A Comparative Study of Hybrid, Neural Networks and Nonparametric Regression Models in Time Series Prediction

DURSUN AYDIN
Department of Statistics
Muğla University
48000 Kötekli / Muğla
TURKEY
duaydin@mu.edu.tr

MAMMADAGHA MAMMADOV
Department of Statistics
Anadolu University
26470 Tepebaşı / Eskişehir
TURKEY
mmammadov@anadolu.edu.tr

Abstract: - This paper presents a comparative study of the hybrid models, neural networks and nonparametric regression models in time series forecasting. The components of these hybrid models are consisting of the nonparametric regression and artificial neural networks models. Smoothing spline, regression spline and additive regression models are considered as the nonparametric regression components. Furthermore, various multilayer perceptron algorithms and radial basis function network model are regarded as the artificial neural networks components. The performances of these models are compared by forecasting the series of number of produced Cars and Domestic product per capita (GDP) data occurred in Turkey. This comparisons show that hybrid models proposed in this paper have denoted much more excellent performance than the hybrid models in literature.

Key-Words: - Time series, Neural networks, Multilayer perceptrons, Radial basis function, Nonparametric regression, Additive regression model, Hybrid models.

1 Introduction

In order to forecast time series apart from Autoregressive integrated moving average (ARIMA) and artificial neural networks (ANN), a hybrid approach that uses ARIMA and ANN models together is recommended by Zhang [1] and Tseng et al.,[3]. Experimental results with real data sets in paper Zhang [1] indicate that a hybrid methodology that combines both ARIMA and ANN models can be an effective way to improve forecasting accuracy achieved by either of the models used separately. In addition, Aslanargun et al., [2] demonstrated that hybrid models combines models with two nonlinear components have had the best performance for time series forecasting. Zhang [1] explains the reasons of using hybrid models in detail.

In recently, nonparametric regression methods have become a very useful tool for non-linear data such as time series (E. Ferreira et al., [4]). However, these approaches perform poorly when seasonality is present. To overcome this problem, two alternatives methods have been proposed in literature. In both approaches the trend is specified as nonparametric, but the seasonal component specification is different. First, we discussed a semi-parametric model where the parametric part is a dummy-variable specification for the seasonality. Secondly, we considered the seasonal component to be a smooth function of time and, the model falls within the class of additive models. The nonparametric regression models are discussed in detail by Wahba [5]; Hardle [6]; Green and Silverman [7]; Hastie and Tibshirani [8]; Hardle et al. [9].

Although there are numerous studies combining different ANN and conventional statistical techniques to forecast time series, so far a study that combines ANN and nonparametric regression models has not been made yet. In this paper, we generalized the hybrid models studied by Zhang [1] for nonparametric regression models. We proposed a hybrid models combining neural networks and nonparametric regression models called as semi-parametric and additive regression. It is seen that hybrid models obtained by combining neural networks and nonparametric regression models indicated the better performance than individual models for time series forecasting.

2 Methodology

This paper will use two real data sets, the number of produced cars and GDP in Turkey, in order to estimate and evaluate of models. Firstly, ANN approach is considered and then, the nonparametric regression models and hybrid models, one of their components is nonparametric regression and the

other is artificial neural networks model, are discussed in the next sections.

2.1 The ARIMA Model

ARIMA models are usually used to predict a univariate time series. In these models, any observed value of the series in any time period is defined as the linear component of the several past observations and random errors. The general form of ARIMA model is given by Box and Jenkins [10]

ARIMA (p, d, q)(P, D, Q)$_s$,

where p is the number of parameters of the autoregressive (AR) model, d is degree of difference, q is the number of parameters in the moving average (MA) model, P is the number of parameters in AR seasonal model, D is the seasonal degree of difference, Q is number of parameters in MA seasonal model, and s is the period of seasonality (s = 4 for quarterly data and s = 12 for monthly data).

2.2 ANN Approach

2.2.1 Standard backpropagation training algorithm.

Multilayer perceptrons (MLPs) are used in a variety of problems, especially in forecasting. Backpropagation (BP) is the widespread approximation approach for training of the multi-layer feedforward neural networks based on Widrow-Hoff training rule (Bishop [11]; Haykin [12]). The main idea here is to adjust the weights and the biases that minimize the sum of square error by propagation the error back at each step. To minimize the sum of square error, different BP algorithms are constructed by applying different numeric optimization algorithms among gradient and Newton methods class.

The sum of squares for *qth* training sample in supervised training situations with n inputs, m outputs, and p hidden units, in a two-layers (with single hidden layer) neural network is calculated as follows:

$$E^q(w)=\sum_{k=1}^{m}\left[y_k-t_k\right]^2=\sum_{k=1}^{m}\left[f_o(\sum_{j=1}^{p}w_{jk}^o f_h(\sum_{i=1}^{n}w_{ij}^h x_i+b_j^h)+b_k^o)-t_k\right]^2 \quad (1)$$

where t_k is the kth target y_k is the kth output value; w_{ij}^h is the weight that connects the ith input and jth hidden units, w_{jk}^o is the weight that connects the jth hidden and kth output units, b_j^h is the bias for jth hidden unit, b_k^0 is the bias for kth output unit, $f_h(\cdot)$ is activation function applied to the hidden units, and $f_o(\cdot)$ is the activation function that applied to the output units. Here, w is the vector of all weight and bias components. For simplicity, the q indices are not shown.

A nonlinear $f_h(\cdot)$ function is taken. $f_o(\cdot)$ can also be taken as a linear function. For all N training sample, the mean square error (MSE) is defined as:

$$E(w)=\frac{1}{n}\sum_{q=1}^{N}E^q(w)\,. \quad (2)$$

Gradient descent numeric optimization method is used to decrease error in standard BP training algorithm ([11]; [12]). The iteration of this method is as follows:

$$w_{k+1}=w_k-\alpha\cdot g_k$$

where w_k is the current vector of weights and biases, g_k is the current gradient of error function (2) at the point w_k, and α is the learning (training) rate. The learning rate is crucial for BP since it determines the magnitude of weight changes.

A standard BP training algorithm that includes *momentum* (Rumelhart et al., [15]) is given by the improved gradient descent with momentum formula:

$$\Delta w_{k+1}=-\alpha\cdot g_k+\mu\Delta w_k\,,$$

where $\Delta w_k=w_k-w_{k-1}$ is the weight change in the previous iteration and μ is the momentum coefficient. The weight change in BP training algorithm with momentum is made by the combination of current gradient vector and the previous gradient vector. This change is better for the behavior of the algorithm since it provides the chance of escaping surface local minimums.

