

Differential Evaluation Learning of Fuzzy Wavelet Neural Networks for Stock Price Prediction

Rahib H. Abiyev¹, Vasif Hidayat Abiyev²

¹Near East University, Department of Computer Engineering, P.O. Box 670, Lefkosa, TRNC, Mersin-10, Turkey, rahib@neu.edu.tr

²Aksaray University, Department of Economics, Aksaray, Turkey

(Received November 7, 2011, accepted December 2, 2011)

Abstract. Prediction of a stock price movement becomes very difficult problem in finance because of the presence of financial instability and crisis. The time series describing the movement of stock price are complex and non stationary. This paper presents the development of fuzzy wavelet neural networks that combines the advantages of fuzzy systems and wavelet neural networks for prediction of stock prices. The structure of Fuzzy Wavelet Neural Networks (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 rules. The proposed FWNN structure is trained with differential evaluation (DE) algorithm. The use of DE allows quickly train the FWNN system than traditional genetic algorithm (GA). FWNN is used for modelling and prediction of stock prices. Stock prices are changed every day and have high-order nonlinearity. The statistical data for the last three 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 other related models.

Keywords: Prediction of stock price; Fuzzy wavelet neural networks; Differential Evolution

1. Introduction

Financial market is characterized with complex, stochastic, nonstationary processes [1-3]. The chaotic behaviour of the stock price movement complicates the stock price prediction. The development an effective model for prediction of a stock price is one of important problem in finance. Depend on results of prediction system, the trading systems implement market actions of buy, sell or hold. Here the goal of is to choose the best stocks when making an investment and to decide when and how many stocks to sell or buy. Timely decisions must be made which result in buy signals when the market is low and sell signals when the market is high. Profitability of the trading system is very much related to the accuracy of the forecasts, the trading strategy used and magnitude of the transaction costs [1].

Numerous techniques have been developed to predict nonlinearity of time series and improve the accuracy of prediction [2-4]. These are well-known Box-Jenkins method [2], Linear Regression (LR), autoregressive random variance (ARV) model, autoregressive conditional heteroscedasticity (ARCH), general autoregressive conditional heteroskedasticity (GARCH). While these techniques may be good for a particular situation, they do not give satisfactory results for the nonlinear time-series [4]. The 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 which have shortcomings. Hence the idea of applying non-linear models, like soft computing technologies, such as neural networks (NNs), fuzzy systems (FSs), genetic algorithms (GAs), Support Vector Machines (SVMs), has become important for time series prediction. During the last decade, stocks and futures traders have come to rely upon various types

of intelligent systems to make trading decisions. These methods have shown clear advantages over the traditional statistical ones.

NNs have shown to provide better predictive performance in technical analysis and in predicting future stock price movements by analyzing the past sequence of stock prices. NNs can substantially outperform the conventional statistical models [4-6]. NNs do not require strong model assumptions and can map any nonlinear function without any priori assumption about the properties of the data. The radial based network

algorithms for time-series prediction has been considered in [7,8]. Support vector machine (SVM) based on statistical learning theory is a novel neural network algorithm and has been applied in developing an accurate stock market prediction model [9].

During a day, stock prices change from morning to night. In this changeable traffic stock prices are usually described by these four parameters: Open, High, Low, Close. These correspond to market opening price, high and low range of price volatility within a day and closing price. In this paper the integration of fuzzy logic, neural networks and wavelet technology is used in order to describe the stock price values of prediction system. The use of such approach provides better prediction accuracy for achieving the optimal solution, as this was demonstrated in the simulation section of this paper.

