by using wavelets are used. They have the following form.

$$\begin{aligned}
 &\text{If } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{12} \text{ and } \dots \text{ and } x_m \text{ is } A_{1m}, \\
 &\text{then } y_1 \text{ is } \sum_{i=1}^m w_{i1} (1 - z_{i1}^2) e^{-\frac{z_{i1}^2}{2}} \\
 &\text{If } x_1 \text{ is } A_{21} \text{ and } x_2 \text{ is } A_{22} \text{ and } \dots \text{ and } x_m \text{ is } A_{2m}, \\
 &\text{then } y_2 \text{ is } \sum_{i=1}^m w_{i2} (1 - z_{i2}^2) e^{-\frac{z_{i2}^2}{2}} \\
 &\dots \\
 &\text{If } x_1 \text{ is } A_{n1} \text{ and } x_2 \text{ is } A_{n2} \text{ and } \dots \text{ and } x_m \text{ is } A_{nm}, \\
 &\text{then } y_n \text{ is } \sum_{i=1}^m w_{in} (1 - z_{in}^2) e^{-\frac{z_{in}^2}{2}} \tag{1}
 \end{aligned}$$

Here, x_1, x_2, \dots, x_m are input variables, y_1, y_2, \dots, y_n are output variables that include Mexican Hat wavelet functions, A_{ij} is a membership function for the i -th rule of the j -th input defined as Gaussian membership function. N is number of fuzzy rules. Conclusion parts of rules contain wavelet neural networks (WNNs). The WNNs include wavelet function. Wavelets are defined in the following form

$$\Psi_j(x) = \frac{1}{\sqrt{|a_j|}} \psi\left(\frac{x - b_j}{a_j}\right), \quad a_j \neq 0, \quad j = 1, \dots, n \tag{2}$$

$\Psi_j(x)$ represents the family of wavelet obtained from the single $\Psi(x)$ function by dilations and translations, where $a_j = \{a_{1j}, a_{2j}, \dots, a_{mj}\}$ and $b_j = \{b_{1j}, b_{2j}, \dots, b_{mj}\}$ are the dilation and translation parameters, respectively. $x = \{x_1, x_2, \dots, x_m\}$ are input signals. $\Psi(x)$ is localized in both time space and frequency space and is called a mother wavelet. The output of WNN is calculated as

$$y = \sum_{j=1}^k w_j \Psi_j(x) = \sum_{j=1}^k w_j |a_j|^{-\frac{1}{2}} \psi(a_j^{-1}x - d_j) \tag{3}$$

Here, $d_j = a_j^{-1} * b_j$. $\Psi_j(x)$ is the wavelet function of j -th unit of the hidden layer, w_j are weight coefficients between input and hidden layers, a_i and b_j are parameters of the wavelet function. Wavelet networks include wavelet functions in the neurons of hidden layer of network. WNN has good generalization ability, can approximate complex functions to some precision very compactly, and can be easily trained than other networks, such as multilayer perceptrons and radial-based networks [32, 53]. A good initialization of the parameters of WNNs allows to obtain fast convergence. A number of methods are implemented for initializing wavelets, such as orthogonal least square procedure, clustering method [25, 26]. The optimal dilation and translation of the wavelet increases training speed and obtains fast convergence. The approximation and convergence properties of WNN are presented in [27].

In formula (1), fuzzy rules provide the influence of each WNN to the output of FWNN. The use of WNN with different dilation and translation values allows to capture different behaviors and essential features of the nonlinear model under these fuzzy rules. The proper fuzzy model that is described by IF–THEN rules can be obtained by learning-dilation and translation parameters of conclusion parts and the parameters of membership function of premise parts. Here, because of the use of wavelets, the computational strength and generalization ability of FWNN are improved, and FWNN can describe the nonlinear processes with desired accuracy.

The structure of fuzzy wavelet system is given in Fig. 1. The fuzzy wavelet neural network includes six layers. In the first layer, the number of nodes is equal to the number of input signals. These nodes are used for distributing input signals. In the second layer, each node corresponds to one linguistic term. For each input signal entering to the system, the membership degree to which input value belongs to a fuzzy set is calculated. To describe linguistic terms, the Gaussian membership function is used.

$$\eta_j(x_i) = e^{-\frac{(x_i - c_{ij})^2}{2\sigma_{ij}^2}} \quad i = 1, \dots, m, \quad j = 1, \dots, n \tag{4}$$

Here, m is the number of input signals, n is the number of fuzzy rules (hidden neurons in third layer). c_{ij} and σ_{ij} are

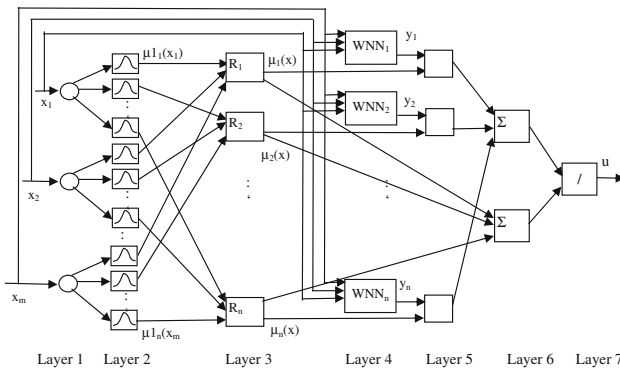


Fig. 1 Structure of Fuzzy wavelet neural network

center and width of the Gaussian membership functions, respectively. $\eta_j(x_i)$ is membership function of i -th input variable for j -th term.

In the third layer, the number of nodes corresponds to the number of rules R_1, R_2, \dots, R_n . Each node represents one fuzzy rule. The number of fuzzy rules and the number of membership functions are determined using clustering algorithm which is described Sect. 3. The third layer realizes the inference engine. In this layer, the t -norm prod operator is applied to calculate the membership degree of the given input signals for each rule.

$$\mu_j(x) = \eta_j(x_1) * \eta_j(x_2) * \dots * \eta_j(x_m), \quad j = 1, \dots, n \quad (5)$$

Here, $*$ is t -norm prod operator.