2.2.2 Conjugate Gradient (CG) algorithms.

In CG algorithms, search is made in *conjugate directions* [14] A set of nonzero n-dimensional vectors $\{p_0,p_1,...,p_{n-1}\}$ is said to be *conjugate* with respect to the symmetric positive definite $n\times n$ matrix A if

$$p_i^TAp_j=0 \text{ for all } i\neq j\,.$$

First, we should be taken into account as the minimum problem of square function:

$$F(w)=\frac{1}{2}w^THw-b^Tw\,, \quad (3)$$

where $w\in R^n$ and H is the $n\times n$ symmetric positive definite matrix. For a starting point $w_0\in R^n$, the method defined by the equations given subsequently is called the *conjugate direction method* [14]:

$$w_{k+1}=w_k+\alpha_k p_k \quad (4)$$

$$\alpha_k=-\frac{p_k^Tg_k}{p_k^THp_k}\,. \quad (5)$$

where, equation (5) is defined by the minimum problem of one variable function of $\varphi(\alpha)=F(w_k+\alpha p_k)$.

For any starting point $w_0\in R^n$, the sequence $\{w_k\}$ generated by the conjugate direction algorithm (4) and

(5) converges to the minimum point w^* of the problem (3) in at most n steps [14]. The conjugate gradient method is a conjugate direction method; start out by searching in the gradient direction on the first iteration

$$p_0 = -g_0, \quad (6)$$

The conjugate current p_k vector is calculated by using the previous p_{k-1} vector and current gradient

$$p_k = -g_k + \beta_k p_{k-1} \quad (7)$$

The coefficient β_k in (7) is selected by the condition of p_{k-1} and p_k vectors, being conjugate with respect to the symmetric positive definite $n \times n$ matrix H ($p_{k-1} H p_k = 0$):

$$\beta_k = \frac{g_k^T H p_{k-1}}{p_{k-1}^T H p_{k-1}}. \quad (8)$$

There are various CG algorithms depending on different calculation formulas of β_k coefficients (without computing Hessian matrix H): *Fletcher–Reeves*, *Polak and Ribere* conjugate gradient algorithms ([11];[12];[14]).

Now, error function (2) is taken into account. As activation functions are nonlinear, sum of mean squared error function $E(w)$ is the nonlinear function of w. Using first two terms of the Taylor-series expansion of $E(w)$ around w_k, function $E(w)$ may be approximated to a squared function

$$E(w_{k+1}) - E(w_k) = \nabla E(w_{k+1}) \approx g_k^T \Delta w_k + \frac{1}{2} \Delta w_k^T H_k \Delta w_k \quad (9)$$

where $H_k = \frac{\partial^2 E(w_k)}{\partial w^2}$ is the Hessian matrix in current point w_k. Hence, in each current step, taking function (9) instead of function (3), a suitable minimization problem is taken into account and a local CG is realized by using Eq. (4)-(8). In this case the Hessian matrix H_k is assumed positive definite.

Another algorithm of CG algorithms is *Scaled Conjugate Gradients (SCG) algorithm* [14]. The basic idea of SCG is to combine the model *trust region approach* with the CG approach. The problem can be overcomes by modifying the Hessian matrix to ensure that it is positive definite [14].

2.2.3 Quasi-Newton (QN) Algorithms

The basic step of Newton's method is

$$w_{k+1} = w_k - H_k^{-1} g_k, \quad (10)$$

where H_k is the Hessian matrix in current point w_k. Newton's method often converges faster than conjugate gradient methods. Unfortunately, it is complex and expensive to compute the Hessian matrix. QN methods are based on Newton's method, which does not require calculation of second derivatives however. They update (the update is computed as a function of the gradient) an approximate Hessian matrix at each iteration of the algorithm [14].

The Broyden, Fletcher, Goldfarb, and Shanno (BFGS) algorithm is a successful QN algorithm. In BFGS algorithm, instead of the inverse of Hessian matrix in (10), its G approach is constructed by the following recursive way:

$$G_{k+1} = G_k + \frac{pp^T}{p^T v} - \frac{G_k v v^T G_k}{v^T G_k v} + (v^T G_k v) u u^T,$$

where

$$p = w_{k+1} - w_k, \; v = g_{k+1} - g_k, \; u = \frac{p}{p^T v} - \frac{G_k v}{v^T G_k v}.$$

Since G matrix is positive definite, -Gg will be the decrease direction of the algorithm. Weight update is made as follows:

$$w_{k+1} = w_k + \alpha_k G_k g_k,$$

where α_k is found by line minimization.

The one-step secant (OSS) algorithm is an attempt to bridge the gap between the CG algorithms and QN algorithms. This algorithm accepts that the previous Hessian is the identity matrix, and hence, inverse matrixes are not calculated to determine the new search direction. This algorithm requires very light storing and requires less storing at each step than the CG algorithms.

The *Levenberg-Marquart (LM) algorithm* is one of the required QN algorithms [11]. Hessian matrixes are not calculated, and being second order. The Hessian matrices can be approximated as $H = J^T J$, where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases. The gradient can be computed as $g = J^T \varepsilon$, where ε is a vector of network error. The LM algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$w_{k+1} = w_k + [J^T J + \mu I]^{-1} J^T \varepsilon.$$

This algorithm runs fast for moderate-dimensioned feedforward neural networks for regression problems.

2.2.4 Radial Basis Function Networks (RBF)

RBF is also used besides MLP networks in regression and classification problems ([11];[12]). In RBF, one hidden layer with required number of units is enough in order to model a function. The activations of hidden (radial) units are defined depending on the distance of the input vector and the center vector. Typically, the radial layer has exponential activation functions and the output layer a linear activation function. Appropriate y output vector for the x input vector is calculated as follows for n input, m output, and p radial units for RBF:

$$y_k(x)=\sum_{j=0}^{p} w_{kj}\phi_j(x),\ k=1,2,...,m\ , \qquad (11)$$

where w_{kj}, $j=1,2,...,p$ are the appropriate weights for kth output unit, $\phi_j(x)$, $j=1,2,...,p$, is the basis function of jth radial unit, w_{k0}, $k=1,2,...,m$ are the appropriate deviations for kth output unit, ϕ_0 is an extra basis function with activation value fixed at $\phi_0=1$. Usually, more attention paid for the following Gaussian basis function:

$$\phi_j(x)=\exp(-\|x-\mu_j\|^2/2\sigma_j)\ ,\quad j=1,2,...,p \qquad (12)$$

where $\mu_j=(\mu_{j1},...,\mu_{jn})$ vector is center for $\phi_j(x)$, and σ_j is deviation (or width) parameters of that function. The basis function of the unit is defined using those two parameters. Equation (11) can be written in matrix notation as

$$y(x)=W\cdot\phi\ , \qquad (13)$$

where $W=(w_{kj})$ and $\phi=(\phi_j)$. As can be seen from (11), the linear activation function is used in RBF for output layer.