Fuzzy theory is applied to stock price forecasting [10-13]. Heuristic model [12], adaptive expectation model [10] for stock price forecasting is proposed to improve stock price forecasting performance. During development of a fuzzy system one of basic problem is generation of IF-THEN rules. Nowadays, the use of neural networks takes more importance for this purpose [14,15,16]. In this paper the integration of NNs 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 NNs shows advantages of wavelet neural networks (WNNs) over NNs in complex nonlinear system modeling. The WNN has been proposed by researchers for solving approximation, classification problems and modeling problems [17-21]. Fuzzy wavelet neural networks (FWNN) combine wavelet theory, fuzzy logic, and neural networks. The different FWNN models are proposed in literature [22-29]. In [22] the membership functions are selected from the family of scaling functions, and a fuzzy system is constructed using wavelet techniques. FWNN is applied for prediction of electricity consumption [26,27]. The combination of wavelet networks 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 non-stationary signals to discover their local details. Fuzzy logic allows us to reduce the complexity of the data and to deal with uncertainty. Neural networks has a self-learning characteristic that increases the accuracy of the prediction. In this paper, to increase prediction accuracy and reduce search space and time for achieving the optimal solution, the combination of wavelet neural networks with a fuzzy knowledge base is used for financial time series prediction, in particular, for the prediction of stock prices. The advantage of FWNN prediction system over other prediction systems is given in [27,28]. For designing neuro-fuzzy systems different approaches are used. These are gradient algorithms, clustering algorithms. These algorithms suffer of having local minima problems. GA and DE can solve local minima problem and find global optimal solution [30-35]. Sometimes GA needs more time for parameter updating. In this paper to speed learning and find optimal solution DE algorithm is used.

The paper is organized as follows: Section 2 presents the structure of FWNN prediction model. Section 3 presents parameter update rules of the FWNN system. The brief descriptions of the DE algorithm for learning of FWNN are given. Section 4 contains simulation results of the FWNN used for prediction of stock prices. Comparative results of different models for time series prediction are given. Finally, a brief conclusion is presented in section 5.

2. Fuzzy Wavelet Neural Networks

The knowledge base of fuzzy systems is 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 [14]. This paper presents fuzzy wavelet neural networks that integrate 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 modelling 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.

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

where 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

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

where $d_j = a_j^{-1} * b_j$. $\Psi_j(x)$ 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.

