

Using Support Vector Machines in Financial Time Series Forecasting

Mahmoud K. Okasha

Department of Applied Statistics, Al-Azhar University – Gaza, Palestine

Abstract Forecasting financial time series, such as stock price indices, is a complex process. This is because financial time series are usually quite noisy and involve ambiguous seasonal effects due to holidays, weekends, irregular closure periods of the stock market, changes in interest rates, and announcements of macroeconomic and political events. Support vector machines (SVM) and Artificial neural networks (ANN) have been used in a variety of applications, mainly in classification, regression, and forecasting problems. In the SVM method for both regression and classification, data is mapped to a higher-dimensional space and separated using a maximum-margin hyperplane. This paper investigated the application of SVM in financial forecasting. The autoregressive integrated moving average (ARIMA), ANN, and SVM models were fitted to Al-Quds Index of the Palestinian Stock Exchange Market time series data and two-month future points were forecast. The results of applying SVM methods and the accuracy of forecasting were assessed and compared to those of the ARIMA and ANN methods through the minimum root-mean-square error of the natural logarithms of the data. We concluded that the results from SVM provide a more accurate model and a more efficient forecasting technique for such financial data than both the ANN and ARIMA models.

Keywords ARIMA model, Artificial neural networks, Back-propagation, Forecasting, Kernel function, Nonlinear time series, Support vector machine

1. Introduction

Recently, forecasting of future observations based on time series data has received great attention in many fields of research. Several techniques have been developed to address this issue in order to predict the future behaviour of a particular phenomenon. The traditional approach based on Box and Jenkins' Autoregressive Integrated Moving Averages (ARIMA) models is commonly used because the resulting models are easy to understand and interpret. Support vector machines (SVM) and artificial neural networks (ANN) are alternative methods that can be used for forecasting in nonlinear time series and can overcome the problems of nonlinearity and nonstationarity. The use of SVM and ANN is increasing rapidly because of their ability to form complex nonlinear systems for forecasting based on sample data. In particular, in recent years, SVM and ANN have been applied in economic forecasting to predict stock market indicators in line with economic growth in various countries[1]. When fitting ARIMA models to economic and financial data that are either nonlinear or non-stationary time series, the results of forecasting are expected to be

inaccurate. The forecast values tend to converge to the mean of the series after a few forecast values. Thus, alternative forecasting methods such as SVM need to be examined on non-linear and non-stationary time series.

The research problem in this study involves the applicability of the SVM method and its ability to forecast financial time series data; to investigate this, we compare SVM with those of the ARIMA and ANN techniques. The data used in this investigation is a time series that represents the daily scores of Al-Quds index of the Palestine Stock Exchange (PSE), published in the Palestine Stock Exchange [2]. The number of observations in the series is 1,321, representing daily scores in the period from August 1, 2007 to December 31, 2012, a period which includes the recent economic crises in the global financial market. The PSE operates five days per week, excluding national and religious holidays. Al-Quds index is the main indicator used to describe changes in stock prices in the market. This is an index number that measures the overall level of rise and decline in the prices of companies trading on the PSE. It is easy to see that the time series is not a stationary one.

Several studies have been conducted on the comparison between ARIMA models, SVM, and ANN in forecasting using time series data. Most of these studies have been data-based and many have used economic data. Kuan and White[3] discussed the possibility of using ANN for economic variables and the usability of traditional time

* Corresponding author:

Mahmoud K. Okasha (m.okasha@palnet.com)

Published online at <http://journal.sapub.org/statistics>

Copyright © 2014 Scientific & Academic Publishing. All Rights Reserved

series models, and emphasized the similarities between the two methods. In a similar study, Yao and Tan[4] used neural networks to predict several kinds of long-term exchange rates, whereas Tkacz[5] compared the forecasting abilities of time series models, linear models, and ANN models using Canadian gross domestic product (GDP) data and financial variables. Zhang[6] used a hybrid approach that combined the ARIMA and ANN models. Junoh[7] forecast the GDP of the Malaysian economy using information based on economic indicators. Mohammadi, et al.[8] compared several methods of forecasting the spring inflow to the Amir Kabir reservoir in the Karaj river watershed. Rutka[9] conducted a study to forecast network traffic using ARIMA and ANN. Some of these studies showed that ANN has limitations, such as an overtraining problem that emerges from the implementation of empirical risk minimization principles, making it fall into a local optimal solution. It is also necessary to select a large number of controlling parameters, which is very difficult to carry out.

SVMs were originally developed by Vapnik[10] for pattern recognition problems to provide a novel approach to improve the generalization property of neural networks. Recently, with the introduction of the e-insensitive loss function, SVMs have been extended to solve nonlinear regression, classification, and time series forecasting problems[11, 12, 13, 14]. SVMs' ability to solve nonlinear regression estimation problems makes it a promising technique in time series forecasting[15, 16]. This has become a topic of intense interest due to its successful application in classification and regression tasks. Studies on SVM have shown some success in application to certain fields, such as pattern recognition and function regression. In terms of the application of SVM to financial time series forecasting, Kim[17] applied SVM to predict the stock price index for South Korea, while Tay & Cao[18] used SVM to predict five kinds of exchange rates, including GBP/USD, for the purpose of comparing between SVM and neural networks.

In this paper, we compare the performance of SVM with the ARIMA models and ANN learning theory, using a real-world dataset to train the models and to create a two-month forecast (10% of the number of observations). The results show that the tendencies of the predicted value curve using SVM are basically identical to those of the actual value curve. In addition, since there is no structured way to choose the optimal parameters of SVM, this study investigates the variability in performance with respect to the parameters.

2. Artificial Neural Networks

Rosenblatt[19, 20] developed the first single feed-forward network. Here, the output obtained from this single layer was the weighted sum of various inputs. A major development in ANN occurred when Cowan[21],

introduced new functions, such as activation of the smooth sigmoid function, which have the capacity to deal with nonlinear functions more effectively than the learning process "perceptron" model. The procedure which uses the gradient-descent learning technique for multilayer feed-forward ANN is known as back-propagation, or the generalized delta rule, as set forth by Rumelhart, et al.[22] and developed by Zou et al.[23]. Initial weights are selected randomly between -1 and $+1$ and the power of NN models largely depends on how their layer-connection weights are adjusted over time. The weight adjustment process is known in NN methodology as training of the network. The objective of the training process is to update the weights in such a way as to facilitate learning of the patterns inherent in the data. The data is divided into two groups—the training group and the test group, where the training group is used to estimate the weights in the model.