Education is made in three stages in RBF. In the first stage, by unsupervised education, radial basis function centers (in other words μ_j) are optimized using all $\{x^{(i)}\}$, $i=1,2,...,N$, education data. Centers can be assigned by a number of algorithms: Sub-sampling, K-means, Kohenen training, or learned vector quantization. In the second stage σ_j, $j=1,2,...,p$, parameters can be assigned by algorithms explicit, isotropic and K-nearest neighbor. In the third stage of education, the basis functions that are obtained for adjusting the appropriate weights for output units are taken as fixed and deviation parameters are added to linear sum. Optimum weights are obtained by minimization of the sum of square errors,

$$E(w)=\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{m}[y_k(x^{(i)})-t_k^{(i)}]\ . \qquad (14)$$

In equation (14), $t_k^{(i)}$ is the target value for output unit k when the network is presented with input vector $x^{(i)}$, $i=1,2,...,N$. Since the equality in (13) is the quadratic function of the weights, optimum weights can be found as the solution of the linear equations system. The output layer is usually optimized using the pseudo-inverse technique.

MLP, with a defined architecture, is given by the appropriate weights and the biases of the units, but in RBF, it is given by the center and the deviation of the radial units and by the weights and biases of the output units. As the point is given by n coordinates in n dimensional space, the number of the coordinates are equal to the linear input units n. Hence, in Statistica Neural Network software, the coordinates of the center radial unit are taken as weights and the deviation of the radial unit is taken as bias. As a result, radial weights denotes the center point, radial bias denotes the deviation. Having only one hidden layer and making faster education than MLP, can be taken as advantages of RBF. As the linear modeling methods are more useful in output layers of RBF, the difficulties that occur about the local minimums in MLP are removed.

However, RBF has some disadvantages in comparison with MLP. In order to correctly model a typical function in RBF, many more hidden (radial) units may be required than appropriate MLP model. That may cause the model slow down, and more memory may be required. RBF is very sensitive to any increment of the network dimension and some difficulties may occur as a result of an increment in the number of input units.

RBF is unsuccessful in extrapolation in its nature. MLP networks are more successful in extrapolation problems than RBF because when the input data are far from the radial centers, the output signal is 0, and this may not show the required result.

2.3 The Nonparametric Regression Approach

The following general model form has been considered

$$y(t_i)=s(t_i)+z(t_i)+e(t_i),\ i=1,...,n \qquad (15)$$

where t_i's are knot points spaced in time interval [a, b], $s(t_i)$ is denote the seasonal component, $z(t_i)$ is represent the trend, and $e(t_i)$ is indicate the terms of error with zero mean and common variance σ_e^2. The model (15) can be also written as,

$$y_i=s_i+z_i+e_i,\ i=1,2,...,n. \qquad (16)$$

It is assumed that the following model structure for the trend:

$$z_i=f(t_i)+\varepsilon_i,\ i=1,2,...,n \qquad (17)$$

where f is a smooth function in [a, b], and ε_i's are assumed to be with zero mean and common variance σ_ε^2, and different from e_i's. The basic aim is to estimate the functions f and s. The function f is estimated as a smooth function, but the estimation of the function s is different due to seasonality (E. Ferreira et al, [4]). Therefore, it is considered two alternative models for the estimation of s. Firstly, we considered a semi-parametric model where parametric component is dummy variable for the seasonality. Secondly, we discussed the seasonal component to be a smooth function of time, and use a nonparametric method.

2.3.1 Semi-parametric regression model

It is assumed that the seasonality is build as follows:

$$s_i = s(t_i) = \sum_{k=1}^{r-1} \beta_k D_{ki}^* + v_i,\ i = 1,...,n \qquad (18)$$

where r is the number of annual observations (r=12) and v_i's are assumed to be with zero mean and common variance σ_v^2, and different from the errors in (16) and (17). D_{ki}^*'s are dummy variable that denotes the seasonal effects, and β_k's are parametric coefficients. Dummy variables are denoted by $D_{ki}^* = D_{ki} - D_{ri}$ (where $D_{ki} = 1$ if i. observation correspond to the kth month of year, and $D_{ki} = 0$ otherwise) for cancels the seasonal effects when a year is completed (E. Ferreira et al, [4]). By substitution equations (18) and (17) in (16), the semi-parametric regression model is obtained as

$$y_i = \sum_{k=1}^{r-1} \beta_k D_{ki} + f(t_i) + e_i,\ i = 1,...,n \qquad (19)$$

where r is the number of annual observations (r=12 for monthly data), D_{ki}'s are dummy variable that denotes the seasonal effects, and β_k's are parametric coefficients. Dummy variables are denoted by $D_{ki} = D_{ki}^* - D_{ri}^*$, where $D_{ki}^* = 1$ if i. observation correspond to the kth month of year, and $D_{ki}^* = 0$ otherwise, for cancels the seasonal effects when a year is completed [4]. Eq. (19) in vector-matrix form can be expressed as

$$\mathbf{y} = D\boldsymbol{\beta} + \mathbf{f} + \mathbf{e} \qquad (20)$$

where $\boldsymbol{\beta} = (\beta_1,...,\beta_{r-1})^T$, $\mathbf{y} = (y_1,...,y_n)^T$, $\mathbf{f} = (f(t_1),...,f(t_n))^T$ $\mathbf{e} = (e_1, e_2, ..., e_n)^T$ and D is the $n \times (r-1)$ matrix, so that $D^T = \{D_{ki}\}_{k=1,...,r-1}^{i=1,...,n}$.

Therefore,

$$D^T = \begin{bmatrix} 1 & 0 & . & . & . & 0 & -1 & 1 & 0 & . & . & . \\ 0 & 1 & . & . & . & 0 & -1 & 0 & 0 & . & . & . \\ & & . & & & & & & & & & \\ & & & . & & & & & & & & \\ & & & & . & & & & & & & \\ 0 & 0 & . & . & . & 1 & -1 & 0 & 0 & . & . & . \end{bmatrix}$$

Model (19) is a semi-parametric model due to consist of parametric linear component and nonparametric component. The main purpose is to estimate the parameter vector $\boldsymbol{\beta}$ and function f at sample points $t_1,...,t_n$. For this aim, two estimation methods, called as *smoothing spline* and *regression spline*, have been considered by Wahba [5]; Green and Silverman [7]; Hastie and Tibshirani [8]; Hardle et al., [9]; Eubank [17].

