Comparison between ARIMA Models and Artificial Neural Networks in Forecasting Al-Quds indices of Palestine Stock Exchange Market

Mahmoud K Okasha
Al-Azhar University - Gaza

COMPARISON BETWEEN ARIMA MODELS AND ARTIFICIAL NEURAL NETWORKS IN FORECASTING AL-QUDS INDICES OF PALESTINE STOCK EXCHANGE MARKET

Mahmoud K. Okasha,

Department of Applied Statistics
Al-Azhar University – Gaza, Palestine
m.okasha@palnet.com

Assem A. Yaseen,
Lecturer,
The University of Palestine, Gaza; Palestine
Email: assem-yassen@hotmail.com

Abstract

The accuracy of forecasts obtained by fitting time series models to time series data, such as stock price indices, received a great attention by many statisticians and economists. The Box-Jenkins "Autoregressive Integrated Moving Average" (ARIMA) models gave good forecasts for future observations in many cases but they were not always so accurate in many other cases. This is because the forecasts converge to the mean of the series after three or four forecast values. Artificial Neural Networks (ANN) can be a good alternative to the Box-Jenkins approach, particularly in the case of nonlinear data and for forecasting many future values.

In this study the forecasting capabilities of both Artificial Neural Networks and ARIMA models are studied and compared. The comparison has been conducted through studying their efficiency in modeling and forecasting the daily data of Al-Quds Index of the Palestinian Stock Exchange Index for 3 years. The result was that the ANN provides forecasts so close to the actual ones when the logarithmic transformation of the original time series is used. However, the best traditional Box-Jenkins model for the logarithmic transformation the data was ARIMA(0,1,1), but the results were not as satisfactory.

Key words: *ARIMA models, nonlinear time series, Artificial Neural Networks, Forecasting, backpropagation.*

1. Introduction

1.1 Rationale

Several forecasting methods have recently been developed to forecast future observations of time series especially of economic nature, such as inflation, stock market returns and stock indices. The difficulty of forecasting economic time series is the complex interactions between factors affecting the market and the unknown random processes such as unprecedented news or changes in other factors (Clement and Hendry, 2002). Well-known models based on the autoregressive integrated moving average (ARIMA) time series models were commonly used.

The parameters of the ARIMA models are usually estimated by the method of least squares or the maximum likelihood estimation method. However, ordinary least squares method requires imposing, strict assumptions on the model specifications during the estimation of the parameters to achieve meaningful results (Box *et al*., 1994) and thus it is inefficient to use with complex or nonlinear models. Since most economic relations are usually non-linear in either parameter and can be nonstationary, nonlinear least squares method is the most commonly used method to estimate the parameters in non-linear economic models.

The Artificial Neural Networks (ANN) is an alternative method that can be used for forecasting in such non-linear time series and can overcome the problems of non-linearity and nonstationarity. The use of ANN has been rapidly increasing because of their ability to form complex non-linear systems for forecasting based on sample data. Applications of ANN received a great attention in recent years because of their enormous storage capacity and their capabilities of learning and prediction. In particular, ANN has been applied in economic forecasting to predict stock markets indicators in line with economic growth in various countries in the past few years (Trippi and Turban; 1996).

1.2 Research problem

When fitting ARIMA models on economic and financial data that are either non-linear or nonstationary time series, the results of forecasting are expected to be inaccurate and do not give an appropriate picture of what could be the future values. This is because of the fact that the forecasts converge to the mean of the series after three or four forecasted values. An alternative method often used for forecasting and produce good forecasts when the data is non-linear or nonstationary is the Artificial Neural Networks (ANN). The problem of this study is to apply both ARIMA model and ANN method for forecasting time series using Al-Quds Index of the Palestine Stock Exchange Index data and to evaluate the performance of both methods.

1.3 Research Methodology

In this study we apply two different methods of forecasting future values for the same time series of the daily values of Al-Quds Index in Palestine Stock Exchange market and we conduct a comparison between the results of the two methods in order to determine the best method to use in similar situations. To achieve this we perform the following steps:

- Finding the best order of ARIMA model to forecast future values of Al-Quds Index data.

- Finding the most suitable ANN model to predict future values from the time series.
- Comparing the two methods based on Forecast and the Akaike Information Criterion, (AIC), the Bayesian Information Criterion, (BIC), and the minimum Root Mean Squares Error (RMSE).
- Finally, we give the relevant recommendations based on the results of the above comparisons.

1.4 Background

Several studies have been conducted on the comparison between ARIMA models and ANN in forecasting using time series data. Most of those comparisons were data based and many of them used economic data. Kuan and White (1994), discussed the possibility of using ANN in economic variables and the usability of traditional time series models and emphasized the similarities between the two methods. In a similar study by Maasoumi et.al (1994) the methods were applied on a group of 14 different time series in macroeconomics and found that the ANN method performed better than other methods in their forecasting abilities. Kohzadi *et.al*. (1995), compared ANN and ARIMA, in forecasting of Egypt's cereal future and found out that standard error of forecasts of neural network are less than those of ARIMA models. Swanson and White (1997) used ANN approach in forecasting macroeconomic variables. They compared different linear and non-linear models by using a large sample size data. They found that the performance of multivariate linear models is marginally better than other univariate models. Hansen, *et.al*. (1999), found that ANN outperform ARIMA forecasting models in six different time series originally published in Box and Jenkins (1968) and more recently by McDonald and Xu (1994). The developments in ARIMA model improved forecasting performance over standard ordinary least squares estimation by 8% to13%. In contrast, ANN achieve dramatic improvements of 10% to 40%. Tkacz and Hu (1999), examined whether artificial ANN can be used in modeling the increase in production based on monetary and financial variables. The results indicated that ANN forecasting performances were better than those of the linear models. Tkacz (2001) compared the forecasting abilities of both time series models (ARIMA and exponential smoothing) and linear models on the one hand and ANN models on the other hand using the Canadian GDP data and monetary and financial variables.

Zhang (2003) has used hybrid approach in a combination of ARIMA and ANN models. The results obtained were very encouraging. Junoh (2004) forecasted the GDP of Malaysian economy using information based on economic indicators. In this study, the author compared ANN and econometric approaches and showed that ANN had better results in GDP forecasting. Mohammadi, et.al; (2005) applied different methods of forecasting spring inflow to the Amir Kabir reservoir in the Karaj river watershed. Three different methods, artificial neural network, ARIMA time series and regression analysis between some hydroclimatological data and inflow were used to forecast the spring inflow. The performances of the models were compared and the ANN proved to be an effective tool for reservoir inflow forecasting in the Amir Kabir reservoir using snowmelt equivalent data. Rutka (2008) conducted a study to forecast the network traffic using ARIMA Model and ANN. The author concluded that ARIMA models are easier to use for training and forecasting, but the prediction results showed that they are not very accurate. In contrary, the ANN models are more complex in training and simulation but they give much better results compared with ARIMA models.

2. The Box-Jenkins Approach to fitting ARIMA Models and Forecasting:

The Box Jenkins approach in fitting time series models is through identifying the best model for the analysis which is the most comprehensive, most effective and has the fewest parameters. If we assume that this series has mixed parts of Autoregressive and moving average, we obtain a quite general time series model. Suppose that the series {rt}, t =... -1, 0, 1 ... is an equally spaced weakly stationary or covariance stationary, time series, then a famous linear model for time series analysis in the time domain belongs to an autoregressive moving average and can be expressed in the form:

$$r_t = \phi_1 r_{t-1} + \phi_2 r_{t-2} + \ldots + \phi_p r_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \qquad (2.1)$$

We say that {rt} is a mixed autoregressive moving average process of orders p and q , respectively ; and referred simply as ARMA(p ,q). For the general ARMA(p ,q) model, we state that εt, is independent of rt−1, rt−2, rt−3,…, a stationary solution to equation (2.1) exists if and only if all the roots of the AR characteristic equation ϕ, (x) = 0 are outside the unit circle .

