ANN Models and Bayesian Spline Models for Analysis of Exchange Rates and Gold Price

Ozer Ozdemir® **Memmedaga Memmedli** **Akhlitdin Nizamitdinov**

Anadolu University Anadolu University Anadolu University

ABSTRACT

ANN (Artificial Neural Network) models and Spline techniques have been applied to economic analysis, to handle economic problems, evaluate portfolio risk and stock performance, and to forecast stock exchange rates and gold prices. These techniques are improving nowadays and continue to serve as powerful predictive tools.
In this study, we compare the performance of ANN models and Bayesian Spline models in forecasting economic datasets. We consider the most commonly used ANN models, which are Generalized Regression Neural Networks (GRNN), Multilayer Perceptron (MLP), and Radial Basis Function Neural Networks (RBFNN). We compare these models using BayesX and Statistica software with three important economic datasets: on the exchange rate of Turkish Liras (TL) to Euro, exchange rate of Turkish Liras (TL) to United States Dollars (USD), and Gold Price for Turkey. With these three economic datasets, we made a comparative study of these models, using the criterions MSE and MAPE to evaluate their forecasting performance. The results demonstrate that the penalized spline model performed best amongst the spline techniques and their Bayesian versions. Amongst the ANN models, the MLP model obtained the best performance criterion results.

Key words: *Artificial Neural Networks, Bayesian Spline Models, Exchange Rates*
JEL Classifications: C11, C45, C53

1. INTRODUCTION

Alfred Greiner (2009) conducted a study analyzing primary-surplus–to-GDP ratio and the debt ratio using the penalized spline estimation technique. This technique indicates that fiscal policy is sustainable if the primary–surplus-to-GDP ratio is a linear or convex function of the debt ratio. Greiner and Kauermann (2007) estimated these economic variables using a nonlinear technique and showed that the relation between the surplus and the debt ratio could be nonlinear.

Another study utilizing the spline technique, by Audrino and Bühlmann (2009), proposes a flexible generalized autoregressive conditional heteroscedasticity type model for predicting volatility in financial time-series, using two financial datasets: 3376 daily log-returns from the US Standard and Poor's index S&P500 and yields from the 30-years US Treasury bonds between 01.1990 and 10.2003.

® Ozer Ozdemir, Anadolu University, Faculty of Science, Department of Statistics, 26470, Eskisehir, Turkey, (email: ozerozdemir@anadolu.edu.tr), Tel: +902223350580/4668, Fax: +902223204910.
Memmedaga Memmedli, Anadolu University, Faculty of Science, Department of Statistics, 26470, Eskisehir, Turkey, (email: mmammadov@anadolu.edu.tr).
Akhlitdin Nizamitdinov, Anadolu University, Faculty of Science, Department of Statistics, 26470, Eskisehir, Turkey, (email: ahlidin@gmail.com).

In a study using regression splines to model time-series in finance applications, Wong and Lai (2004) demonstrated the flexibility and computational tractability of regression splines in modeling American option prices, nonlinear autoregressions with exogenous inputs and occasional parameter jumps, and asymmetric volatility clusters.

For this paper, which conducts an empirical analysis of financial time-series, forecasting time-series data is of obvious interest, as empirical financial analysis is highly important for prediction in the economic world.

Unfortunately, we are often not able to generate a very accurate prediction. Financial time-series typically display nonlinear effects. For this reason, we consider nonlinear forecasting models here, based on nonparametric regression. The primary goal of this study is to compare the two proposed techniques: ANN models and Spline techniques.

There exist various techniques in nonparametric statistics, like regression splines (Wood, 2003), smoothing splines (Green and Silverman, 1994), B-splines (de Boor, 1978; Dierckx, 1993), and P-splines (Eilers and Marx, 1996). At this time Bayesian versions of the spline approach are also widely used in nonparametric problems, such as the Bayesian penalized spline (Lang and Brezger, 2001), and adaptive Bayesian regression spline (Biller, 2000). These techniques are all aimed at one problem: to make a precise prediction while allowing for flexibility in models.

In this paper we also use artificial neural networks, which are mathematical models imitating biological neural networks. Artificial neural networks are defined by three fundamental elements: architecture structure, learning algorithm, and activation function (Egrioglu et al., 2009). These elements determine the forecasting performance of artificial neural networks, and thus appropriately determining them is an issue that should be considered carefully (Aladag et al., 2009). One important parameter is network architecture, and although there are no general rules for determining the best architecture, different types of architecture should be tried for the most accurate results. Of the various types of artificial neural networks in existence, in this paper we use multilayer perceptron (MLP), Radial Basis Function (RBF) and Generalized Regression Neural Networks (GRNN) in forecasting time-series data.

In our previous studies involving time-series data, we compared the cubic smoothing spline and penalized spline models (Nizamitdinov et al, 2010a). In another study, we compared artificial neural network models and the fuzzy time-series approach with spline functions using daily, weekly, and monthly closing prices of Istanbul Stock Exchange (ISE) national-100 index (Nizamitdinov et al., 2010b).

In this paper we apply nonparametric analysis using P-splines and cubic regression spline, Bayesian penalized spline and adaptive Bayesian regression spline, and neural network models such as Generalized Regression Neural Networks (GRNN), Multi-Layer Perceptron (MLP), and Radial Basis Function (RBF) using three different time-series datasets: Weekly Exchange Rate of Euro/TL during 06.01.2006-31.12.2010, Weekly Exchange Rate of Dollar/TL during 06.01.2006-31.12.2010, and Monthly Gold Price during 2001-2010. Forecasting results of models are then compared using mean squared error (MSE) and mean absolute percentage error (MAPE) performance criterions.

The regression spline, penalized spline, and Bayesian splines are briefly introduced and described in Section 2. Section 3 offers a review of artificial neural networks. Section 4 presents the comparative empirical analysis. Finally, Section 5 concludes the paper.