Estimation with smoothing spline method (SSM): Estimation of the parameters of interest in equation (20) can be performed using smoothing spline. Mentioned here the vector parameter $\boldsymbol{\beta}$ and the values of function f at sample points $t_1,...,t_n$ are estimated by minimizing the penalized residual sum of squares

$$PSS(\boldsymbol{\beta}, \mathbf{f}) = \sum_{i=1}^{n} \{y_i - d_i^T \boldsymbol{\beta} - f(t_i)\}^2 + \lambda \int_0^1 (f''(u))^2\, du \qquad (21)$$

where $f \in C^2[0,1]$ and d_i is the ith row of the matrix D. When the $\boldsymbol{\beta} = 0$, resulting estimator has the form $\hat{\mathbf{f}} = \left(\hat{f}(t_1),...,\hat{f}(t_n)\right) = S_\lambda \mathbf{y}$, where S_λ a known positive-definite (symmetric) smoother matrix that depends on λ and the knots $t_1,...,t_n$ (see, [5] and [17])

For a pre-specified value of λ the corresponding estimators for $\mathbf{f}$ and $\boldsymbol{\beta}$ based on Eq. (20) can be obtained as follows (E. Ferreira et al, [4]): Given a smoother matrix S_λ, depending on a smoothing parameter λ, construct $\tilde{D} = (I - S_\lambda)D$. Then, by using penalized least squares, mentioned here estimator are given by

$$\hat{\boldsymbol{\beta}} = \left(D^T \tilde{D}\right)^{-1} \tilde{D}^T \mathbf{y} \qquad (22)$$

$$\hat{\mathbf{f}} = S_\lambda \left(\mathbf{y} - D\hat{\boldsymbol{\beta}}\right) \qquad (23)$$

Evaluate some criterion function (such as cross validation, generalized cross validation) and iterate changing λ until it is minimized.

Estimation with regression spline method (RSM): Smoothing spline become less practical when sample size n is large, because they use n knots. A more general approach to spline fitting is regression spline. Smoothing spline require that many parameters be estimated, typically at least at many parameter as observations. A regression spline is a piecewise polynomial function whose highest order nonzero derivative takes jumps at fixed "knots". Usually regression splines are smoothed by deleting nonessential knots. When the knots have been selected, regression spline can be fit by ordinary least squares. For further discussion on selection of knots, see study of Ruppert and Carrol [18]).

$f(t_i)$ in (19) is approximated by

$$f(t_i) = f(t_i, \gamma) = \gamma_0 + \gamma_1 t_i + ... + \gamma_p t_i^p + \sum_{k=1}^{K} b_k (t_i - \kappa_k)_+^p,\ i = 1,...,n \qquad (24)$$

where $p \geq 1$ is an integer (order of the regression spline and usually chosen a priori), $b_1,...,b_K$ are independently and identically distributed (i.i.d) with $N(0, \sigma_b^2)$, $(t)_+ = t$ if $t > 0$ and 0 otherwise and $\kappa_1 < .... < \kappa_k$ are fixed knots ($\min(t_i) < \kappa_1,...,< \kappa_K < \max(t_i)$).

In matrix notation model (19) can be written as

$$\mathbf{y} = D\boldsymbol{\beta} + Z\mathbf{b} + \boldsymbol{\eta} \quad (25)$$

where

$$D^T = \begin{bmatrix} 1 & 0 & \dots & 0 & -1 & 1 & 0 & \dots & 1 & t_1 & \dots & t_n \\ 0 & 1 & \dots & 0 & -1 & 0 & 1 & \dots & 1 & t_1^{p-1} & \dots & t_n^{p-1} \\ & & \cdot & & & & & & \cdot & \cdot & & \cdot \\ & & & \cdot & & & & & \cdot & \cdot & & \cdot \\ & & & & \cdot & & & & \cdot & \cdot & & \cdot \\ 0 & 0 & \dots & 1 & -1 & 0 & 0 & \dots & 1 & t_1^{p} & \dots & t_n^{p} \end{bmatrix}$$

and

$$Z = \begin{bmatrix} (t_1 - \kappa_1)_+^p & \dots & (t_1 - \kappa_K)_+^p \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ (t_n - \kappa_1)_+^p & \dots & (t_n - \kappa_K)_+^p \end{bmatrix}$$

$\mathbf{b} = (b_1, \dots, b_K)^T$ is vector of coefficients and $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)^T$ is a vector of the random error. Predicted value of $\hat{y}$ in (17) is given by

$$\hat{\mathbf{y}} = \hat{\gamma}_0 + \hat{\gamma}_1 t_i + \dots + \hat{\gamma}_p t_i^p + \hat{\beta}_1 D_1 + \dots + \hat{\beta}_{r-1} D_K + (Z_1, \dots, Z_K)\left(\hat{b}_1, \dots, \hat{b}_K\right)^T . \quad (26)$$

Regression spline estimators $\left(\hat{\boldsymbol{\beta}} = \left(\hat{\gamma}_0, \hat{\gamma}_1, \dots, \hat{\gamma}_p, \hat{\beta}_1, \dots, \hat{\beta}_{r-1}\right)^T, \hat{\mathbf{f}} = (\hat{b}_1, \dots, \hat{b}_k)\right)$ of $(\boldsymbol{\beta}, \mathbf{f})$ are defined as the minimizer of

$$PSS(\boldsymbol{\beta}, \mathbf{f}) = \sum_{i=1}^{n} \left\{ y_i - d_i^T \boldsymbol{\beta} - f(t_i) \right\}^2 + \lambda \sum_{k=1}^{K} b_k^2 \quad (27)$$

where $\lambda > 0$ is a smoothing parameter such as in (21). As $\lambda \to \infty$, the regression spline converges to a pth degree polynomial fit. As $\lambda \to 0$, the regression spline converges to the ordinary least squares (OLS) fitted spline. For a pre-specified value of λ the corresponding estimators for $\boldsymbol{\beta}$ and $\mathbf{f}$ based on Eq. (25) can be obtained as follows (Ruppert et al., [20]):

$$\hat{\boldsymbol{\beta}} = \left(D^T \hat{\Sigma}^{-1} D \right)^{-1} D \hat{\Sigma}^{-1} \mathbf{y} , \quad (28)$$

where, $\hat{\Sigma} = ZZ^T \hat{\sigma}_b^2 + diag(\hat{\sigma}_{\eta i}^2), \quad i = 1, 2, \dots, n$,

$$\hat{\mathbf{f}} = (\hat{b}_1, \dots, \hat{b}_k)^T = \hat{\sigma}_b^2 Z^T \hat{\Sigma}^{-1} \left(\mathbf{y} - D\hat{\boldsymbol{\beta}} \right) \quad (29)$$

The smoothing parameter (penalty parameter λ) and the number of knots K must be selected in implementing the regression spline. However, λ plays a more important role (see, Ruppert [18] for more details on discussion of the knot selection). The solution can be obtained by S-Plus software.