For inevitability we have to assume that the roots of $\theta(x) = 0$ are outside the unit circle. Where {εt} is a sequence of uncorrelated variables, also referred to as a white noise process, and $(\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$ are unknown constants or parameters. The Box-Jenkins model can then be expressed as:

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^P) r_t = (1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q) \varepsilon_t \qquad (2.2)$$

where B is the backshift operator, that is BXt = Xt−1.

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_P B^P$$
$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q$$

In general, the ARMA (p,q) is a combination of an AR(p), and a MA(q) and can be written as:

$$r_t = \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \qquad (2.3)$$

In practice, it is often impossible to directly apply the ARMA model on a given time series because it is nonstationary but it needs to be transformed. It is often the case, that the time series of differences is a stationary in spite of the nonstationarity of the basic process. That leads to the application of the (ARIMA) model.

ARIMA is one of the most popular models of nonstationary time series analysis used for forecasting. This model does not take the form of a single equation or simultaneous equation models, but it focuses on the analysis of the probabilistic or stochastic properties of the time series itself. In contrast to the regression models, the ARIMA models allows values of Yt to be explained from the past, or lagged values of Y itself and stochastic error terms. The time series {rt} which is classified as an integrated autoregressive moving average of the dth difference and expressed as:

$$w_t = \nabla^d r_t = (1-B)^d r_t \tag{2.4}$$

is a stationary ARMA process, where $\nabla = B - 1$ is the difference operator. If {wt} follows the ARMA (p, q) model, we say that { rt } is an ARIMA (p, d, q) process. For practical purposes, we usually take d = 1 or 2 at most. We can write model (2.4) as :

$$\phi(B) w_t = \theta_0 + \theta(B)\,\varepsilon_t \tag{2.5}$$

where $\phi(B)$ is a stationary autoregressive operator, $\theta(B)$ is a stationary moving average operator, and {εt } is white noise and θ0 is a constant usually referred to as a trend parameter. The model is called "integrated" since rt can be thought of as the summation (integration) of the stationary series wt. The previously mentioned models are built on assumptions that the time series involved are weakly stationary.

But, as it is well known that many economic time series are nonstationary, they require integrating. If the integration of a time series is of order d i.e. I(d), after differencing it d times we obtain an I(0) series. That is to say if we need to difference a time series d times to make it stationary and then apply the ARMA (p, q) process, we say that the original time series is ARIMA(p,d,q) process, that is, it is an autoregressive integrated moving average time series.

The first step in estimating the ARIMA model is to determine (p,d,q) where p denotes the number of autoregressive terms, q denotes the number of moving average terms and d denotes the number of times a series must be differenced to induce stationarity. (For further details on ARIMA models see Box, *et al.*, 1994; Hamilton, 1994; Brockwell and Davis,1996; and Cryer and Chan, 2008).

The final model is usually used to generate predictions about the future values of and then calculate the forecast errors as developments of new values watch from the time series and control of these errors in the so-called Control Charts and developed for the acceptance by a specific error if exceeded prediction error re-examined in the form and returned the cycle again by selecting another candidate model. (Barry, 2002).

3. Artificial Neural Networks

3.1 Introduction

The human brain as has the ability to learn from the past, led many researchers to the establishment of the cognitive sciences, known as artificial intelligence and building a network, known as Artificial Neural Network (ANN), which mimic the properties of brain neurons, and is not a biological fact. Unlike the brain implementation, the ANN separate operations, which are possible in light of the high-capacity electronic computers to perform complex operations quickly. McCulloch and Pitis (1943) developed, the first computing machines that mimic the biological nervous system, and can perform the functions of the logic of learning as shown in figure (3.1). The consequence of this network was a set of logic functions that were used to transfer information from one nerve cell to another. This led ultimately to the development of a binary model possibility. According to this model, the unit can be nerve switch either on or off depending on whether the function is activated or not.

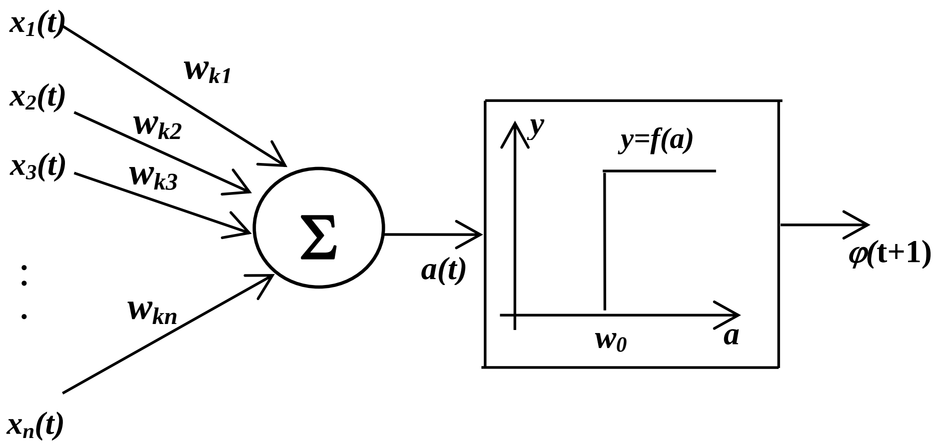

Figure (3.1): Artificial neuron of McCuUoch and Pitts.

By the well-known advanced models that have been developed for the learning process model “perception", Rosenblatt (1959 and 1962) developed a single feedforward network. The output obtained from this single layer is the weighted sum of different inputs. A major development in ANN was by Cowan (1967), where the introduction of new functions such as activation of the smooth sigmoid function, etc., which have the capacity to deal with nonlinear functions more effectively than the perception model as in fig (3.2) .

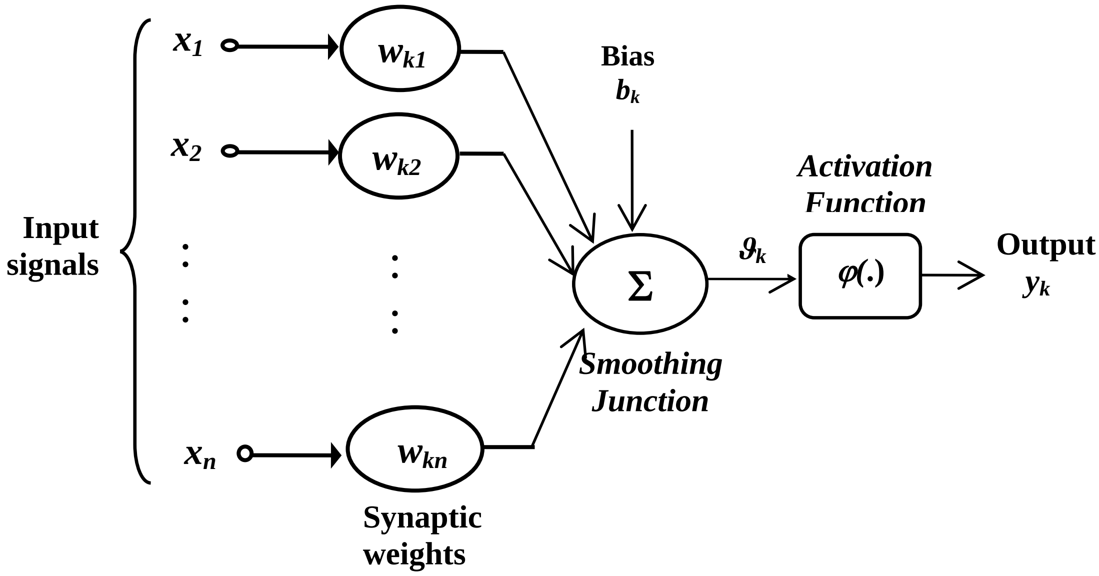

Figure (3.2): Nonlinear model of neuron.

