Assist. Professor Oguz KAYNAR, PhD
Department of Management Information Systems
Cumhuriyet University, Turkey,
E-mail: okaynar@cumhuriyet.edu.tr

FORECASTING INDUSTRIAL PRODUCTION INDEX WITH SOFT COMPUTING TECHNIQUES

Abstract: *The production industry has a dynamic structure that is affected by socioeconomic factors such as economical policies, stabilization and competition increasing with globalization and also incorporates complex manufacturing processes. One of the most important indicators that demonstrates the current situation and development of manufacturing industry in time is manufacturing index. Effective planning for manufacturing industry depends on accurate and realistic predictions for future of sector. Soft computing techniques such as artificial neural networks (ANN) and fuzzy inference systems (FIS) draw attention along with classical time series in prediction applications recently. In this study, monthly production index is forecasted by using adaptive neuro- fuzzy inference systems (ANFIS) and two different learning of algorithms of ANN models (multi layer perceptron -MLP and radial basis function network-RBFN). This index was also predicted by SARIMA model which is one of the classical time series analysis method and prediction performances of classical method and soft computing methods were then compared.*

Keywords: *ANN, RBF, MLP, ANFIS, Forecasting.*

JEL Classification: C45, C22, C63

INTRODUCTION

The most important indicator for monitoring the production in industry in monthly and quarterly terms and directing the decisions about investments is industrial production index. Industrial production index forms a stable series from the point of representing the changes of production in the sector on one hand and allow following the situation of sub sectors in the way of public and private sectors on the other hand. The aim of generating and publishing the industrial production index is to see the course of production industry in time, to follow the changes between terms and the cyclical development of economics and to meet decision making bodies' and scientists' needs in this area

Industrial production index is composed of three basic subsectors: mining, manufacturing industry and energy sector. Food, textile, oil productions and chemistry sub sectors which have an important part in industrial production is 85% of the index. Information for production industry is compiled by using monthly/quarterly polls conducted in all public enterprises in production industry and 89% of private business offices which employ 10 or more employees. The data about basic sectors of industry and sub sectors of production industry is useful from the point of seeing which sectors are in trouble and which try to survive. Besides, by monitoring these indexes, it is possible to follow and evaluate the economics in detail.

In addition to classical time series, Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference System (ANFIS) are used in forecasting of economical and financial series successfully in recent years. ANN, which has an ability of learning from data set and working with lack of data, can model nonlinear relationships besides linear relationships successfully. In contrast to statistical approaches it does not need any assumptions such as normality, linearity and stationarity on data set. It can model many different formed structures and approximate any form of functions in certain accuracy so it is named general function approximator (Cybenko,1989; Hornik et al.,1989; Hornik,1991). Hence ANN becomes an alternative method used in time series analysis. A wide compilation about studies which use ANN in prediction of time series is made by Zhang et al. (1998).

Another new technique which can model complex and nonlinear relationships on input output data set is ANFIS. ANFIS is a system which models relationships between input and output data set by conveying the opinion and thoughts of experts to system in a mathematical way by using fuzzy logic and fuzzy set theory. The biggest disadvantages of fuzzy logic are not having a general approach to determine input membership functions, need of determine these functions through experiment by experts and taking much more time. ANFIS which is a combination of fuzzy system and ANN deals with this problem by using the learning ability of ANN. ANFIS is proposed by Jang first and is used to model Mackey-Glass chaotic time series (Jang, 1993). Then it gains popularity and is used to model time series in different areas such as energy, engineering, business and finance (Kaynar et al., 2011; Yilmaz and Kaynar, 2010;Yilmaz et al.,2010; Ying and Pan, 2008; Azadeh et al., 2009; Sfetsos, 2000; Nayak et al., 2004; Chang and Chang, 2006; Cheng et al., 2009; Atsalakis and Valavanis, 2009).

ANN and ANFIS which are frequently used in prediction of exchange rate, inflation, recession, national income, stock certificate and stock market index are used in a few studies related with forecasting of production index. Moody et al. forecast industrial production index for USA by using three different methods: ANN, linear regression analysis and a classical time series analysis method (ARIMA) and indicate that ANN outperforms (Moddy et al., 1993). Utans et al. (1995) forecast production index by using 480 observations of a variety of economical and financial time series from 1950 January to November 1989 by means of ANN. 48 different time series, which are formed of different economical and financial indicators and can be used in input of network, are examined by means of sensitive based reducing algorithm and tried to ascertain efficient time series to determine production index (Utans et al., 1995). Dilli and Wang (2002) predict the level of production industry of China with classical time series analysis methods in their study (Dilli and Wang, 2002). They use ANN to predict industrial product index in another study. By creating different number of three-layers ANN model, they compared the results of the model having the best performance among these with ARIMA model. They indicate that the RMSE performance criteria of the model having the best performance is 49% better than ARIMA model (Dili ve Wang, 2003). Atsalakis et al. predict Greek industry index for the sectors which don't cover metal industry with ANFIS method (Atsalakis et al., 2009). Different models are created by means of different type and number of membership functions and predictions are made by using these models. The model having minimum error in different ANFIS models is determined and it is compared with AR and ARMA models by means of MAPE, MAE and RMSE performance criteria. For all criteria AR and ARMA models outperforms.

In this study Multilayer Perceptron (MLP), Radial Basis Function Network (RBFN) and ANFIS are used to forecast industrial production index. The results obtained by using these methods have been compared with the ones by using classical time series analysis method, ARIMA and evaluated performance of soft computing techniques in prediction of industrial production index. In the second section information about methods that are used in this study has been given. In third section a model has been created for each method and predictions have been made by means of these models. Conclusions have been discussed in the last section

2. Overviews of the Models

2.1 Artificial Neural Networks

Artificial neural networks are data processing systems devised via imitating brain activity and have performance characteristics such as biological neural networks. ANN has many important capabilities such as learning from data, generalization, working with unlimited number of variable. ANN is typically composed of several layers of many computing elements are called nodes. Each node receives an input signal from the other nodes or external inputs. Then after processing the signals locally through a transfer function, it outputs a transformed signal to other nodes or final result (Zhang et al, 1998). Different neural network models can be issued in time series analysis. The most commonly used among these are multilayer feed forward artificial neural network (Multiple Layer Perceptron-MLP) and radial basis function networks ((RBFN) which are also used in this study.

2.1.1 Multi Layer Feed Forward Artificial Neural Networks (MLP)

Neurons and layers are organized in a feed forward manner in MLP networks. In MLP, first layer is an input layer where external information about problem wanted to be solved is received. The last layer is output layer that data manipulated in network is obtained. The layer exists between input and output layers is called hidden layer. There can be more than one hidden layer in MLP networks. Figure 1 shows the architecture of typical MLP network.

