An Empirical Study of Fuzzy Approach with Artificial Neural Network Models

Memmedaga Memmedli and Ozer Ozdemir

Abstract— Time series forecasting based on fuzzy approach by using artificial neural networks is a significant topic in many scientific areas nowadays. Artificial neural network models are sufficient due to their abilities to solve nonlinear problems especially financial researches in recent years. For these reasons, in this paper we made a forecasting study for weekly closed prices of the exchange rate of Turkish Liras (TL) to Euro between 2005 and 2009 which has important effect in economical and industrial areas. We applied the best four networks which are called multilayer perceptron (MLP), radial basis function (RBF) neural network and generalized regression neural network (GRNN) to improve forecasting fuzzy time series with different degrees of membership by using MSE performance measure. Empirical results show that the MLP outperforms others to forecast neural network based-fuzzy time series.

Keywords—Artificial neural network models, Exchange rate, Forecasting, Fuzzy approach, Time series.

I. Introduction

FUZZY set theory was introduced by Zadeh [1]. In recent years, fuzzy time series approach introduced by Song and Chissom[2],[3],[4]. Since then, many fuzzy time series models have been proposed such as first order models [5],[6],[7], high-order models [8],[9], hybrid models[10],[11], seasonal models [12],[13], bivariate models [14],[15] and multivariate models[16],[17],[18]. These fuzzy time series models have been applied to various problem domains, such as enrollment [19],[22], temperature [23,[24] and stock index [25],[26].

Neural networks have become popular due to their abilities to solve nonlinear problems in recent years. So, in this study, a neural network based fuzzy time series model has been applied to solve nonlinear problems. We made a comparison study to improve forecasting performance of exchange rate of TL to Euro with different neural network architectures. Forecasting results of all neural network models including multilayer perceptron, radial basis function and generalized regression neural networks are finally compared with each other and multilayer perceptron is the best to forecast fuzzy time series according to other artificial neural network models. To show these things, the remainder of this paper is organized as follows. Section 2 clearly reviews fuzzy time series. Section 3 briefly explains most commonly used artificial neural networks. Section 4 includes all empirical analysis. Finally, section 5 concludes the paper.

Prof. Dr. M. Memmedli is with the Statistics Department, Faculty of Science, Anadolu University, Eskisehir, TURKEY. (e-mail: mmammadov@anadolu.edu.tr).

Lecturer O. Ozdemir is with the Department of Statistics, Faculty of Science, Anadolu University, Eskisehir, TURKEY. (corresponding author to provide phone: +902223350580-4677; e-mail: ozerozdemir@anadolu.edu.tr).

II. Fuzzy Time Series

Song and Chissom first proposed the definitions of fuzzy time series [2],[3]. The some general definitions of fuzzy time series are given as follows:

Let U be the universe of discourse, where $U=\{u_1,u_2,...,u_n\}$. A fuzzy set A_i of U is defined by

$$A_i = f_{Ai}(u_1)/u_1 + f_{Ai}(u_2)/u_2 + ... + f_{Ai}(u_n)/u_n ,$$

where f_{Ai} is the membership function of the fuzzy set A_i, $f_{Ai}:U \to [0,1]$. u_k is an element of fuzzy set A_i and $f_{Ai}(u_k)$ is the degree of belongingness of u_k to A_i. $f_{Ai}(u_k)\in[0,1]$ and $1 \le k \le n$.

Definition 1: $Y(t)$ $(t=...,0,1,2,...)$ is a subset of real numbers. Let $Y(t)$ be the universe of discourse defined by the fuzzy set $f_i(t)$. If $F(t)$ consists of $f_i(t)$ $(i=1,2,...)$, $F(t)$ is defined as a fuzzy time series on $Y(t)$ $(t=...,0,1,2,...)$ [2].

Definition 2: If there exists a fuzzy relationship $R(t-1,t)$, such that

$$F(t)=F(t-1)\times R(t-1,t),$$

where $\times$ is an operator, then $F(t)$ is said to be caused by $F(t-1)$. The relationship between $F(t)$ and $F(t-1)$ can be denoted by

$$F(t-1)\to F(t).$$

Definition 3: Suppose $F(t-1)=A_i$ and $F(t)=A_j$, a fuzzy logical relationship is defined as

$$A_i \rightarrow A_j,$$

where A_i is named as left-hand side of the fuzzy logical relationship and A_j the right-hand side. Note the repeated fuzzy logical relationships are removed [3], [43].