These $\mu_j(x)$ signals are input signals for the next layer. This layer is a consequent layer. It includes n wavelet neural networks that are denoted by $WNN_1, WNN_2, \dots, WNN_n$. In the fifth layer, the output signals of third layer are multiplied by the output signals of wavelet networks. The output of j -th wavelet network is calculated as

$$y_j = w_j \Psi_j(z); \quad \Psi_j(z) = \sum_{i=1}^m \frac{1}{\sqrt{|a_{ij}|}} (1 - z_{ij}^2) e^{-\frac{z_{ij}^2}{2}} \quad (6)$$

Here, $z_{ij} = \frac{x_i - b_{ij}}{a_{ij}}$. Here, a_{ij} and b_{ij} are parameters of the wavelet function between i -th ($i = 1, \dots, n$) input and j -th ($j = 1, \dots, m$) WNN. In sixth and seventh layers, defuzzification is made to calculate the output of whole network. In this layer, the contribution of each WNN to the output of the FWNN is determined.

$$u = \frac{\sum_{j=1}^n \mu_j(x) y_j}{\sum_{j=1}^n \mu_j(x)} \quad (7)$$

Here, y_j are the output signals of wavelet neural networks.

After calculating the output signal of the FWNN, the training of the network is started. The training includes the adjustment of the parameters of the membership functions c_{ij} and σ_{ij} in the second layer and the parameters of the wavelet functions $w_j, a_{ij}, b_{ij}, (i = 1, \dots, m, j = 1, \dots, n)$ of

network in the fourth layer. In the next section, the learning of type-2 FWNN is derived.

3 Parameter update rules

The design of FWNN (Fig. 1) includes determination of the unknown parameters that are the parameters of the antecedent and the consequent parts of the fuzzy IF–THEN rules (1). In the antecedent parts, the input space is divided into a set of fuzzy regions, and in the consequent parts the system behavior in those regions is described. As mentioned earlier, recently, a number of different approaches have been used for designing fuzzy IF–THEN rules based on clustering [43–47], the table look-up scheme [48, 49], the least-squares method (LSM) [24, 47], gradient algorithms [24, 39–42], and genetic algorithms [24, 42, 50]. In [47], the use of LSM and clustering algorithm is used for updating parameters of the neuro-fuzzy inference system. In this paper, the fuzzy clustering is applied to design the antecedent (premise) parts, and the gradient algorithm is applied to design the consequent parts of the fuzzy rules. Fuzzy clustering is an efficient technique for constructing the antecedent structures. The aim of clustering methods is to identify a certain group of data from a large data set, such that a concise representation of the behavior of the system is produced. Each cluster center can be translated into a fuzzy rule for identifying the class. Different clustering algorithms are developed. For fuzzy systems, subtractive clustering [44, 46] and fuzzy c -means [43] clustering have been developed recently. Subtractive clustering [44] is an extension of the grid-based mountain clustering [45]. It is unsupervised clustering, in which the number of clusters for input data points is determined by the clustering algorithm. Sometimes we need to control the number of clusters in an input space. In these cases, the supervised clustering algorithms are of primary concern. Fuzzy c -means clustering is one of them. It can efficiently be used for fuzzy systems [43] with simple structure and sufficient accuracy. In this paper, the fuzzy c -means clustering technique is used for structuring the premise part of the fuzzy system.

At the first step, fuzzy c -means classification is applied in order to partition input data and construct antecedent part of fuzzy IF–THEN rules. Fuzzy c -means classification is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m d_{ji}^2, \quad \text{where } d_{ji} = \|x_i - c_j\|, \quad 1 \leq m < \infty \quad (8)$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i -th of d -dimensional measured data, c_j is the k -dimension center

of the cluster, and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown earlier, with the update of membership u_{ij} and the cluster centers c_j . The algorithm is composed of the following steps.

1. Initialize $U = [u_{ij}]$ matrix, $U^{(0)}$
2. Calculate the centers vectors $C^{(t)} = [c_j]$ with $U^{(t)}$

$$c_j = \left(\sum_{i=1}^N u_{ij}^m \cdot x_i \right) / \sum_{i=1}^N u_{ij}^m$$

3. Update $U^{(t)}, U^{(t+1)}$

$$u_{ij} = 1 / \sum_{k=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}$$

4. If $\{|U^{(t+1)} - U^{(t)}|\} < \varepsilon$ then stop; otherwise set $t = t + 1$ and return to step 2.

After partition, each cluster center will correspond to center of the membership function used in input layer of type-2 FWNN. The width of the membership function is determined using distance between cluster centers.

After the design of the antecedents parts by fuzzy clustering, the gradient descent algorithm is applied to design the consequent parts of the fuzzy rules. In what follows, the parameter update rules are derived for the FWNN. At first, the output error is calculated. At the beginning, the parameters of consequent parts FWNN are generated randomly. To generate a proper FWNN model, the training of the parameters has been carried out. In this paper, we applied gradient learning with adaptive learning rate. The adaptive learning rate guarantees the convergence and speeds up the learning of the network.