2. SOME THEORY ABOUT NONPARAMETRIC SPLINES

The nonparametric regression model has the following form:

$$y_i = f(x_i) + \varepsilon_i, \quad a < x_1 < ... < x_n < b \tag{2.1}$$

where $f \in C^2(a,b)$ is an unknown smooth function; y_i, $i = 1,...,n$ are observation values of the response variable y; x_i, $i = 1,...,n$ are observation values of the predictor variable x and ε_i normal distributed random errors with zero mean and common variance σ^2.

The basic aim of the nonparametric regression is to estimate unknown function $f \in C^2(a,b)$ (of all functions f with continuous first and second derivatives) in model (2.1). In the nonparametric regression, function f is some unknown smooth function.

Smoothing spline (Green and Silverman, 1994) estimate of the function solves the following minimization problem:

Find $\hat{f} \in C^2(a,b)$ that minimizes the penalized residual sum of squares, defined as:

$$S(f) = \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda \int_a^b \{f''(x)\}^2 dx \tag{2.2}$$

for some value $\lambda > 0$. The first term in equation (2.2) denotes the residual sum of the squares, and it penalizes the lack of fit. The second term which is weighted by λ denotes the roughness penalty. In other words, it penalizes the curvature of the function f. The λ in (2.2) is known as the smoothing parameter.

The solution based on smoothing spline for minimum problem in the equation (2.2) is known as a "natural cubic spline" with knots at $x_1,...,x_n$. From this point of view, a special structured spline interpolation which depends on a chosen value λ becomes a suitable approach of function f in model (2.1).

Let $f = (f(x_1),...,f(x_n))$ be the vector of values of function f at the knot points $x_1,...,x_n$. The smoothing spline estimate $\hat{f}_\lambda$ of this vector or the fitted values for data $y = (y_1,...,y_n)'$ are given by:

$$\hat{f}_\lambda = \begin{bmatrix} \hat{f}_\lambda(x_1) \\ \hat{f}_\lambda(x_2) \\ . \\ . \\ . \\ \hat{f}_\lambda(x_n) \end{bmatrix}_{(n\times 1)} = (S_\lambda)_{(n\times n)} \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_n \end{bmatrix}_{(n\times 1)} \tag{2.3}$$

where $\hat{f}_\lambda$ is a natural cubic spline with knots at $x_1,...,x_n$ for a fixed smoothing parameter $\lambda > 0$, and S_λ is a positive-definite smoother matrix, which depends on λ and the knot points $x_1,...,x_n$. For general references about smoothing spline, see Green and Silverman (1994).

The problem of choosing the smoothing parameter is one of the main problems in curve estimation. If we use fitting curves by polynomial regression, the choice of the degree of the fitted polynomial is essentially the equivalent of the choice of a smoothing parameter. There are a number of different methods to choose a smoothing parameter. Probably the most well-known is generalized cross-validation. The generalized cross-validation (GCV), a modified form of cross-validation, is a popular method for choosing the smoothing parameter. The first reference was given by Craven and Wahba (1979).

The basic idea of generalized cross validation is to replace the denominators $1-(S_\lambda)_{ii}$ of Cross Validation by their average $1-n^{-1}tr(S_\lambda)$, producing the Generalized Cross Validation score function defined below:

$$GCV(\lambda)=\frac{1}{n}\frac{\sum_{i=1}^{n}\left\{y_i-\hat{f}_\lambda(x_i)\right\}^2}{\left\{1-n^{-1}tr(S_\lambda)\right\}^2} \tag{2.4}$$

the residual sum of squares about $\hat{f}$ divided by a correction factor of $n\{1-n^{-1}tr(S_\lambda)\}^2$. Just as in cross-validation, the GCV choice of smoothing parameter is then carried out by minimizing the function GCV(λ) over λ.

There are many types of cross validation, detailed explanations of which can be found in Basci et al. (2010). For a justification of choosing GCV over CV, as well as information about different types of cross validation see Basci et al. (2010). The next technique that we used in study is P-splines (Eilers and Marx, 1996). Eilers and Marx (1998) make some significant changes in the smoothing spline technique. They made the following two assumptions: First, they assume that $E(y)=Ba$ where $B=(B_1(x), B_2(x),...,B_k(x))$ is a $n\times k$ matrix of B-splines, and that a is the vector of regression coefficients; secondly, they suppose that the coefficients of adjacent B-splines satisfy certain smoothness conditions that can be expressed in terms of finite differences of the a_is. Thus, from a least-squares perspective, the coefficients are chosen to minimize the following:

$$S=\sum_{i=1}^{m}\left\{y_i-\sum_{j=1}^{n}a_jB_j(x_i)\right\}^2+\lambda\sum_{j=k+1}^{n}(\Delta^k a_j)^2 \tag{2.5}$$

For least squares smoothing we have to minimize S in (2.5). The system of equations that follows from minimization of S can be written as:

$$B'y=(B'B+\lambda D_k'D_k)a \tag{2.6}$$

where D_k is a matrix representation of the difference operator Δ^k, and the elements of **B** are $b_{ij}=B_j(x_i)$. The B-spline $B_{i,k+1}$ of degree with knots $\tau_i,...,\tau_{i+k+1}$ is defined as:

$$B_{i,k+1}(x)=(\tau_{i+k+1}-\tau_i)\Delta_t^{k+1}(\tau_i,\ldots,\tau_{i+k+1})(t-x)_+^k \tag{2.7}$$

Using the explicit expression property, the B-spline representation can be expressed as follows:

$$B_{i,k+1}(x)=(\tau_{i+k+1}-\tau_i)\sum_{j=0}^{k+1}\frac{(\tau_{i+j}-x)_+^k}{\prod_{\substack{l=0\\l\neq j}}^{k+1}(\tau_{i+j}-\tau_{i+l})} \tag{2.8}$$

which shows that this function is indeed a spline with $\tau_i,...,\tau_{i+k+1}$ as active knots.

The Bayesian P-splines approach developed by Stefan Lang and Andreas Brezger (2001) for additive models and extensions by replacing difference penalties with their stochastic analogues, i.e. Gaussian (intrinsic) random walk priors which serve as smoothness priors for

the unknown regression coefficients. The approach generalizes work by Fahrmeir and Lang (2001) based on simple random walk priors. A closely related approach based on a Bayesian version of smoothing splines can be found in Hastie and Tibshirani (1990). Compared to smoothing splines, in a P-splines approach a more parsimonious parameterization is possible, which is of particular advantage in a Bayesian framework where inference is based on Markov Chain Monte Carlo (MCMC) techniques.