Definition 4: Fuzzy logical relationships can be further grouped together into fuzzy logical relationship groups according to the same left-hand sides of the fuzzy logical relationships.

For example, there are fuzzy logical relationships with the same left-hand sides (A_i):

$$A_i \rightarrow A_{j1},$$
$$A_i \rightarrow A_{j2}$$
$$\cdots$$

These fuzzy logical relationships can be grouped into a fuzzy logical relationship group as follows:

$$A_i \rightarrow A_{j1}, A_{j2}, \ldots$$

Definition 5: Suppose $F(t)$ is caused by $F(t-1)$ only, and $F(t) = F(t-1) \times R(t-1,t)$. For any t, if $R(t-1,t)$ is independent of t, then $F(t)$ is named a time-invariant fuzzy time series, otherwise a time-variant fuzzy time series.

Song and Chissom applied both time-invariant and time-variant models to forecast the enrollment at the University of Alabama [3],[4]. The time-invariant model includes the following steps:

(1) define the universe of discourse and the intervals,
(2) partition the intervals,
(3) define the fuzzy sets,
(4) fuzzify the data,
(5) establish the fuzzy relationships,
(6) forecast,
(7) defuzzify the forecasting results.

The time-variant model includes the following steps:

(1) Define the universe of discourse and the intervals (the same as step (1) in the time-invariant model).
(2) Partition the intervals (the same as step (2) in the time-invariant model).
(3) Define the fuzzy sets (the same as step (3) in the time-invariant model).
(4) Fuzzify the data (the same as step (4) in the time-invariant model).
(5) Establish the fuzzy relationships and forecast.
(6) Defuzzify the forecasting results.

In both models, note that the establishment of fuzzy relationships, $R(t-1,t)$, and defuzzification were the critical steps for forecasting [3],[4],[5],[8],[18], [44].

We used the following steps in problem solution:

Step 1. Defining and partitioning the universe of discourse.
Step 2. Fuzzification.
Step 3. Neural Network Training.
Step 4. Neural Network Forecasting.
Step 5. Defuzzification.
Step 6. Performance Evaluation.

III. Artificial Neural Networks

In forecasting, artificial neural networks are mathematical models that imitate biological neural networks. Artificial neural networks consist of some elements. Determining the elements of the artificial neural networks issue that affect the forecasting performance of artificial neural networks should be considered carefully [27],[28]. One of them is network architecture. However, there are not general rules for determining the best architecture. So, much architecture should be tried for the correct results. There are various types of artificial neural networks. Let give an overview of the networks which is indicated in the best three networks for the related data of the recent study [20], [21].

Multi Layer Percepteron (MLP): MLP networks are constructed of multiple layers of computational units. Each neuron in one layer is directly connected to the neurons of the subsequent hidden layer. In many applications, the frequently used activation function is sigmoid function. Multi-layer networks use a variety of learning techniques, the most popular being back-propagation.

Artificial Neural Networks usually refer to Multilayer Perceptron Neural Networks and are a popular estimator to construct nonlinear models of data. A MLP distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons of hidden units. For example, a three-layer MLP is given in (Fig. 1). The function of hidden neurons is to intervene between the external input and the network output in some useful manner [29].

MLP has been applied successfully to difficult problems by training in a supervised algorithm known as the error backpropagation algorithm. This learning algorithm consists of two directions through the different layers of the network: forward and backward directions. In the forward direction, an input data is applied to the input nodes of the network and its error propagates through the network layer by layer. Finally, a set of outputs is produced as an actual response of the network. During the forward direction, the synaptic weights of the networks are not changed while, during the backward direction, the synaptic weights are altered in accordance with an error correction rule. The definite response of the output layer is subtracted absolutely from an expected response to produce an error signal. This error signal is then propagated backward through the network [21].