A new method of learning has been proposed by Werbos (1974), which is backpropagation, but has been published in few reports at the time where estimation errors back to the hidden layer. This method and has been used successfully in many applications such as playing table, and handwriting recognition, filtration processes, control system, economic and financial forecasting, etc (Kabundi, 2002). The process of estimating the parameters is the most important step to build this model. Power of NN models depends to a large extent on the way their layer connection weights are adjusted over time. The weights adjustment process is known in NN methodology as *training* of the network. The objective of the training process is that the weights are updated in the way to facilitate learning of the patterns inherent to the data.

3.2 The Backpropagation Learning Algorithm

The backpropagation procedure uses the gradient descent learning technique for multilayer feedforward ANN. It is also called generalized delta rule as stated by Rumelhart *et al*. (1986). Figure (3.3) shows that training network consists of two main parts, the input and output parts. Initial weights are selected randomly between –1 and +1. The network outputs depend on the input units, hidden units, weights of the network, and the activation function. The difference between the computed and the actual output (target) is known as network error. Backpropagation method takes the network error and propagates it backward into the network. Errors are used at each neuron to update weights. This process is repeated until the total network error becomes the smallest.

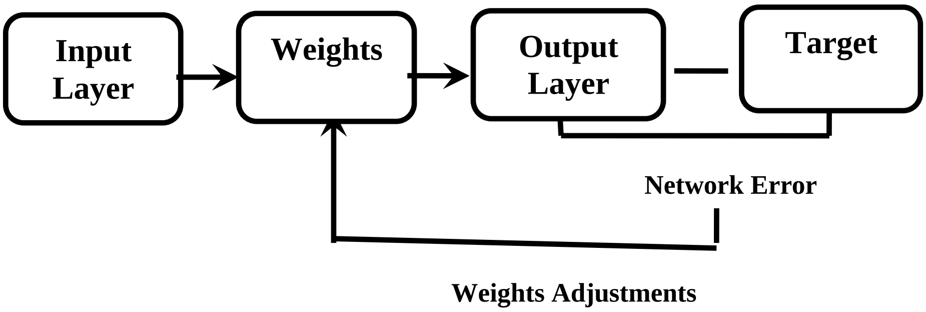

Figure 3.3: Feedforward backpropagation technique

ANN method uses the error or cost function to measure the difference between the target value and the output value. Weights of the network are frequently adjusted in such a way that the error or objective function becomes as small as possible. The target function can be written as:

$$E_t = T_t - Y_t \tag{3.1}$$

where T_t is the actual or targeted output value of the t^{th} iteration, and Y_t is the computed output of the t^{th} iteration. The most common cost functions used are the mean absolute error (MAE) and the mean squared error (MSE). The mean squared error (MSE) is expressed as:

$$MSE = \frac{1}{N}\sum_{t=1}^{N} (\mathrm{T}_t - Y_t)^2 \tag{3.2}$$

The network is most commonly trained using the MSE error function. But in NN modeling this differs slightly from equation (3.2) where the network objective function is:

$$E = \frac{1}{2}\sum_{t=1}^{N} (\mathrm{T}_t - Y_t)^2 \tag{3.3}$$

The constant ½ is used to facilitate the computation of the derivative for the error function, which is essential in the estimation of the parameters. In the derivative of (3.3) the constant ½ disappears, while if (3.2) is used, 1/N could not be reduced and we would end up with the factor 2/N; hence making it hard to determine the network parameters.

3.3 Backpropagation for multilayer feedforward

The backpropagation algorithm is the approach which is commonly used to work with network weights for a multilayered feedforward ANN. This algorithm is conclusive in NN modeling as it improves the learning process of the NN models. Application of the backpropagation algorithm becomes possible for the ANN to learn from the past experience in a way similar to the human brain. If E is the value of the cost function, then the rate of change in E with respect to the weights θ is given by :

$$\nabla E(\theta) = \frac{\partial E}{\partial \theta_t} \tag{3.4}$$

where θ_t is the vector of all weights of the network at t^th^ iteration. When applying the backpropagation rule, knowledge is accumulated through a learning process. The network weights are determined by:

$$\theta_{t+1} = \theta_t + \Delta(\theta)_t \tag{3.5}$$

where: θ_t are network weights of the t^{th} iteration, θ_{t+1} are parameters of $(t+1)^{th}$ iteration , and $\Delta(\theta)_t$ is the learning process. The NN learning experience entails update of the $\Delta(\theta)_t$ in order to decrease the error made at each period so that NN learning method is similar to the learning method of the biological nervous systems.

The function $\Delta(\theta)_t$ can be written as:

$$\Delta(\theta)_t = -a\nabla E(\theta) \tag{3.6}$$

where a is a positive constant called the learning rate. Return to the error function (3.3), where

$$Y_t = f\left(\sum_{j=1}^{m} \sum_{i=1}^{n} X_i w_{ji}\right) \quad \text{or simply} \quad Y_t = f(X_t, \theta_t) \tag{3.7}$$

From (3. 7) equation (3.4) can be written as:

$$\nabla E(\theta) = \frac{\partial E}{\partial \theta_t} = \frac{\partial E}{\partial w_{ji}} \tag{3.8}$$

By using a sigmoid function as the activation function in the hidden layer, and a linear activation function in the output layer, we will have:

$$f(u) = \frac{1}{1+e^{-u}} \tag{3.9}$$

where,

$$u = \sum_{j=1}^{m} \sum_{i=1}^{n} X_i w_{ji} \tag{3.10}$$

Using the chain rule, the gradient $\dfrac{\partial E}{\partial w_{ji}}$ can be written as:

$$\begin{aligned} \frac{\partial E}{\partial w_{Ji}} &= \frac{\partial E}{\partial f(u)} \frac{\partial f(u)}{\partial (u)} \frac{\partial (u)}{\partial w_{Ji}} \\ &= -f(u)(1-f_t(u))(T_t - f_t) \end{aligned} \tag{3.11}$$

From equation (3.10), we get

$$\frac{\partial u}{\partial w_{ji}} = X_i \tag{3.12}$$

Analysis of the residuals involves determining the value of

$$\frac{\partial E}{\partial f(u)}\ \frac{\partial f(u)}{\partial u}$$
.