Priors for the regression parameters of nonlinear functions are defined by replacing the difference penalties of penalized splines based on B-spline basis by their stochastic analogues. First differences correspond to a first order random walk and second differences to a second order random. Thus, the following is obtained:

$$\beta_{jp} = \beta_{j,p-1} + u_{jp} \text{ or } \beta_{jp} = 2\beta_{j,p-1} - \beta_{j,p-2} + u_{jp} \tag{2.9}$$

The priors can be equivalently written in the form of global smoothness priors.

$$\beta_j \left| \tau_j^2 \right. \propto \exp(-\frac{1}{2\tau_j^2} \beta_j' K_j \beta_j) \tag{2.10}$$

with appropriate penalty matrix $\mathbf{K}_j$. For full Bayesian inference, the unknown variance parameters τ_j^2 are also considered as random and estimated simultaneously with the unknown β_j. Therefore, hyperpriors are assigned to the variances (and the overall variance parameter σ^2) in a further stage of the hierarchy by highly dispersed (but proper) inverse Gamma priors $p(\tau_j^2) \sim IG(a_j,b_j)$. The prior for must not be diffuse in order to obtain a proper posterior for β_j, see (Hobert and Casella, 1996) for the case of linear mixed models. A common choice for the hyperparameters is and a small value for b_j, e.g. $b_j = 0.005$, $b_j = 0.0005$ or $b_j = 0.00005$, leading to almost diffuse priors for τ_j^2.

Adaptive Bayesian regression spline was introduced by Biller (2000). He supposed a fully Bayesian approach to regression splines with automatic knot selection in generalized semi parametric models. As a basis function representation of the regression spline he used B-spline basis. The reversible jump Markov chain Monte Carlo method allows for estimation of both the number of knots and the knot placement, together with the unknown basis coefficients determining the shape of the spline (Biller, 2000).

3. ARTIFICIAL NEURAL NETWORKS

McCulloch and Pitts (1943) first presented a mathematical model of a neuron. Since then many artificial neural networks have been developed building on the well-known McCulloch and Pitts model (see for example, Rumelhart and McClelland, 1986; Mammone and Zeevi, 1991; and Zhang and Zhang, 1999).

The following three sub-sections present an overview of the three neural networks used in this study.

3.1. Multilayer Perceptron (MLP) Neural Network

Since the development of the back-propagation learning algorithm in the mid-1980s, the multilayer perceptron neural network (MLP) is one of the most widely used neural networks in the field of neural computing (Rumelhart et al., 1986; Vaughn, 1999). The weights and thresholds learned by the network during supervised training are not fully understood however. Furthermore, neural networks are currently undermined by their inability to explain

or justify their output classifications, and the MLP network is widely regarded as a black box (Vaughn, 1999).

In multilayer perceptron networks (MLP-networks), the processing units are arranged vertically in several layers. Connections exist only between units in adjacent layers. The bottom layer is called the input layer, because the activations of the units in this layer represent the input of the network. Correspondingly, the top layer is called the output layer. Any layers between the input layer and output layer are called hidden layers; their activations are not visible externally.

The MLP is a very simple model of biological neural networks; based on the principle of a feedforward flow of information, i.e., the network is structured in a hierarchical way. The MLP consists of different layers where information flows only from one layer to the adjacent layer. From a theoretical point of view, it is not necessary to consider more than one output unit, because two or more output units could be realized by considering two or more MLPs in parallel. However, if the outputs are correlated, it may be possible to achieve the same approximation results with fewer hidden units. The input units play no active role in processing the information flow, because they just distribute the signals to the units of the first hidden layer. All hidden units work in an identical way, and the output unit is a simpler version of a hidden unit. In an MLP, each hidden unit transforms the signals from the former layer to one output signal, which is distributed to the next layer. Each hidden unit has an activation function, which is, in general, nonlinear and is the same for all hidden units. The output of a hidden unit is determined by applying the activation function on the sum of the weighted signals from the former layer and an individual bias. In the output unit, the activation function is the identity function (Trenn, 2008).

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

$$net_j = \sum_i a_i w_{ij} + \theta_j \tag{3.11}$$

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

$$a_j = \frac{1}{1+e^{-net_j}} \tag{3.12}$$

The network learns by adapting the weights of the connections between units until the correct output is produced. MLP networks use a variety of learning techniques, the most popular being back-propagation (Haykin, 1999:178-278), which performs a gradient descent search on the error surface. The weight update Δw_{ij}, i.e. the difference between the old and the new value of weight w_{ij}, is defined as:

$$\Delta w_{ij} = \eta a_{pi} \delta_{pj,} \tag{3.13}$$

where

$$\delta_{pj} = \begin{cases} a_{pj}\left(1-a_{pj}\right)\left(t_{pj}-a_{pj}\right), & \text{if } j \text{ is an output unit} \\ a_{pj}\left(1-a_{pj}\right)\sum_k \delta_{pk} w_{jk}, & \text{if } j \text{ is a hidden unit} \end{cases} \tag{3.14}$$

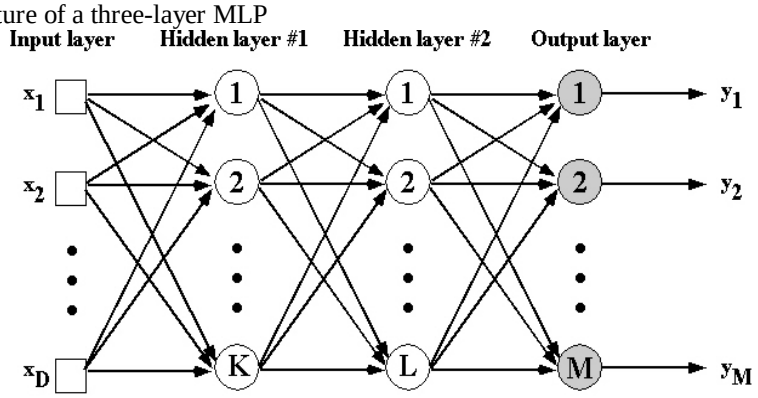
here, t_p is the target output vector that the network must learn.