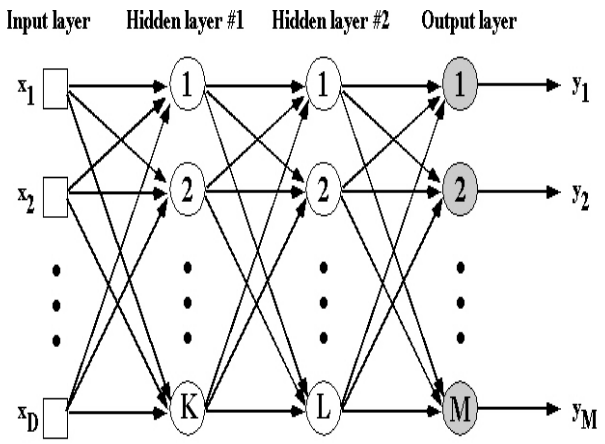

Figure 1. Architecture of a three-layer MLP

During the processing in a MLP-network, activations are propagated from input units through hidden units to output units. At each unit j, the weighted input activations $a_i w_{ij}$ are summed and a bias parameter θ_j is added.

$$net_j = \sum_i a_i w_{ij} + \theta_j \tag{1}$$

The resulting network input net_j is then passed through a sigmoid function (the logistic function) in order to restrict the value range of the resulting activation a_j to the interval [0, 1].

$$a_j = \frac{1}{1 + e^{-net_j}} \tag{2}$$

The network learns by adapting the weights of the connections between units, until the correct output is produced. MLP networks use a variety of learning techniques, the most popular being back-propagation [21]. It performs a gradient descent search on the error surface. The weight update Δw_{ij}, i.e. the difference between the old and the new value of weight w_{ij}, is here defined as:

$$\Delta w_{ij} = \eta a_{pi} \delta_{pj}, \tag{3}$$

where

$$\delta_{pj} = \begin{cases} a_{pj}\left(1 - a_{pj}\right)\left(t_{pj} - a_{pj}\right), & \text{if } j \text{ is an output unit} \\ a_{pj}\left(1 - a_{pj}\right)\sum_k \delta_{pk} w_{jk}, & \text{if } j \text{ is a hidden unit} \end{cases} \tag{4}$$

here, t_p is the target output vector which the network must learn.

Training the MLP-network with the backpropagation rule guarantees that a local minimum of the error surface is found, though this is not necessarily the global one. In order to speed up the training process, a momentum term is often introduced into the update formula [42]:

$$\Delta w_{ij}\left(t+1\right) = \eta a_{pi} \delta_{pj} + \alpha \Delta w_{ij}\left(t\right) \tag{5}$$

Radial Basis Function (RBF): That network type is consisting of an input layer, a hidden layer of radial units and an output layer of linear units. Typically, the radial layer has exponential activation functions and the output layer a linear activation functions.

RBF network is an alternative to the more widely used MLP network and is less computer time consuming for network training. RBF network consists of three layers: an input layer, a hidden layer and an output layer. The nodes within each layer are fully connected to the previous layer. The input variables are each assigned to the nodes in the input layer and they pass directly to the hidden layer without weights. The transfer functions of the hidden nodes are RBF [30].

RBF networks are being used for function approximation, pattern recognition and time series prediction problems. Such networks have the universal approximation property [31], are dealt well in the theory of interpolation [32] and arise naturally as regularized solutions of ill-posed problems [33]. Their simple structure enables learning in stages, gives a reduction in the training time, and this has led to the application of such networks to many practical problems [34].

RBF networks have traditionally been associated with radial functions in a three-layer network (see Fig. 2) consisting of an input layer, a hidden layer of radial units and an output layer of linear units [35], [36].

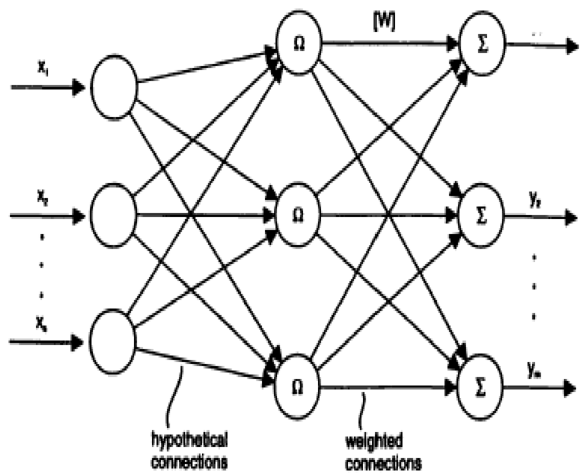

Figure 2. Architecture of a RBF Network.