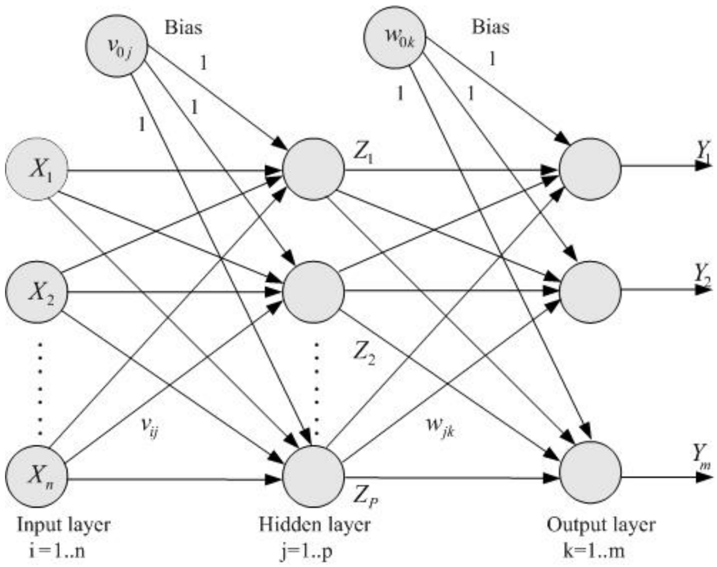

Figure 1: Structure of multi layer perceptron neural network

Technically, an ANN’s main point of view is learning the structure of sample data set, to generalize it. For doing this, network is made to be able to generalize by training with samples of the case (Öztemel 2003, s.30). During the training process, input patterns or examples are presented to the input layer of a network. The activation values of the input nodes are weighted and accumulated at each node in the hidden layer. The weighted sum is transferred by an appropriate transfer function to produce the node's activation value. The output of hidden layer j can be calculated as follows:

$$z_j = f\left(v_{0j} + \sum_{i=1}^{N} v_{ij} x_i\right) \tag{1}$$

where ($x_1, x_2 .... x_n$) are inputs values, v_{ij} is weight that connects i th input to hidden node j, v_{0j} is bias, f is nonlinear transfer function such as sigmoid or hyperbolic tangent functions.

Output of hidden nodes becomes the input of the output layer. Finally an output value of the k th node in the output layer is obtained as followed:

$$y_k = f\left(w_{0k} + \sum_{j=1}^{p} w_{jk} z_j\right) \tag{2}$$

Similarly z_j is j th hidden layer output, w_{jk} is weight that connects hidden layer node j to output layer k. Generally linear transfer function is used in the output layer nodes of network.

The aim of training is to minimize the differences between the ANN output values and the known target values for all training patterns. The most popular algorithm for training is the well-known back-propagation which is basically a gradient steepest descent method with a constant step size (Zhang et al., 1999). This algorithm is named as back-propagation because it tries to reduce errors from output to input backwardly. In supervised learning algorithms, a sample data set that consists of input and output values is given to network for training. The given target output values are named as supervisor or teacher in ANN literature. In supervised learning algorithms, weights are adjusted by minimizing error function given in Equation 3 in training level.

$$E = \frac{1}{2}\sum_{k=1}^{m}(y_k - t_k)^2 \tag{3}$$

where y_k represents the output that network produce and t_k as real output value. Connection weights are updated for minimizing error. So it is aimed that the network produces closest output values to real output values. Details of back-propagation algorithm can be examined in Fauset (Fauset 1994, s.294-296).

2.1.2 Radial Basis Function Networks

The radial basis function network (RBFN) is traditionally used for strict interpolation problem in multi-dimensional space and has similar capabilities with MLP neural network which solves any function approximation problem (Park and Sandberg, 1993). Advantages of RFBN are that it can be trained in a short time according to MLP (Moody and Darken, 1989) and approximate the best solution without dealing with local minimums (Park and Sandberg, 1991). Additionally, RBFN are local networks compared to the feed-forward networks that perform global mapping. This means that RBFN uses a single set of processing units, each set is most receptive to a local region of the input space (Xu et al., 2003). Because of its features mentioned above, RBFN were used as alternative neural network model in applications such as function approximation, time series forecasting as well as classifying task in recent years (Bianchini et al., 1995; Chen et al., 1991; Sheta and Jong, 2001; Harpham and Dawson, 2006; Rivas et al., 2004; Yu et al., 2008; Foody, 2004; Sarimveis et al., 2006; Zhang et al., 2007).

The structure of RBFN is composed of three layers as can be seen in Figure 2. The main distinction between MLP and RBFN is that RBFN have one hidden layer which contains nodes called RBF units. As its name implies, radially symmetric basis function is used as activation functions of hidden nodes.

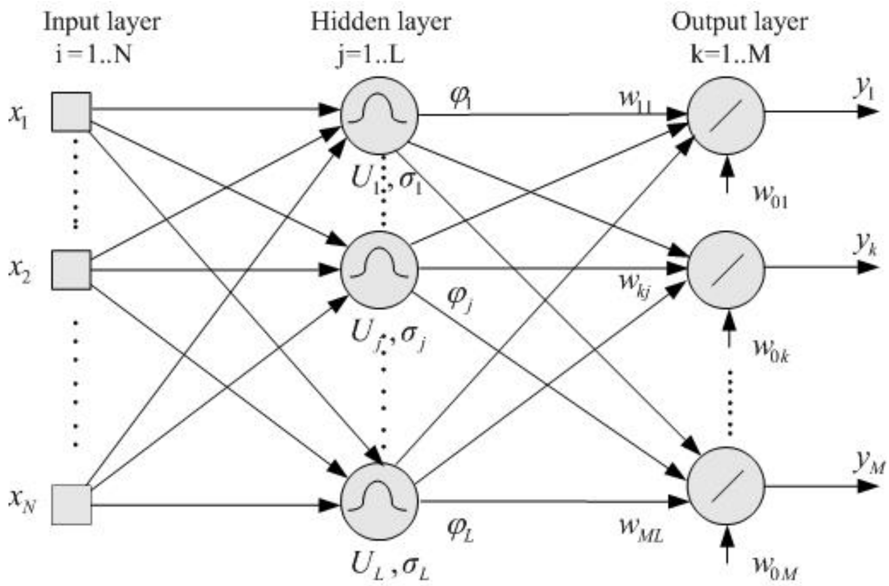

Figure 2: Structure of radial basis function neural network

The input layer serves only as input distributor to the hidden layer. Differently from MLP, the values in input layer are forwarded to hidden layer directly without being multiplied by weight values. The hidden layer unit measures the distance between an input vector and the centre of its radial function and produces an output value depending on the distance. The center of radial basis function is called reference vector. The closer input vector is to the reference vector, the more the value is produced at the output of hidden node. Though a lot of radial basis functions are suggested for using in hidden layer (Gausian, Multi-Quadric, Generalized Multi-Quadric, Thin Plate Spline), Gaussian function is the most widely used in applications. Chen et al. indicate that the choice of radial basis function used in network does not effect the networks performance significantly (Chen et al., 1991). The activation function of the individual hidden nodes defined by the Gaussian function is expressed as follows:

$$\varphi_j = e^{\left[-\frac{\|X-C_j\|^2}{\sigma_j^2}\right]} \qquad j = 1,2....,L \tag{4}$$

where φ_j denotes the output of the j th node in hidden layer, $\|.\|$ is Euclidian distance function which is generally used in applications, X is the input vector, C_j is center of the j th Gaussian function, σ_j is radius which shows the width of the Gaussian function of the j th node and L denotes the number of hidden layer nodes.