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

$$\Delta w_{ij}(t+1) = \eta a_{pi}\delta_{pj} + \alpha \Delta w_{ij}(t) \tag{3.15}$$

For example, a three-layer MLP is given in (Figure 3.1).

Figure 3.1 Architecture of a three-layer MLP



3.2. Radial Basis Function (RBF) Neural Network

The locally tuned and overlapped receptive field is a well-known structure that has been studied in regions of the cerebral cortex, visual cortex, and so on. Based on the biological receptive fields, Moody and Darken (1989) proposed a network structure, namely, a Radial Basis Function network that employs local receptive fields to perform function mappings (Trenn, 2008; Lee and Choi, 2001; Jang and Sun, 1993). The RBF network is known in the field for approximation of non-linear functions and pattern recognition. Due to its simple architecture, the RBF network has an especially faster convergence property than the multiplayer neural network (Narendra and Parthasarathy, 1990; Wang, 1994; Wang, 1996; Jang et al., 1997).

RBF networks traditionally have been associated with radial functions in a three-layer network (see Figure 3.2) consisting of an input layer, a hidden layer of radial units, and an output layer of linear units (Moody and Darken, 1989; Renals and Rohwer, 1989).

The radial basis function determines the output with input variable x and distance from center μ. As the input variable approaches the center, the output becomes larger. If the basis functions are appropriately selected, RBF network proves notably adept at approximating any nonlinear function. The Gaussian function is often used as the radial basis function, written as:

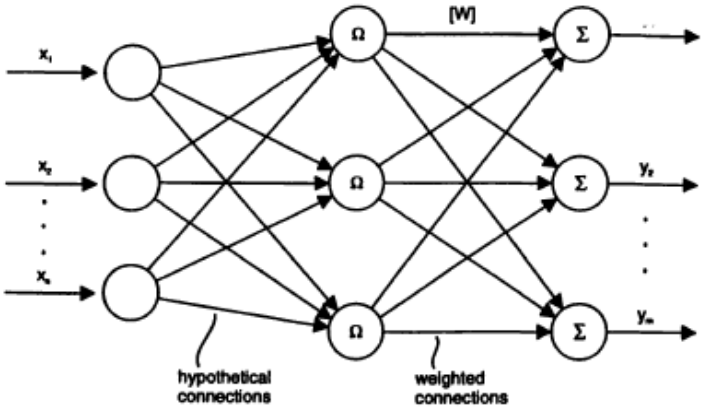
$$f(x) = \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \tag{3.16}$$

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

$$y = \sum_{j=1}^{n} w_j \phi_j \tag{3.17}$$

where w_j: connection weight, ϕ_j: output of basis function. The clustering method is used to determine center μ, and width σ is determined by a heuristic method. For example, the average distance between the center and N neighborhood data is used (Mori and Awata, 2007; Wasserman, 1993). Parameter σ adjusts the response of output to input.

Figure 3.2 Architecture of a RBF Network

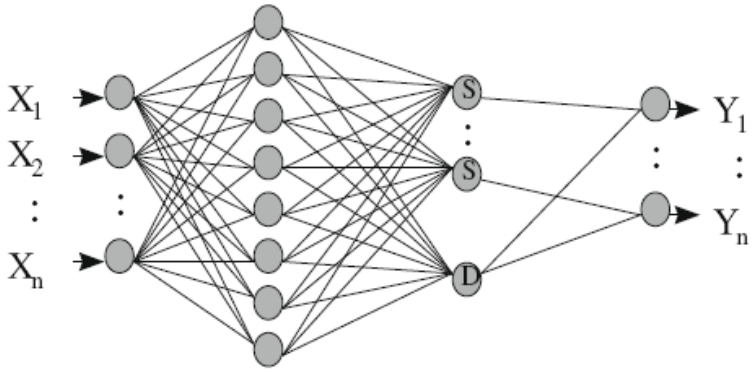


After evaluating the parameters μ and σ, the RBF network needs to determine the weight between the hidden and the output layers, which is evaluated by the steepest descent method (Mori and Awata, 2007). The RBF network is easier to tune up compared to the MLP, since unlike the MLP, RBF network has the weight of unity between the input and the hidden layers (Bishop, 1995).

3.3. Generalized Regression Neural Network (GRNN)

GRNN can be used for function approximation (Wasserman, 1993), and when enough neurons are available GRNN can approximate a continuous function to any level of accuracy (Wagener et al., 2004). GRNN has exactly four layers: input, a layer of radial centers, a layer of regression units, and output. This network must be trained by a clustering algorithm. This network can be thought of a normalized RBF network in which there is a hidden unit centered at every training case. Figure 3.3 shows the general structure of the GRNN (Mostafa, 2010).

Figure 3.3 General structure of the GRNN



4. DESIGN OF THE ANALYSIS

In this section: The three datasets used for analysis are described in Section 4.1. Section 4.2 reviews the performance criterions used for evaluating the different techniques and finally, Section 4.3 talks about ANN and Spline modeling for economic datasets and presents the results from the analysis (see Makridakis et al., 1993; Ord et al., 2000).

4.1. Data description

In this section three time-series datasets are presented: (1) the dataset on the Exchange Rate of Euro/TL in Turkey (see TCMB, 2011), which consists of weekly time-series, starting 06.01.2006 and ending 31.12.2010, and comprises *n*=261 observations; (2) the dataset on the Exchange Rate of Dollar/TL in Turkey (see TCMB, 2011), which consists of weekly time-series, starting 06.01.2006 and ending 31.12.2010, and comprises *n*=261 observations; (3) is the dataset on Gold Prices in Turkey (see TCMB, 2011), which consists of monthly time-series, starting January 2001 and ending December 2010, and comprises *n*=120 observations.