The network outputs depend on the input units, hidden units, weights of the network, and the activation function. The ANN method uses the error or cost function to measure the difference between the target value and the output value. The back-propagation method takes the network error and propagates it backward into the network. Errors are used at each neuron to update the weights. The weights of the network are frequently adjusted in such a way that the error or objective function becomes as small as possible.

The output of ANN, assuming linear output neuron j , a single hidden layer with h sigmoid hidden nodes, and an input variable (x_i), is given by:

$$x_j = g \left(\sum_{j=1}^h w_j f(s_j) + b_k \right) \quad (1)$$

where $g(\cdot)$ is the linear transfer function of output neuron k and b_k is its bias; w_j is the connection weight between hidden layers and output units; and $f(\cdot)$ is the transfer function of the hidden layer.

Transfer functions can take several forms. The most widely used transfer functions are:

$$\text{Log-sigmoid: } f(s_i) = \log \text{sig}(s_i) = \frac{1}{1 + \exp(-s_i)}$$

$$\text{Linear: } f(s_i) = \text{purelin}(s_i) = s_i \quad (2)$$

Hyperbolic tangent sigmoid

$$f(s_i) = \text{tansig}(s_i) = \frac{2}{1 + \exp(-2s_i)} - 1$$

where $S_i = \sum_{i=1}^n w_i x_i$ is the input signal, referred to as the weighted sum of incoming information.

The gradient descent method is utilized to calculate the weights of the network and to adjust the weight of interconnection to minimize the sum of the squared error (SSE) of the network. This is given by:

$$SSE = \frac{1}{2} \sum_k (Y_k - \hat{Y}_k)^2 \quad (3)$$

where Y_k and \hat{Y}_k are the true and predicted output vectors, respectively, of the k^{th} output node. The constant $\frac{1}{2}$ is used to facilitate computation of the derivative for the error function, which is essential in estimating the parameters[24].

For a univariate time series forecasting problem, the inputs of the network are the past lagged observations $(x_{t-1}, x_{t-2}, \dots, x_{t-p})$ and the output is the predicted value (x_t) [25]. Hence, ANN can be written as:

$$x_t = g(x_{t-1}, x_{t-2}, \dots, x_{t-p}, w) + \varepsilon_t \quad (4)$$

where w is a vector of all parameters and $g(\cdot)$ is a function determined by the network structure and connection weights. For more detailed information on the use and application of ANN for a time series, see Okasha & Yassin[26], and Tseng, Yu & Tzeng[27].

3. Support Vector Machines

SVM is used for a variety of purposes, particularly classification and regression problems. SVM can be especially useful in time series forecasting, from the stock market to chaotic systems[28]. The method by which SVM works in time series is similar to classification: Data is mapped to a higher-dimensional space and separated using a maximum-margin hyperplane. However, the new goal differs in that our goal is to find a function that can accurately predict future values[29].

Consider a given training set of n data points $\{x_i, y_i\}_{i=1}^n$ with input data $x_i \in \mathfrak{R}^p$, where p is the total number of data patterns and the output is $y_i \in \mathfrak{R}$. Generally, the idea of building SVM to approximate a function involves mapping the data x into a high-dimensional feature space via nonlinear mapping and performing a linear regression in the feature space. SVM approximates the function in the following form:

$$y(x) = w^T \varphi(x) + b \quad (5)$$

where, $\varphi(x)$ represents a high-dimensional feature space that is nonlinearly mapped from the input space x [30]. The coefficients w and b are estimated by minimizing the following function:

$$\text{Min} \left(\frac{1}{2} w^T w \right) \quad (6)$$

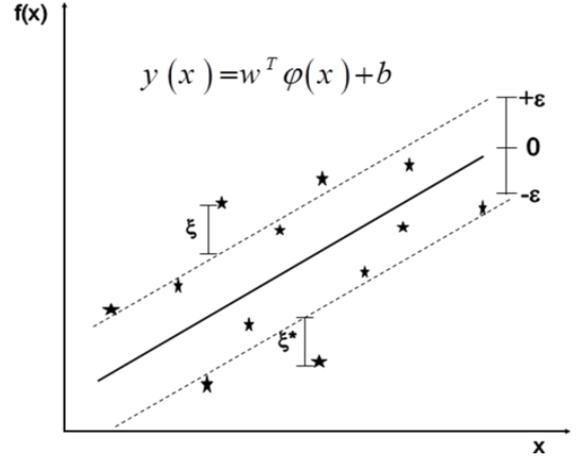
subject to the following constraints:

$$\begin{aligned} y_i - w^T \varphi(x_i) - b &\leq \varepsilon \\ w^T \varphi(x_i) + b - y_i &\leq \varepsilon \end{aligned} \quad (7)$$

This gives:

$$y_i (w x_i + b) - 1 \geq 0 \quad \text{for } i = 1, 2, \dots, n \quad (8)$$

To estimate w and b , the above equation is transformed to the prime function below by introducing the positive slack variables ξ and ξ^* as follows:



Minimize

$$R(w, \xi, \xi^*) = \frac{1}{2} w^T w + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (9)$$

subject to the constraints:

$$\begin{aligned} y_i - w^T x_i - b &\leq \varepsilon + \xi_i \\ w^T x_i + b - y_i &\leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0 \end{aligned}$$

The first term $(1/2) w^T w$ in Eq. (9) is the weights vector norm, y_i is the desired value, and C is referred to as the regularized constant, determining the tradeoff between the empirical error and the regularized term. ε is called the tube size of SVM and is equivalent to the approximation accuracy placed on the training data points. Here, the slack variables ξ and ξ^* are introduced. Using Lagrange multipliers and exploiting the optimality constraints, the decision function given by Eq. 9 takes the following explicit form:

$$\begin{aligned} L &= \frac{1}{2} w^T w \\ &+ C \sum_{i=1}^N (\xi_i + \xi_i^*) - \sum_{i=1}^N \alpha_i (\varepsilon + \xi_i - y_i + w^T x_i + b) \\ &- \sum_{i=1}^N \alpha_i^* (\varepsilon + \xi_i^* - y_i + w^T x_i + b) - \sum_{i=1}^N (\eta_i \xi_i + \eta_i^* \xi_i^*) \end{aligned} \quad (10)$$

with the constraints:

$$\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \alpha_i^*,$$

$$0 \leq \alpha_i^* \leq C, \quad i = 1, 2, \dots, n$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n$$

This can also be expressed in the form:

$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot (\varphi(x_i), \varphi(x)) + b \quad (11)$$

or more generally as:

$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b \quad (12)$$

where $K(x_i, x)$ is defined as the kernel function [31, 32]. The value of the kernel is equal to the inner product of two vectors, X_i and X_j , in the feature space $\varphi(x_i)$ and $\varphi(x_j)$, that is, $K(x_i, x) = \varphi(x_i) \times \varphi(x_i)$. Typical examples of the kernel function are:

Linear: $K(x, x_i) = x_i^T x_j$

Sigmoid: $K(x, x_i) = \tanh(\gamma x_i^T x_j + r)$

Polynomial: $K(x, x_i) = (\gamma x_i^T x_j + r)^d, \quad \gamma > 0$

Gaussian function:

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\gamma}\right) \quad \gamma > 0$$

$$\text{Exponential RBF: } K(x, x_i) = \exp\left(-\frac{\|x - x_i\|}{2\gamma}\right)$$

Here, γ , r , and d are kernel parameters; the kernel parameter need to be chosen carefully, as they implicitly define the structure of the high dimensional feature space $\varphi(x)$ and thus controls the complexity of the final solution [33].