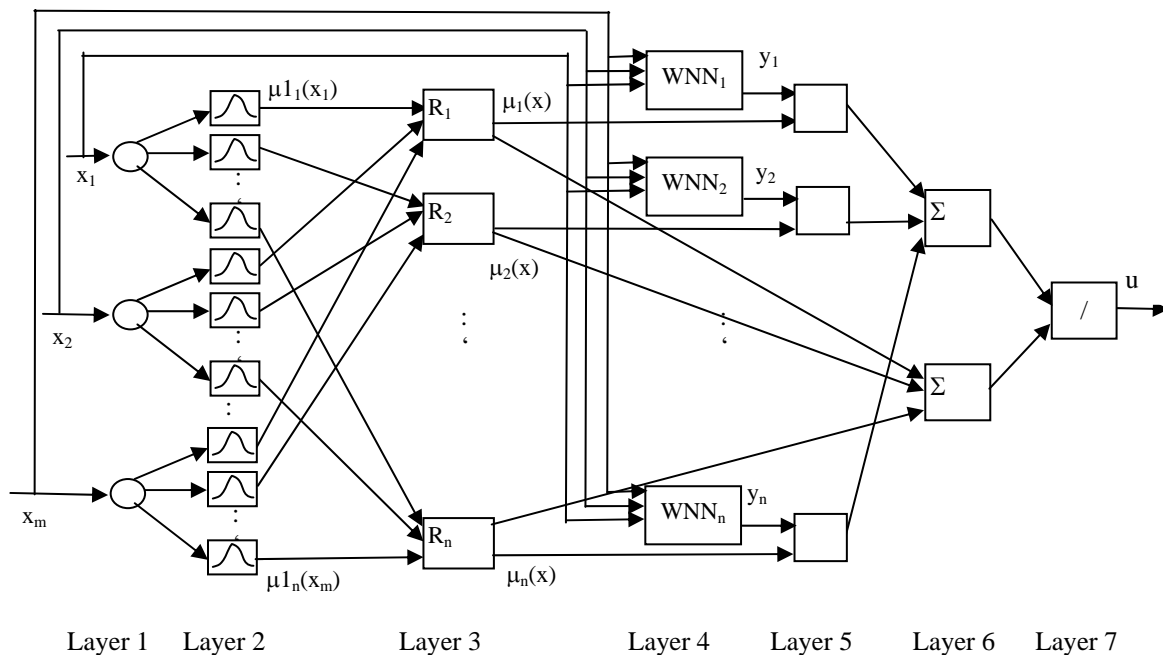


Figure 1. Structure of Fuzzy wavelet neural networks

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 [17-19]. A good initialization of the parameters of WNNs allows to obtain fast convergence. A number of methods is implemented for initializing wavelets, such as orthogonal least square procedure, clustering method [18,19]. 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 [18].

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 behaviours 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 is improved, and, FWNN can describe the nonlinear processes with desired accuracy.

The structure of fuzzy wavelet system is given in Figure 1. The FWNN 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.

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

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

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

where $*$ 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

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

where $z_{ij} = \frac{x_i - b_{ij}}{a_{ij}}$. 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 networks. In this layer the contribution of each WNN to the output of the FWNN is determined.

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

where y_j are the output signals of wavelet neural networks.

After calculating the output signal of the FWNN, the training of the networks 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 networks in the fourth layer. In the next section, the learning of type-2 FWNN is derived.

3. Parameter Update Rules

3.1. GA operators

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). Sometimes the learning of networks using the gradient method for nonlinear processes has “local minima” problem and could not find a global optimal solution. That is, the gradient method might find the set of sub-optimal weight from which it cannot escape. Evolution Algorithms (EAs) is effective optimization technique that can be used to improve training of the FWNN and avoid “local minima” problem. Example of such EAs are Genetic algorithm (GA), Differential Evolution (DE) algorithm. GA is a directed random search method that exploits historical information to direct the search into the region of better performance within the search space [30]. In this paper, the real coded genetic algorithm (GA) with differential evaluation (DE) algorithm is applied for searching optimal values of the parameters of the FWNN.

During optimization, number of the network parameters, which is defined as population size, is generated randomly. These parameters characterize the parameters of antecedent and consequent parts of the FWNN. GA learning is applied to train the parameter values. GA learning is carried out by GA operators. The main operations in GA are selection, crossover and mutation [30-32]. The aim of the selection is to give more reproductive chance to population members (or solutions) that have higher fitness. The tournament selection is applied for selection of new generation. In this method, two members of the population are selected and their fitness values are compared. The member with high fitness is selected for the next generation.

Crossover and mutation are two main components in the reproduction process in which selected pairs mate to produce the next generation. Crossover and mutation allows producing new solutions combining and modifying parent solutions to inherit and reinforce their best characteristics.

The real coded crossover operation is used for the correction of individuals. According to crossover rate, the individuals are selected for the crossover operation in order to generate a new solution. The high value of the crossover rate led to a quick generation of a new solution. The typical value of the crossover rate is selected in the interval [0.1,1]. In crossover operation two parent members $X=(x_1, x_2, \dots, x_n)$ and $Y=(y_1, y_2, \dots, y_n)$ are selected. After crossover operation, the new members will have the form $X'=(x'_1, x'_2, \dots, x'_n)$ and $Y'=(y'_1, y'_2, \dots, y'_n)$. The crossover operation has been performed using the following formula.

$$\begin{aligned} x'_i &= x_i + \delta(y_i - x_i) \\ y'_i &= x_i + \delta(x_i - y_i) \end{aligned} \quad (8)$$

when $F(X) > F(Y)$. Here x_i and y_i are the i -th genes of the parents X and Y , x'_i and y'_i are the i -th genes of the parents X' and Y' . The value δ is changed between 0 and 1.