4.2. Analysis method

The performance of each model is evaluated based on the proximity of the prediction values for test data and the observed values. The mean square error (*MSE*) and mean absolute percentage error (*MAPE*) consistency criterions are used in order to compare the performances of results obtained from spline methods and artificial neural network models. These criterions are defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.18}$$

$$MAPE = \frac{1}{n}\sum\left|\frac{e_i}{y_i}\right|(100), \tag{3.19}$$

There are various other performance measures of accuracy in the forecasting literature without *MSE*, each with particular advantages and limitations (Makridakis et al., 1983). The most frequently used are:

$$\begin{aligned} &\text{The mean absolute deviation } (MAD) = \sum|e_i|/n; \\ &\text{The sum of squared error } (SSE) = \sum(e_i)^2; \\ &\text{The root mean squared error } (RMSE) = \sqrt{MSE}; \end{aligned} \tag{3.20}$$

where $e_i = (y_i - \hat{y}_i)$ and is the individual forecast error; y_i is the actual value; $\hat{y}_i$ is the forecasted value and n is the number of error terms (Zhang et al., 1998).

In this study, we used *MSE* and *MAPE*, which are more commonly used performance measures (and also easy to transform into *RMSE*) in the forecasting literature, especially for ANN models. Also, we found similar results for these performance measures for forecasting the three datasets at hand. We have chosen *MSE* and *MAPE* as the performance criterions in this study for the interest of efficiency of the analysis for all techniques and datasets, and legibility and conciseness of tables and results.

4.3. ANN and Spline modeling for economic datasets

Because the datasets contain time-series data, it is necessary to determine if these data are stationary or not. So, firstly, we use the time-series toolbox and the intelligent problem solver (IPS) module of the STATISTICA to test the datasets for stationarity. After doing the necessary transformation for the given datasets, for time-series forecasting, and after obtaining stationary data, the empirical analysis step is started.

Neural networks are much more complex models than the linear techniques used in traditional statistical modeling. They are more difficult to optimize and require difficult design decisions, such as determining the right type and complexity of network for the problem and the right input variables to use (Asma et al., 2012). The intelligent problem solver (IPS) module of the STATISTICA uses some highly sophisticated algorithms to solve these problems automatically (Hill and Lewicki, 2007).

For our empirical analysis, we used the mgcv package of S. N. Wood (2006) for cubic regression and penalized spline, and the functions from the R program for choosing the smoothing parameter. For the analysis with Bayesian penalized splines we used BayesX software (BayesX, 2011), and the 'BVCM' package (Biller, 2000) for analysis using adaptive Bayesian regression spline, available at SFB386 (2011). Forecasting with artificial neural network models was utilized in STATISTICA.

First, we analyzed the time-series datasets using the following methods:

- Cubic penalized splines with second order difference penalty, with 20, 30, and 40 knots. The smoothing parameter was forecasted using generalized cross validation, where the optimal smoothing parameter was chosen by the R program.

- Cubic regression splines with second order penalty, with 20, 30, and 40 knots. The smoothing parameter was forecasted using generalized cross validation, where the optimal smoothing parameter was chosen by the R program.

- Bayesian cubic penalized spline with second order random walk penalty and 20 knots. We forecasted the models using three different choices for the hyperparameters a and b. We used $a = 1$, $b = 0.005$, $a = 1$, $b = 0.0005$ and $a = 1$, $b = 0.00005$ as in the paper by Lang and Brezger (2001).

- The adaptive Bayesian regression spline by Biller (2000) was used as an example of a competing Bayesian approach. We used 20 knots, which was default in the "BVCM" package (Biller, 2000), available at SFB386 (2011). Experiments with 30 and 40 knots showed worse results than with 20 knots.

In the following tables, we summarized the performance criterions (*MSE* and *MAPE*) for all datasets (out of sample). Table 4.1 contains information about the performance criterions of the forecasting results of the Gold time-series dataset. As we can see from this table, the best result is attained using the cubic penalized spline according to *MSE* and *MAPE* performance criterions. Among the Bayesian spline techniques, the Bayesian penalized spline outperforms the adaptive Bayesian regression spline method, according to both *MSE* and *MAPE* results.

In Table 4.2, we disclose the forecasting results of Exchange rate of Euro/TL time-series dataset. As we can see from this table, in general, we can say that the best performance

criterion is attained using the cubic penalized spline technique. Amongst the Bayesian spline methods, the Bayesian penalized spline shows better results according to *MSE* and *MAPE* performance criterions.

Methods	***MSE*** *k*=20 knots	***MAPE*** *k*=20 knots	***MSE*** *k*=30 knots	***MAPE*** *k*=30 knots	***MSE*** *k*=40 knots	***MAPE*** *k*=40 knots
Cubic regression spline	2.755	17.01	1.170	16.12	0.659	16.02
Cubic penalized spline	2.258	14.44	0.961	14.37	0.733	14.35
Adaptive Bayesian regression spline	5.013	23.05	4.055	24.06	4.326	25.12
For Bayesian penalized splines selection of hyper parameters						
	MSE *a*=1, *b*=0.005	***MAPE*** *a*=1, *b*=0.0005	***MSE*** *a*=1, *b*=0.0005	***MAPE*** *a*=1, *b*=0.000	***MSE*** *a*=1, *b*=0.00005	***MAPE*** *a*=1, *b*=0.00005
Bayesian penalized splines	3.349	18.3	3.348	17.8	3.319	17.3

Table 4.1 The performance criterions (*MSE* and *MAPE*) of forecasting results of Gold time-series dataset.

Methods	***MSE*** *k*=20 knots	***MAPE*** *k*=20 knots	***MSE*** *k*=30 knots	***MAPE*** *k*=30 knots	***MSE*** *k*=40 knots	***MAPE*** *k*=40 knots
Cubic regression spline with	0.0021	30.06	0.0019	30.79	0.0012	29.98
Cubic penalized spline	0.0023	30.04	0.0018	30.06	0.0011	29.76
Adaptive Bayesian regression spline	0.0025	36.45	0.0021	37.8	0.0019	38.9
For Bayesian penalized splines selection of hyper parameters						
	MSE *a*=1, *b*=0.005	***MAPE*** *a*=1, *b*=0.005	***MSE*** *a*=1, *b*=0.0005	***MAPE*** *a*=1, *b*=0.0005	***MSE*** *a*=1, *b*=0.00005	***MAPE*** *a*=1, *b*=0.00005
Bayesian penalized splines	0.0020	31.5	0.0019	30.8	0.0018	30.7

Table 4.2 The performance criterions (*MSE* and *MAPE*) of forecasting results of Exchange rate Euro/TL dataset.