4. Application of the Box-Jenkins Methodology

The available time series data was composed of 1,321 observations representing Al-Quds daily stock price index for Palestine. The series was transformed using natural logarithms to stabilize the time series. Figure 1 represents the original time series and indicates that the time series was non-stationary and involved a sharp decline in stock market indices at the end of 2008; this was a period of worldwide economic crisis, which influenced all global financial markets. The natural logarithms of the series are displayed in Fig. 2 and the first order differences of the natural logarithms are shown in Fig. 3. No clear seasonal fluctuations in the series are observed, and the seasonal effects, if any, are disregarded.

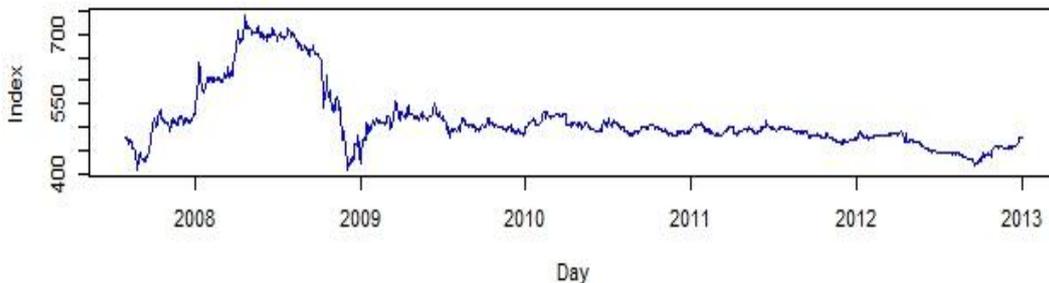


Figure 1. Al-Quds Index Daily Data

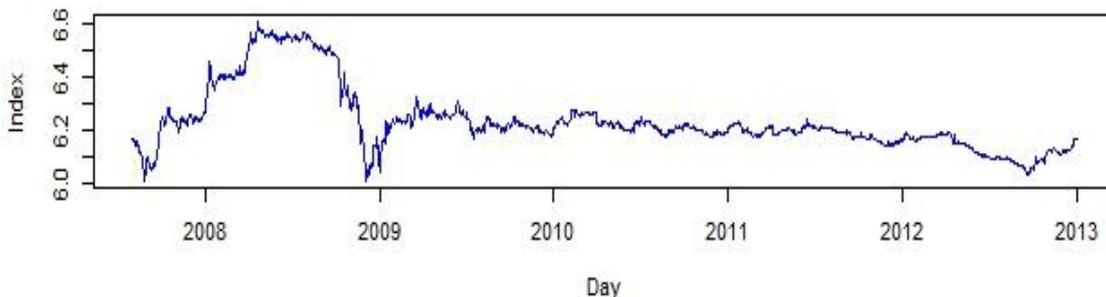


Figure 2. The Logarithms of Al-Quds Index Daily Data

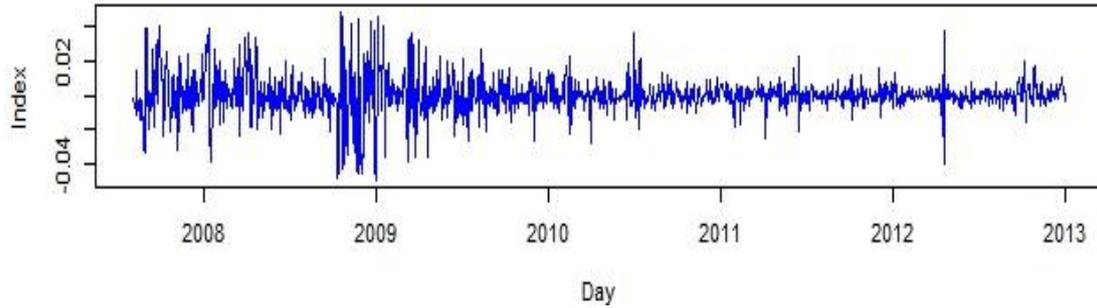


Figure 3. The 1st Differences of the Logarithms of Al-Quds Index

The sample autocorrelation and partial autocorrelation functions of the transformed Al-Quds index of the PSE time series, as shown in Fig. 4 and 5, indicated that the series had been stabilized, the transformed series was stationary, and some autocorrelations were significantly different from zero.

The stationarity of the transformed series was tested using Kwiatkowski, et al.[34] method, and the results indicated that the KPSS level was 0.1542, while the truncation lag parameter was 8, and the p-value was approximately 0.1. This indicates that the first difference of the natural logarithms of Al-Quds index of PSE series was stationary.

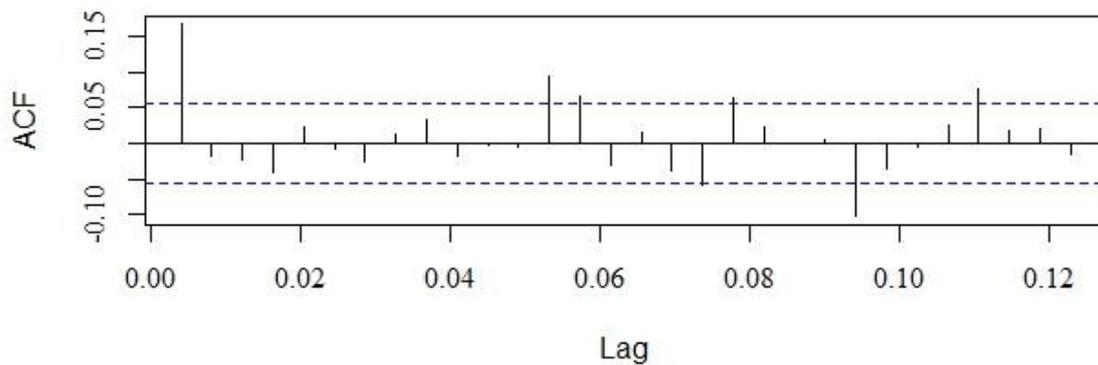


Figure 4. The Autocorrelation Function of the Differenced Logarithms of the Series