The radial basis function determines the output with input variable x and distance from center μ. As the input variable approaches the center, the output becomes larger. As the radial basis function, the Gaussian function is often used. It may be written as

$$f(x) = \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \tag{6}$$

where x : input, μ : center, σ : width of receptive field. Output of RBF network is expressed by a linear combination of the radial basis functions. It may be written as

$$y = \sum_{j=1}^{n} w_j \phi_j \tag{7}$$

where w_j : connection weight, ϕ_j : output of basis function [45].

Generalized Regression Neural Networks (GRNN): That type of networks is a kind of Bayesian network. GRNN has exactly four layers: input, a layer of radial centers, a layer of regression units, and output. This layer must be trained by a clustering algorithm. Think of it as a normalized RBF network in which there is a hidden unit centered at every training case.

A GRNN is based on nonlinear regression theory and often used as a popular statistical tool for function approximation. GRNN is one variant of the RBF network, unlike the standard RBF, the weights of these networks can be calculated analytically [37].

GRNN was devised by Specht [38], casting a statistical method of function approximation into a neural network form. The GRNN, like the MLP, is able to approximate any functional relationship between inputs and outputs [39]. Structurally, the GRNN resembles the MLP. However, unlike the MLP, the GRNN does not require an estimate of the number of hidden units to be made before training can take place. Furthermore, the GRNN differs from the classical MLP in that every weight is replaced by a distribution of weight which minimizes the chance of ending up in local minima. Therefore, no test and verification sets are required [40].

GRNN has exactly four layers: input, a layer of radial centers, a layer of regression units, and output. This layer must be trained by a clustering algorithm. Think of it as a normalized RBF network in which there is a hidden unit centered at every training case. Figure 3 shows the general structure of the GRNN [41].

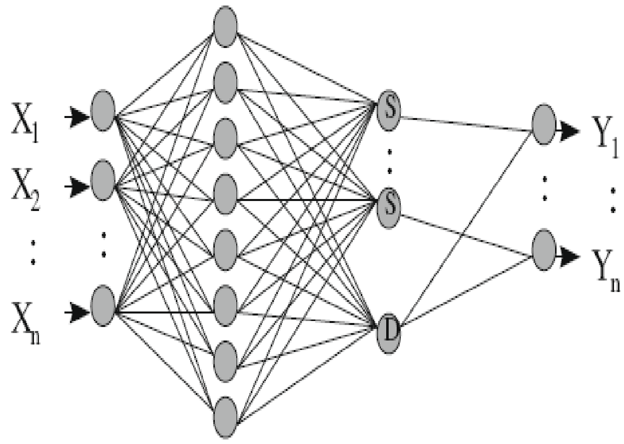

Figure 3. General structure of the GRNN.

IV. Empirical Analysis

This study uses weekly closed prices of the exchange rate of TL to Euro between 2005 and 2009 as the forecasting target. Empirical analysis shows preparing a neural network based fuzzy time series model to improve forecasting performance and show forecasting performance year by year according to performance measure called mean square error (MSE). After obtaining all results, we compared the results for all artificial neural networks by using MSE from 2005 to 2009 and calculated overall results.

We can explain all steps for problem solution.

Step 1. Defining and partitioning the universe of discourse

The universe of discourse for observations, $U = [starting, ending]$, is defined. After the length of intervals, l, is determined, the U can be partitioned into equal–length intervals $u_1, u_2, \ldots, u_b$, $b = 1, \ldots$ and their corresponding midpoints $m_1, m_2, \ldots, m_b$ respectively.

$$u_b = \left[starting + (b-1)\times l, starting + b \times l\right],$$
$$m_b = \frac{\left[starting + (b-1)\times l, starting + b \times l\right]}{2}.$$

We can show different length of intervals with starting and ending points according to exchange rate of TL to Euro for all years in Table 1.

Table 1. Length of intervals for all years

	Start	End	Interval	Length
2005	1.58	1.86	0.28	0.02
2006	1.55	2.11	0.56	0.04
2007	1.67	1.89	0.22	0.02
2008	1.7	2.22	0.52	0.04
2009	2.05	2.29	0.24	0.02

For example, for the year 2005, we have $U = [1.58, 1.86]$

and the length of interval is set to 0.02.