As we can see from the Table 4.3, all methods have similar results to each other. For the Exchange rate of Dollar/TL dataset all methods perform well according to the performance criterion.

Methods	***MSE*** *k*=20 knots	***MAPE*** *k*=20 knots	***MSE*** *k*=30 knots	***MAPE*** *k*=30 knots	***MSE*** *k*=40 knots	***MAPE*** *k*=40 knots
Cubic regression spline with	0.0016	43.4	0.0013	43.2	0.0006	42.91
Cubic penalized spline	0.0019	43.05	0.0013	42.8	0.0004	42.68
Adaptive Bayesian regression spline	0.012	53.42	0.0113	52.4	0.0103	52.6
For Bayesian penalized splines selection of hyper parameters						
	MSE *a*=1, *b*=0.005	***MAPE*** *a*=1, *b*=0.005	***MSE*** *a*=1, *b*=0.0005	***MAPE*** *a*=1, *b*=0.0005	***MSE*** *a*=1, *b*=0.00005	***MAPE*** *a*=1, *b*=0.00005
Bayesian penalized splines	0.0016	44.65	0.0016	44.52	0.0016	43.32

Table 4.3 The performance criterions (*MSE* and *MAPE*) of forecasting results of Exchange rate of Dollar/TL dataset.

Because of the nonlinear structure of the corresponding datasets, ANN is a suitable tool for accurate modeling. The users can determine the number of inputs and outputs, the activation functions, algorithm for obtaining the weights of the network, the number of hidden layers, and the number of neurons inside the hidden layers. Since the combination of all the different choices available would result in various ANN models, the analysis would become overly complicated. Thus, for the sake of simplicity, we utilize the intelligent problem solver (IPS) module of the STATISTICA for deciding the best ANN model for the given datasets. The

corresponding program gives the users the opportunity to construct numerous ANN models at a time and to select the best model. So we run 100000 ANN models to select the most appropriate one for the three datasets: the dataset of the exchange rate of Turkish Liras (TL) to Euro, the dataset of the exchange rate of Turkish Liras (TL) to United States Dollars (USD), and the dataset of Gold Prices for Turkey.

To the best of our knowledge, there are no theoretical reasons for invalidating neural network models built with nonstationary time-series, and in fact many successful neural network approaches for financial forecasting applications do not explicitly address it (Refenes, 1995; Weigend et al., 1992; White, 1988; Virili and Freisleben, 2000). Because of this reason, we did not take into account nonstationary data for neural networks. If we use Box-Jenkins models for forecasting, stationarity should be obtained. There are many studies that consider these situations and compare these models aspect of stationarity (see e.g. Virili and Freisleben, 2000; Zhang and Qi, 2005).

Generalized regression neural network (GRNN), Multilayer Perceptron (MLP), and Radial Basis Neural Network (RBF) are used to analyze these economic datasets. Then, *MSE* and *MAPE* are calculated for each model respectively. The best ones for each model according to MSE and MAPE values with their corresponding ANN model architecture are shown in Table 4.4, 4.5, and 4.6, respectively

Methods	Architecture	*MAPE*	*MSE*
Radial basis function	1:5-11-1:1	265.76	46.03
Multilayer perceptron	1:5-16-16-1:1	59.65	11.65
Generalized regression neural networks	1:10-47-2-1:1	1171.71	228.83

Table 4.4 The performance criterions (*MSE* and *MAPE*) of ANN models of Gold price time-series dataset.

Methods	Architecture	*MAPE*	*MSE* (Multiplied by 10000)
Radial basis function	1:5-8-1:1	11.78	5.80
Multilayer perceptron	1:5-16-16-1:1	13.17	6.59
Generalized regression neural networks	1:10-100-2-1:1	107.67	54.42

Table 4.5 The performance criterions (*MSE* and *MAPE*) of ANN models of Euro/TL time-series dataset.

Methods	Architecture	*MAPE*	*MSE* (Multiplied by 10000)
Radial basis function	1:5-13-1:1	12.43	5.96
Multilayer perceptron	1:4-16-16-1:1	11.94	5.94
Generalized regression neural networks	1:7-100-2-1:1	32.19	18.67

Table 4.6 The performance criterions (*MSE* and *MAPE*) of ANN models of Dollar/TL time-series dataset.

For example, in Table 4.4, the best result for the performance criterions (*MSE* and *MAPE*) of ANN models of Gold price time-series dataset is that obtained by the MLP model, with 1-16-16-1 architecture. Here, MLP has 1 input, 16 hidden nodes in the first hidden layer, 16 hidden nodes in the second hidden layer, and 1 output. All inputs for all of the models are taken at time (*t*-1), and the outputs are taken at time t. We show this model's architecture in Table 4.7.

Layer	Units	Synaptic function	Activation function
1	5	Linear	Linear
2	16	Linear	Hyperbolic
3	16	Linear	Hyperbolic
4	1	Linear	Linear

Table 4.7 MLP 1:5-16-16-1:1 model's architecture.

MLP 1:5-16-16-1:1 model has a total of (5×16+16×16+16×1+33 biases) 385 weights, including 33 biases. It is not necessary to give all weights' and biases' values for all models. Other models in these results can also be interpreted as the MLP 1:5-16-16-1:1 model.