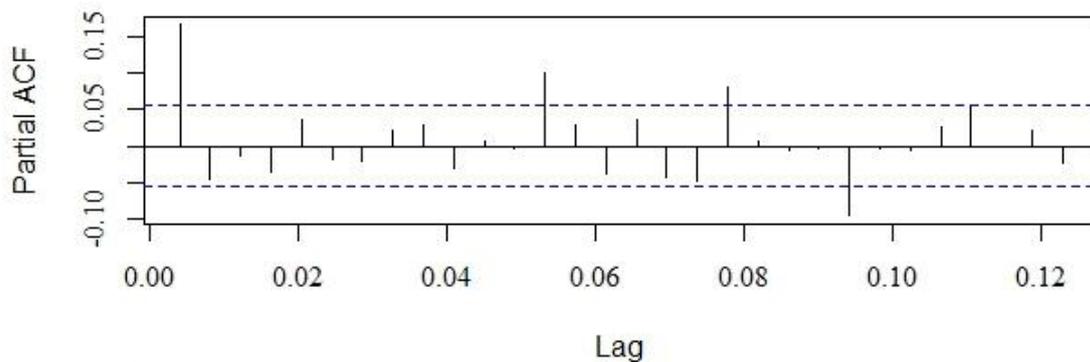


Figure 5. The Partial Autocorrelation Function of the Differenced Logarithms of the Series

To fit the Box-Jenkins ARIMA model for the transformed series, we identified the orders p and q of the ARIMA model while fixing d at 1. The correlogram of the transformed series given in Fig. 4 and 5 above enables us to identify the values of these parameters. In this correlogram, we noted a significant autocorrelation and partial autocorrelation at lag 1. There are several graphical tools to

facilitate identification of the ARMA orders. These include the corner method[35] and the extended autocorrelation (EACF) method[36, 37].

We applied the EACF method to the underlying differenced time series and compared the results of different estimates of p and q . The result of this comparison was that an appropriate model for the series could be ARIMA(0,1,1),

ARIMA(1,1,2), or ARIMA(2,1,2). Therefore, the parameters for the three models were estimated, and the best model that could predict future values for stock prices among the models was identified. The results of this analysis revealed that the best model is ARIMA (0, 1, 1), since this had the lowest Akaike information criterion (AIC), and Bayesian information criterion (BIC).

The parameters of the ARIMA(0,1,1) model, as the best one, were estimated and the following model was obtained:

$$\hat{Y} = 0.1769 \varepsilon_{t-1}$$

with AIC = -7717.59 and RMSE= 0.0282, where Y_t denotes the differenced natural logarithm of Al-Quds index of the PSE series. Note that the intercept was omitted from this model, since it was not significant and equalled zero.

Figure 6 displays three diagnostic tools for the fitted ARIMA(0,1,1) model. These are plots of the standardized residuals, the sample ACF of the residuals, and the p-values for the Ljung-Box test statistic for a whole range of values of K from 2 to 12. The horizontal dashed lines at 5% help determine the size of the p-values. These plots and the significance test of the coefficients suggest that the ARIMA(0,1,1) model fits the natural logarithms of Al-Quds index of the PSE time series adequately.