2.3.2 Additive regression model (ARM)

In the previous section, it was used the semi-parametric model for estimation of the parameters in (19). However, there are situations in which a dummy variable specification does not capture all fluctuations because of the seasonal effects. For this reason, in this section, it is considered a more general case for seasonal component as follows:

$$s_i = g(t_i) + \nu_i , \; i = 1, \dots, n \quad (30)$$

where g is an (0,1) and $g \in C^2[a,b]$, ν_i's are denote the terms of random error with zero mean and common variance σ_ν^2. By substitution of the equations (17) and (30) in (16), it is obtained as

$$y_i = g(t_i) + f(t_i) + u_i , \; i = 1, \dots, n , \quad (31)$$

where u_i's are the terms of random error with zero mean and constant variance $\sigma_u^2 = \sigma_e^2 + \sigma_\varepsilon^2 + \sigma_v^2$.

Model presented in (31) has a fully nonparametric model because of the parametric component is missing. These models are called as additive nonparametric regression models. In order to estimate model in (31), it can be generalized the criterion (21) and (31) in an obvious way. Estimator of the model (31) is based on minimum of the penalized residual sum of squares [8]

$$PSS(\mathbf{f}, \mathbf{g}) = \sum_{i=1}^{n} \{ y_i - f(t_i) - g(t_i) \}^2 + \lambda_1 \int_0^1 \left(f''(u) \right)^2 du + \lambda_2 \int_0^1 \left(g''(u) \right)^2 du \quad (32)$$

where the first term denotes the residual sum of the squares (RSS) and this term penalizes the lack of fit. The second term multiplicand by λ_1 is denote the roughness penalty for the f, and the third term multiplicand by λ_2 is denote the roughness penalty for g. Firstly, Eq. (32) can be written as

$$PSS(\mathbf{f}, \mathbf{g}) = (\mathbf{y} - \mathbf{f} - \mathbf{g})^T (\mathbf{y} - \mathbf{f} - \mathbf{g}) + \lambda_1 \mathbf{f}^T K_f \mathbf{f} + \lambda_2 \mathbf{g}^T K_g \mathbf{g} \quad (33)$$

Here K_f is a penalty matrix for $\mathbf{f}$ and K_g is a penalty matrix for $\mathbf{g}$. Then, by differentiating to $\mathbf{f}$ and $\mathbf{g}$:

$$PSS(\mathbf{f}, \mathbf{g}) / \mathbf{f} = -2(\mathbf{y} - \mathbf{f} - \mathbf{g}) + 2\lambda_1 K_f \mathbf{f} \quad (34)$$

$$PSS(\mathbf{f}, \mathbf{g}) / \mathbf{g} = -2(\mathbf{y} - \mathbf{f} - \mathbf{g}) + 2\lambda_2 K_g \mathbf{g} \quad (35)$$

Afterwards, by making (34) and (35) setting to zero, the estimators of $\mathbf{f}$ and $\mathbf{g}$ are defined as:

$$\hat{\mathbf{f}} = (I + \lambda_1 K_f)^{-1} (\mathbf{y} - \mathbf{g}) = S_{\lambda_1} (\mathbf{y} - \mathbf{g}) \quad (36)$$

$$\hat{\mathbf{g}} = \left(I + \lambda_2 K_g \right)^{-1} \left(\mathbf{y} - \mathbf{f} \right) = S_{\lambda_2} \left(\mathbf{y} - \mathbf{f} \right) \quad (37)$$

2.4 The hybrid methodology

Suppose observed are measurements y_t, $t = 1, 2, \dots, N$. The forecasts of the combined model in hybrid methodology are defined as follows:

$$\hat{y}_t = \hat{y}_t^1 + \hat{y}_t^2$$

where superscripts denote the row number of the hybrid model, $\hat{y}_t^1$ and $\hat{y}_t^2$ are estimations of the appropriate first and second combined models at the time point t. Firstly, to obtain the forecast values $\hat{y}_t^1$ of the first model at time point t, the model with 1-indiced is applied to the observation data y_t, $t = 1,2,...,N$. If the first model contains m_1 input units, the forecast values of the first model are calculated as follows:

$$\hat{y}_t^1 = f_1(y_{t-1}, y_{t-2}, ..., y_{t-m_1})$$

where f_1 is the function obtained from the first model. After this stage, the input data are calculated as $e_t^1 = y_t - \hat{y}_t^1$ for the second model. In this case, the number of e_t^1 will be $N - m_1$ for this model. If the second model contains m_2 input units, the number of $\hat{y}_t^2$ forecast (improved residuals) values will be $\mathrm{N} - \mathrm{m}_1 - \mathrm{m}_2$. In this case, the forecast values that are appropriate for the second model are calculated as follows:

$$\hat{y}_t^2 = f_2(e_{t-1}, e_{t-2}, ..., e_{t-m_2}),$$

where f_2 is the function obtained from the second model.

In case of the nonparametric of the first model, $m_1 = 0$ and the number of e_t^1 units will be N. If the second model is nonparametric, the number of $\hat{y}_t^2$ forecast values will be $N - m_1$.

3 Experimental Evaluations

In this section, two different real data sets occurred in Turkey is discussed as experimental. Appropriate ANN, nonparametric regression models and hybrid models were chosen by doing experiments to forecast, and these models are also compared with each others. In this study, STATISTICA Neural Networks, S-Plus, and R-Programs are used.

3.1 Data Sets

The first real data set is from Central Bank of the Republic of Turkey. The data can be found in www.tcmb.gov.tr, and denotes the number of produced cars in Turkey for January 1989–December 2006 period [21]. The data set is divided into two parts to use in training and forecasting. In the first part, 216 monthly data are taken into account for the January 1989–December 2006 period. These data are used in training to construct the models. In the second part, by using the models obtained in the first part, the performances of these models are calculated for the 24 monthly test data in the January 2007–December 2008 period.

The second real data set is also taken from Central Bank of the Republic of Turkey. The data can be found in www.tcmb.gov.tr and indicates GDP occurred in Turkey for January 1989–December 2006 period [21]. The data set is divided into two parts for the use in training and forecasting. In the first part, 156 quarterly data are taken into account for January 1989–December 2004 period. These data are used for constructing of the models. In the second part, by using the models obtained in the first part, the performances of these models are calculated for the 24 quarterly test data in the January 2001–December 2006 period.