The simple mutation operation is applied. In this operation, for each gene, a random number is generated. If this random number is less than the mutation rate, then the corresponding gene is selected for mutation. During mutation a small random number, taken from the interval [0,1], is added to the selected gene in order to determine its new value. A large value of the mutation rate leads to a purely random search. The typical value of mutation rate is selected from the interval [0,0.1].

3.2. Differential Evaluation

Differential Evolution is one of the population based stochastic evolutionary optimization techniques which is used for minimizing non-linear and non-differentiable continuous space functions [33,34]. DE includes Evolution Strategies (ES) and conventional GAs. Here the main operation is based on the differences of randomly sampled pairs of solutions in the population. Like other evolutionary algorithms, the first generation is initialized randomly and further generations evolve through the application of certain evolutionary operator until a stopping criterion is reached.

The optimization process in DE is carried with four basic operations: DE, selection, crossover, mutation. Each individual or candidate solution is a vector that contains as many parameters as the problem decision variables. The algorithm starts by creating a population vector $P^G = [X_1^G, X_2^G, \dots, X_{NP}^G]$ of size NP composed of individuals that evolve over G generation. Each individual is a vector that contains as many elements as the problem decision variable $X_i^G = [x_{1i}^G, x_{2i}^G, \dots, x_{Di}^G]$, $i=1, 2, \dots, NP$. The population size is an

algorithm control parameter selected by the user. In Differential Evolution the NP remains constant throughout the optimization process.

The first step in the DE optimization process is to create an initial population of candidate solutions by assigning random values to each decision parameter of each individual of the population. In each iteration of the algorithm to change each population member $X_i(t)$, a Donor vector $V_i(t)$ is created [33-35]. Here DE1 algorithm [33] was used. In this scheme, to create $V_i(t)$ for each i th member, three other parameter vectors (say the r_1 , r_2 , and r_3 th vectors) are chosen in a random fashion from the current population [1, NP]. Next, a scalar number F scales the difference of any two of the three vectors and the scaled difference is added to the third one whence we obtain the donor vector $V_i(t)$. The process for the j th component of each vector can be expressed as

$$v_{i,j}(t+1) = x_{r_1,G}(t) + F \cdot (x_{r_2,j}(t) - x_{r_3,j}(t)) \quad (9)$$

F is a real and constant factor which controls the amplification of the differential variation $(x_{r_2,j}(t) - x_{r_3,j}(t))$.

In order to increase the diversity of the parameter vectors, the vector $U = (u_1, u_2, \dots, u_D)$ with

$$u_j = \begin{cases} v_j & \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D \\ (x_{j,G})_j, & \text{otherwise} \end{cases} \quad (10)$$

is formed where the acute brackets $\langle \rangle_D$ denote the modulo function with modulus D .

In order to decide whether the new vector shall become a population member of generation G , it will be compared to $x_{i,G}$. If vector u_j yields a smaller objective function value than $x_{i,G}$, $x_{i,G+1}$ is set to u_j , otherwise the old value $x_{i,G}$ is retained

Using above described operators selection, crossover, mutation and differential evaluation the training of FWNN parameters, that are parameters of 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 networks in the fourth layer have been performed.

4. Simulation Study. Stock Price Prediction

The FWNN structure and its learning algorithms are applied for modelling and predicting the future values of chaotic time series. The FWNN system is applied for constructing a prediction model of stock prices. Stock price series are high order nonlinear. Appropriate prediction of stock price is one of the most important financial problems and it is very crucial for the success of many business and fund managers.