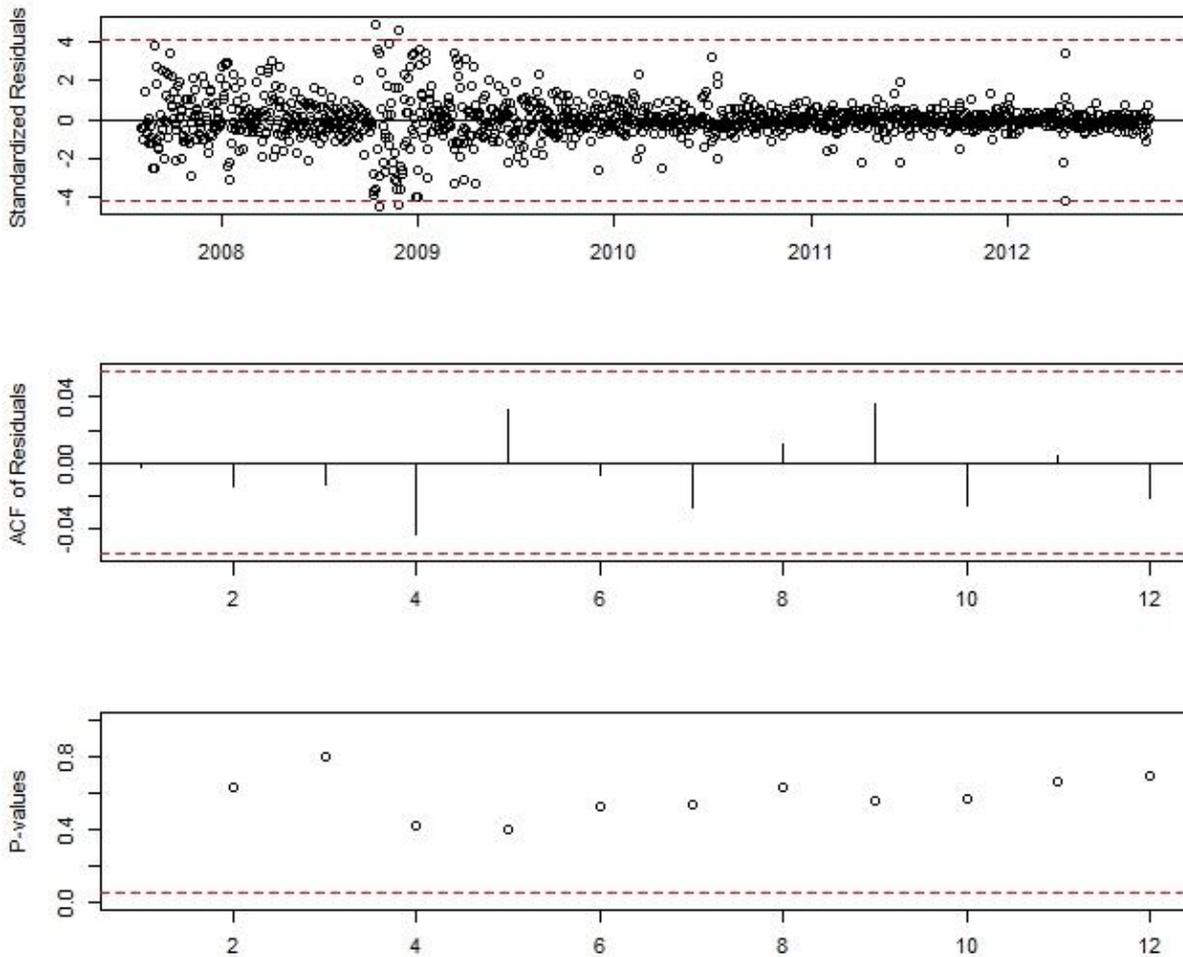


Figure 6. Diagnostic Plots for the Residuals of the ARIMA(0,1,1) Model

Using the model in Eq. 13 above to forecast two-month future values yields, the forecast values displayed in Fig. 7 were obtained. The first two values were close to the actual values, while the rest quickly settled to the mean of the series; the 95% forecasting confidence limits contained all of the actual values. This shows that ARIMA models may be suitable for forecasting a few, but not many, future values.

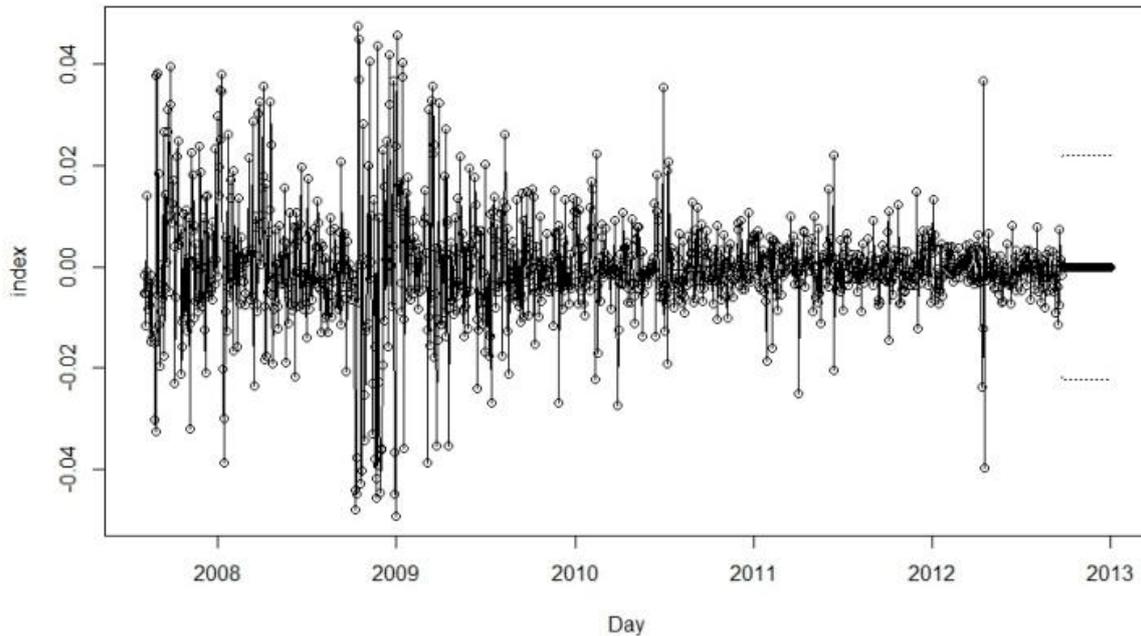


Figure 7. Actual, Forecast and Forecast Limits for Differenced Logarithms of the Series

5. Fitting the Artificial Neural Network Model to the Data

This section focuses on fitting the ANN model described in Section 2 into the time series data for Al-Quds daily stock price index for Palestine. The data had 1,321 points, as shown in Fig. 1 and described in section 4. The number of observations in the training set was the same as the number of observations used in fitting the ARIMA model. That is, 90% (1,255 observations) of the series was considered a training set, and 10% (66 observations—representing more than a two-month period) was used as a test set. We assumed a continuous learning rate throughout the training of the network. *R* statistical software was used for all computations.

The selection of hidden layers for the network is not straightforward. When the number of hidden layer units is too small, correlation of the output and input cannot be assessed properly, and errors increase. However, when the number of hidden layer units is sufficiently large, unrelated noise and the correlation of both input and output can be examined, as the error increases accordingly. Many methods have been developed to identify the number of hidden layer units, but there is no ideal solution to this problem[38]. Therefore, in our analysis, we started with one hidden layer and gradually increased the number to 15 layers; we then attempted to find the network with the least RMSE for the residuals.

Since the series may also contain seasonality effects, different numbers of seasonal lags were used as inputs. When the feed-forward back-propagation network for the stock price index data was applied with one unit in the hidden layer associated with different values of lags and different learning rates, 90 results produce 90 networks. The best network with the minimum RMSE of the residuals in the various runs with different numbers of seasonal lags was that with one unit in the hidden layer and five seasonal lags. The minimum RMSE of the natural logarithms of Al-Quds index of the PSE for the final network equalled 0.02990. Taking into consideration the independence of the learning rates, the number of lags considered and the number of hidden layers, the RMSE value did not change a great deal.

Using the above ANN model, we obtained the forecasting results for Al-Quds index of the PSE time series as shown in Fig. 8. Moreover, through this figure we can observe that the values of forecasting were almost identical to the actual values of the time series, although ANN does not require the time series to be stationary.

Figure 9 shows the residuals of the final network and indicates that they were very small, with the majority close to zero, and most falling in the interval between -0.05 and 0.05. We may conclude that the best network to forecast the logarithms of the Al-Quds index of the PSE time series is that which uses the back-propagation algorithm with 15 units in the hidden layer, five seasonal lags used as inputs, and a learning rate of 0.01.