3.2 Choice of Appropriate ANN Models

The choice of the best ANN models depends on a comparison of statistics such as the MSE (RMSE), MAE, and MAPE. As the initial weight and bias values of the network were random, 150 replications were made for the same network structure, and the models giving the best forecasts were determined.

For the first data set, the 216 monthly data were used in training stage of the network, whereas the 156 quarterly data were used in training of the ANN models for the second data set. For all data sets, an evaluation of the model was made depending on the forecasts for the 24 observations, test data set. As the initial weight and bias values of the network were random, experiments with 150 replications were made for the same network structure, and the models giving the best forecasts were determined. Since the mentioned times series data sets include the seasonality, after trying many neural networks with different numbers of input units, as expected, the number of input units was determined as 12 for the first data set. On the other hand, the number of input units was determined as 4 for the second data sets During these experiments, various, with one or two hidden layers multilayer feed-forward (MLP) neural network algorithms and the RBF models were applied on the data set. For the first data set, as the initial 12 data were lost because of the seasonality, 204 from the 216 data were used to adjust the weights. In the training stage of the network, data were divided into two parts: 132 of the 204 data were used for training and 72 data were used for validation. This division was used to restrict memorization of the network and provided for better forecasts ([11]; [12]): For the second data set, as the initial 4 data were lost because of the seasonality, 152 from the 156 data were used to adjust the weights. In the training stage of the network, data were divided into two parts: 98 of the 152 data were used for training and 54 data were used for validation

For the first data, the MLP(12:7:1) model showed the best performance among the MLP networks. The CG algorithm is used to train the network, and found the best performance on the 24th epoch. A hyperbolic tangent function is applied in the hidden unit and the linear activation function is applied in the output unit. The weights and biases of the MLP(12:7:1) model are given in Table 1:

Table 1. The weights and biases of the MLP(12:7:1)

	2.1	2.2	2.3	2.4	2.5	2.6	2.7	3.1
Thresh	-0,795	0,780	0,184	-0,325	-0,032	0,385	0,867	0,325
1.1	-0,392	0,035	0,188	0,539	-0,024	-0,735	0,398	
1.2	0,882	0,124	0,480	-0,267	0,479	0,558	0,073	
1.3	-0,502	-0,587	-0,341	0,338	-0,348	0,015	0,250	
1.4	-0,396	-0,426	-0,562	0,773	0,480	-0,469	0,099	
1.5	0,636	0,456	-0,767	-0,599	-0,549	0,557	0,219	
1.6	0,003	0,007	-0,341	-0,374	-0,747	-0,364	-0,170	
1.7	0,077	0,617	-0,115	-0,018	-0,483	-1,028	-0,455	
1.8	0,111	0,801	0,792	0,783	-0,527	-0,571	-0,283	
1.9	0,507	0,290	0,464	0,242	-0,835	0,049	-0,350	
1.10	0,123	-0,819	0,647	0,609	-0,758	0,790	0,358	
1.11	0,363	-0,906	0,752	0,957	-0,237	0,167	-0,280	
1.12	-0,427	-0,239	0,157	-0,854	0,691	0,318	0,896	
2.1								0,463
2.2								-0,406
2.3								0,449
2.4								-0,513
2.5								-0,272
2.6								-0,723
2.7								0,597

Note: The row and column header numeric terminology first lists the layer, then the unit number within the layer. For example, 2.1 stands for unit 1in layer 2

Among the MLP networks, the MLP(4:3:1) model showed the best performance with respect to the second data set. The CG algorithm was used to train network, the best network discovered during that run was selected, and this network was found on the 46th epoch. A hyperbolic tangent function is applied in the hidden unit and the linear activation function is applied in the output unit. The weights and biases of the MLP(4:3:1) model are given in Table 2.

Table 2: The weights and biases of the MLP(4:3:1)

	2.1	2.2	2.3	3.1
Thresh	-0.96938	-0.645298	0.742079	-0.38392
1.1	0.15117	0.791275	-0.724751	
1.2	-0.19540	0.952034	0.523114	
1.3	0.17326	0.555809	-0.544963	
1.4	-1.08251	0.943615	-0.081043	
2.1				-1.05406
2.2				0.22117
2.3				-0.47403

3.3 Constructing Appropriate Nonparametric Models

We used two nonparametric regression models. Firstly, we considered a semi-parametric model to estimate the parametric vector β, and nonparametric function f. The estimators of the β and f are obtained by using (5). We need to select the smoothing parameter λ presented in (5). In practice, a value of the smoothing parameter can be chosen by specifying degrees of freedom $\left(df = trace(S_\lambda)\right)$ for the nonparametric components [8]. Therefore, we used the df in order to select the smoothing parameter λ in smoothing spline. On the other hand, both the smoothing parameter λ and the number of knots K must be selected in implementing the regression spline. Secondly, we considered an additive regression model to estimate f and g in eq. (10). The estimators of f and g are obtained by using eq. (11). We need to select the soothing parameters called as λ_1 and λ_2 in (11). We select the both of the smoothing parameters by specifying the df. For both data sets, observed and their estimated results obtained by appropriate nonparametric models are given in Figure 1 and 2, respectively.

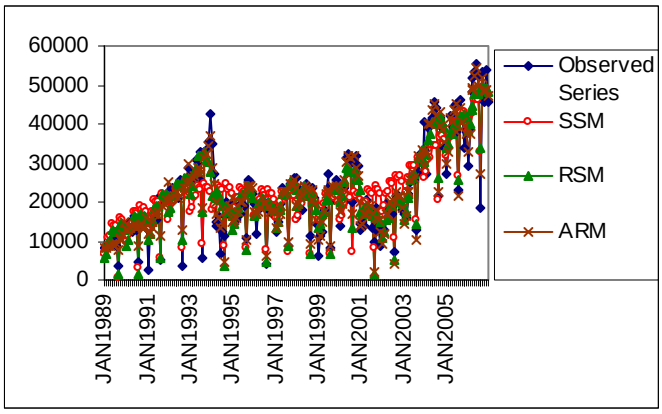

Figure 1. The number of produced cars, and their predicted values by SSM, ARM, and RSM, respectively