In the next step, the neurons of the output layer perform a weighted sum using the hidden layer outputs and the weights which connect hidden layer to output layer. Output of network can be presented as linear combination of the basis functions:

$$y_k = \sum_{j=1}^{L} \varphi_j w_{kj} + w_{k0} \tag{5}$$

where w_{kj} is the weight that connects hidden neuron j and output neuron k, w_{k0} is bias for the output neuron.

Training of RBF networks contains process of determining of centre vector (C_j), radius value (σ_j) and linear weight values (w_{kj}). Generally two stage hybrid learning algorithm is used to train RBF networks. In the first stage of hybrid learning algorithm, center parameters of RBFs in hidden layer are determined by using unsupervised clustering algorithms such as K-means or randomly selected from given input data set. Width parameters of RBF are also important and very narrow width will lead to over fitting, and radius wider than necessary can give even worst results. This value can be determined by user or computed with equation below (Haykin 1999, p.321):

$$\sigma = d_{\max} / \sqrt{2M} \tag{6}$$

where $d_{\max}$ is maximum distance of centers, M is the number of clusters. After centers and the radius have been fixed in first stage, output layer weights can be calculated by using ordinary least square algorithm at one stage solving linear equation system presented as below :

$$W = \varphi^{\dagger} Y \tag{7}$$

where $\varphi^{\dagger}$ is the generalized inverse of φ.

2.2 Neuro-Fuzzy Modeling and ANFIS

The ANFIS is a FIS implemented in the framework of an adaptive fuzzy neural network, and is a very powerful approach for building complex and nonlinear relationship between a given set of input and output data (Jang, 1993; Jang et al., 1997). FIS utilizes human expertise and is composed of three conceptual components. The first component is a rule base, which contains a selection of fuzzy rules. The second component is a database which defines the membership functions (MF) used in the fuzzy rules. The third component is a reasoning mechanism, which performs the inference procedure upon the rules to derive an output. FIS implements a nonlinear mapping from its input space to the output space by using a number of fuzzy if-then rules, each of which describes the local behavior of the mapping. The parameters of the if-then rules (referred to as antecedents or premises in fuzzy modeling) define a fuzzy region of the input space, and the output parameters (also consequents in fuzzy modeling) specify the corresponding output (Nayak et al, 2004). The main problem in FIS is that there is no systematic procedure to define the membership function parameters. Hence, the efficiency of the FIS depends on the estimated parameters. The construction of the fuzzy rule necessitates the definition of premises and consequences of fuzzy sets. On the other hand, an ANN has the ability of learning from input and output pairs and adapting them to an interactive manner. Main idea of combining fuzzy system and neural networks is to design an architecture that uses fuzzy system to represent knowledge in interpretable manner, in addition to possessing the learning ability of neural network to optimize its parameter (Jang, 1993). Thus ANFIS eliminates the basic problem in fuzzy system design, defining the membership function parameters and design of fuzzy if-then rules, by effectively using the learning capability of ANN for automatic fuzzy rule generation and parameter optimization (Yurdusev and Fırat, 2009).

Two types of FIS have been widely used in various applications: the Mamdani model (Mamdani and Assilian, 1975) and the Takagi Sugeno model (Takagi and Sugeno, 1985). The difference between two models lies in the consequent part of their rules. The consequence parameter in Sugeno FIS is either a linear equation of premise parameter, called "first-order Sugeno model", or constant coefficient, "zero order Sugeno model" (Jang et al., 1997). For two inputs first-order Sugeno fuzzy model, if –then rules can be expressed as:

$$\text{Rule i: If } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } f_i = p_i x + q_i y + r_i \tag{8}$$

where p_i , q_i and r_i are linear parameters of consequent part of first-order Sugeno model.

As it can be seen in Figure 3, five layers are used to construct this inference system. Each layer contains several nodes having different shapes and functions. There are two types of nodes which are fixed nodes denoted by circles and square shaped adaptive nodes. An adaptive node is the node whose parameters are adjusted in the training phase of the system. Functions of each layer are briefly described as follow:

Layer1 (Input layer): Every node of this layer is adaptive node which generates membership grades related with appropriate fuzzy sets using membership function.

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x) \quad i = 1, 2 \\ O_{1,i} &= \mu_{B_{i-2}}(y) \quad i = 3, 4 \end{aligned} \tag{9}$$

where x , y crisps inputs of node i and A_i , B_i are linguistic labels, μ_{A_i} , μ_{B_i} are membership functions. Different membership functions such as trapezoidal, triangular, bell-shaped, gaussian function, etc. can be used to determine the membership grades. Bell-shaped function is the most popular among these and it is used in this study. This function is given in Equation 10:

$$\mu(x) = \frac{1}{1 + \left(\dfrac{x - c_i}{a_i} \right)^{2b_i}} \tag{10}$$

$\{a_i, b_i, c_i\}$ is parameter set of membership function and called premise parameters. These parameters are tuned during the training phase.

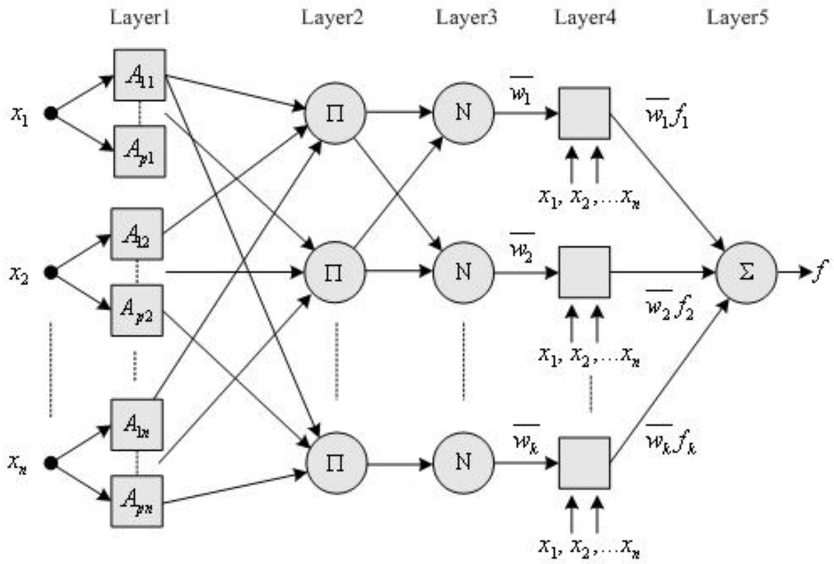

Figure 3: Structure of ANFIS having two inputs and four rules

Layer 2 (Rule layer): In this layer, all nodes are fixed and labeled as Π. In this layer, the T-Norm operator (AND or product) is applied to get one output that represents the results of the antecedent for a fuzzy rule, that is, firing strength. Firing strength is weight degree of if-then rule in the premise part and it indicates the shape of the output function for that rule. The outputs of the second layer are the products of the corresponding degrees obtaining from previous layer and calculated as:

$$O_{2,i} = w_i = \mu_{Aj}(x) \times \mu_{B_j}(x) \qquad i = 1,..,4\ ;\ j = 1,2 \tag{11}$$