At first, on the output of network, the value of cost function is calculated.

$$E = \frac{1}{2} \sum_{i=1}^O (u_i^d - u_i)^2 \tag{9}$$

Here, O is the number of output signals of the network (in given case $O = 1$), u_i^d and u_i are desired and current output values of the network, respectively. Using gradient algorithm, the parameters $w_j, a_{ij}, b_{ij}, (i = 1, \dots, m, j = 1, \dots, n)$ of wavelet neural network and parameters of membership function c_{ij} and $\sigma_{ij} (i = 1, \dots, m, j = 1, \dots, n)$ of neuro-fuzzy structure are adjusted using the following formulas.

$$\begin{aligned} w_j(t+1) &= w_j(t) + \gamma \frac{\partial E}{\partial w_j}; & a_{ij}(t+1) &= a_{ij}(t) + \gamma \frac{\partial E}{\partial a_{ij}}; \\ b_{ij}(t+1) &= b_{ij}(t) + \gamma \frac{\partial E}{\partial b_{ij}}; \end{aligned} \tag{10}$$

$$c_{ij}(t+1) = c_{ij}(t) + \gamma \frac{\partial E}{\partial c_{ij}}, \quad \sigma_{ij}(t+1) = \sigma_{ij}(t) + \gamma \frac{\partial E}{\partial \sigma_{ij}}, \tag{11}$$

Here, γ is the learning rate, $i = 1, \dots, m, j = 1, \dots, n, m$ is the number of input signals of the network (input neurons), and n is the number of fuzzy rules (hidden neurons).

The values of derivatives in (10) are computed using the following formulas.

$$\begin{aligned} \frac{\partial E}{\partial w_j} &= \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_j} \frac{\partial y_j}{\partial w_j} = (u(t) - u^d(t)) \cdot \mu_j \cdot \psi_j(z) / \sum_{j=1}^n \mu_j \\ \frac{\partial E}{\partial a_{ij}} &= \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_j} \frac{\partial y_j}{\partial \psi_j} \frac{\partial \psi_j}{\partial z_{ij}} \frac{\partial z_{ij}}{\partial a_{ij}} = \delta_j (3.5 z_{ij}^2 - z_{ij}^4 - 0.5) e^{-\frac{z_{ij}^2}{2}} / \left(\sqrt{a_{ij}^3} \right), \\ \frac{\partial E}{\partial b_{ij}} &= \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_j} \frac{\partial y_j}{\partial \psi_j} \frac{\partial \psi_j}{\partial z_{ij}} \frac{\partial z_{ij}}{\partial b_{ij}} = \delta_j (3 z_{ij} - z_{ij}^3) e^{-\frac{z_{ij}^2}{2}} / \left(\sqrt{a_{ij}^3} \right) \end{aligned}$$

here $\delta_j = \frac{\partial E}{\partial u} \frac{\partial u}{\partial y_j} \frac{\partial y_j}{\partial \psi_j} = (u(t) - u^d(t)) \cdot \mu_j \cdot w_j / \sum_{j=1}^n \mu_j,$

$$i = 1, \dots, m, j = 1, \dots, n \tag{12}$$

The derivatives in (11) are determined by the following formulas.

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial E}{\partial u} \frac{\partial u}{\partial \mu_j} \frac{\partial \mu_j}{\partial \eta_{ij}} \frac{\partial \eta_{ij}}{\partial c_{ij}} \tag{13}$$

$$\frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial E}{\partial u} \frac{\partial u}{\partial \mu_j} \frac{\partial \mu_j}{\partial \eta_{ij}} \frac{\partial \eta_{ij}}{\partial \sigma_{ij}} \tag{14}$$

Here,

$$\frac{\partial E}{\partial u} = u(t) - u^d(t); \quad \frac{\partial u}{\partial \mu_j} = \frac{y_j - u}{\sum_{j=1}^n \mu_j}; \quad \frac{\partial \mu_j}{\partial \eta_{ij}} = \prod_{\substack{k=1 \\ k \neq i}}^{M1} \eta_{ik};$$

$$i = 1, \dots, m; j = 1, \dots, n; k = 1, \dots, m \tag{15}$$

\prod is t -norm product operator.

$$\frac{\partial \eta_j(x_i)}{\partial c_{ij}} = \eta_j(x_i) \frac{(x_i - c_{ij})}{\sigma_{ij}^2}; \quad \frac{\partial \eta_j(x_i)}{\partial \sigma_{ij}} = \eta_j(x_i) \frac{(x_i - c_{ij})^2}{\sigma_{ij}^3} \tag{16}$$

The parameters of the type-2 FWNN can thus be updated using (10, 11) together with (12–16).

One important problem in learning algorithms is convergence. The convergence of the gradient descent method depends on the selection of the initial values of the learning rate. Usually, this value is selected in the interval [0–1]. A large value of the learning rate may lead to unstable learning, a small value of the learning rate results in a slow learning speed. In this paper, an adaptive approach is used for updating these parameters. That is, the learning of the FWNN parameters is started with a small value of the

learning rate γ . During learning, γ is increased if the value of change of error $\Delta E = E(t) - E(t + 1)$ is positive, and decreased if negative. This strategy ensures a stable learning for the FWNN, guarantees the convergence, and speeds up the learning.

The optimal value of the learning rate for each time instance can be obtained using a Lyapunov function [54]. Let $\gamma(t)$ be the learning rate for the weights $W = [w_j, a_{ij}, b_{ij}, c_{ij}, \sigma_{ij}]$ of the FWNN, trained using (10) and (11). The convergence is guaranteed if the following condition is satisfied.

$$0 < \gamma(t) < \frac{2}{\left(\max_t \left\| \frac{\partial u(t)}{\partial W} \right\| \right)^2} \quad (17)$$

The derivation of this condition is given in [Appendix](#).

4 Simulation

4.1 Time series prediction

The FWNN structure and its learning algorithms are applied for modeling and predicting the future values of chaotic time series. As an example, Box and Jenkins gas furnace data (series J) [1] prediction problem is considered. The data set was taken from a combustion process of a machine—air mixture. This is a benchmark problem used for testing identification and prediction algorithms. The data set includes 296 pairs of input–output measurements. The input as $x(t)$ is the gas flow into the furnace, and the output as $y(t)$ is the concentration of carbon dioxide (CO₂) in outlet gas. The sampling interval is 9 s. Following previous researches, the inputs of the prediction model are selected as $x(t - 4)$ and $y(t - 1)$, and the output is $y(t)$.