In the prediction problem, it is needed to predict the value of stock price 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. Three input data points $[x(t-2) \ x(t-1) \ x(t)]$ are used as input to the prediction model. The output training data corresponds to $x(t+3)$. In other words, since the stock price 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 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. 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 , b and w). The design of prediction system is accomplished in four ways: FWNN using traditional GA operators, FWNN using DE, Adaptive Neuro-Fuzzy inference System (ANFIS) [14] and feedforward NNs. At first stage GA operators described in Section 3 are applied in order to determine the parameters of FWNN. The obtained trained parameters are then used in order to design FWNN. 3 fuzzy rules are used for FWNN design, that is three input neurons, three hidden neurons and one output neuron is used to design the structure of FWNN. The structure is used for prediction purpose.

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

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

For training of the system, the statistical data describing daily stock prices for last three years is considered. The data set consists of 1000 data. The 950 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 200 epochs. The values of the parameters of the FWNN system were determined at the conclusion of training. The simulations were performed using three and eight hidden neurons. Once the FWNN has been successfully trained, it is then used for the prediction of the daily stock prices. During learning, the value of RMSE for training data was 0.018655. After learning, the RMSE values for test data was 0.015854. The simulation was performed for two cases: using only GA operators and using DE algorithm that includes selection, crossover, mutation and DE operators described above. Fig. 2 depicts RMSE values obtained during training. As shown in figure the learning curve using DE is quickly converge than learning curve of GA algorithm. The use of differential operator speeds up the training of FWNN model.

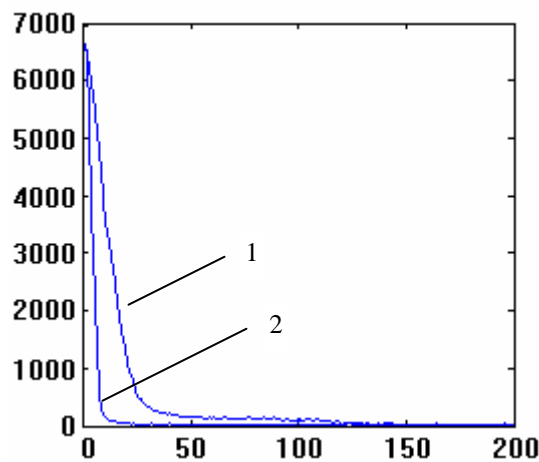


Fig. 2. SME values obtained during training. 1- Using Selection-Crossover-Mutation operators, 2- Using Differential Evaluation algorithm

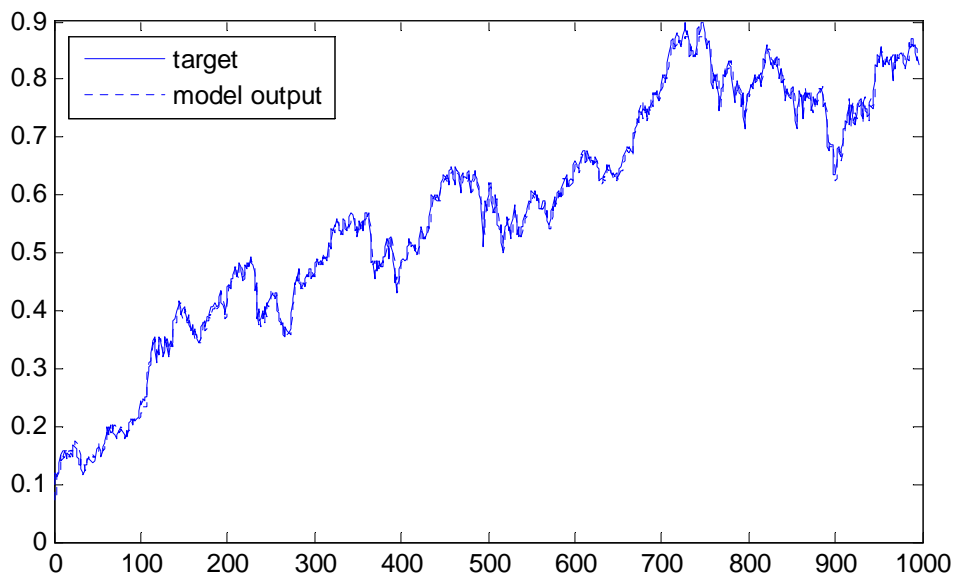


Fig. 3. Three step ahead prediction. Plot of output signals: generated by FWNN (dotted line) and predicted signal (solid line)

In Fig. 3, the output of the FWNN system for three-step ahead prediction of stock price for learning 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. 4. Fig. 5 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 NNs and ANFIS based prediction models also. To estimate the performance of the neural and FWNN prediction systems, the RMSE values of errors between predicted and current output signal are compared.