Layer 3 (Normalize layer): Every node in this layer is a fixed node, it is marked by a circle and labeled N, with the node function to normalize the firing strength by calculating the ratio of the i th node firing strength to the sum of all rules' firing strength. Consequently, $\overline{w_i}$ is given as the normalized firing strength:

$$O_{3,i} = \overline{w_i} = \frac{w_i}{\sum_{k=1}^{4} w_k} \quad i = 1,..,4 \tag{12}$$

Layer 4 (Consequent layer): The node function of the fourth layer computes the contribution of each *i*th rule's toward the total output and the function defined as:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i + q_i + r_i) \quad i = 1,..,4 \tag{13}$$

where $\overline{w_i}$ is the i th node's output from the previous layer. $\{p_i, q_i, r_i\}$ is an adjustable parameter set that is referred as consequent parameters and also the coefficients of linear combination in Sugeno inference system.

Layer 5 (Output layer): The single node computes the overall output by summing all the incoming signals. Accordingly, the defuzzification process transforms each rule's fuzzy results into a crisp output in this layer.

$$O_{5,i} = \sum_{i=1}^{4} \overline{w_i} f_i = \frac{\sum_{i=1}^{4} w_i f_i}{\sum_{i=1}^{4} w_i} \tag{14}$$

ANFIS uses either gradient descent alone or hybrid learning algorithm, which is the combination of gradient descent and least square estimation (LSE), to adapt the parameters. Gradient method can be used to identify all the parameters in an adaptive network but this method is generally slow and likely to become trapped in local minima. Therefore, Jang (1993) proposed two pass hybrid learning algorithm to identify parameters. In the forward pass of hybrid learning algorithm, antecedent parameters (a_i, b_i, c_i) are fixed and node outputs go forward until layer 4 and the consequent parameters (p_i, q_i, r_i) are identified by the least square estimate approach.

2.3 ARIMA and SARIMA Models

Box-Jenkins model is a statistical forecasting method that is used in prediction and controlling of single variable time series in short and middle term successfully for a long time. Box-Jenkins approach becomes popular with these features, such as being able to be used with different time series, having strong teorical basics and success in applications (Frechtling 1996, p.96) Box-Jenkins approach is an integration of Autoregressive (AR) and Moving Average (MA) prediction methods can be applied stationary series. Differentiating method is used to provide stationary nonstationary time series. ARIMA, which expresses the d order differentiated variable in terms only of its own past value along with current and past errors, is a combination of Autoregressive (AR) and Moving Average (MA) processes. Models are represented as ARIMA (p, d, q) generally. Here, p and q are degrees of Autoregressive (AR) Model and Moving Average (MA) respectively and d is degree of differentiating.

In practice, mostly time series contain a seasonal periodic component that appear by natural and socio-economic and repeated for every observation s (Chatfield 2003, s.66). Box-Jenkins proposed a model named as seasonal ARIMA (SARIMA) that is generalized version of ARIMA with the aim of modeling seasonal components (Franses 1996, s.41). A SARIMA model consists of seasonal and non-seasonal parts. Seasonal and non-seasonal parts have their own autoregressive and moving average parameters P, Q and p, q. Seasonal differentiating number is defined as D and non-seasonal d in SARIMA model. Seasonal ARIMA models are expressed in the shape of ARIMA(p,d,q)(P,D,Q)s and can be written as in Equation 15 (Kuvulmaz et al.,2005):

In the model Φ and Θ represent B^s 's polynomial degree of P and Q , φ and θ B's polynom in degree of p and q. s is seasonal distance and s=12 for monthly data, s=4 for quarterly data. $(1-B^s)^D$ and $(1-B)^d$ are seasonal and non-seasonal differencing operators respectively. ε_t represents error.

$$\begin{aligned}
&\varphi_{\mathrm{p}}(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t \\
&\varphi_{\mathrm{p}}(B) = 1-\varphi_1 B-\varphi_2 B^2-.....-\varphi_{\mathrm{p}} B^p \\
&\Phi_P(B) = 1-\Phi_s B^s-\Phi_{2s} B^{2s}-.....-\Phi_{Ps} B^{Ps} \\
&\theta_q(B) = 1-\theta_1 B-\theta_2 B^2-.....-\theta_q B^q \\
&\Theta_Q(B) = 1-\Theta_s B^s-\Theta_{2s} B^{2s}-.....-\Theta_{Qs} B^{Qs}
\end{aligned} \tag{15}$$

In seasonal ARIMA models, forecasting happens in four levels. Firstly, stationary of series is controlled and it is providing stationary of non-stationary series appropriate Box-Jenkins model that is determined. In the second level parameters of this model is predicted. Then, models correspondence with data set is tested by statistical methods. In third level, if the model determined in previous level is approved it is passed to last level, if not; it is turned into first level. In the last level the most appropriate model is used for prediction

3. Data Processing, Application and Results

Data used in this study is monthly time series that shows production index for Turkey between January 1992 and December 2008 and is obtained from Turkish Statistical Institute. It can be seen that series has an increasing trend. In addition, it is not stationary if graph given in Figure 4 is examined.