Step 2. Fuzzification

Each linguistic observation, A_i, can be fuzzified into a set of degrees of membership,

$$A_1 = \mu_1^1 / u_1 + \mu_1^2 / u_2 + ... + \mu_1^b / u_b$$

Step 3. Neural Network Training

A large data set is necessary for training a neural network and we used weekly closing prices of the exchange rate of TL to Euro for the years from 2005 to 2009. Many studies have used a convenient ratio to separate in-samples from out-of samples ranging from 70%:30% to 90%:10%. Hence, we chose the data from January to October for our training (in-sample) and November and December for forecasting (out-sample). So the ratio is about 83%:17%. $F(t-1) = A_i$ is taken as input and $F(t) = A_j$ is taken as output since fuzzy logical relationship is defined as $A_i \rightarrow A_j$.

Step 4. Neural Network Forecasting

By applying the process, we use all training data for training the neural network, so we can forecast all the degrees of membership for out of sample data.

Step 5. Defuzzification

We used weighted averages to defuzzify the degrees of membership as follows;

$$forecast(t) = \frac{\sum_{k=1} \mu_t^k m_k}{\sum_{k=1} \mu_t^k}$$

μ_t^k shows the forecasted degrees of membership and m_k represents corresponding midpoints.

Step 6. Performance Evaluation

We choose MSE for performance measure for performance evaluation. It is calculated as follows;

$$MSE = \frac{\sum_{t=k+1}^{n} \left(forecast(t) - observation(t) \right)^2}{n-k}$$

n is the number of observations, including k in-sample and n-k out-of-sample observations.

We trained and forecasted the data for all years with MLP, RBF and GRNN. The best results of all are obtained with MLP and its results by all years and overall performance can be seen in Section 5.

V. Conclusion

Fuzzy approach and artificial neural networks become effective tool for researchers by forecasting fuzzy time series. The relation of these has advantage to improve forecasting performance especially in handling nonlinear systems. Hence, in this study we aimed to handle a nonlinear problem to apply neural network-based fuzzy time series model. Differing from previous studies, we used various degrees of membership in establishing fuzzy relationships and we performed different neural network models to improve forecasting performance. To demonstrate comparison between these models we used a data set of exchange rate of Turkish Liras (TL) to Euro for the years 2005-2009. Empirical results show that the multilayer perceptron is the best to forecast fuzzy time series in most commonly used artificial neural network models.

Time series forecasting by using artificial neural networks is an important issue in many scientific researches in recent years. Artificial neural networks are sufficient due to their abilities to solve nonlinear problems nowadays.

In this paper we made a forecasting study for weekly closed prices of the exchange rate of TL to Euro between 2005 and 2009 which has important effect in economical and industrial areas. We applied the best four networks which are called MLP, RBF and GRNN to improve forecasting fuzzy time series with different degrees of membership by using MSE performance measure.

First, by using the exchange rate of TL to Euro for the years 2005-2009 which is a large data set for training a neural network is used for forecasting target and separated into in-sample (estimation) and out-of-sample (forecasting). The ratio was 83%:17%. Second, we used the best four networks for training and forecasting.

In Table 2, 3 and 4, we can see the results of MSE for various artificial neural network models after forecasting for financial time series. In Table 2, all results are shown according to years for GRNN best models in all other GRNN models.

Table 2. MSE values for neural network models for GRNN

Years	MSE
2005	0,005409
2006	0,000980
2007	0,002482
2008	0,027661
2009	0,002905
Overall	0.0078874

In Table 3, all results are shown according to years for MLP best models in all other MLP models.

Table 3. MSE values for neural network models for MLP

Years	MSE
2005	0,004193
2006	0,000405
2007	0,001098
2008	0,030090
2009	0,003024
Overall	0.007762

According to the first two tables, in general, we can say that MLP models are better results by comparing all years and overall results. In Table 4, all results are shown according to years for RBF best models in all other RBF models.

Table 4. MSE values for neural network models for RBF

Years	**MSE**
2005	0,004615
2006	0,000414
2007	0,002648
2008	0,032995
2009	0,002741
Overall	0.0086826

After Table 4 and according to overall results in Table 5, we can finally express that MLP models without fuzzy approach for a large data set of exchange rate outperform other model types by yearly comparison and overall results.