Since E is the objective function and represented by the formula (3.3), we have:

$$\frac{\partial E}{\partial f(u)} = \frac{\partial \frac{1}{2}\left(T_t - f_t(u)\right)^2}{\partial f\ (u)}$$

$$= \frac{1}{2} \times 2\left(T_t - f_t(u)\right) \frac{\partial \frac{1}{2}\left(T_t - f_t(u)\right)}{\partial f\ (u)}$$

$$= -\left(T_t - f_t(u)\right) \qquad (3.13)$$

Similarly, the partial derivative of (3.9) with respect to u is

$$\frac{\partial f(u)}{\partial u} = -f_t\ (u)\,(1_t - f_t(u)) \qquad (3.14)$$

Substituting (3.12), (3.13), and (3.14) into (3.11), we have:

$$\frac{\partial E}{\partial w_{Ji}} = \frac{\partial E}{\partial f(u)}\ \frac{\partial f(u)}{\partial (u)}\ \frac{\partial (u)}{\partial w_{Ji}}$$

$$= -f_t(u)(1 - f_t(u))\,(T_t - f_t\ (u))\,X_i \qquad (3.15)$$

Equation (3.11) becomes

$$\nabla \mathrm{E}(\theta) = \frac{\partial E}{\partial \theta_t} = \frac{\partial E}{\partial w_{Ji}} =$$

$$= -X_i\, f_t(u)(1 - f_t(u))\,(T_t - f_t\ (u)) \qquad (3.16)$$

Substituting (3.16) into (3.8), and thereafter into (3.7), we get the number of iterations and $\Delta(\theta)_t$ of the learning process as follows:

$$\theta_{t+1} = \theta_t + a\,X_i\ f_t(u)\,(1 - f_t(u))\,(T_t - f_t(u)) \qquad (3.17)$$

Or

$$\theta_{t+1} = \theta_t + a\,X_i\ f_t(X_t, \theta_t)\,(1 - f_t(X_t, \theta_t))\,(T_t - f_t(X_t, \theta_t))$$

The NN learning experience updates the $\Delta(\theta)_t$ in order to decrease the error made at each period so that the NN learning method becomes similar to the learning method of the biological nervous systems.

From (3.17) we conclude that the backpropagation method is based on three factors: the learning rate a, the distance between the actual output and predicted output $(T_t - f_t\ (X_t, \theta_t))$, and the activation function $f_t\ (X_t, \theta_t)$.

The learning rate controls the size of change in weights in each step. If it is too small, the ideal point of convergence may be small. But in the case if the learning rate is too large, the algorithm might not converge at all. The learning rate should fall in the range $0 \leq a \leq 1$. One of the reasons for the success of the backpropagation procedure is the use of a nonlinear differentiable activation function. This algorithm is similar to a large extent to the style gradient descent, which is used heavily in the field of modeling biological systems and dynamics.

4. Data and Results

4.1 The data

The data used in this study is a time series that represent the daily scores of Al-Quds index of Palestine Stock Exchange (PSE) and published in the Palestine Stock Exchange . The number of observations in the series is 1194, representing daily scores in the period from the first of August 2007 until the end of May 2012. The market works only five days a week excluding national and religious holidays.

PSE has been established in 1995 and transferred into a public shareholding company in 2007. The PSE market is subject to governmental control and supervision of the Palestinian capital market. The PSE market helps investors through awareness, training and information provision to make investment decisions based on sound information, through different activities.

Al-Quds Index is the main indicator used in the market that gives a general idea about the direction of changes in stock prices in the market. It is a measure that helps the investor to recognize the pulse of the market and determine the direction of supply and demand, and the overall level of rise and decline in the prices of companies traded in the PSE market. Al-Quds index is calculated using the following formula:

$$\text{Al} - \text{Quds index} = \frac{\text{sum}\left(\text{number of shares subscribed} \times \text{the trading price}\right) \times 100}{\text{sum}\left(\text{number of shares subscribed} \times \text{primary price per share}\right)}$$

4.2 ARIMA and ANN Results

Descriptive statistics for the daily scores of Al-Quds index of PSE have been estimated for the purpose of illustration. The mean of the time series is estimated at 521.96 , the median of the time series is estimated at 499.9 and the standard deviation is estimated at 66.989. The original time series data has been transformed using the natural logarithmic transformation in order to reduce the effects of outliers on the analysis and to stabilize the time series. Therefore, all the analysis below were conducted on the natural logarithms of the time series of the Stock price index of Palestine (LOGPAL). Figure (4.1) below represents the time series that we analyzed and it indicates that the time series is nonstationary. This series varies randomly over time and there is no global trend or seasonal note. We noted here the sharp decline in the stock market at the end of the year 2008 as it was during the period of the world economic crisis which affected all global financial markets.

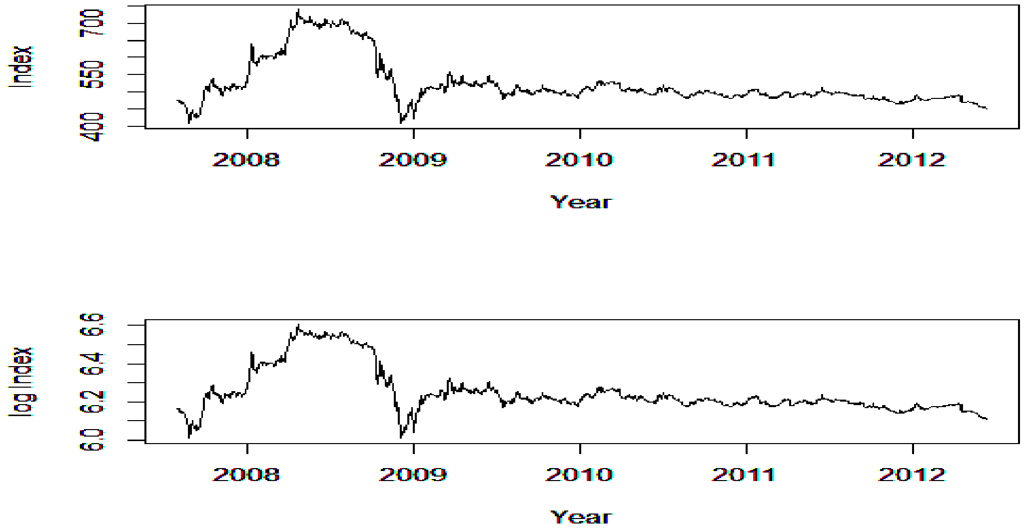

Figure 4.1: The logarithmic transformation of the Al-Quds Index time series.

The autocorrelation function of Al-Quds index of PSE time series refers to all values shown are "significantly far from zero", and the only pattern is a linear decrease with increasing lag the sample Partial Autocorrelation Function is also indeterminate, and cut-off after the second lag (figure 4.2). This means that we are dealing with a typical correlogram of a nonstationary time series. We use the first difference of the natural logarithm of the original time series in order to achieve a stationary series. Figure 4.3 below illustrates the first difference of the natural logarithm of the original Al-Quds index of PSE time series.