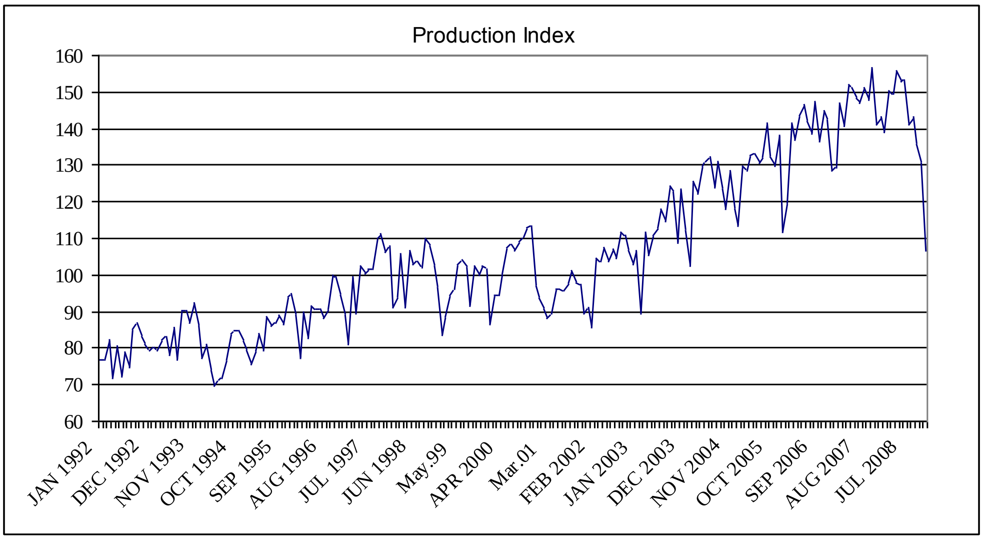

Figure 4. Production index

Starting with Box Jenkins methodology, firstly, stationary of series is examined to determine an appropriate ARIMA model for data set. In this step, sample autocorrelations (SAC) and sample partial autocorrelations (SPAC) of the historical data were plotted to observe the pattern. Three periodical datas were selected to illustrate the plot. The result is shown in Fig. 5a. Based on Fig. 5a, it could be observed that the correlogram of time series is likely to have seasonal cycles especially in SAC which implied level non-stationary. Therefore, the regular differencing and seasonal differencing were applied to the original time series as presented in Fig. 5b and 5c.

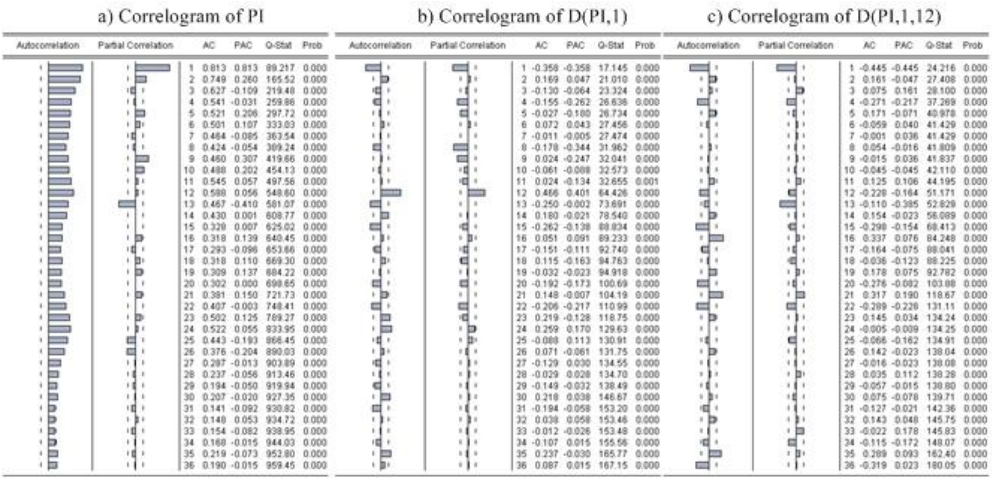

Figure 5. SAC and SPAC graph of Production index

a) Original PI b) Regular differencing PI c) regular differencing and seasonal differencing PI

An Augmented Dickey-Fuller (ADF) test was performed to determine whether a data differencing is needed and result was given in Table 1. The null hypothesis of the Augmented Dickey-Fuller t-test is:

• H0: $\theta = 0$ then the data needs to be differenced to make it stationary series, versus the alternative hypothesis of

• H1: $\theta < 0$ then the series is stationary and it doesn't need to be differenced.

The results have been compared with the 1%, 5%, and 10% critical values to indicate non-rejection of the null hypothesis. The ADF test statistic values for Trend, Trend and intercept and None have t-Statistic values of -1.228,-2.572 and 0.932, respectively. P values of models are 0.660, 0.294 and 0.905. The critical values reported at 1%, 5%, and 10%. It can be seen in Table 1 for different models. These values show that α value is greater than the critical values that provide evidence to accept the null hypothesis, then the time series need to be differencing

Table 1. ADF test Statistic PI

		Trend	Prob.	Trend and Intercept	Prob.	None	Prob.
Test statistic		-1.228	0.660	-2.572	0.294	0.932	0.905
Test critical values	1% level	-3.486		-4.037		-2.584	
	5% level	-2.886		-3.448		-1.943	
	10% level	-2.580		-3.149		-1.614	

The regular differencing and seasonal differencing have been applied to the original time series. The ADF test results are showed in Table 2. P-value for three models is 0.000. While the probability value of 0.000 provided evidence to reject the null hypotheses. It shows the stationarity of the series.

Table 2. ADF test statistic regular and seasonal differencing PI

		Trend	Prob.	Trend and Intercept	Prob.	None	Prob.
Test statistic		-17.386	0.000	-17.328	0.000	-17.457	0.000
Test critical values	1% level	-3.487		-4.0377		-2.585	
	5% level	-2.886		-3.448		-1.944	
	10% level	-2.580		-3.149		-1.6165	

Different ARIMA models have been applied to find the best fitting model .These models are given in Table 3. The most appropriate model was selected by using BIC and AIC, R^2, MAPE values. ARIMA(0,1,4)(1,1,1)12 model which has the biggest R^2 and smallest MAPE, RMSE and BIC values is chosen as the best model. Parameters and statistic of the model can be seen in Table 4.

Table 3. ARIMA models

	R-squared	RMSE	MAPE	BIC
ARIMA(0,1,0)(0,1,0)12	.897	6.546	5.085	3.787
ARIMA(0,1,0)(0,1,1)12	.902	6.375	4.997	3.734
ARIMA(0,1,0)(1,1,1)12	.908	6.196	4.850	3.706
ARIMA(0,1,1)(1,1,1)12	.929	5.459	4.218	3.482
ARIMA(1,1,1)(1,1,1)12	.931	5.410	4.087	3.493
ARIMA(0,1,2)(1,1,1)12	.930	5.436	4.135	3.503
ARIMA(0,1,3)(1,1,1)12	.933	5.336	4.009	3.495
ARIMA(0,1,4)(1,1,1)12	**.940**	**5.036**	**3.763**	**3.379**
ARIMA(0,1,5)(1,1,1)12	.935	5.269	3.958	3.528

Table 4. ARIMA(0,1,4)(1,1,1)12 model and statistic

		prediction	Std. error	t	Prob.
Difference		1			
MA	Lag 1	0.523	0.072	7.244	0.000
	Lag 2	-0.170	0.072	-2.360	0.019
	Lag 4	0.318	0.064	4.997	0.000
AR. Seasonal	Lag 1	0.404	0.126	3.215	0.002
Seasonal Difference		1			
MA. Seasonal	Lag 1	0.915	0.153	5.987	0.000