Table 5. MSE overall values for all neural network models

Models	**MSE**
GRNN	0.0078874
MLP	**0.0077620**
RBF	0.0086826

Now, by using neural network-based fuzzy time series models, we can obtain Table 6, 7 and 8. So, we can see the results of MSE for taking into account fuzzy approach by using various artificial neural network models in proposed method to forecast this time series. In Table 6, we can see the all results according to all years for GRNN-based fuzzy time series best models in all other GRNN-based fuzzy time series models which are generated by STATISTICA software.

Table 6. MSE values for GRNN-based fuzzy time series models

Years	**MSE**
2005	0,003448
2006	0,000314
2007	0,003727
2008	0,056690
2009	0,002614
Overall	0.0133586

In Table 7, all results are shown according to years for MLP-based fuzzy time series best models in all other MLP-based fuzzy time series models which are generated by STATISTICA software.

According to the first two tables, in general, we can say that MLP-based fuzzy time series models are better results by comparing all years and overall results according to GRNN-based fuzzy time series models.

In Table 8, all results are shown according to years for RBF-based fuzzy time series best models in all other RBF-based fuzzy time series models.

Table 7. MSE values for MLP-based fuzzy time series models

Years	**MSE**
2005	0,003255
2006	0,000310
2007	0,000771
2008	0,021647
2009	0,001501
Overall	0.0054968

Table 8. MSE values for RBF-based fuzzy time series models

Years	**MSE**
2005	0,016142
2006	0,002519
2007	0,001521
2008	0,022613
2009	0,001852
Overall	0.0089294

After Table 8 and according to overall results in Table 9, we can finally express that MLP-based fuzzy time series models with fuzzy approach for a large data set of weekly closed prices of the exchange rate of TL to Euro between 2005 and 2009 outperform other model types with using fuzzy approach by yearly comparison and overall results.

Table 9. MSE overall values for all neural network-based fuzzy time series models

Models	**MSE**
F-GRNN	0.0133586
F-MLP	**0.0054968**
F-RBF	0.0089294

All empirical results show that the MLP is the best to forecast fuzzy time series in most commonly used artificial neural network models. Also, MLP-based fuzzy time series models outperform artificial neural network models without fuzzy approach for Table 10. We can see the general results of all model types in Table 10.

Table 10. MSE overall values for both neural network-based fuzzy time series models and artificial neural network models

Models	**MSE**
GRNN	0.0078874
MLP	0.0077620
RBF	0.0086826
F-GRNN	0.0133586
F-MLP	**0.0054968**
F-RBF	0.0089294

In terms of the yearly comparison, the MLP model performed better than other models for demonstrating 100000 networks by using STATISTICA 6.0 software. We can see the same results from the overall performance. Average of all years' performance of neural network based fuzzy time series by MLP model is better than average of all years' performance of the other. We can say that MLP neural networks without fuzzy approach are the second and other neural network-based fuzzy time series models are the last for performance evaluations of forecasting according to weekly closed prices of the exchange rate of TL to Euro between 2005 and 2009 by using STATISTICA 6.0 software. Therefore, fuzzy approach with neural network based is better by using just MLP architecture. Without MLP, artificial neural network models can have better results according to other neural network-based fuzzy time series models such as GRNN or RBF.

REFERENCES

[1] L.A. Zadeh, "Fuzzy Sets", *Information and Control,* 8, 1965, pp. 338-353.

[2] Q. Song and B. S. Chissom, "Fuzzy Time Series and its Models", *Fuzzy Sets and Systems,* 54, 1993, pp. 269-277.

[3] Q. Song and B. S. Chissom, "Forecasting Enrollments with Fuzzy Time Series, Part 1", *Fuzzy Sets and Systems,* 54, 1993, pp. 1-10.

[4] Q. Song and B. S. Chissom, "Forecasting Enrollments with Fuzzy Time Series, Part 2", *Fuzzy Sets and Systems,* 62, 1994, pp. 1-8.

[5] S.-M. Chen, "Forecasting Enrollments Based on Fuzzy Time Series", *Fuzzy Sets and Systems,* 81(3), 1996, pp. 311-319.

[6] S.-M. Chen and J. R. Hwang, "Temperature Prediction Using Fuzzy Time Series", *IEEE Transactions on Systems, Man, and Cybernetics Part B,* 50(2), 2000, pp. 263-275.

[7] J. Sullivan and W. H. Woodall, "A Comparison of Fuzzy Forecasting and Markov Modeling", *Fuzzy Sets and Systems,* 64(3), 1994, pp. 279-293.