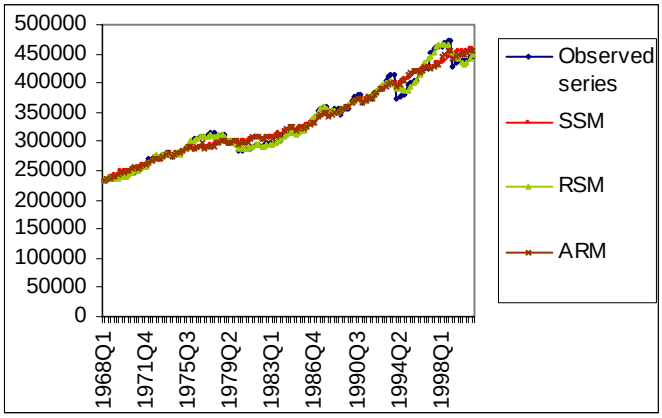

Figure 2. GDP and estimated values obtained by SSM, RSM, and ARM respectively

3.4 Constructing Appropriate Hybrid Models

In this study, we discussed hybrid models where components are nonparametric regression and ANN. In determining hybrid models whose first component is nonparametric regression, firstly, the nonparametric regression model was applied to the 216 real data, and then, the 216 residuals were obtained. At the next step, as the second component, ANN was applied to the 216 residuals data. As shown Table 3, according to the forecasts resulting from experiment, the models with the best performance among hybrid models whose first component is nonparametric regression are ARM&RBF (12:4:1) ARM&MLP(12:8:1) RSM&RBF(4:9:1) RSM&MLP(6:8:1) SSM&RBF(12:3:1) SSM&MLP (12:6:1). As for the second data sets, among hybrid models whose first component is nonparametric regression, SSM&MLP(4:8:1) SSM&RBF(4:10:1) RSM&MLP(6:8:1) RSM&RBF(4:9:1) ARM&MLP (4:8:1) ARM&RBF(4:4:1) models showed a good empirical performance (see in detail Table 4).

For the first data, the models denoting a good performance out of 150 replicated models among hybrid models where first component is ANN and second component is nonparametric regression are the MLP(12:7:1)&ARM MLP(12:7:1)&RSM MLP(12:7:1) &SSM RBF(12:18:1)&SSM RBF(12:18:1)&ARM, and RBF(12:18:1)&RSM (see Table 3). For the second data, among hybrid models, whose first component is ANN and second component is nonparametric regression, RBF(3:18:1)&SSM RBF(3:18:1)&RSM RBF(3:18:1)& ARM MLP(1:8:1)&SSM MLP(1:8:1)&RSM MLP(1:8:1) &ARM models denoted a good performance (see Table 4).

For test data composed of the 24 values, the observed and forecasted values obtained by different models are calculated, but they are only given as graphically for three models since they would occupy very much place. Observed and their forecasted values by the best SSM, ANN and hybrid models are given in the following figure 3 and 4, respectively.

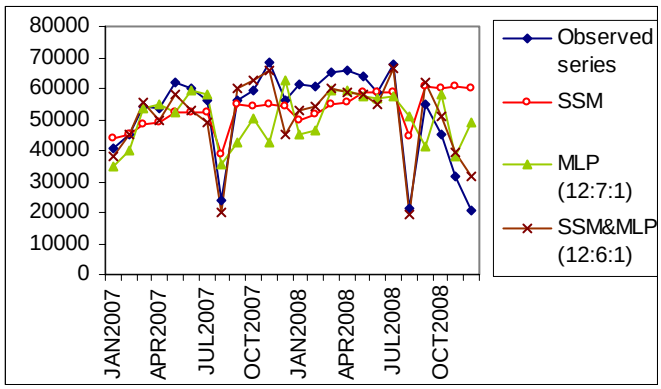

Figure 3. Observed and forecasted values for the January 2007–December 2008 period.

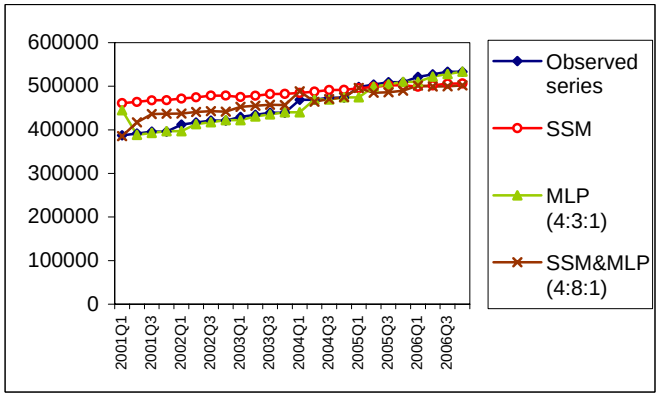

Figure 4: Observed and forecasted values obtained by SSM, MLP (4:3:1) and SSM&MLP (4:8:1) models for the 2001Q1–2008Q2 period.

3.5 Comparisons of the Models

We used the test data to compare the performances of the hybrid and the others models. The performances of models are evaluated the criterion such as the mean square error (MSE), the root mean square error (RMSE), the mean absolute error (MAE) and the mean absolute percentage error (MAPE). (see in detail Carey and Rob [22]).

By using test data, for the number of produced Cars in Turkey, the performance values of the nonparametric regression, ANN and hybrid models are presented in Table 3. For this data, the SSM among the nonparametric models, and the MLP (12:7:1) among the ANN models have also indicated the best empirical performance scores. The SSM and MLP models individually indicate good performance. According to the values of MSE, RMSE, MAE and MAPE, the SSM&MLP(12:6:1) hybrid model, whose components are SSM and MLP, have indicated the best empirical performance scores.

Table 3. Performance values for selected models

Models	MSE	RMSE	MAE	MAPE
ARIMA$(1,1,1)(1,1,0)_{12}$	2,16E+08	14707,63	12345,00	33,00
MLP (12:7:1)	1,69E+08	13010,87	10197,49	26,55
RBF(6:9:1)	2,22E+08	14907,37	13349,63	32,62
SSM	1,85E+08	13609,15	10025,80	28,60
RSM	2,86E+08	16921,10	10353,00	33,50
ARM	2,33E+08	15255,52	11385,80	31,00
SSM&MLP(12:6:1)	**1.2E+08**	**11032.22**	**7807.3**	**23.70**
SSM&RBF (12:3:1)	2.81E+09	53008.07	51335.79	99.54
RSM&MLP (6:8:1)	2.3E+08	15157.17	8826.4	27.80
RSM&RBF (4:9:1)	2.78E+08	16668.61	10207.21	32.65
ARM&MLP (12:8:1)	2.2E+08	14922.17	10968.2	30.10
ARM&RBF (12:4:1)	2.23E+08	14936.24	11131.065	30.32
RBF (12:18:1)&SSM	2.73E+09	52264.65	49797.48	94.31
RBF(12:18:1)&RSM	6.71E+09	81941.47	79833.04	162.11
RBF(12:18:1)&ARM	5.89E+09	76774.33	73385.10	140.77
MLP (12:7:1)&SSM	5.11E+08	22611.99	20233.27	39.36
MLP (12:7:1)&RSM	5.57E+08	23597.74	21125.97	40.48
MLP (12:7:1)&ARM	4.50E+08	21204.30	17744.20	41.23

(*) Indicates the model having best performance

For the GDP test data set, the performance values of the nonparametric regression, ANN and hybrid models are given in Table 4. As can be seen Table 4, the SSM among the nonparametric models, and the MLP (4:3:1) among the ANN models have denoted the best performance. According to the values of MSE, RMSE, MAE and MAPE, the SSM&MLP(4:8:1) hybrid model has denoted the best empirical performance among the all models for the GDP data set.