Now, we compare all the techniques that we used in this analysis. Table 4.8 displays the mean squared error criterion for all the utilized methods, and Table 4.9 shows the *MAPE* results for all methods. The *MSE* and *MAPE* results are similar so we can express same comments for both of them to all datasets and methods. As Table 4.3 previously demonstrated, the *MSE* criterion for the cubic regression spline, penalized spline, and adaptive Bayesian spline was minimized when we used 40 knots. Different hyperparameters were used to obtain the best approximation by the Bayesian penalized spline technique, namely, $b = 0.05$, $b = 0.005$, $b = 0.0005$, and $b = 0.00005$. The Bayesian penalized spline method shows the best results for the hyperparameters $a = 1$ and $b = 0.00005$.

Methods/ Dataset	Gold price	Ex. rate of Dollar/TL (Multiplied by 10000)	Ex. rate of Euro/TL (Multiplied by 1000)
Cubic regression spline	0.66	6.14	1.29
Penalized spline	0.73	4.76	1.17
Adaptive Bayesian regression spline	4.33	103.20	1.92
Bayesian penalized spline	3.32	16.32	1.81
Radial basis function	46.03	5.96	0.58
Multilayer perceptron	11.65	5.94	0.65
Generalized regression neural networks	228.83	18.67	5.44

Table 4.8 Performance results of all techniques for *MSE*.

Methods/ Dataset	Gold price	Ex. rate of Dollar/TL	Ex. rate of Euro/TL
Cubic regression spline	16.02	42.91	29.98
Penalized spline	14.35	42.68	29.76
Adaptive Bayesian regression spline	23.05	52.40	38.90
Bayesian penalized spline	18.30	43.32	30.70
Radial basis function	265.76	12.43	11.78
Multilayer perceptron	59.65	11.94	13.17
Generalized regression neural networks	1171.71	32.19	107.67

Table 4.9 Performance results of all techniques for *MAPE*.

5. CONCLUSION

In this paper we analyzed three different time-series datasets: the Weekly Exchange Rate of Euro/TL between 06.01.2006 and 31.12.2010, the Weekly Exchange Rate of Dollar/TL between 06.01.2006 and 31.12.2010, and the Monthly Gold Price between 2001 and 2010. The methods that we used in this study included the following: cubic regression spline, penalized splines, Bayesian penalized splines, adaptive Bayesian regression spline, and artificial neural networks, such as Radial Basis Function (RBF), multilayer perceptron (MLP), and generalized regression neural networks (GRNN). Based on the empirical results of the analysis, we conclude the following:

- For the Bayesian penalized splines method the performance criterion varied depending on the hyperparameters selected. As we can see from the table of results for Gold prices, the performance criterion are worse when the hyperparameters are $a = 1$ and $b = 0.005$ than when they are $a = 1$ and $b = 0.00005$. The table on the Exchange rate Euro/TL dataset, however, highlights that the differences between the results of the different parameters are not so large.

- The adaptive Bayesian regression spline method shows the worst result among all of the techniques used in analysis.

- The penalized spline method had the best performance criterion values between the spline techniques and their Bayesian versions. However, the mean squared error criterion results of the penalized splines, regression spline, and Bayesian penalized spline techniques are very close when used with both the Exchange rate of Dollar/TL dataset and the Exchange rate of Euro/TL dataset. The results of artificial neural networks show that the best performance criterion result was obtained using the multilayer perceptron model. Actually, the results of the multilayer perceptron model are close to that of the radial basis function.

- Among all techniques that we have used in this study, for the Gold price and the Dollar/TL datasets, the penalized cubic spline model had the best performance criterion results, whereas for the Euro/TL dataset, the multilayer perceptron method showed the best forecasting performance.

REFERENCES

Aladag, C.H., M.A. Basaran, E. Egrioglu, U. Yolcu and V.R. Uslu (2009). Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations. *Expert Systems with Applications*, 36, 4228–4231.

Asma, S., A. Sezer and O. Ozdemir (2012). MLR and ANN models of significant wave height on the west coast of India. *Computers & Geosciences*, 49, 231–237.

Audrino, F. and P. Bühlmann (2009). Splines for financial volatility. *Journal of the Royal Statistical Society*. Series B, 71, 655–670.

Basci, S., A. Zaman and A. Kiraci (2010). Variance Estimates and Model Selection. *International Econometric Review*, 2 (2), 57–72.

BayesX (2011). http://www.stat.uni-muenchen.de/~bayesx/ (accessed January 20, 2011).

Biller, C. (2000). Adaptive Bayesian Regression splines in semi parametric generalized linear models. *Journal of Computational and Graphical Statistics*, 9 (1), 122–140.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

Craven, P. and G. Wahba (1979). Smoothing Noisy Data with Spline Functions. *Numerische Mathematik*, 31, 377-403.

De Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Oxford: Clarendon Press.

Egrioglu, E., C.H. Aladag, U. Yolcu, V.R. Uslu and M.A. Basaran (2009). A new approach based on artificial neural networks for high order multivariate fuzzy time-series. *Expert Systems with Applications*, 36, 10589–10594.

Eilers, P.H.C. and B.D. Marx (1996). Flexible smoothing using B-splines and penalized likelihood (with comments and rejoinders). *Statistical Science*, 11 (2), 89–121.

Eilers, P.H.C. and B.D. Marx (1998). Direct generalized additive modeling with penalized likelihood. *Computational Statistics and Data Analysis*, 28, 193–209.

Fahrmeir, L. and S. Lang (2001). Bayesian Inference for generalized additive mixed models based on Markov Random Field Priors. *Journal of the Royal Statistical Society C* (*Applied Statistics*), 50, 201–220.

Green, P.J. and B.M. Silverman (1994). *Nonparametric regression and generalized linear models*. London: Chapman and Hall.

Greiner, A. and G. Kauermann (2007). Sustainability of U.S. Public Debt: Estimating smoothing spline regressions. *Economic Modelling*, 24 (2), 350–364.

Greiner, A. (2009). Estimating penalized spline regressions: theory and application to economics. *Applied Economic Letters*, 16 (28), 1831–1835.

Hastie, T.J. and R.J. Tibshirani (1990). *Generalized Additive Models*. Chapman & Hall/CRC.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1–823.

Hill, T. and P. Lewicki (2007). *STATISTICS: Methods and Applications, StatSoft*. Tulsa: OK.