[8] S.-M. Chen, "Forecasting Enrollments Based on High-Order Fuzzy Time Series", *Cybernetics and Systems,* 33(1), 2002, pp. 1-16.

[9] K. Huarng and H.-K. Yu, "An Information Criterion for Fuzzy Time Series to Forecast the Stock Exchange Index on the TAIEX", *In Ninth International Conference on Fuzzy Theory and Technology,* 2003, pp. 187-189.

[10] K. Huarng and H.-K. Yu, "An N-th Order Heuristic Fuzzy Time Series Model for TAIEX Forecasting", *International Journal of Fuzzy Systems,* 5(4), 2003, pp. 247-253.

[11] F.-M. Tseng and G.-H. Tzeng, "A Fuzzy Seasonal ARIMA Model for Forecasting", *Fuzzy Sets and Systems,* 126(3), 2002, pp. 367-376.

[12] P.T. Chang, "Fuzzy Seasonality Forecasting", *Fuzzy Sets and Systems*, 90(1), 1997, pp. 1-10.

[13] Q. Song, "Seasonal Forecasting in Fuzzy Time Series", *Fuzzy Sets and Systems,* 107(2), 1999, pp.235.

[14] Y.-Y. Hsu, S.-M. Tse and B. Wu, "A New Approach of Bivariate Fuzzy Time Series Analysis to the Forecasting of a Stock Index", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 11(6), 2003, pp. 671-690.

[15] H.-K. Yu and K.-H. Huarng, "A Bivariate Fuzzy Time Series Model to Forecast the TAIEX", *Expert Systems with Applications,* 34(4), 2008, pp. 2945-2952.

[16] T. A. Jilani, and S. M. A. Burney, "Multivariate Stochastic Fuzzy Forecasting Models", *Expert Systems with Applications,* 35(3), 2008, pp. 691-700.

[17] C.-H. Cheng, G.-W. Cheng and J.-W. Wang, "Multi-Attribute Fuzzy Time Series Method Based on Fuzzy Clustering", *Expert Systems with Applications,* 34(2), 2008, pp. 1235-1242.

[18] K. Huarng, "Heuristic Models of Fuzzy Time Series for Forecasting", *Fuzzy Sets and Systems,*123(3), 2001, pp. 369-386.

[19] K. Huarng, "Effective Lengths of Intervals to Improve Forecasting in Fuzzy Time Series", *Fuzzy Sets and Systems*, 123(3), 2001, pp. 387-394.

[20] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press, 1995.

[21] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1999.

[22] K. Huarng and H.-K. Yu, "Ratio-Based Lengths of Intervals to Improve Fuzzy Time Series Forecasting", *IEEE Transactions on Systems, Man, and Cybernetics Part B,* 36(2), 2006, pp. 328-340.

[23] L.-W. Lee, L.-H. Wang and S.-M. Chen, "Temperature Prediction and TAIFEX Forecasting Based on Fuzzy Logical Relationships and Genetic Algorithms", *Expert Systems with Applications,* 33, 2007, pp. 539-550.

[24] L.-W. Lee, L.-H. Wang, S.-M. Chen and Y.-H. Leu, "Handling Forecasting Problems Based on Two Factors High-Order Fuzzy Time Series", *IEEE Transactions on Fuzzy Systems,* 14(3), 2006.

[25] K. Huarng and H.-K. Yu, "A Dynamic Approach to Adjusting Lengths of Intervals in Fuzzy Time Series Forecasting", *Intelligent Data Analysis,* 8(1), 2004, pp. 3-27.

[26] K. Huarng and H.-K. Yu, "A Type 2 Fuzzy Time Series Model for Stock Index Forecasting", *Physica A,* 353, 2005, pp. 445-462.

[27] J. M. Zurada, *Introduction of Artificial Neural Systems,* St. Paul: West Publishing, 1992.

[28] C. H. Aladag, M. A. Basaran, E. Egrioglu, U. Yolcu, V. R. Uslu, "Forecasting in High Order Fuzzy Time Series by Using Neural Networks to Define Fuzzy Relations", *Expert Systems with Applications,* 36, 2009, pp. 4228-4231.

[29] Pinar E., Paydas K., Seckin G., Akilli H., Sahin B., Cobaner M., Kocaman S., Akar M. A., 2010, Artificial neural network approaches for prediction of backwater through arched bridge constrictions, Advanced in Engineering Software, (41), pp. 627-635.