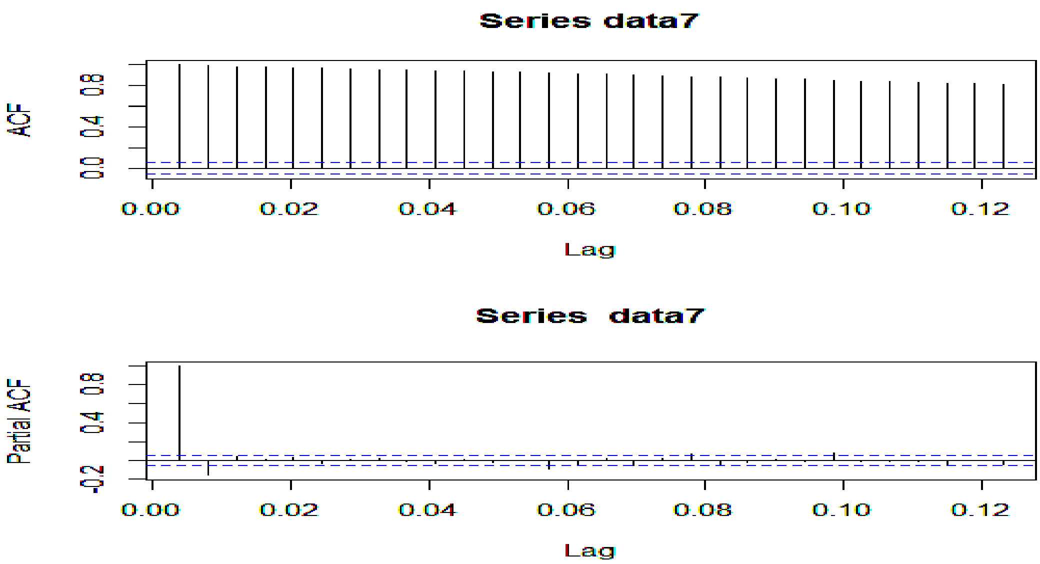

Figure 4.2: The correlogram of LOGPAL

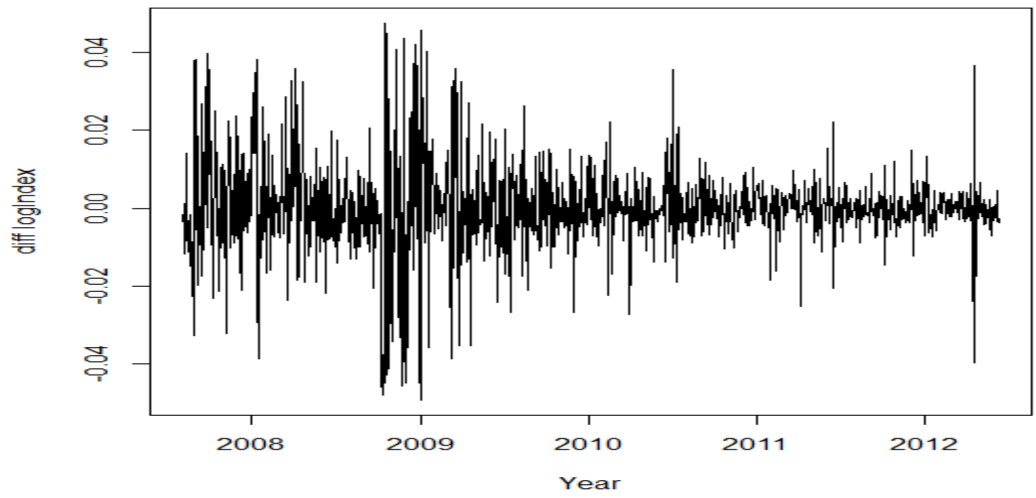

Figure 4.3: Graphical display of the 1st differences of the natural logarithm of Al-Quds index of PSE.

By looking at the correlogram of the first difference of natural logarithm series in figure (4.4) we can say it became stationary. Also we can test the stationarity through the unit root test, which has recently become popular in econometrics, being quite efficient and easy to be interpreted. There are several tests that allowed to get the unit root . One of the most well known is the Augmented Dickey-Fuller (ADF) test. That is based on the following equation:

$$\Delta y_t = \beta_{1+}\beta_2 + (\rho - 1)y_{t-1} + \sum_{i=1}^{m} \alpha_i \Delta y_{t-i} + \varepsilon_t \qquad (4.1)$$

where ε_t is a pure white noise error term and where:

$$\Delta y_{t-1} = y_{t-1} - y_{t-2},\ \Delta y_{t-2} = y_{t-2} - y_{t-3}, \text{ etc..}$$

The number of lagged difference terms to include is often determined empirically, where the idea is to include enough terms so that (4.1) is serially uncorrelated. Usually it is used the Akaike Information Criterion, AIC, or the Schwarz or Bayesian Information Criterion, SBC, in order to have so many lags as need for$\{\varepsilon_t\}$~$WN(0,\sigma^2)$. The ADF test follows the same asymptotic distribution as Dickey-Fuller (DF) statistic, (the τ (tau) statistic) and the null hypothesis is $\rho = 1$, that is, if there is a unit root – then the time series is nonstationary. The alternative hypothesis is that ρ is less than 1, that is, the time series is stationary. The case of being bigger than 1 is out of possible, because in that case the original time series will be explosive.

	Augmented Dickey-Fuller (ADF) test	
	Test for unit root in level	Test for unit root in 1st difference
Nul Hypothesis	LOGPAL has unit root	Diff(LOGPAL) has unit root
Drift	p-value is greater than printed p-value	p-value is smaller than printed p-value
Drift & Trend	p-value is greater than printed p-value	p-value is smaller than printed p-value
None	p-value is greater than printed p-value	p-value is smaller than printed p-value
Result	LOGPAL time series nonstationary	Diff(LOGPAL) time series stationary

Table 4.1: The results for the Unit Root Test

Once again, in the time series we don't reject, in levels, the null hypothesis, that is, there is a unit root – the time series are nonstationary, although we reject this hypothesis in first difference model.
In this way, after differencing the LOGPAL, obtain stationary time series.

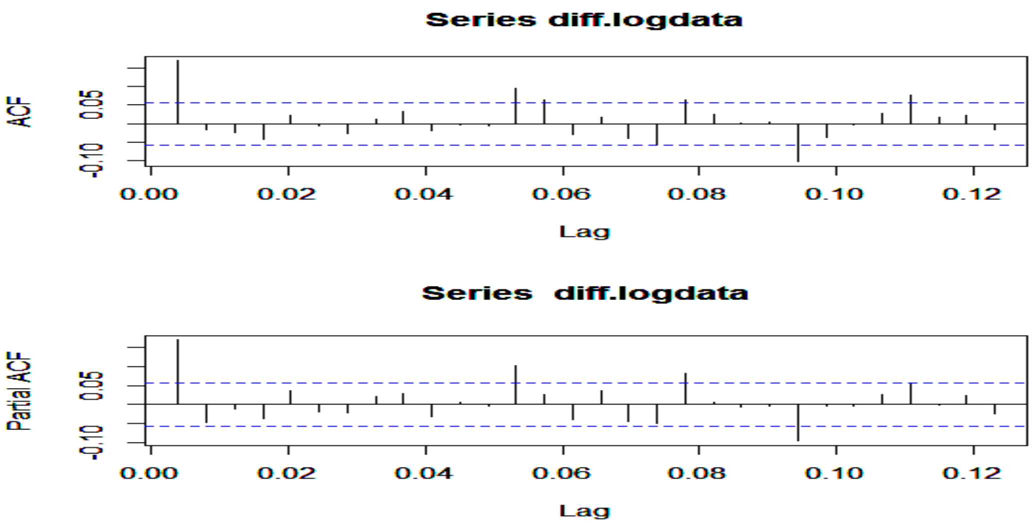

Figure 4.4: The correlogram of the natural logarithm of Al-Quds index of PSE time series (LOGPAL)

4.3 Fitting ARIMA model to the data

When we extend the model by allowing the AR polynomial to have one characteristic root, the model then becomes the ARIMA model. As it was seen, the time series are nonstationary and this fact implies the necessity to use the first-difference form data. Now, the time series is integrated of order 1, which means that we will have d=1 in the ARIMA(p,d,q) model. Next, we need to determine the order of autoregressive (p) and moving average (q) parameters that are necessary to give an effective model of the process. The correlograms given in figure (4.4) enables us to estimate these parameters. The decision in less typical cases requires not only experience but also a good deal of experimentation with alternative models.