Diagnostic check has been made for the selected model. Correlogram and the residual plots have been investigated. There is not a significant correlation between error terms and all of them have random values. Additionally, Ljung-Box shows that there is no significant correlation between lags. So it is decided that the model is valid. Equation of this model is given below:

$$\begin{aligned} y_t &= y_{t-1} + y_{t-12} - y_{t-13} + 0.404\,y_{t-12} - 0.404\,y_{t-13} - 0.404\,y_{t-24} + 0.404\,y_{t-25} \\ &\quad - 0.523e_{t-1} + 1.70e_{t-2} - 0e_{t-3} - 0.318e_{t-4} - 0915e_{t-12} + 0.915x0{,}523e_{t-13} \\ &\quad - 0.915x1.70e_{t-14} + 0.915x0e_{t-15} + 0.915x0.404e_{t-16} + e_t \end{aligned} \tag{16}$$

The last 14 observation of data set are used to test predictions of MLP, RBF and ARIMA models. 80% of the rest of the data is used for training and 20% for validation in MLP and ANFIS models. For RBF, all data except test data is used for training. The data set is normalized to range [-1, 1] to prevent saturation of hidden nodes before feeding into the neural network and ANFIS models. MATLAB ‘premnmx’ function is used to normalize data.

In this study, software code is developed using Matlab program to create artificial neural network and ANFIS models. All created MLP models within the study have three layers architecture. By means of changing the number of neurons in input layer from 1 to 12 and the number of neurons used in hidden layer from 1 to 10, 120 different MLP neural network models are obtained. Linear transfer function is used in the output layer node, whereas tangent-sigmoid transfer function is preferred in the hidden layer nodes. Levenberg-Marquardt back-propagation algorithm is used to train all MLP models. Network training parameters, epoch number and goal error rate, which stop training, are chosen 400 and 0.01 respectively. Besides, to achieve a good accuracy and avoid over fitting, validation vectors are used to stop training early, if the network performance on the validation vectors fails to improve. Feeding network with training data, training process is implemented and the artificial neural network model has minimum squared error (MSE) for testing data is chosen from 120 models. MLP model has 2 input neurons and 3 hidden neurons is determined as the most appropriate model. Some of these models are shown in table 5

Three different approaches are used to make predictions with RBF networks. In the first approach, the number of neurons in hidden layer is set to the number of observation in training data set, reference vectors are equalized with input values in data set. This model was called exact RBF. In the second approach, growing method is used to determine number of hidden layer nodes. 30 models are created by increasing the number of neurons in hidden layer from 1 to 30. The center of radial functions in hidden layer is determined by choosing from training data set randomly. In the last approach k-means clustering algorithm is used to determine centers of RBF. It is the same as the previous approach cluster number is changed 1 to 30 and k-means clustering algorithm is applied. The appropriate number of input nodes is not easy to determine because it is not known the value of series at time t is affected by how many past lag values. So, for all approaches the number of input neurons is changed from 1 to 12. As a result 12 models are tried for the first approach and 12x30 models for the second and the third one. The RBF model having minimum MSE for test data is chosen as the best performance model from created models. The best RBF model having 2 inputs and 16 hidden neurons is given in Table 6 and this model's center is determined by k-means clustering algorithm.

Table 5. Some of MLP models and their performance criteria

	Input neuron	Hidden neuron	MAPE(%)	MSE
MLP_Model1	1	1	5.74	92.3423
MLP_Model2	**2**	**4**	**5.21**	**70.1890**
MLP_Model3	3	1	6.10	106.6444
MLP_Model4	4	1	5.67	98.8320
MLP_Model5	5	9	5.93	99.0534
MLP_Model6	6	1	6.10	122.1384
MLP_Model7	7	10	5.63	103.4808
MLP_Model8	8	4	6.03	121.4640
MLP_Model9	9	2	5.43	104.1711
MLP_Model10	10	10	5.90	80.9518
MLP_Model11	11	5	5.97	107.8836
MLP_Model12	12	4	6.36	147.9047

Table 6. RBF models and their performance criteria

	Input neuron	Hidden neuron	MAPE(%)	MSE
RBF (exact)	2	190	6.01	117.5564
RBF (center is determined randomly)	2	15	5.42	87.71503
RBF (center is determined k-means)	**2**	**16**	**5.40**	**83.68073**

In constructing ANFIS model, The FIS architecture must be determined before network is trained. This process contains determining number of membership functions of each input variable, rules and values of parameters belong to these functions. There are two functions in Matlab that are called genfis1 and genfis2 to determine initial FIS architecture. Basic difference between two functions is how the rules that partition the input space are created.

Genfis1 make grid partition of input space by using all possible combinations of membership functions of each variable. The most important disadvantage of this method is to produce many rules if it has many parameters to be trained. It is considered that there are N numbers of variables, P numbers of membership functions for each variable and L parameters belong to each membership function, P^N numbers of rules are created, so $P^N(N+1)$ numbers of linear parameters(consequent) and $NxPxL$ numbers of nonlinear parameters (antecent) are needed to be trained. When the number of input variables is big, the data set is small. This situation is curse of dimensionality problem. This causes that parameters are not able to be computed. It is needed to reduce the number of rules to deal with curse of dimension problem. Genfis2 uses subtractive clustering algorithm to reduce the number of rules and creates a rule for each cluster. Moreover, centers of obtained clusters are initial values of membership functions parameters of input variables. In this study, genfis1 and genfis2 rule creating methods are used to determine initial FIS structure and all models are trained by hybrid learning algorithm.

In the FIS models which use genfis1 functions, network structures have different lag and the different number of membership functions which is increased starting from 2 to the certain value in the way that curse of dimension problem does not occur, are created. The model having the best prediction accuracy among these models is selected. This model has two inputs and five membership functions for each input, so there are 25 rules. Models that is created by means of genfis2 are formed similarly by increasing the number of input variables starting from 1 to the highest value in the way that curse of dimension problem does not occur. By changing cluster spread value, used in genfis2 function that determines the rules by means of subtractive clustering algorithm, from 0.1 to 1 step by step with the value of 0.05, sub models having different number of valid rules are created. The Model that has the best performance has four inputs and seven rules. ANFIS model which uses genfis2 function is better than model which uses genfis1 function.

Table 7. ANFIS Models and their statistics

	ANFIS model1 (grid partition)	ANFIS model2 (Sub-clustering partition)
Number of input	2	4
Number of linear parameters	75	35
Number of nonlinear parameters	30	56
Total number of parameters	105	91
Number of fuzzy rules	25	7
MAPE(%)	89.29	64.54
MSE	5.84	4.97

In this study, Mean Absolute Percentage Error (MAPE) and Mean Square Error (MSE) criteria are used to compare various models obtained from ARIMA, MLP, RBF and ANFIS. MAPE is a relative measurement and it is easy to interpret. MAPE is also an independent of scale, reliable and valid (Law and Au, 1999). The smaller the values of MAPE and MSE are, the closer the forecasted values are to the actual values. The results are given in Table 8. It can be seen in the table that MAPE values of ARIMA, MLP, RBF and ANFIS models are calculated as 8.26%, 5.21%, 5.41%, and 4.97% respectively. Similarly, MSE values are 271.07, 70.18, 83.68, and 64.54. The best result can be obtained from ANFIS model. Observation values of data and graph about prediction values are shown in figure 6. Scatter diagrams of model predicted values to observed value are also given in Figure 7.