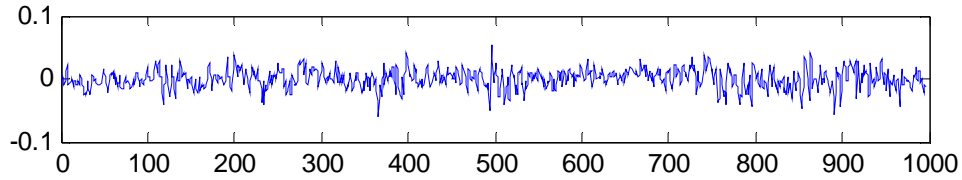


Fig. 4. Plot of prediction error

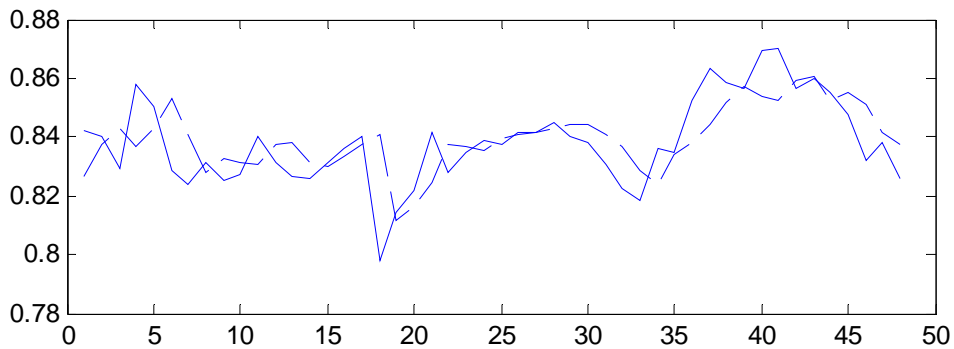


Fig. 5. Three step ahead prediction. Curves describing testing data

In Table 1, the comparative results of simulations, averaged over 10 runs, are given. As shown in the table the performance of FWNN (using DE) prediction is better than the performance of the NNs and ANFIS based prediction models.

Table 3. Comparative results of simulation for three step ahead prediction

Method	Number of rules	Epochs	RMSE	
			Train	Test
Feed-forward NN	3	500	0.023306	0.022998
	8	500	0.022452	0.021507
ANFIS	8	200	0.020206	0.017954
FWNN using GA	3	200	0.022060	0.020419
	8	200	0.020126	0.017867
FWNN using DE	3	200	0.020118	0.017391
	8	200	0.018655	0.015854

In next experiment $x(t+1)$ one step ahead prediction of stock price is performed. The data points $[x(t-2) x(t-1) x(t)]$ are used as input for the system. The 950 data points are used for learning, and last 50 days are used for testing. Simulations are performed using 3 and 8 hidden neurons. In the result of learning the parameters of FWNN were found. During learning the RMSE value was 0.013921. After learning, for the test data, the value of RMSE was 0.012438. The simulation was performed using FWNN with GA operators, using feedforward NNs and using ANFIS based models also. Table 2 demonstrate the comparative simulation results of used models. As shown in the table the performance of FWNN (using DE) prediction is better than the performance of other models. The simulation results satisfy the efficiency of the application of FWNN technology in constructing a prediction model of stock price.