Using statistical data, the training of FWNN has been carried out. The training input/output data for the prediction system will be a structure whose first component is the two-dimension input vector, and the second component is the predicted output.

The unknown parameters of fuzzy FWNN are the parameters of the membership functions of the second layer (σ and c) and the parameters of the wavelet function (a and b). The learning of the parameters is accomplished by using fuzzy c -means clustering and the gradient descent algorithms. In prediction, the clustering involves the determination of clusters in the input data space and the translation of these clusters into fuzzy rules. At first, the fuzzy clustering approach described in Sect. 2 is applied to the input data points of the plant in order to determine the cluster centers. In identification, the clustering involves the determination of clusters in data space for the input signals $x(t - 4)$ and $y(t - 1)$ and the translation of these clusters

into fuzzy rules such that the model obtained is close to the identified system. The obtained cluster centers are then used in order to organize the premise parts of the fuzzy rules. It is to be noted that with such an approach, the number of parameters to be determined for the antecedent part are reduced significantly. That is to say, with two cluster centers for each input, a total of four membership functions exist with 2 parameters each and thus having 8 parameters in the antecedent part of FWNN in total. Using these four membership functions, two for each input signal, four rules could be constructed. On the other hand, if the generation of the membership functions is done completely randomly, then 8 membership functions are needed for the 4 rules, and this would require the determination of 16 parameters.

After the determination of the parameters of the antecedent part, the gradient algorithm is applied for the learning of the parameters in the consequent part. These are the parameters of the wavelet functions of the layer 4. The initial values of a and b are selected randomly, in the interval $[-1, 1]$. Using the parameter update rules derived earlier, they are updated for the given input signals.

During the learning, the data points were partitioned in 200 data points as a training set, and the remaining 92 points as a test set for testing the performance of the evolved model. All data set is scaled between 0 and 1. The training of FWNN is carried out for 500 epoch.

Simulation was performed for two cases: when the number of clusters is 2 and 3. At first stage, the fuzzy rules are constructed using two clusters for each input variable and taking all possible combinations. Four fuzzy rules are constructed using different combination of clusters for two inputs. After clustering of input space gradient, decent algorithm is used for learning of consequent parts of the fuzzy rules, that is the parameters of wavelet function of 4th layer of the FWNN.

For comparative analysis, the obtained results are compared with existing online models applied to the same task. As a performance criteria, the root mean square error (RMSE) is used

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i^d - x_i)^2}$$

During learning, the value of RMSE was 0.014. After learning, in the generalization step, the value of RMSE was 0.019. Total number of parameters will be determined as 8(premise part) + 20(consequent part) = 28. Figure 2 depicts RMSE values obtained during training. In Fig. 3, the trajectories of desired and predicted values and the prediction error for both training and checking data are shown. Here, the solid line indicates the trajectory of statistical data, and the dashed line indicates the predicted

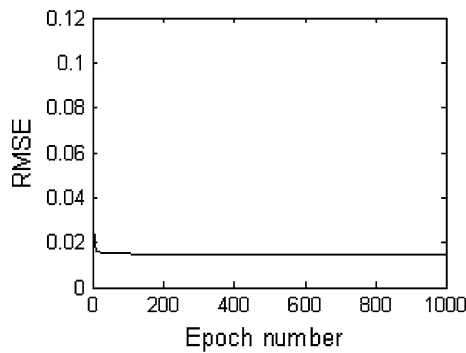


Fig. 2 RMSE values obtained during training

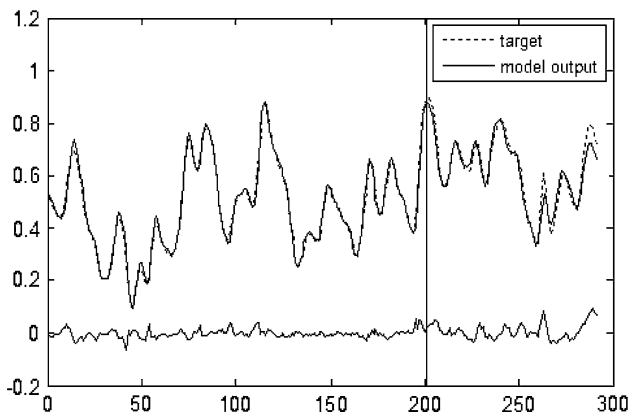


Fig. 3 Box–Jenkins time series data, model output and prediction error for training and test samples (first 200 points is used for training, second 92 data points for testing)

value of time series. Here, 200 data points is used as a training set, and the remaining 92 points as a test set. As shown in Fig. 3, the value of RMSE for training data (until iteration 200) is small than for checking data (after iteration 200). In second stage, the training of prediction model is carried out using 3 cluster centers for each input. Using the combination of these cluster centers, the 9 fuzzy rules were generated. Total number of parameters will be determined as 12(premise part) + 45(consequent part) = 57. After clustering input space gradient, decent algorithm is applied for learning of consequent parts of the fuzzy rules, that is parameters of wavelet function of 4th layer of FWNN. During learning, the values of RMSE for training

and testing data were 0.013 and 0.016, respectively. The results of simulations for different number of fuzzy rules are given in Table 1. In the paper, because of the approach used in learning, the computational time of FWNN system would be decreased. The parameters of the antecedent part are determined using a clustering technique that takes a very small time (0.15–0.2 s). What takes time is the gradient technique. That is the training time of the FWNN is basically the time required to learn the parameters of the consequent parts. Because of the fact that clustering takes less time than the gradient technique, the training time of a FWNN that uses clustering for the antecedent part and the gradient approach for the consequent part will be less than the training time of FWNN that uses the gradient technique for both parts with the same number of rules (Table 1).