Hobert, J. and G. Casella (1996). The Effect of Improper priors on Gibbs Sampling in Hierarchical Linear Mixed Models. *Journal of the American Statistical Association*, 91, 1461–1473.

Jang, J.S.R. and C.T. Sun (1993). Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems. *IEEE Transactions on Neural Networks*, 4 (1), 156–159.

Jang, J.S.R., C. T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. New Jersey: Prentice-Hall.

Lang, S. and A. Brezger (2001). Bayesian P-splines. *Journal of Computational and Graphical Statistics*, 13, 183–212.

Lee, M.-J. and Y.-K. Choi (2001). An Adaptive Control Method for Robot Manipulators using Radial Basis Function Networks. In *Industrial Electronics, IEEE International Symposium, Pusan, South Korea*, 3, 1827–1832.

Makridakis, S., S.C. Wheelwright and V.E. McGee (1983). *Forecasting: Methods and Applications*. 2nd ed. New York: John Wiley.

Makridakis, S., C. Chatfield, M. Hibon, M. Lawrence, T. Mills, K. Ord and L.F. Simmons (1993). The M2-Competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting*, 9, 5–22.

Mammone, R.J. and Y.Y. Zeevi (1991). *Neural Networks: Theory and Applications*. San Diego, CA: Academic, 1–376.

McCulloch, W.S. and W. Pitts (1943). A logical calculus of the ideas immanent in neurons activity. *Bulletin of Mathematical Biophysics*, 5, 115–133.

Moody, T. and C. Darken (1989). Fast learning in networks of locally tuned processing units. *Neural Computation*, 1, 281–194.

Mori, H. and A. Awata (2007). Data Mining of Electricity Price Forecasting with Regression Tree and Normalized Radial Basis Function Network. In *IEEE Transactions on Systems, Man, and Cybernetics, IEEE International Conference*, 3743–3748.

Mostafa, M.M. (2010). Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait. *Expert Systems with Applications*, 37 (9), 6302–6309.

Narendra, K.S. and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1 (1), 4–27.

Nizamitdinov A., M. Memmedli and O. Ozdemir (2010a). Comparison Study of P-Spline and Univariate Additive Model (Cubic Smoothing Spline) in Time-Series Prediction. In *24th. Mini EURO Conference on Continuous Optimization and Information-Based Technologies in the Financial Sector (MEC EurOPT 2010), İzmir, TURKEY*, Vilnius "Technika" 2010 (ISI proceedings), 34-39.

Nizamitdinov A., M. Memmedli and O. Ozdemir (2010b). Time Series Forecasting Using Fuzzy Time Series Approach, Neural Network Models and Regression Splines. In *24th European Conference on Operational Research (EURO XXIV LISBON), Lisbon, Portugal*, 257.

Ord, K., M. Hibon and S. Makridakis (2000). The M3-Competition. *International Journal of Forecasting*, 16, 433–436.

Refenes, A.P.N. (1995). *Neural Networks in the Capital Markets*. New York: John Wiley & Sons.

Renals, S. and R. Rohwer (1989). Phoneme classification experiments using radial basis functions. In *Proceedings of International Joint Conference on Neural Networks*, Washington, DC, 461–467.

Rumelhart, D.E. and J.L. McClelland (1986). *Parallel Distributed Processing*, Cambridge, MA: MIT Press.

Rumelhart, D.E., G.E. Hinton and R. J. Williams (1986). Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition*, ed. D.E. Rumelhart and J.L. McClelland. Cambridge, MA: MIT Press, 318–362.

SFB386 (2011). http://www.stat.uni-muenchen.de/sfb386/ (accessed January 20, 2011).

TCMB (2011). http://www.tcmb.gov.tr (accessed January 20, 2011).

Trenn, S. (2008). Multilayer Perceptrons: Approximation Order and Necessary Number of Hidden Units, *IEEE Transactions on Neural Networks*, 19 (5), 836–844.

Vaughn, M.L. (1999). Derivation of the Multilayer Perceptron Weight Constraints for Direct Network Interpretation and Knowledge Discovery. *Neural Networks*, 12, 1259–1271.

Virili, F. and B. Freisleben (2000). Nonstationarity and data preprocessing for neural network predictions of an economic time-series. In *IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 5, 129–134.

Wagener, T., H.S. Wheater and H.V. Gupta (2004). *Rainfall-Runoff Modelling in Gauged and Ungauged Catchments*. London: Imperial College Press.

Wang, L.-X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Wang, L.-X. (1996). Stable adaptive fuzzy controllers with application to inverted pendulum tracking. *IEEE Transactions on Systems, Man, and Cybernetics- Part b: Cybernetics*, 26 (5), 677–691.

Wasserman, P.D. (1993). *Advanced Methods in Neural Computing*. New York: Van Nostrand Reinhold.

Weigend, A., B.A. Huberman and D.E. Rumelhart (1992). *Predicting Sunspots and Exchange Rates with Connectionist Networks, Nonlinear Modelling and Forecasting*. Redwood City, CA: Addison-Wesley.

White, H. (1988). Economic Prediction Using Neural Networks: the Case of IBM Daily Stock Returns. In *Proceedings of the 1988 IEEE International Conference on Neural Networks*, 2, 451–458.

Wong, S.P. and T.-L. Lai (2004). Spline models in time-series analysis. In *Abstract Book of International Conference on Threshold Models and New Developments in Time-series*. Department of Statistics and Actuarial Science University of Hong Kong, 27.

Wood, S.N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65, 95–114.

Wood, S.N. (2006). *Generalized additive models: an introduction with R*. London: Chapman and Hall.

Zhang, G., B.E. Patuwo and M.Y. Hu (1998). Forecasting with artificial neural networks: The state of art. *International Journal of Forecasting*, 14, 35–62.

Zhang, L. and B. Zhang (1999). A Geometrical Representation of McCulloch-Pitts Neural Model and Its Applications. *IEEE Transactions on Neural Networks*, 10 (4), 925–929.

Zhang, G.P. and M. Qi (2005). Neural network forecasting for seasonal and trend time-series. *European Journal of Operational Research*, 160 (2), 501–514.