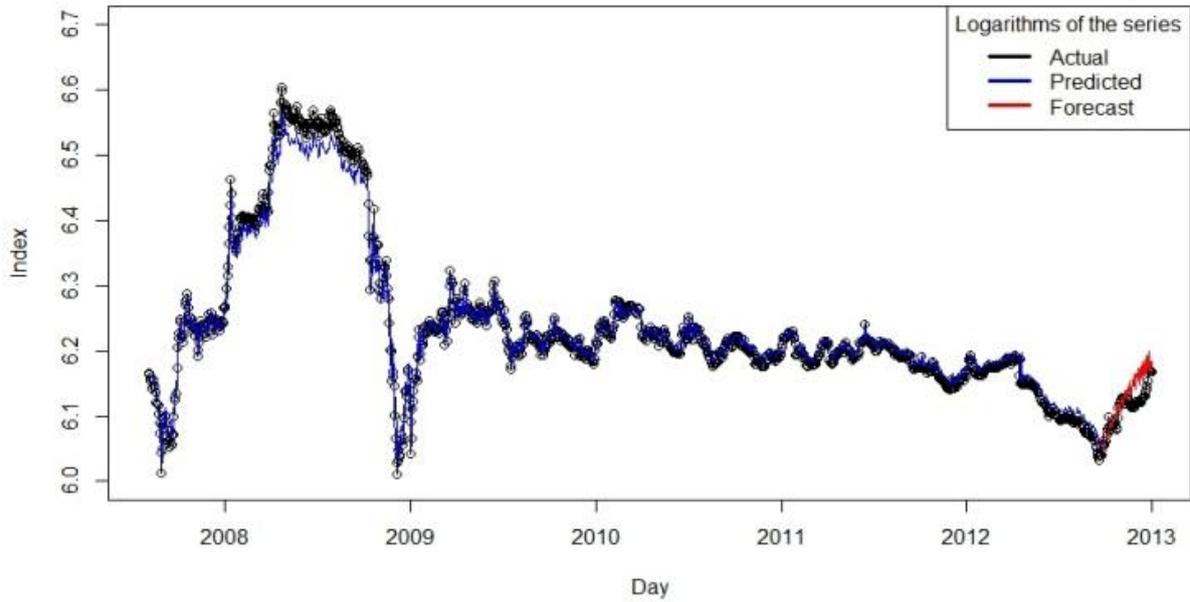


Figure 8. Observed and Fitted Values of the Logarithms of the Series Using ANN

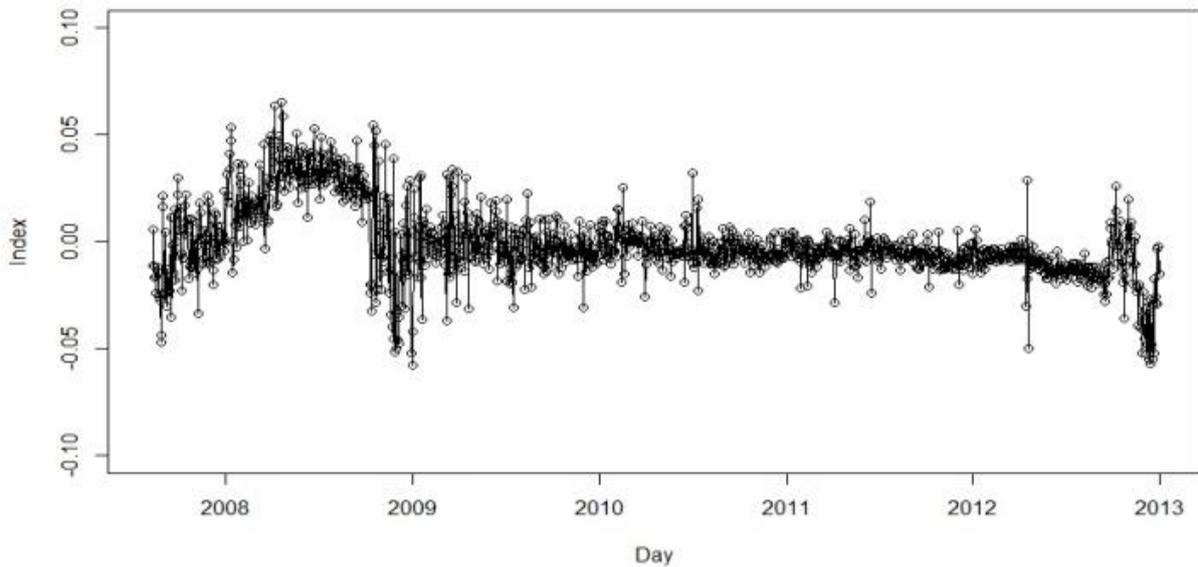


Figure 9. Residuals of the ANN model of the Logarithms of the Series

6. Fitting the SVM to Data

The time series consisted of 1,321 points of daily Al-Quds indices of the PSE, as shown in Fig. 1. The series was transferred to natural logarithms to stabilize its variance, as shown in Fig. 2. First, 90% of the points in the series were used for training and the rest for testing the SVM. Ten-fold cross-validation was also performed.

Now, given a time series $\{x_1, x_2, \dots, x_n\}$, to make forecasting about it using SVM, the time series needed to be transferred into an autocorrelated dataset. That is to say, if $\{x_t\}$ is the goal value of forecasting, the previous values

$\{x_{t-1}, x_{t-2}, \dots, x_{t-p}\}$ should be the correlated variables of the input. From this, we were able to map the autocorrelated input variables $\vec{x}_t = \{x_{t-1}, x_{t-2}, \dots, x_{t-p}\}$ to the goal variable, $y_t = \{x_t\}$. Here p is an embedding dimension. We considered the effect of the forecasting horizon and the embedding dimension on the performance of SVM. As to the choice of embedding dimension, this needed to be made in accordance with practical problems. Transferring the data in this way, we obtained the time series data suitable for SVM learning. The prediction performance was evaluated using RMSE.

Because we did not know the optimal embedding dimension p , we first had to determine this value. With the other conditions fixed, we used $p=\{2,3,4,5,6,7,8,9\}$ to carry out our preliminary experiments. From these experiments, we found that when $p=5$, RMSE was the lowest, so we concluded that five-days lagged daily indices was most suitable for forecasting the next-day's index. We divide the time series data into two parts. The first included 1,256 points (90% of the series), which were used for both training and validation—to train the SVM and to find its optimal parameters. The test set was composed of the remaining 66 data points (10% of the series) which was used to check the predictive power of SVM.

Since there is no structured method for selecting the free parameters of SVMs, the generalization error and number of support vectors with respect to C and ϵ were examined. The kernel parameters γ and C were selected based on the validation set. RMSE and the number of support vectors with respect to the free parameters were investigated. In this investigation, the Gaussian function was used as the kernel function of the SVMs. Our experiments showed that a width value of the Gaussian function of $\gamma=0.02$ produced the best possible results. Figures 10 shows the logarithms of the series together with its predicted values for the training set and the forecast values of the test set, with the values of RMSE indicated below each figure for different experimental values of γ . C and ϵ were arbitrarily set at 10 and 1,000, respectively. Table (1) shows the values of the

RMSEs at various experimental values of γ with C fixed at 100. The table shows that when $\gamma \in (0.00001, 0.02)$, the RMSE decreases as γ increases, while $\gamma \in (0.02, 10000)$, it increases as γ increases. This indicates that too small a value of $\gamma \in (0.00001, 0.02)$, or too large a value of $\gamma \in (0.02, 10000)$ can cause the SVM to under-fit. An appropriate value for γ would be approximately 0.02 for this time series. This is because the value of $\gamma=0.02$ produce the minimum RMSE and hence prove to be the best possible value for γ and provide the best possible forecasts. Only the results of γ are illustrated; the same approach can be applied to the other two parameters.