4.3.1 Fitting ARIMA(p,1,q) for the Stock price index

For the correlogram of the differenced series of LOGPAL, presented in Figure (4.4), it is possible to note a significant autocorrelation (ACF) and partial autocorrelation (PAC) at lag 1. But, for a mixed ARMA model, its theoretical ACF and PACF have infinitely many nonzero values, making it difficult to identify mixed models from the sample ACF and PACF.

There are several graphical tools to facilitate identifying the ARMA orders. Those include the corner method (Becuinj et al., 1980), the extended autocorrelation (EACF) method (Tsay and Tiao, 1984), and the smallest canonical correlation method (Tsay and Tiao, 1985), among others. We applied the (EACF) method for the underlying differenced time series and the results of different estimates of p and q are given in table (4.2) below.

Table 4.2: The Theoretical Extended ACF (EACF)

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	0	0	0	0	0	0	0	0	0	0	0	x	x
1	x	x	0	0	0	0	0	0	0	0	0	0	x	x
2	x	x	0	0	0	0	0	0	0	0	0	0	x	x
3	x	x	x	x	0	0	0	0	0	0	0	0	0	x
4	x	x	x	x	0	0	0	0	0	0	0	0	0	0
5	x	x	x	x	x	0	0	0	0	0	0	0	x	0
6	x	x	x	0	x	x	0	0	0	0	0	0	x	0
7	x	x	x	0	x	x	0	0	0	0	0	0	x	0

From the previous table we can observe that an appropriate models for the series may be ARIM(0,1,1) , ARIM(1,1,2), ARIM(2,1,2). Therefore we will estimate parameters for the three models, and diagnose the best model that may predict future values for the stock prices among these models. The results for this analysis showed that the best model is the ARIMA (0, 1, 1) or IMA(1,1) as can be seen table (4.3), since this model has the lowest Akaike Information Criterion, (AIC), and the Bayesian Information Criterion, (BIC).

Table 4.3: The value for (AIC, AICc) for different ARIMA models

ARIMA	AIC	AICc	BIC
ARIMA(0,1,1)	-7154.48	-7154.46	-7139.28
ARIMA(2,1,2)	-7149.42	-7149.35	-7119.01
ARIMA(1,1,2)	-7150.77	-7150.72	-7125.43

Using R software, we estimate the parameters of ARIMA(1,1) model as the best model and we get the following model:

$$\hat{Y} = 0.2203\varepsilon_{t-1} \qquad (4.2)$$

with AIC = -7154.48 ; RMSE = 1.14e^{-02}, where Y_t denotes the differenced natural logarithm of Al-Quds index of PSE series.

4.4 Diagnostic testing for the model

Figure 4.5 shows three of our diagnostic tools in one display. These are a plot of the standardized residuals, the sample ACF of the residuals, and p-values for the Ljung-Box test statistic "LB.test" for a whole range of values of K from 2 to 12. The horizontal dashed line at 5% helps judge the size of the p-values. In these plots, the suggested model looks to fit the modeling time series very well. Therefore the estimated ARIMA(0,1,1) model seems to be capturing the dependence structure of the Diff(LOGPAL) time series quite well.

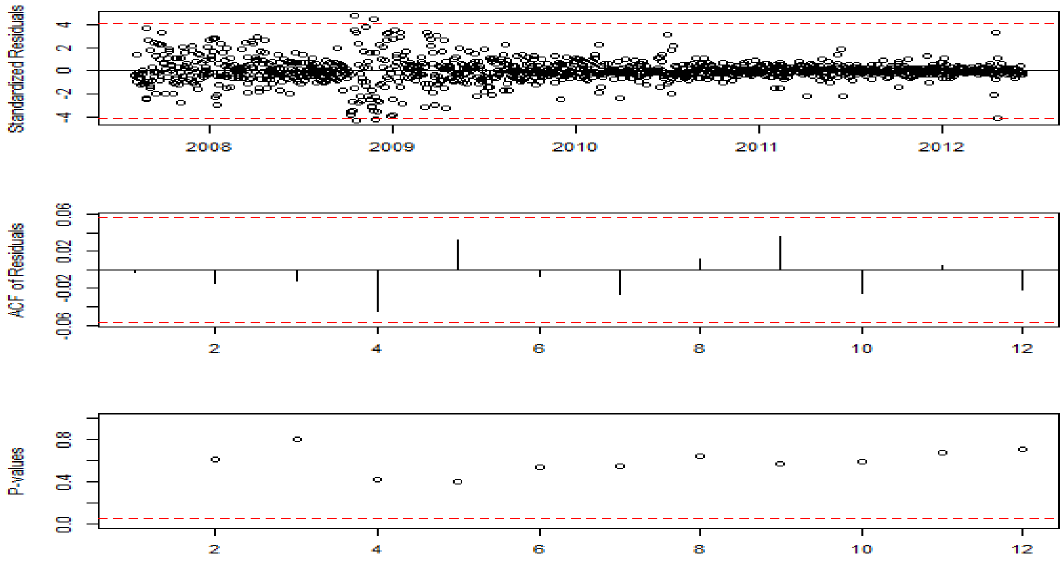

Figure 4.5: Diagnostic Display for the ARIMA(0,1,1) Model of Diff(LOGPAL)

Now the ARIMA(0,1,1) model has been fitted to the series of stock price index. Investigating the results of this fit, resulted that all coefficients are significant and the diagnostic model suggests that this model is suitable , Let Yt denote the Diff(LOGPAL), then our tentatively identified ARIMA model is:

$$\hat{Y} = 0.1792\varepsilon_{t-1} \qquad (4.3)$$

Where Se = 0.0290 ; t = 6.1793 .

Using this model for forecasting we get the following results:

From the results in the table(4.4) and figure(4.6) below which illustrate the prediction of the last ten values of the time series and compare them with last ten actual values with 95% forecast limits we note that the first three values only close to the actual values and then the rest of the forecast shall revert to the mean of the series, Since the model does not contain a lot of autocorrelation , the forecasts, quickly settle down to the mean of the series ,the forecast limits contain all of the actual values.

Table 4.4: Forecasting results of ARIMA model for LOGPAL time series

year	actual	forecast
2012.98	2.68480	2.72123
2012.99	2.68466	2.68466
2012.100	2.68345	2.68356
2012.101	2.68460	2.68356
2012.102	2.68571	2.68356
2012.103	2.69127	2.68356
2012.104	2.69245	2.68356
2012.105	2.68935	2.68356
2012.106	2.69182	2.68356
2012.107	2.68851	2.68356

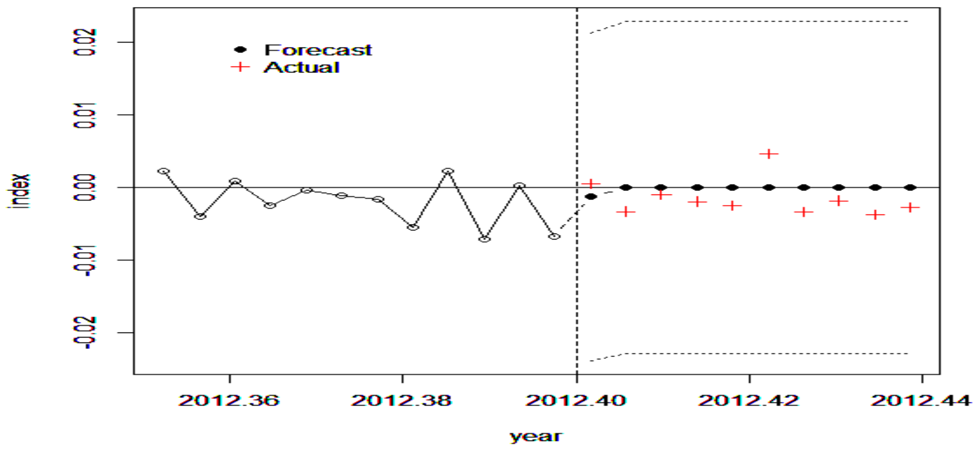

Figure 4.6: Actual , forecast and forecast limits for D(LOGPAL index)

5. Fitting the Artificial Neural Network model for the time series:

For fitting ANN model for the time series, as described previously we may use Matlab software. In ANN modeling, through the Matlab functions forecasts time series with minimum Root Mean Squares Error (RMSE) which is used as stopping criteria in the network. It should be used here because we look for differences between target and the output series. It is determined by the value of the square root of Mean Square Error (MSE). Smaller values of RMSE indicate higher accuracy in forecasting. The measure of dispersion between the target and the output is the MSE is given by:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(T_t - Y_t)^2{}_t \qquad (5.1)$$

where T_t is the actual or targeted output value of the t^{th} iteration and , Y_t is the computed output of the same t^{th} iteration.