Table 4: Performance values of the selected models

Models	MSE	RMSE	MAE	MAPE
ARIMA(1,1,1)(1,1,0)$_{12}$	1,87E+11	432997,2	431235,0	94,20
MLP (4:3:1)	4,49E+08	21192,5	15210,66	3,20
RBF(4:6:1)	2,06E+09	45469,8	42273,5	9,06
SSM	1,93E+09	43892,7	37243,5	8,70
RSM	3,38E+09	58096,6	48340,2	10,20
ARM	2,49E+09	49916,7	44345,0	10,20
SSM&MLP (4:8:1)	**3.2E+08**	**18017.92**	**13785.8**	**3.20**
SSM&RBF (4:10:1)	6.70E+08	25890.43	20064.64	4.76
RSM&MLP (6:8:1)	2.1E+09	45460.88	37720.6	7.80
RSM&RBF (4:9:1)	2.79E+09	52795.12	46986.78	9.91
ARM&MLP (4:8:1)	1.1E+09	33301.77	24650.6	5.70
ARM&RBF (4:4:1)	3.04E+09	55152.88	47265.68	11.03
RBF (3:18:1)&SSM	7.69E+09	87696.31	82650.59	17.68
RBF(3:18:1)&RSM	9.87E+08	31418.91	25572.10	5.93
RBF(3:18:1)&ARM	5.21E+08	22815.30	18534.60	4.17
MLP (1:8:1)&SSM	1.12E+09	33402.51	29412.32	6.22
MLP (1:8:1)&RSM	1.37E+09	36995.60	27762.00	5.71
MLP (1:8:1)&ARM	1.40E+09	37370.19	27877.08	5.73

***** Indicates the model having the best performance

4 Conclusions

It is known that hybrid models indicate very good performance in time series forecasting problems. Zhang [1] reported that hybrid models where a component is linear and the other is nonlinear have demonstrated a good performance in time series forecasting. Then, a study has been made by Aslanargun et al., [2] and found that usage of hybrid models, whose components are nonlinear, are more effective.

In this paper, in the time series forecasting problems based on two real data sets, we observed that hybrid models where one of its components is nonparametric regression model and the other is ANN, make more better forecasting than hybrid models discussed in Zhang [1] and Aslanargun et al., [2]. We noticed that hybrid models, whose first component is SSM and second component is MLP, have indicated very good performance. At the same time, the hybrid models, whose second component is particularly MLP, have denoted good results too.

As a result, our opinion is that, using hybrid models, whose components are nonparametric regression and ANN, can be more useful in time series forecasting problems included seasonality and trend. Especially, for prediction problems of the time series data included trend and seasonality, our sight is that, hybrid models having the form SSM&MLP will be having much better performance.

As individual models, the SSM among the nonparametric models, and the MLP (CG algorithm) among the ANN models, can be used as model having a good performance.

References:

[1] Zhang, G.P., Time-series forecasting using a hybrid ARIMA and neural network model, Neurocomputing, Vol.50, 2003 159–175.

[2] Aslanargun A., Mammadov M., Yazıcı B., and Yolacan, S. Comparison of ARIMA, Neural Networks and Hybrid Models in Time Series: Tourist Arrival Forecasting, JSCS, Vol.77, 2007, pp. 29-53.

[3] Tseng, F.M. Yu, H.C. and Tzeng G.H., ombining neural network model with seasonal time series ARIMA model. Technological Forecasting & Social Change, Vol. 69, 2002, pp. 71–87.

[4] E. Ferreira, V. Nunez-Anton, and J. Rodriguez-Poo., Semi-parametric Approaches to Signal Extraction Problems in Economic Time Series, CSDA, Vol.33**,** 2000**,** pp. 315-333.

[5] Wahba, G., Spline Models Of Observational Data*,* University Of Winconsin At Madison, Pensilvenya. 1990.

[6] Hardle W., Applied Nonparametric Regression, Cambridge University Press, Cambridge, 1991.

[7] Green P.J., and Silverman B.W., Nonparametric Regression and Generalized Linear Models*,* Chapman &Hall, London, 1994.

[8] Hastie, T., and Tibshirani, R.J., Generalized Additive Models*,* Chapman & Hall, London, 1999.

[9] Hardle W., Müler M., Sperlich S., and Werwatz A., Nonparametric and Semiparametric Models*,* Springer, New York, 2004.

[10] Box, G.P. and Jenkins, G.M., Time Series Analysis, Forecasting and Control, CA: Holden-Day , 1970.

[11] Bishop, C.M., Neural Networks for Pattern Recognition Oxford: Clarenden Press, 1995.

[12] Haykin, S. Neural Networks: A Comprehensive Foundation*,* Prentical Hall, 1999.

[13] Moller, M., A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks, Vol. 6(4), pp. 525–533. 1993.

[14] Nocedal, J. and Wright, S.J., Numerical Optimization*,* NewYork: Springer-Verlag, 1999.

[15] Rumelhart, D.E., Hinton, G.E. and Willams, R.J., Learning representation by backpropagating errors. Nature, Vol.323, 1986, .pp. 533–536.

[16] Zhang, G.P., Patuwo, E.B. and Hu, M.Y., Forecasting with artificial neural networks: the state

of the art. International Journal of Forecasting, Vol. 14, 1998, pp. 35–62.
[17] Eubank, R.L. Nonparametric Regression and Smoothing Spline, Marcel Dekker Inc., 1999.
[18] Ruppert, D.. Selection the number of knots for penalized spline, Journal of Computational and Graphical Statistics, 11, 2002, pp.735-757.
[19] Ruppert, D., Carrol, R., Spatially-adaptive penalties for penalized spline, New Zeland J.statist, Vol.42, 2000, pp. 205-224..
[20] Ruppert, D., Wand, M.P., Carrol, R.J. Semi-parametric regression, Cambridge university press, Cambridge, 2003.
[21] The central bank of the republic of Turkey: www.tcmb.gov.tr.
[22] Carey G, Rob L., Modeling and forecasting tourism demand for arrivals with stochastic nonstationary seasonality and intervention, Tourism Management Vol.23, 2002, pp. 499–510.