[30] Ghorbanian K., Gholamrezaei M., 2009, An artificial neural network approach to compressor performance prediction, Applied Energy, (86), pp. 1210-1221.

[31] Park J., Sandberg I. W., 1991, Universal approximation using radial basis function networks, Neural Comput., 3 (2), pp. 246-257.

[32] Powell M. J. D., 1987, Radial basis function for multivariate interpolation, in: J. C. Mason, M. G. Cox (Eds.), A Review Algorithm for the Approximation of Functions and Data, Clarendon, Oxford, UK.

[33] Poggio T., Girosi F., 1990, Networks for approximation and learning, Proc. IEEE 78 (9), pp. 1481-1497.

[34] Celikoglu H. B., 2006, Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modeling, Mathematical and Computer Modelling, 44, pp. 640-658.

[35] Moody T., Darken C., 1989, Fast learning in networks of locally tuned processing units, Neural Computation, 1, pp. 281-194.

[36] Renals S, Rohwer R., 1989, Phoneme classification experiments using radial basis functions, In Proceedings of International Joint Conference on Neural Networks, Washington, DC, pp. 461-467.

[37] Bagherieh A. H., Hower J. C., Bagherieh A. R., Jorjani E., 2008, Studies of the relationship between petrography and grindability for Kentucky coals using artificial neural network, International Journal of Coal Geology, 73, pp. 130-138.

[38] Specht D., 1991, A general regression neural network, IEEE Trans. Neural Networks, 2 (6), pp. 568-576.

[39] Wasserman P.D., 1993, Advanced Methods in Neural Computing, New York, Van Nostrand Reinhold, NY, USA.

[40] Mostafa M. M., 2011, A neuro-computational intelligence analysis of the global consumer software piracy rates, Expert Systems with Applications, 38, pp. 8782-8803.

[41] Mostafa, M. M., 2010, Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait. Expert Systems with Applications.

[42] M. Mammadov, E. Tas, "An Improved Version of Backpropagation Algorithm with Effective Dynamic Learning Rate and Momentum", *Proceedings of the 9th WSEAS International Conference on Applied Mathematics,* Istanbul, Turkey, May 27-29, 2006, pp. 356-361.

[43] M. Memmedli, O. Ozdemir, "A Comparison Study of Performance Measures and Length of Intervals in Fuzzy Time Series by Neural Networks", *Proceedings of the 8th WSEAS International Conference on System Science and Simulation in Engineering,* Genova, Italy, October 17-19, 2009, pp. 211-214.

[44] M. Memmedli, O. Ozdemir, "An Application of Fuzzy Time Series to Improve ISE Forecasting", *WSEAS Transactions on Mathematics,* Issue 1, Volume 9, January 2010, pp. 12-21.

[45] D. Aydın, M. Mammadov, "A Comparative Study of Hybrid, Neural Networks and Nonparametric Regression Models in Time Series Prediction", *WSEAS Transactions on Mathematics,* Issue 10, Volume 8, October 2009, pp.593-603.

Memmedaga Memmedli was born in Azerbaijan in 1947. He received a diploma at Baku State University in 1971. He received a PhD diploma from Russian Academy of Science in 1977. From 1985-2006, he has worked as a Associate Professor; from 2007, as a Professor Dr. at Department of Statistics at Anadolu University, Eskisehir, Turkey. Now, he is head of Department of Statistics, Faculty of Science, Anadolu University, Eskisehir, Turkey.
He has published many journals and conference papers in his research areas. His research interests are Game Theory, Control Theory, Optimization, Artificial Neural Networks, Fuzzy Modeling and Nonparametric Regression Analysis.

Ozer Ozdemir – Ozer Ozdemir was born in Turkey in 1982. He received his B.Sc., M.Sc. degrees in statistics in the Department of Statistics at Anadolu University, Turkey, respectively in 2005 and in 2008. He is currently doing his Ph.D. in the same place. He has worked as a Research Assistant from 2006-2008 and as a Lecturer from 2008 in the Department of Statistics at Anadolu University, Turkey.

He has published over 20 international conference papers and journals in his research areas. His research interests include Simulation, Artificial Neural Networks, Fuzzy Logic, Fuzzy Modeling, Time Series, Computer Programming, Statistical Software and Computer Applications in Statistics.