Applying ANN, the percentage of observations for training, which must have the same number of observations, 1194, as we have in ARIMA for training is determined, so we have increased in a series of 131 observations. Thus, we have an input string of 1325 observations, 90% for training, and 10% for comparison in the prediction. The learning rate assumed a continuous learning rate throughout the training. The performance of the algorithm is very sensitive to the proper setting for the learning rate. If you have chosen too high learning rate, the algorithm may oscillate and becomes unstable. If the selected learning rate is very small, the algorithm would take a long time to converge.

Selection of hidden layers need to experience more than the mathematical technique. When the number of hidden layers units is small, the correlation of the output and input cannot be studied well and the errors increase. Moreover, when the number of hidden layers units is more than adequate, even an unrelated noise can be studied as well as the correlation of both input and output, and the error increase accordingly. There are some methods to get the number of hidden layer units, however, there is no general solution for this problem (Kermanhahi, et al, 2002). Therefore, we decided to start with one hidden layer and gradually increasing the number of hidden layers to a fifteen layers. Attention should be paid to different number of hidden layers, and different numbers of lags.

We started with one unit in the hidden layer considering one and two lags, with different learning rates. Considering the subsequent results, ninety times produced ninety networks. Figure (5.1), shows that residuals are very small, with the majority of them are close to zero, and falling in the interval [-0.05, 0.05]. The lowest MinRMSE in each run is showed in table (5.1) below, under the number of units in the hidden layer. It indicates the lowest MinRMSE of all ninety runs.

Table 5.1: Lowest MinRMSE results of ANN model for LOGPAL time series.

Maximum number of Lags	1			2		
Learning Rate	0. 1	0.05	0.01	0. 1	0.05	0.01
min RMSE with 1 unit in hidden layer	0.0048999	0.0048377	0.0048393	0.0048898	0.0048144	0.0048184
min RMSE with 2 unit in hidden layer	0.0048430	0.0049081	0.0048445	0.0048520	0.0048541	0.0048050
min RMSE with 3	0.0048391	0.0048999	0.0048328	0.0047974	0.0048387	0.0049706

unit in hidden layer						
min RMSE with 4 unit in hidden layer	0.0048631	0.0049598	0.0048252	0.0048296	0.0048543	0.0048309
min RMSE with 5 unit in hidden layer	0.0049630	0.0048591	0.0050220	0.0048318	0.0048600	0.0050568
min RMSE with 6 unit in hidden layer	0.0048317	0.0049145	0.0049927	0.0049382	0.0048344	0.0047996
min RMSE with 7 unit in hidden layer	0.0048379	0.0049323	0.0049090	0.0049891	0.0048439	0.0050574
min RMSE with 8 unit in hidden layer	0.0048436	0.0049507	0.0048816	0.0048598	0.0048510	0.0047872
min RMSE with 9 unit in hidden layer	0.0049133	0.0048771	0.0049465	0.0049808	0.0049155	0.0048474
min RMSE with 10 unit in hidden layer	0.0048672	0.0049598	0.0049819	0.0049341	0.0048357	0.0048683
min RMSE with 11 unit in hidden layer	0.0050351	0.0049652	0.0049798	0.0049436	0.0048076	0.0050568
min RMSE with 12 unit in hidden layer	0.0050365	0.0050227	0.0050327	0.0049468	0.0049208	0.0048968
min RMSE with 13 unit in hidden layer	0.0050344	0.0050183	0.0050445	0.0049024	0.0048902	0.0049127
min RMSE with 14 unit in hidden layer	0.0048610	0.0049866	0.0049257	0.0048862	0.0049989	0.0049317
min RMSE with 15 unit in hidden layer	0.0051196	0.0051333	0.0050858	0.0048203	0.0048622	0.0047682

From table (5.1) we can see that the minimum RMSE of the natural logarithms of Al-Quds index of PSE equals 0.0047682 and the maximum RMSE of the series equals 0.0051333. We may then conclude from the table that the values of the RMSE obtained are very similar. Taking into consideration the independence of the learning rates, the number of lags considered and the number of hidden layers, the value of RMSE does not change more considerably. We have the best model to forecast the logarithms of Al-Quds index of PSE time series is the model that use backpropagation algorithm with fifteen units in the hidden layer, two lags and learning rate equals 0.01.

Using the above ANN model we obtain the forecasting results for Al-Quds index of PSE time series as shown in table (3.4) below. Moreover, through this table we can see that the values of forecasting are almost close to the actual values for time series; although ANN does not require that the time series is stationary.

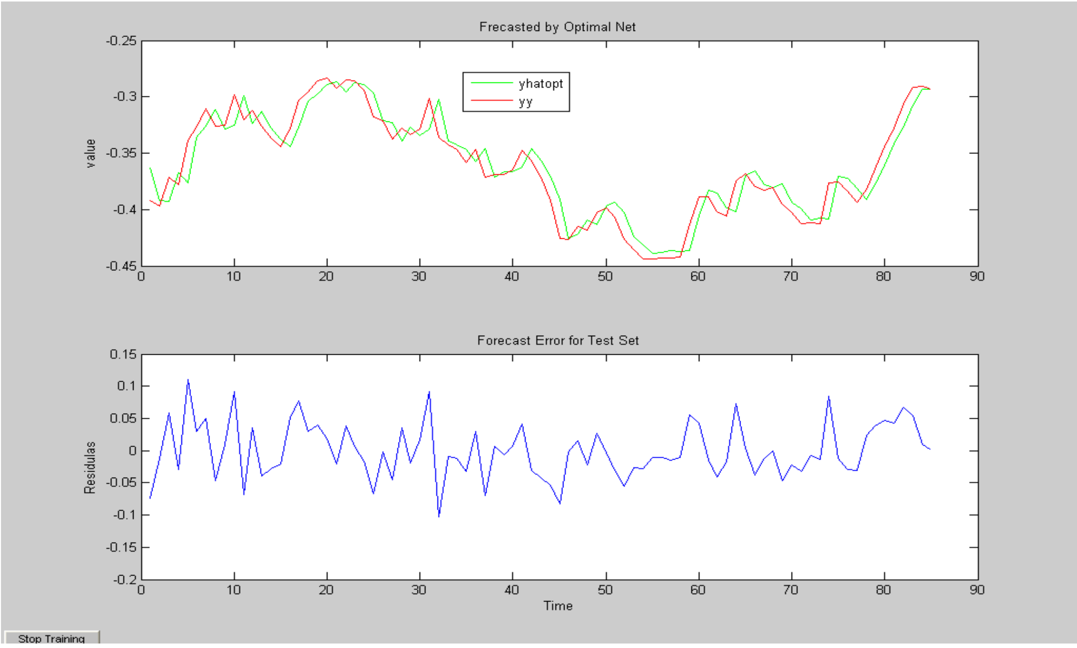

Figure 5.1: Forecasting outputs of LOGPAL (ANN –Matlab)

In spite of getting the lowest Min RMSE when considering the model with fifteen units in the hidden layer, two lags and the learning rate equal to 0.01.