Table 2 shows the comparison of test results of different models for Box–Jenkins data series. The papers [55–57] use reference fuzzy sets in premise parts; therefore, the total number of parameters are omitted and the direct comparison makes no sense.

4.2 Forecasting of exchange rates

The FWNN system is applied for constructing a prediction model of exchange rate USA/TL. Exchange rates play an important role in controlling dynamics of exchange market. Exchange rate series is characterized by complexity, volatility, and unpredictability. Exchange rate series is high-order nonlinear. Prediction of exchange rate is one of the important financial problems. Appropriate prediction of exchange rate is very important for the success of many business and fund managers.

The FWNN structure and its learning algorithm are used to construct the prediction model. In the prediction problem, it is needed to predict the value of exchange rate in the near future $x(t + pr)$ on the base of sample data points $\{x(t - (D - 1)\Delta), \dots, x(t - \Delta), \dots, x(t)\}$. Here, pr is the prediction step. Four input data points $[x(t - 3) x(t - 2) x(t - 1) x(t)]$ are used as input to the prediction model. The output training data corresponds to $x(t + 3)$. In other words, since the exchange rates is considered daily, the value that is to be predicted will be after $pr = 3$ day. The training input/output data for the prediction system will be a structure whose first component is the four dimension

Table 1 Prediction results for gas furnace data

Models	Number of rules	Number of parameters	RMSE		Computational time (s)	
			Train	Test	Train	Test
FWNN (with FCM and gradient learning)	4	28	0.014	0.019	40.029	0.0156
	9	57	0.013	0.016	53.820	0.0312
FWNN (with gradient learning)	4	36	0.0151	0.020	51.729	0.0156
	9	81	0.0133	0.0167	67.657	0.0312

Table 2 Comparisons of test results of different prediction models for Box–Jenkins time series

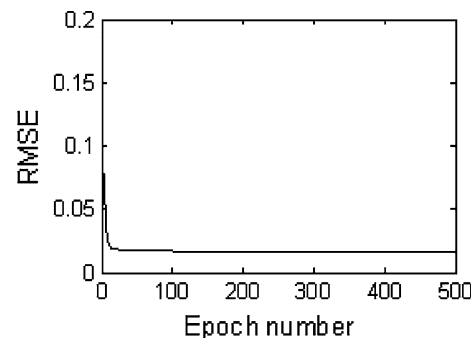
Method	Number of inputs	Number of rules	Number of parameters	Prediction error, RMSE
Box and Jenkins [1]	6	–	10	0.449
Tong [55]	2	19	–	0.685
Pedrycz [56]	2	81	–	0.566
Xu and Lu [57]	2	25	–	0.573
Sugeno and Yasukawa [58]	3	6	96	0.436
Sugeno and Tanaka [59]	6	2	17	0.261
Lin and Cunningham [60]	5	4	354	0.266
Kim et al. [61]	6	2	38	0.234
Kim et al. [62]	6	2	110	0.219
ANFIS model [24]	2	25	75	0.027
Kim and Kasabov [63]	2	15	>75	0.020
Neural tree model [7]	2	–	16	0.026
WNN	2	9	45	0.035
FWNN	2	4	28	0.019

input vector and the second component is the predicted output.

To start the training, the FWNN structure is generated. It includes four input and one output neurons. As before, fuzzy classification is applied in order to partition input space and select the parameters of the premise parts, that is the parameters of Gaussian membership functions used in the second layer of FWNN. Fuzzy *c*-means clustering is used for the input space with 2 clusters for each input. Sixteen fuzzy rules are constructed using different combination of these clusters for four inputs. After clustering input space gradient, decent algorithm is used for learning of consequent parts of the fuzzy rules, that is parameters of wavelet function of 4th layer of FWNN. The initial values of the parameters wavelet functions are randomly generated in the interval $[-1, 1]$ and, using the gradient algorithm derived earlier, they are updated for the given input–output training pairs. As a performance criterion RMSE is used.

For training of the system, the statistical data describing daily exchange rates from January 2007 to April 2009 are considered. The data set consists of 557 data. The 500 data are used for training, and next 50 data are used for diagnostic testing. All input and output data are scaled in the interval $[0, 1]$. The training is carried out for 500 epochs. The values of the parameters of the FWNN system were determined at the conclusion of training. Once the FWNN has been successfully trained, it is then used for the prediction of the daily exchange rates. During learning, the value of RMSE was 0.0164. After learning, for the test data the value of RMSE was 0.0226. Figure 4 depicts RMSE values obtained during training.

In Fig. 5, the output of the FWNN system for three-step ahead prediction of exchange rates for learning and

**Fig. 4** RMSE values obtained during training

generalization step is shown. Here, the solid line is desired output, and the dashed line is the FWNN output.

The plot of prediction error is shown in Fig. 6. As shown in the figure, in generalization step (end part of error curve), the value of error increases. Figure 7 demonstrates the three-step ahead prediction of FWNN for test data. The result of the simulation of the FWNN prediction model is compared with result of simulation of the NN-based prediction model. To estimate the performance of the neural and FWNN prediction systems, the RMSE values of errors between predicted and current output signal are compared.

In next experiment, $x(t + 6)$ six-step ahead prediction of exchange rates is performed. The data points $[x(t - 6) x(t - 2) x(t - 1) x(t)]$ are used as input for the system. The 500 data points are used for learning, and last 50 days are used for testing. In the result of learning, the parameters of FWNN were found. Training and test values of RMSE were 0.0228 and 0.0332, correspondingly.