Table 8. Observed and predicted values for production index

OBSERVED	ARIMA	MLP	RBF	ANFIS
156.7	149.93	147.33	149.55	147.08
140.9	148.85	149.89	150.12	150.42
143.4	138.38	146.09	149.48	145.76
139.2	136.88	142.47	145.13	144.59
150.5	151.49	141.35	144.60	137.66
149.3	146.65	145.06	145.56	145.63
155.8	155.33	147.76	149.66	147.12
152.8	154.74	149.94	150.45	152.84
153.4	153.72	150.49	151.93	150.17
141.1	151.96	149.95	151.13	151.25
143.2	157.69	145.22	148.58	144.48
135.7	154.8	142.45	145.17	142.32
130.4	155.02	139.51	140.85	137.39
106.2	154.62	126.25	131.07	121.17
MAPE	271.07	70.18	83.68	64.54
MSE	8.26%	5.21%	5.41%	4.97%

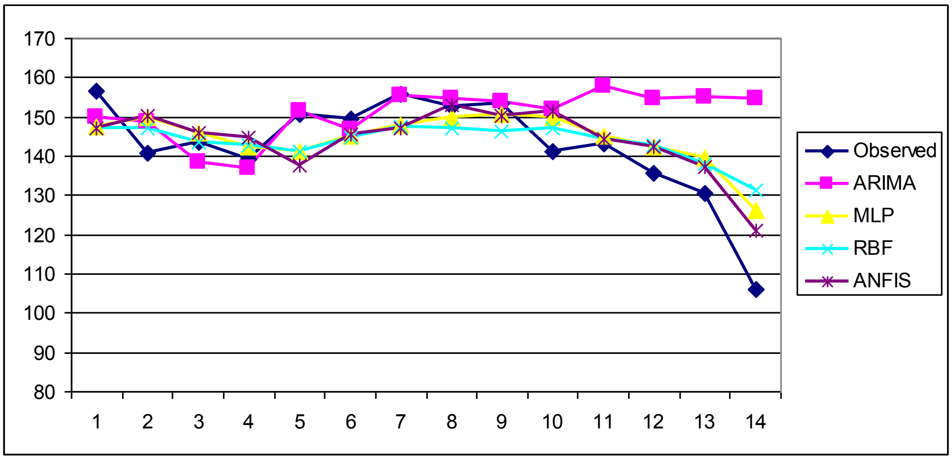

Figure 6. The graph of observed and predicted values for production index

4. Conclusion

In this research, the production index of Turkey is examined using statistical time series analysis, ANN and ANFIS methods. Data set is monthly production index time series that are obtained from Turkey Statistical Institute. Different models are built and best performance model which has lowest RMSE value for training and testing data are determined as final model for each technique. Then predictions are made for the last 15 observations by using final models of each technique and prediction accuracies of these models are compared by using MAPE performance criteria.

The empirical results indicate that MAPE values of final models are very small and they are very close to each other. So, all models provide reasonably good forecasts and they are suitable for predict monthly production index. Among the ANN models, MLP model has slightly better performance than RBF model. In ANFIS models, ANFIS final model which use sub clustering algorithm to generate initial FIS structure has better performance than ANFIS final model using grid partition. So sub clustering algorithm can be used to reduce the number of fuzzy rules and parameters to be calculated, to deal with calculation complexity and curse of dimensionality problem. When it is compared to Neural Networks Model, ANFIS has better performance than RBF neural network and MLP neural network. When the features compared, ANFIS is less time consuming and more flexible than ANN, by employing fuzzy rules and membership functions incorporating with real-world systems, it can be used as an alternative method to neural network.

It can be concluded that both ANN and ANFIS models outperform the ARIMA model when their MAPE and MSE values are compared. Based on the forecast results, when appropriate network structure and enough number of data are used, ANN and ANFIS, which do not depend on meeting statistical conditions such as the type of relation between variables or the type of data distribution, can be used as an alternative method to statistical methods for prediction of economic and financial time series.

REFERENCES

[1] **Atsalakis, G. S., Valavanis, K. P. (2009*), Forecasting Stock Market Short-term Trends Using a Neuro-fuzzy Based Methodology*.** *Expert Systems with Applications, 3*6, 10696–10707;

[2] **Atsalakis, G., Ucenic, C., Skiadas, C. H. (2009), *Time Series Prediction of the Grek Manufacturing Index for the Non-metallic Minerals Sector Using a Neuro-fuzzy Approach (ANFIS).*** Proceeding of Applied Stochastic Models and Data Analysis, 1361–1369;

[3] **Azadeh, A., Saberi, M., Gitiforouz, A., Saberi, Z. (2009), *A Hybrid Simulation-adaptive Network Based Fuzzy Inference System for Improvement of Electricity Consumption Estimation*.** Expert Systems with Applications, 36, 11108–11117;

[4] **Bianchini, M., Frasconi, P., Gori, M. (1995), *Learning without Local Minima in Radial Basis Function Networks*.** *IEEE Trans. Neural Networks,* 6(3), 749-755;

[5] **Chang, F-J., Chang, Y-T. (2006), *Adaptive Neuro-fuzzy Inference System for Prediction of Water Level in Reservoir*;** *Advances in Water Resources,* 29, 1–10;

[6] **Chatfıeld, C. (2003), *The Analysis of Time Series: An Introduction*.** London: CRC Press;

[7] **Chen, S., Cowan, C. F. N., Grant, P. M. (1991), *Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks*.** *IEEE Trans. Neural Networks,* 2(2), 302-309;

[8] **Cheng, C-H., Wei, L-Y., Chen, Y-S. (2009), *Fusion ANFIS Models Based on Multi-stock Volatility Causality for TAIEX Forecasting*.** *Neurocomputing,* 72, 3462–3468;

[9] **Cybenko, G. (1989), *Approximation by Superpositions of a Sigmoidal Function*.** *Mathematical Control Signals Systems*, 2, 303–314;

[10] **Dilli, R. Wang, Y. W. (2002), *An Application of the ARIMA Model for Forecasting the Production Level of Construction Industry*.** *Journal of Harbin Institute of Technology* (New series), 9, 39–45;

[11] **Dilli, R. Wang, Y. W. (2003),** ***Neural Network Forecasting of the Production Level of Chinese Construction Industry***. *Journal of Comparative International Management*, 6, 45 – 64;

[12] **Fausett, L. (1994),** ***Fundamentals of Neural Networks: Architectures, Algorithms and Applications***. *NJ*, USA: Prentice Hall;