Table 4. Comparative results of simulation for one step ahead prediction

Method	Number of rules	Epochs	RMSE	
			Train	Test
Feed-forward NN	3	500	0.022563	0.020455
	3	500	0.021844	0.018776
ANFIS	8	200	0.011449	0.012239
FWNN using GA	3	200	0.015667	0.013126
	8	200	0.013483	0.012266
FWNN using DE	3	200	0.013921	0.012438
	8	200	0.011905	0.010829

5. Conclusion

The time series prediction model is developed by integrating fuzzy logic, neural networks and wavelet technology. FWNN prediction model is constructed using DE algorithm. The structure and parameter update rules of the FWNN system is applied to develop a model for predicting future values of stock prices. This process is high order nonlinear. Using statistical data, the prediction model is constructed. The training was performed using feedforward NN, ANFIS and using FWNN. The training of FWNN was performed with traditional GA algorithm that uses selection, crossover and mutation procedure and using DE algorithm. It was obtained that using DE algorithm the training of networks was performed faster than traditional GA algorithm. Comparative simulation results demonstrate that the proposed FWNN system with DE training has better performance than other models.

6. References

- [1] M. Qi and G. S. Maddala. Economic Factors and the Stock Market: A New Perspective. *Forecasting*. 1999, **18**: 151-166,
- [2] G.E.P. Box. *Time series analysis, forecasting and control*. San Francisco: Holden Day, 1970.
- [3] L. Yu, S. Wang, and K. K. Lai. An Online Learning Algorithm with Adaptive Forgetting Factors for Feedforward Neural Networks in Financial Time Series Forecasting. *Nonlinear Dynamics and Systems Theory*. 2007, **7**: 51 - 66.
- [4] R.John, B. Guerard and Eli Schwartz. Regression Analysis And Forecasting Models. *Quantitative Corporate Finance*. DOI: 10.1007/978-0-387-34465-2_12
- [5] A. Vellido, P. J. G. Lisboa, J. Vaughan. Neural networks in business: a survey of applications (1992-1998). *Expert Systems with Applications*. 1999, **17**: 51-70.
- [6] Y. Chen, B. Yang, J. Dong, A. Abraham. Time-series forecasting using flexible neural tree model. *Information Sciences*. 2005, **174**: 219-235
- [7] M. Marcek and D. Marcek. Granular RBF Neural Network Implementation of Fuzzy Systems: Application to Time Series Modeling. *Journal of Multiple-Valued Logic and Soft Computing*. 2008, **14**(3-5).
- [8] I. Rojas, J. Gonzalez, A. Canas, A. F. Diaz, F. J. Rojas, M. Rodrigues. Short-term prediction of chaotic time series by using RBF network with regression weights. *Int. J. Neural Sys*. 2000, **10**(5): 353-364
- [9] F. E. H. Tay, L. J. Cao. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*. 2003, **14**: 1506-1518.
- [10] C-H. Cheng, T-L. Chen, H.J. Teoh and C-H. Chiang. Fuzzy time-series based on adaptive expectation model for TAIEX forecasting. *Expert Systems with Applications: An International Journal*. 2008, **34**(2): 1126-1132.
- [11] K. Huarng, H.-K. Yu. An N-th order heuristic fuzzy time series model for TAIEX forecasting. *Int. J. Fuzzy Systems*. 2003, **5**(4).
- [12] K. Huarng. Heuristic models of fuzzy time series for forecasting. *Fuzzy Sets and Systems*. 2002, **123**: 369-386.
- [13] H.S. Lee and M.T. Chou. Fuzzy forecasting based on fuzzy time series. *International Journal of Computer Mathematics*. 2004, **81**(7): 781-789.
- [14] Jang, J-S. R. *Neuro-Fuzzy and Soft computing. A computational approach to learning and machine intelligence*. Prentice-Hall, 1997.
- [15] T. H-K. Yu, K-H. Huarng. A neural network-based fuzzy time series model to improve forecasting. *Expert Systems with Applications: An International Journal*. 2010, **37**(4): 3366-3372.