The simulation results satisfy the efficiency of the application of FWNN technology in constructing a prediction model of exchange rates. In Table 3, the comparative

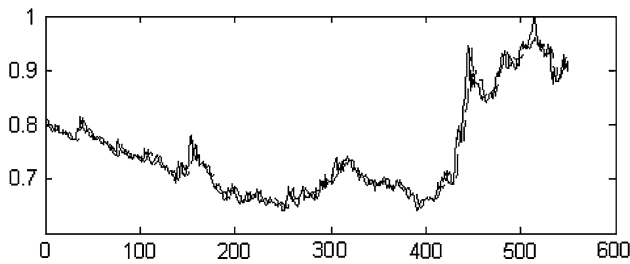


Fig. 5 Three-step ahead prediction. Plot of output signals generated by FWNN (dotted line) and predicted signal (solid line)

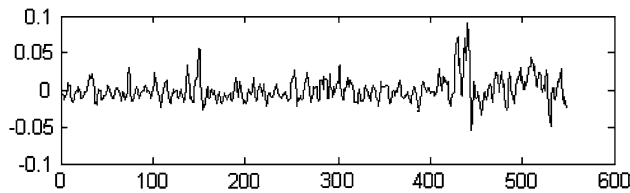


Fig. 6 Plot of prediction error

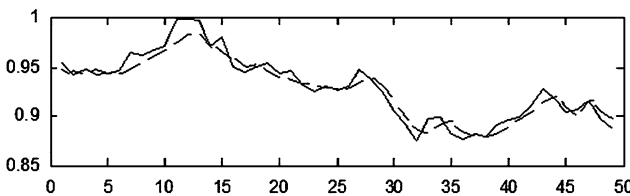


Fig. 7 Three-step ahead prediction. Curves describing testing data

results of simulations are given. As shown in the table, the performance of FWNN prediction is better than the performance of the NN model.

5 Conclusion

The time series prediction model is developed by integrating fuzzy logic, neural networks, and wavelet technology. The wavelet networks are used to construct the fuzzy rules, and the functionality of the fuzzy system is realized by the neural network structure. FWNN prediction model is constructed using fuzzy clustering and gradient algorithm. The fuzzy clustering is applied in order to select the parameters antecedent part of fuzzy rules, that is the parameters of second layer. Gradient algorithm is used for training the parameters

of consequent part-4th layer of FWNN structure. The structure and parameter update rules of the FWNN system are applied for modeling and prediction of complex time series. Simulation results demonstrated that the applied FWNN structure has better performance than other models. The developed FWNN structure is applied to develop a model for predicting future values of exchange rates. This process is high-order nonlinear. Using statistical data, the prediction model is constructed. The test results of the developed system are compared with these obtained from the feed-forward NN-based system, and the first one has demonstrated better performance.

Appendix

As mentioned, the time varying learning rate is used in this paper. Let us consider the derivation of optimal learning rate using Lyapunov function [54]. $\gamma(t)$ denotes learning rates for weight update formulas (10, 11) at discrete time t .

Let $\gamma(t)$ is learning rate for the weights $W = [w_j, a_{ij}, b_{ij}, c_{ij}, o_{ij}]$ and FWNN is trained using (10, 11). The convergence is guaranteed if the following conditions are satisfied

$$0 < \gamma(t) < \frac{2}{\left(\max_t \left\| \frac{\partial u(t)}{\partial W} \right\| \right)^2} \tag{18}$$

Above statement can be proved by choosing a Lyapunov function $V(t) = \frac{1}{2}e^2(t)$, where $e(t) = (u^d(t) - u(t))$.

Let us define Lyapunov function as,

$$V(t) = \frac{1}{2}e^2(t) \tag{19}$$

Here, $e(k)$ represents error function calculated in learning processes.

The change of the Lyapunov function is

$$\begin{aligned} \Delta V(t) &= V(t+1) - V(t) = \frac{1}{2}(e^2(t+1) - e^2(t)) \\ &= \frac{1}{2}((e(t) + \Delta e(t))^2 - e^2(t)) \\ &= \frac{1}{2}(2e(t) \cdot \Delta e(t) + \Delta e^2(t)) \\ &= \frac{1}{2}\Delta e(t)(2e(t) + \Delta e(t)) \end{aligned} \tag{20}$$

Table 3 Comparative results of simulation

Method	Number of hidden neurons	Prediction steps	Epochs	RMSE	
				Training	Testing
Feed-forward NN	16	3-step ahead prediction	1,000	0.0196	0.0352
		6-step ahead prediction	1,000	0.0242	0.0426
FWNN	16	3-step ahead prediction	500	0.0164	0.0226
		6-step ahead prediction	500	0.0228	0.0332

The error difference is determined as

$$\Delta e(t) = \frac{\partial e(t)}{\partial W} \Delta W = \frac{\partial (u^d(t) - u(t))}{\partial W} \Delta W = -\frac{\partial u(t)}{\partial W} \Delta W \tag{21}$$

From the update formula (10, 11)

$$\Delta W = -\gamma \frac{\partial E}{\partial W} = \gamma e(t) \frac{\partial u(t)}{\partial W}$$

$$\frac{\partial E}{\partial W} = \frac{\partial}{\partial W} \left[\frac{1}{2} e^2(t) \right] = e(t) \frac{\partial e(t)}{\partial W} = -e(t) \frac{\partial u(t)}{\partial W} \tag{22}$$

$$\Delta V(t) = \frac{1}{2} \Delta e(t) (2e(t) + \Delta e(t)) = -\frac{1}{2} \left[\frac{\partial u(t)}{\partial W} \right]^T$$

$$\times \gamma(t) e(t) \frac{\partial u(t)}{\partial W} \left(2e(t) - \left[\frac{\partial u(t)}{\partial W} \right]^T \gamma(t) e(t) \frac{\partial u(t)}{\partial W} \right)$$

$$= \frac{1}{2} \gamma(t) e^2(t) \left\| \frac{\partial u(t)}{\partial W} \right\|^2 \left(\gamma(t) \left\| \frac{\partial u(t)}{\partial W} \right\|^2 - 2 \right) \tag{23}$$