From the previous results we have the best model to predict the time series LOGPAL using the following parameters: (LOGPAL, 2, 15, 90, 10, 0.01).

Table 5.2: Forecasting results of ANN model for LOGPAL time series

year	actual	ANN forecast
2012.98	2.68480	2.7013
2012.99	2.68466	2.7013
2012.100	2.68345	2.70098
2012.101	2.68460	2.70096
2012.102	2.68571	2.70096
2012.103	2.69127	2.70064
2012.104	2.69245	2.70065
2012.105	2.68935	2.70065
2012.106	2.69182	2.70033
2012.107	2.68851	2.70035

As well as through the table (5.2) above we can observe that the values of forecasting is almost applicable to the actual values for LOGPAL time series. Although ANN does not require that the time series is stationary.

6. Comparison between ARIMA and ANN results

The results of applying both ARIMA and ANN methods are compared through the results and figures (4.6), and (5.1), as well as tables (4.1), (4.2) and (3.4). The most important result that can be observed is the minimum RMSE of the natural logarithms of Al-Quds index of PSE time series using the ANN model equals 0.0047682 while that of the ARIMA model was 0.01145 and that is equivalent to 240% times RMSE of the ANN model.

Finally, we can conclude from the above dissuasion that the results of ANN model are much better than the ARIMA model results and more efficient.

7. Conclusion

In this study we conduct a comparison of forecast accuracy between the results of ARIMA model and the ANN model. Forecasting financial time series, such as indices and stock prices is a complex process, for several reasons. The most important reason is the fact that financial time series are usually very noisy. There is a large amount of random unpredictable noise day after day. Among other reasons is the existence of different factors, such as interest rates changes, announcement of macroeconomic news and political events that affects the forecasting accuracy. In this study we fitted ANN model for Al-Quds index of PSE time series data and used this model to forecast future observations. For the purpose of comparison we also used the Box and Jenkins approach to attempt to forecast the same points. We then conducted a comparison of forecast accuracy between the traditional ARIMA model of Box and Jenkins and the ANN model.

From all the discussion in this study the following conclusions can be drawn:

- The ARIMA(0,1,1) model is the best fit for Al-Quds Index among other Box-Jenkins models. This result is supported by the ACF,AIC,BIC, and RMSE. However, the use of ARIMA model in forecasting economic and financial data does not give accurate results, for more than 3 future values.
- The ANN model that use backpropagation algorithm with fifteen units in the hidden layer, two lags and learning rate equals 0.01, is the best fit for Al-Quds Index forecasting,
- ANN model can be effectively used in forecasting stock price index for several points. It can also perform very well in economic and financial data, and thus it makes a great contribution as an efficient tool for forecasting in financial markets.

References

[1] Becuinj, M., Gourieroucx, S., and Monfort, A. (1980); *"Identification of a mixed autoregressive-moving average process: The corner method In Time Series"*; edited by O. D. Anderson; 423–436. Amsterdam: North-Holland.

[2] Box, G E P, Jenkins G.M. (1968), "*Some recent advances in forecasting and control",* Applied Statistics, 17(2), 91-109

[3] Box, G.E.P. Jenkins, G.M. and Reinsel, G.C. (1994); *"Time Series Analysis: Forecasting and Control"*; 3rd edition; Prentice Hall: Englewood Cliffs, New Jersey.

[4] Clement, M.P and Hendry, D.F. (2002); *"A companion to economic forecasting"*; Black well publishers ltd ; Massachusetts; USA.

[5] Cowan, J. (1967); *"A Mathematical Theory of Central Nervous Activity"*; unpublished Ph.D. dissertation; University of London.

[6] Cryer J. and Chan K.-S. (2008). "*Time Series Analysis, with Applications in R",* 2nd ed. New York - Berlin: Springer

[7] Hansen, J.V. McDonald, J.B. and Nelson, R.D. (1999); *" Time Series Prediction With Genetic - Algorithm Designed Neural Networks : An Empirical Comparison With Modern Statistical Models"*; Computational Intelligence, 15:171-184.

[8] 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.

[9] Kabundi, A. (2002); *"Macroeconomic Forecasting : A Comparison Between Artificial Neural Networks and Eeconometric Models"*; MA. thesis, Rand Afrikan University .

[10] Kohzadi, N.M.S., Boyd, I. K., Kermanshahi, B. S. and Scuse, D. (1995);" *Neural Networks for Forecasting: An Introduction*"; *Canadian Journal of Agricultural Economics*, 43, 463-474.

[11] Kuan,C.M. and White, H. (1994); *"Artifical Neural Networks: An Econometric Perspective"*; Econometric Reviews; 13:1-91; Marcel Dekker; New York.

[12] Maasoumi, E., Khontazad, A. and Abaye, A. (1994); *"Artifical Neural Networks for Some Macroeconomic Series: A First Report"*; Econometric Reviews, 13: 105-122.

[13] Mc Culloch, W.S. and Pitis, W. (1943); *"A logical calculus of ideas immanent in nervous activity"*. Bulletin of Mathematical Biophysics, 5, 115-133.

[14] McDonald, J. B. and Xu, Y. (1994); "*Some forecasting applications of partially adaptive estimators of ARIMA models*". *Economics Letters* 45, 155-160.

[15] Mohammadi, K. Eslami, H.R. and 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.

[16] Rosenblatt, F. (1962); *"Principles of Neurodynamics",* Spartan, New York.

[17] Rosenblatt, F. (1959); *"The perceptron: a probabilistic model for information storage and organization in the brain"* ; Psychological Review; 65:386-408.

[18] Rumelhart, D.E., Hinton, G.E. and Williams, R.J; (1986); *"Learning Internal Representations by Error Propagation"*; Cambridge: MA:MIT. Press; 1: 318-362.

[19] Rutka, G. (2008); *"Network Traffic Prediction using ARIMA and Neural Networks Models"*; Electronics and Electrical Engineering; Kaunas: Technological; No 4(48):47-52.

[20] Swanson, N. R and White, H. (1997); *"A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks"*; Review of Economics and Statistics, 79: 540-550.

[21] Tkacz, G. (2001); *"Neural Network Forecasting of Canadian GDP Growth"*; International Journal of Forecasting, 17: 57-69.

[22] Tkacz, G. and Hu, S. (1999); *"Forecasting GDP Growth Using Artifical Neural Networks"*; Working Paper / Bank of Canada, 3:1-24.

[23] Trippi, R.R and Turban, E. (1996); *"Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance"*; IRWIN; Chicago.

[24] Tsay, R. S. and 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.

[25] Tsay, R. and Tiao, G. (1985); *"Use of canonical analysis in time series model identification"*; Biometrika; 72; 299–315.

[26] Werbos, P. (1974); *"Beyond Regression: New Tools For Predicting and Analysis in the Behavioral Sciences"*; Ph.D. thesis; Cambridge, MA: Harvard University Committee on Applied Mathematics.

[27] Zhang, G.P. (2003); *"Time Series Forecasting Using A Hybrid ARIMA and Neural Network Model"*; Neurocomputing, 50:159-175.