- [16] A. Gholipour, C. Lucas, B. N. Araabi, M. Mirmomeni and M. Shafiee. Extracting the main patterns of natural time series for long-term neurofuzzy prediction. *Neural computing & applications*. 2007, **16**(4-5): 383-393.
- [17] Kugarajah, T. & Zhang, Q. Multidimensional wavelet frames. *IEEE Transaction on Neural Networks*. 1995, **6**: 1552-1556.
- [18] Zhang, Q., & Benviste, A. Wavelet networks. *IEEE Transaction on Neural Networks*. 1995, **3**: 889-898.
- [19] Zhang, J., Walter, G.G. & Wayne Lee, W.N. Wavelet Neural Networks for Function Learning. *IEEE Transaction on Signal Processing*. 1995, **43**(6): 1485-1497
- [20] S. Postalcioglu and Y. Becerikli. Wavelet networks for nonlinear system modelling. *Neural Computing and Application*. 2007, **16**(4-5): 433-441.
- [21] K. K. Minu, M. C. Lineesh and C. Jessy John. Wavelet Neural Networks for Nonlinear Time Series Analysis. *Applied Mathematical Sciences*. 2010, **4**(50): 2485 – 2495.
- [22] Thuillard, M. *Wavelets in Softcomputing*. World Scientific Press, 2010.
- [23] Daniel, W.C. H., Ping-An, Z., & Jinhua, X. Fuzzy Wavelet Networks for Function Learning. *IEEE Transactions on Fuzzy Systems*. 2001, **9**(1): 200-211.
- [24] 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 Trans. on Industrial Electronics*. 2008, **55**(8): 3133-3140.
- [25] R.H. Abiyev. Controller Based of Fuzzy Wavelet Neural Network for Control of Technological Processes. *CIMSA 2005 – IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. Giardini Naxos, Italy. 2005, pp.215-219.
- [26] R.H. Abiyev. Time Series Prediction Using Fuzzy Wavelet Neural Network Model. *Lecture Notes in Computer Sciences*. Berlin Heidelberg: Springer-Verlag. 2006, pp.191-200.
- [27] R. H. Abiyev. Fuzzy Wavelet Neural Network for Prediction of Electricity Consumption. *AIEDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*. 2009, **23**(2): 109-118.
- [28] R. H. Abiyev. Fuzzy Wavelet Neural Network Based on Fuzzy Clustering and Gradient Techniques for Time Series Prediction. *Neural Computing & Applications*. 2011, **20**(2): 249-259.
- [29] S. Yilmaz and Y. Oysal. Fuzzy Wavelet Neural Network Models for Prediction and identification of Dynamical Systems . *IEEE Tran. On Neural Networks*. 2010, **21**(10).
- [30] Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison –Wesley, 1998.
- [31] Alcalá-Fdez, J.; Alcalá, R.; Gacto, M.J.; Herrera, F. Learning the Membership Function Contexts for Mining Fuzzy Association Rules by Using Genetic Algorithms. *Fuzzy Sets And Systems*. 2009, **160**(7): 905-921
- [32] Xianghe Jing, Xianzhong Zhou, Yangyong Xu. A Hybrid Genetic Algorithm for Bin Packing Problem Based on Item Sequencing. *Journal of Information and Computing Science*. 2006, **1**(1): 61-64
- [33] R. Storn and K. Price. *Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report TR-95-012, Berkeley, USA: International Computer Science Institute, 1995.
- [34] K. Vaisakh and L. R. Srinivas. Differential Evolution Approach for Optimal Power Flow Solution. *Journal of Theoretical and Applied Information Technology*. 2008, **4**(4): 261-268.
- [35] U.K. Chakraborty (Ed.). *Advances in Differential Evolution*. New York: Springer, Heidelberg, 2008.