From the Lyapunov stability theorem, asymptotic stability is granted if $\Delta V(t) < 0$, for all t . The initial values of the learning rates for the parameters $\{c1_{ij}, c2_{ij}, o_{ij}, a_{ij}, b_{ij}, w_j, q\}$ can be taken differently. In the paper, the learning rates for all parameters $W = \{c1_{ij}, c2_{ij}, o_{ij}, a_{ij}, b_{ij}, w_j, q\}$ are chosen to be the same initially, i.e., $\gamma = \gamma^c = \gamma^o = \gamma^a = \gamma^b = \gamma^w$. According to stability condition, from (23), the sufficient condition for convergence can be derived.

$$0 < \gamma(t) < \frac{2}{\left(\max_t \left\| \frac{\partial u(t)}{\partial W} \right\| \right)^2} \tag{24}$$

From (24), it is seen that the upper bound of learning rate is found from an epoch. However, the learning rate that guaranties most rapid or optimal convergence is $\gamma(t) = 1 / \left\| \frac{\partial u(t)}{\partial W} \right\|^2$. The error at the discrete time $t + 1$ can be represented as

$$e(t + 1) = e(t) + \Delta e(t) \approx e(t) + \left[\frac{\partial e(t)}{\partial W} \right]^T \Delta W$$

$$= e(t) - \left[\frac{\partial u(t)}{\partial W} \right]^T \gamma e(t) \frac{\partial u(t)}{\partial W}$$

$$= e(t) \left(1 - \gamma \left\| \frac{\partial u(t)}{\partial W} \right\|^2 \right) \tag{25}$$

If we solve (25) for $\gamma(t)$ in order to minimize the output error $e(t + 1)$, we can get

$$\gamma(t) = \frac{1}{\left\| \frac{\partial u(t)}{\partial W} \right\|^2} \tag{26}$$

References

1. Box GEP (1970) Time series analysis, forecasting and control. Holden Day, San Francisco
2. So MKP, Lam K, Li WK (1999) Forecasting exchange rate volatility using autoregressive random variance model. Appl Financial Econ 9:583–591
3. Hsieh DA (1989) Modeling heteroscedasticity in daily foreign-exchange rates. J Bus Econ Stat 7:307–317
4. Bollerslev T (1990) Modeling the coherence in short-run nominal exchange rates: a multivariate generalized ARCH model. Rev Econ Stat 72:498–505
5. Huang W, Lai KK, Nakamori Y, Wang S (2004) Forecasting the foreign exchange rates with artificial neural networks: a review. Intl J Inf Tech Decis Mak 3(1):145–165
6. Maddala GS (1996) Introduction to econometrics. Prentice-Hall, Englewood Cliffs
7. Chen Y, Yang B, Dong J, Abraham A (2004) Nonlinear system modelling via optimal design of neural trees. Int J Neural Syst 14(2):125–137
8. Kim K-j, Lee WB (2004) Stock market prediction using artificial neural networks with optimal feature transformation. Neural Comput Appl 13(3):255–260
9. Sfetsos A, Siriopoulos C (2004) Combinatorial time series forecasting based on clustering algorithms and neural networks. Neural Comput Appl 13(1):56–64
10. Maqsood I, Khan MR, Abraham A (2004) An ensemble of neural networks for weather forecasting. Neural Comput Appl 13(2):112–122
11. Thomas B, Soleimani-Mohseni M (2007) Artificial neural network models for indoor temperature prediction: investigations in two buildings. Neural Comput Appl 16(1):81–89
12. Góriz JM, Puntonet CG, Salmerón M, de la Rosa JJG (2004) A new model for time-series forecasting using radial basis functions and exogenous data. Neural Comput Appl 13(2):101–111
13. Sun YF, Liang YC, Zhang WL, Lee HP, Lin WZ, Cao LJ (2005) Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting. Neural Comput Appl 14(1):36–44
14. Garg S, Pal SK (2007) Evaluation of the performance of back-propagation and radial basis function neural networks in predicting the drill flank wear. Neural Comput Appl 16(4–5):407–417
15. Hocaoglu FO, Oysal Y, Kurban M (2009) Missing wind data forecasting with adaptive neuro-fuzzy inference system. Neural Comput Appl 18(3):207–212
16. Gholipour A, Lucas C, Araabi BN, Mirmomeni M, Shafiee M (2007) Extracting the main patterns of natural time series for long-term neurofuzzy prediction. Neural Comput Appl 16(4–5):383–393
17. Weigend AS, Huberman BA, Rumelhart DE (1992) Predicting sunspots and exchange rates with connectionist networks. In: Casdagli M, Eubank S (eds) Nonlinear modeling and forecasting. Addison-Wesley, Redwood City, CA, pp 395–432
18. Refenes AN (1993) Constructive learning and its application to currency exchange rate forecasting. In: Trippi R, Turban E (eds) Neural networks in finance and investing: using artificial intelligence to improve real-world performance. Probus, Chicago, pp 777–805
19. Refenes AN, Azema-Barac M, Chen L, Karoussos SA (1993) Currency exchange rate prediction and neural network design strategies. Neural Comput Appl 1:46–58