[13] **Foody, G.M. (2004) ,** ***Supervised Image Classification by MLP and RBF Neural Networks with and without an Exhaustively Defined Set of Classes;*** *International Journal of Remote Sensing*, 25(15), 3091-3104;

[14] **Franses, P. H. (1996),** ***Periodicity and Stochastic Trends in Economic Time Series.*** *Oxford University Press*;

[15] **Frechtling, D. C. (1996),** ***Practical Tourism Forecasting***. *Oxford: Elsevier*;

[16**] Harpham, C., Dawson, C. W. (2006),** ***The Effect of Different Basis Functions on a Radial Basis Function Network for Time Series Prediction: A Comparative Study***. *Neurocomputing*, 69, 2161–2170;

[17] **Haykin, S. (1999),** ***Neural Networks - A Comprehensive Foundation***. NY, USA: *Prentice Hall*;

[18] Hornik, K., Stinchcombe, M., White, H. (1989), ***Multilayer Feedforward Networks are Universal Approximators****; Neural Networks*, 2, 359–366;

[19] **Hornik, K. (1991),** ***Approximation Capability of Multilayer Feedforward Networks****; Neural Networks*, 4, 251–257;

[20**] Jang, J. S. R. (1993),** ***ANFIS: Adaptive-network-based Fuzzy Inference System;*** *IEEE Trans. Syst., Man, Cybernet*, 23(3), 665–685;

[21] **Jang, J. S. R., Sun, C. T., Mizutani, E. (1997),** ***Neuro-Fuzzy and Soft Computing***. USA: *Prentice Hall*;

[22] **Kaynar, O., Yilmaz, I., Demirkoparan, F. (2011),** ***Forecasting of Natural Gas Consumption with Neural Network and Neuro Fuzzy System***; Energy Education Science and Technology, Part A, *Energy Science and Research*, 26(2), 221 -238;

[23] **Kuvulmaz, J., Usanmaz, S., Engin, S. N. (2005) ,** ***Time-Series Forecasting by Means of Linear and Nonlinear Models*** *; MICAI* , 504-513;

[24]**Mamdani, E. H., Assilian, S. (1975),** ***An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller*** *; International Journal of Man-Machine Studies*, 7(1), 1-13;

[25] **Moody, J., Darken, C. J. (1989),** ***Fast Learning in Networks of Locally-tunes Processing Units.*** *Neural Computation*; 1(2), 281-294;

[26] **Moody, J., Levin, A., Rehfuss, S. (1993),** ***Predicting the U.S. Index of Industrial Production****; Neural Network World*, Special Issue: Proceedings of Parallel Applications in Statistics and Economics '93, 3(6), 791–794;

[27] **Nayak, P. C., Sudheer, K. P., Ragan, D. M., Ramasastri, K. S. (2004),** ***A Neuro Fuzzy Computing Technique for Modeling Hydrological Time Series***. *Journal of Hydrology*, 29, 52–66;

[28] **Öztemel, E. (2003),** ***Yapay Sinir Ağları***. İstanbul: Papatya Yayıncılık, [in Turkish];

[29] **Park, J., Sandberg, I. W. (1991),** ***Universal Approximations Using Radial-Basis-Function Network***. *Neural Computation*, 3(2), 246-257;

[30] **Park, J., Sandberg, I. W. (1993),** ***Approximation and Radial Basis Function Networks. Neural Computation***, 5, 305-316;

[31] **Rivas, V. M., Merelo, J. J., Castillo, P. A., Arenas, M. G., Castellano, J. G. (2004),** ***Evolving RBF Neural Networks for Time-series Forecasting with EvRBF;*** *Information Science*, 165, 207-220;

[32] **Sarimveis, H., Doganis, P., Alexandridis, A. (2006),** ***A Classification Technique based on Radial Basis Function Neural Networks***. *Advances in Engineering Software*, 37, 218–221;

[33] **Sfetsos, A. (2000) ,** ***A Comparison of Various Forecasting Techniques Applied to Mean Hourly Wind Speedy Time Series***; *Renewable Energy*, 21, 23-35;

[34] **Sheta, A. F., Jong, K. D. (2001),** ***Time-series Forecasting Using GA-tuned Radial Basis Function***; *Information Seciences*, 133, 221-228;

[35] **Takagi, T., Sugeno, M. (1985),** ***Fuzzy Identification of Systems and its Applications to Modeling and Control***. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 116–132;

[36] **Utans, J., Moody, J., Rehfuss, S., Siegelmann, H. (1995),** ***Input Variable Selection for Neural Networks: Application Topredicting the U.S. Business Cycle***. *Proc. of IEEE/IAFE Computational Intelligence for Financial Engineering*, 118-122;

[37] **Xu, K., Xie, M., Tang, L. C., Ho, S. L. (2003),** ***Application of Neural Networks in Forecasting Engine Systems Reliability***. *Applied Soft Computing*, 2, 255-268;

[38] **Yilmaz I, Kaynar O.,(2010), Multiple Regression,** ***ANN (RBF, MLP) and ANFIS Models for Prediction of Swell Potential of Clayey Soils;*** *Expert Systems with Applications xxx (2010) xxx–xxx*, Article in Press;

[39] **Yilmaz I, Eric, N.,Y., Kaynar O. (2010),** ***Different Types of Learning Algorithms of Artificial Neural Network (ANN) Models for Prediction of Gross Calorific Value (GCV) of Coals***; *Scientific Research and Essays* , Vol. 5(16), pp. 2242-2249;

[40] **Ying, L-C., Pan, M-C. (2008),** ***Using Adaptive Network Based Fuzzy Inference System to Forecast Regional Electricity Loads***. *Energy Conversion and Management*, 49, 205–211;

[41] **Yurdusev, M. A., Firat, M. (2009),** ***Neuro-Fuzzy Inference System and Artificial Neural Networks for Municipal Water Consumption Prediction***; *Journal of Hydroinformatics*, 365, 225-234;

[42] **Yu, L., Lai, K. K., Wang, S. (2008),** ***Multistage RBF Neural Network Ensemble Learning for Exchange Rates Forecasting***. *Neurocomputing*, 71, 3295-3302;

[43] **Zhang, G., Patuwo, B. E., Hu, M. Y. (1998),** ***Forecasting with Artificial Neural Networks: the State of the Art***. *International Journal of Forecasting*, 14, 35-62;

[44] **Zhang, G., Hu, M., Patuwo, B. E., Indro, D. C. (1999),** ***Artificial Neural Networks in Bankruptcy Prediction: General Framework and Cross-validation Analysis***. *European Journal of Operational Research*, 116, 16-32;

[45] **Zhang, R., Huang, G., Sundararajan, N., Saratchandran, P. (2007),** ***Improved GAP-RBF Network for Classification Problems***. *Neurocomputing,* 70, 3011–3018.