Table 1. Values of the RMSE for various experimental values of γ with C fixed at 100

γ -value	RMSE
0.0001	0.02458
0.0005	0.00971
0.001	0.00939
0.005	0.00778
0.01	0.00747
0.02	0.00703
0.05	0.00794
0.1	0.00859
0.5	0.00992
1.0	0.01087
5.0	0.04899
10.0	0.07574
100.0	0.13965

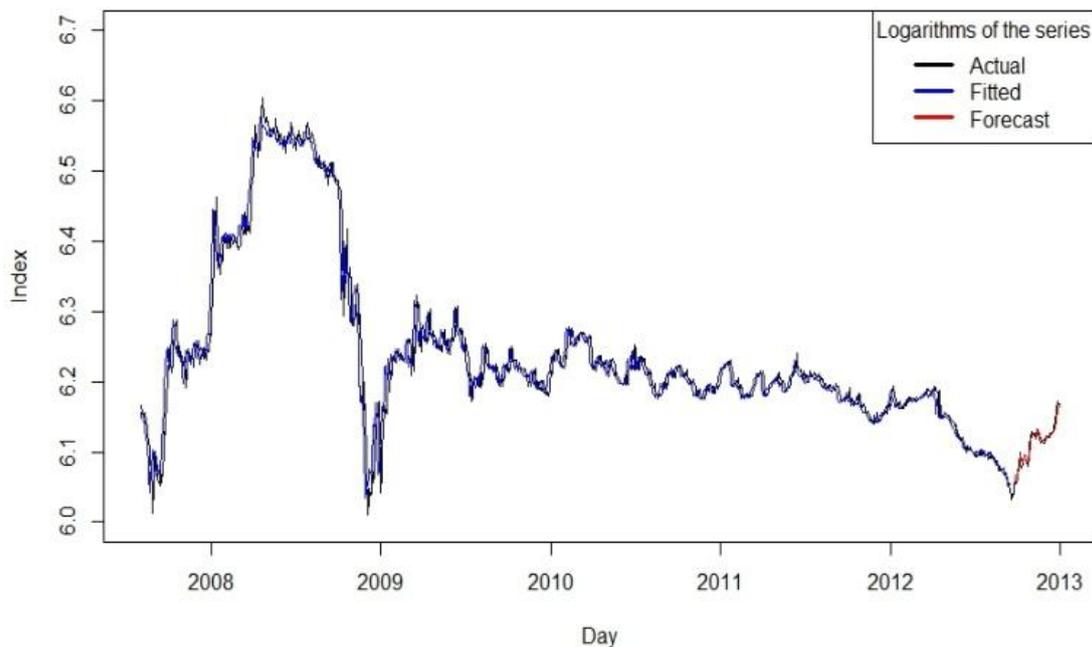


Figure 10. Observed and Fitted Values of the Logarithms of the Series Using SVM ($\gamma=0.02$)

Table 2. Observed and Forecasted Values of Daily Scores of Al-Quds Index for 66 Days at End of 2012 Using the Final SVM

No.	Observed	Forecast	No.	Observed	Forecast
1	423.28	425.67	34	460.36	458.66
2	427.50	427.41	35	457.99	459.70
3	429.17	430.22	36	457.17	458.59
4	427.46	432.05	37	456.79	457.81
5	426.14	431.73	38	453.75	457.36
6	428.68	430.93	39	452.56	455.36
7	432.90	432.07	40	451.79	453.99
8	436.81	434.74	41	453.38	453.09
9	445.76	437.78	42	452.84	453.74
10	438.86	443.90	43	451.40	453.56
11	436.56	442.00	44	451.52	452.66
12	435.82	440.16	45	453.80	452.46
13	438.61	439.16	46	456.34	453.77
14	440.01	440.47	47	455.08	455.67
15	442.50	441.53	48	456.12	455.51
16	442.09	443.34	49	455.64	456.12
17	441.34	443.71	50	455.50	456.06
18	440.85	443.45	51	456.06	455.98
19	439.07	443.09	52	455.22	456.26
20	436.52	441.96	53	457.97	455.86
21	437.31	440.13	54	457.92	457.37
22	444.39	439.99	55	459.20	457.82
23	451.85	443.98	56	458.56	458.75
24	454.05	449.59	57	459.38	458.65
25	457.11	452.77	58	462.41	459.16
26	456.16	455.73	59	464.13	461.13
27	458.84	456.19	60	466.97	462.81
28	458.00	458.02	61	470.88	465.06
29	457.20	458.08	62	475.94	468.19
30	456.95	457.65	63	476.94	472.32
31	454.48	457.34	64	478.81	474.24
32	456.47	455.79	65	477.31	476.03
33	459.76	456.48	66	477.59	475.69

Summing up the analysis above and after several rounds of testing, we identified $\gamma = 0.02$, $C = 100$, and $\varepsilon = 0.001$ as the best selections for our experiment, and used these parameters to train the model again. Following this, we were able to predict the test set. The final result was $RMSE = 0.00703$, with a value of $\gamma = 0.02$. Figure 10 gives a comparison of forecast values with actual values, while Table 2 shows actual observed and forecast values of the daily scores of Al-Quds index of the PSE for 66 points at the end 2012, using consequent one-step-ahead forecasting. The results shown in Figure 10 and Table 2 indicate that the tendencies of the predicted value curve are basically identical to those of the actual value curve, and the predicted values fit the actual values very well.

7. Conclusions

This article examined the application of SVM to financial forecasting. Forecasting financial time series, such as indices and stock prices, is a complex process, mainly

because financial time series are usually very noisy and involve ambiguous seasonal effects due to the influences of holidays, weekends, and irregular closure periods. They also involve other factors such as interest rate changes, announcements of macroeconomic news, and political events that affect forecast accuracy. In this study, we fit the ARIMA, ANN, and SVM models to Al-Quds index of the PSE time series data and used these models to forecast future observations (for 66 days). The results of applying the ARIMA, ANN, and SVM methods were compared through the RMSE results. The most important finding was that the minimum RMSE of the natural logarithms of Al-Quds index of the PSE time series using the SVM model equalled 0.00703, while for the ANN model it was 0.02990 and for the ARIMA model it was 0.0282. The last value was the only one computed from the differenced logarithms of the series. Finally, we can conclude from the above discussion that the results for SVM provided a more accurate and more efficient forecasting technique for such financial data than the ANN and ARIMA models did.