20. Kuan CM, Liu T (1995) Forecasting exchange rates using feed-forward and recurrent neural networks. *J Appl Econom* 10:347–364
21. Hann TH, Steurer E (1996) Much ado about nothing? Exchange rate forecasting: neural networks versus linear models using monthly and weekly data. *Neurocomputing* 10:323–339
22. Episcopos A, Davis J (1996) Predicting returns on Canadian exchange rates with artificial neural networks and EGARCHM-M model. *Neural Comput Appl* 4:168–174
23. Yager RR, Zadeh LA (eds) (1994) *Fuzzy sets, neural networks and soft computing*. Van Nostrand Reinhold, New York
24. Jang J-SR, Sun Ch-T, Muzutani E (1997) *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. Prentice Hall, Upper Saddle River, NJ
25. Kugarajah T, Zhang Q (1995) Multidimensional wavelet frames. *IEEE Trans Neural Netw* 6:1552–1556
26. Zhang Q, Benviste A (1995) Wavelet networks. *IEEE Trans Neural Netw* 3:889–898
27. Zhang J, Walter GG, Wayne Lee WN (1995) Wavelet neural networks for function learning. *IEEE Trans Signal Process* 43(6):1485–1497
28. Postalcioglu S, Becerikli Y (2007) Wavelet networks for non-linear system modelling. *Neural Comput Appl* 16(4–5):433–441
29. Lotric U, Dobnikar A (2005) Predicting time series using neural networks with wavelet-based denoising layers. *Neural Comput Appl* 14(1):11–17
30. Cao L, Hong Y, Fang H, He G (1995) Predicting chaotic time series with wavelet networks. *Physica D* 85:225–238
31. Chang PR, Weihui F, Minjun Y (1998) Short term load forecasting using wavelet networks. *Eng Intell Syst Electr Eng Commun* 6:217–230
32. Khao TQD, Phuong LM, Binh PTT, Lien NTH (2004) Application of wavelet and neural network to long-term load forecasting. *International Conference on Power System technology, POWERCON 2004*, pp 840–844, Singapore
33. Thuillard M (2000) Fuzzy logic in the wavelet framework. *Proc Toolmet'2000*, April 13–14, Oulu
34. Thuillard M (2001) *Wavelets in softcomputing*. World Scientific Press, Singapore
35. Lin CK, Wang SD (1996) Fuzzy modelling using wavelet transform. *Electron Lett* 32:2255–2256
36. Lin Y, Wang FY (2005) Predicting chaotic time series using adaptive wavelet-fuzzy inference system. In: *Proceeding of IEEE intelligent vehicles symposium, Las Vegas, Nevada, USA*, pp 888–893
37. Guo QJ, Yu HB, Xu AD (2005) Wavelet fuzzy network for fault diagnosis. In: *Proceedings of international conference on communications, circuits and systems*. IEEE Press, pp 993–998
38. Daniel WCH, Ping-An Z, Jinhua X (2001) Fuzzy wavelet networks for function learning. *IEEE Trans Fuzzy Syst* 9(1):200–211
39. Abiyev RH, Kaynak O (2008) Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study. *IEEE Trans Ind Electron* 55(8):3133–3140
40. Abiyev RH (2005) Controller based of fuzzy wavelet neural network for control of technological processes CIMSA 2005. In: *IEEE international conference on computational intelligence for measurement systems and applications, Giardini Naxos, Italy*, pp 215–219
41. Abiyev RH (2006) Time series prediction using fuzzy wavelet neural network model. *Lecture Notes in Computer Sciences*, Springer, Berlin, pp 191–200
42. Abiyev RH (2009) Fuzzy model identification based on prediction of electricity consumption. *AIEDAM: Artif Intell Eng Des Anal Manuf* 23(2):109–118
43. Bezdek JC (1981) *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York
44. Chiu SL (1994) Fuzzy model identification based on cluster estimation. *J Intell Fuzzy Syst* 2:267–278
45. Yager RR, Filev DP (1994) Generation of fuzzy rules by mountain clustering. *J Intell Fuzzy Syst* 2:267–278
46. Demirli K, Muthukumar P (2000) Higher order fuzzy system identification using subtractive clustering. *J Intell Fuzzy Syst* 9:129–158
47. Kasabov NK (2002) DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series. *IEEE Trans Syst Fuzzy Syst* 10(2):144–154
48. Wang LX (1997) *A course in fuzzy systems and control*. Prentice Hall, NJ, pp 183–189
49. Wang LX, Wei C (2000) Approximation accuracy of some neuro-fuzzy systems. *IEEE Trans Fuzzy Syst* 8(4):470–478
50. Juang C-F (2002) A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithm. *IEEE Trans Fuzzy Syst* 10:155–170
51. Juang CF, Lin CT (1998) An on-line self-constructing neural fuzzy inference network and its applications. *IEEE Trans Fuzzy Syst* 6(1):12–31
52. Zadeh LA (1975) The concept of linguistic variable and its application to approximate reasoning. *Inf Sci* 8
53. Szu H, Telfer B, Garcia J (1996) Wavelet transforms and neural networks for compression and recognition. *Neural Netw* 9:695–708
54. Ku C-C, Lee KY (1995) Diagonal recurrent neural networks for dynamic systems control. *IEEE Trans Neural Netw* 6:144–156
55. Tong RM (1980) The evaluation of fuzzy models derived from experimental data. *Fuzzy Sets Syst* 4:1–12
56. Pedtycz W (1984) An identification algorithm in fuzzy relational systems. *Fuzzy Sets Syst* 13:53–167
57. Xu CW, Lu YZ (1987) Fuzzy model identification and self-learning for dynamic systems. *IEEE Trans Syst Man Cybernet* 17:683–689
58. Sugeno M, Yasukawa T (1993) A fuzzy logic based approach to qualitative modelling. *IEEE Trans Fuzzy Syst* 1:7–31
59. Sugeno M, Tanaka K (1991) Successive identification of a fuzzy model and its application to prediction of complex system. *Fuzzy Sets and Syst* 42:315–334
60. Lin Y, Cunningham GA III (1995) A new approach to fuzzy-neural system modelling. *IEEE Trans Fuzzy Syst* 3:190–198
61. Kim E, Park M, Ji S, Park M (1997) A new approach to fuzzy modelling. *IEEE Trans Fuzzy Syst* 5:328–337
62. Kim E, Park M, Kim S, Park M (1998) A transformed input-domain approach to fuzzy modelling. *IEEE Trans Fuzzy Syst* 6:596–604
63. Kim J, Kasabov NK (1999) HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Netw* 12:1301–1319