We can also conclude that SVMs provide an alternative, promising technique compared to time series forecasting using Box-Jenkins methodology and ANN. They offer important advantages over other methods, such as having a smaller number of free parameters and producing more accurate forecasts. Although there is little effect on the generalization error with respect to the free parameters of SVMs, we believe that there is still much room for improvement in SVMs with respect to forecasting financial time series. Future work should focus on this possibility.

REFERENCES

- [1] Trippi, R.R. & Turban, E; (1996). Neural networks in finance and investing: Using artificial intelligence to improve real-world performance. IRWIN. Chicago.
- [2] Palestine Stock Exchange. (2013). Al-Quds index of PSE. Retrieved from <http://www.pex.ps/marketwatch/English/CompanyMarketWatch.aspx>
- [3] Kuan, C. M. & White, H. (1994). Artificial neural networks: An econometric perspective. *Econometric Reviews*, 13,1-91.
- [4] Yao, J. & Tan, C.-L. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 34, 79-98.
- [5] Tkacz, G. (2001). Neural network forecasting of Canadian GDP growth. *International Journal of Forecasting*, 17, 57-69.
- [6] Zhang, G.P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Journal Neurocomputing*, 50,159-175.
- [7] Junoh, M. Z. (2004). Predicting GDP growth in Malaysia using knowledge-based economy indicators: A comparison between neural network and econometric approach. *Sunway College Journal*, 1,39-50.
- [8] Mohammadi, K. Eslami,H. R. & Dardashti, Sh. D. (2005). Comparison of regression, ARIMA and ANN models for reservoir inflow forecasting using snowmelt equivalent (a case study of Karaj), *J. Agric. Sci. Technol*, 7,17-30.
- [9] Rutka, G; (2008). Network traffic prediction using ARIMA and neural network models. *Electronics and Electrical Engineering*, 4(48), 47-52.
- [10] Vapnik VN. (1995). *The nature of statistical learning theory*. New York, Springer-Verlag.
- [11] Zhao, Y., Fang, R. Zhang, S. & Luo, S. (2006). Vague neural network controller and its applications. *Lecture Notes in Computer Science*, 4131, 801-810.
- [12] Chen, W-H. & Shih, J.Y. (2006). Comparison of support vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *International Journals Electronics Finance*, 1, 49-67.
- [13] Flake, G.W. & Lawrence, S. (2002). Efficient SVM regression training with SMO. *Machine Learning*, 46, 271-290.
- [14] Cao, L.J. & Tay, E.H. (2001). Support vector with adaptive parameters in financial time series forecasting. *IEEE Trans. Neural Network*, 14, 1506-1518.
- [15] Majhi, B., Rout, M., Majhi, R., Panda, G. & Fleming, P.J. (2012). New robust forecasting models for exchange rate prediction. *Expert Syst. Appl.*, 39(16), 12658-12670.
- [16] Kandananond, K. (2012). A comparison of various forecasting methods for autocorrelated time series. *International Journal of Engineering Business Management*, 4(1), 1-6.
- [17] Kim, K-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55, 307-319.
- [18] Tay, F.E.H. & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, 29, 309-317.
- [19] Rosenblatt, F. (1962). *Principles of neurodynamics*. Spartan, New York.
- [20] Rosenblatt, F. (1959). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
- [21] Cowan, J; (1967). *A mathematical theory of central nervous activity*. (Unpublished doctoral dissertation). University of London.
- [22] Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). *Learning internal representations by error propagation (Vol. 1)*. Cambridge: MA. MIT Press.
- [23] Zou, H.F. Xia, G.P., Yang, F.T. & Wang, H.Y. (2007). An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting. *Neurocomputing*, 70, 2913-2923.
- [24] Liu, H., Chen, C., Tian, H-Q & Li, Y-F. (2012). A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks. *Renewable Energy*, 48, 545-556.
- [25] Zhang, G.P., Patuwo, G.E. & Hu, M.Y. (2001). A simulation study of artificial neural network for non-linear time series forecasting. *Comput. Operat. Res.*, 28:381-396
- [26] Okasha, M. K. and Yaseen, A. (2013). The application of artificial neural networks in forecasting economic time series. *The International Journal of Statistics and Analysis (IJSA)*, 3(2), 123-143.
- [27] Tseng, F-M., Yu, H-C. & Tzeng, G-S. (2002). Combining neural network model with seasonal time series ARIMA model. *Technological Forecasting & Social Change*, 69, 71-87.
- [28] Cao, D-Z., Pang, S.L. & Bai, Y-H. (2005). Forecasting exchange rates using support vector Machines. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005*.
- [29] Huang, H. & Tian, Y. (2013). A novel visual modeling system for time series forecast: application to the domain of hydrology. *Journal of Hydroinformatics*, 15(1), 21-37.
- [30] Zhang, L., Zhou, W-D., Yang, J-W. & Li, F-Z. (2013). Iterated time series prediction with multiple support vector regression models. *Journal Neurocomputing*, 99, 411-422.
- [31] Mellit, A.; Pavan, A.; Benghamem, M. (2013). Least squares

- support vector machine for short-term prediction of meteorological time series. *Theoretical & Applied Climatology*, 111(1), 297-307.
- [32] Wang, J., Li, L., Niu, D. & Tan, Z. (2012). An annual load forecasting model based on support vector regression with differential evolution algorithm. *Applied Energy*, 94, 65-70.
- [33] Mustafa, Z. & Yusof, Y. (2012). A hybridization of enhanced artificial bee colony-least squares support vector machines for price forecasting. *Journal of Computer Science*, 8(10), 1680-1690.
- [34] Kwiatkowski, D., Phillips, P. C. B., Schmidt, P. & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3), 159-178.
- [35] Becuinj, M., Gourieroux, S., & Monfort, A. (1980). Identification of a mixed autoregressive-moving average process: The corner method In time series. Edited by O.D. Anderson. Amsterdam: North-Holland, 423-436.
- [36] Tsay, R. S. & Tiao, G. (1984). Consistent estimates of autoregressive parameters and extended sample autocorrelation function for stationary and nonstationary ARMA Models. *Journal of the American Statistical Association*, 79, 385, 84-96.
- [37] Tsay, R. S. & Tiao, G; (1985). Use of canonical analysis in time series model identification. *Biometrika*, 72, 299-315.
- [38] Kermanshahi, B. & Iwamiya, H. (2002). Up to year 2020 load forecasting using neural net. *Electrical Power & Energy Systems*, 24, 789-797.