The Application of Artificial Neural Networks In Forecasting Economic Time Series

Mahmoud K. Okasha,
Department of Applied Statistics
Al-Azhar University – Gaza, Palestine
Email: m.okasha@palnet.com

Assem A. Yassen,
The University of Palestine, Gaza; Palestine

Abstract

The Box-Jenkins “Autoregressive Integrated Moving Average” (ARIMA) models have been the traditional and most widely used approach for forecasting. These models gave good forecasts for future observations in many cases but they were not always successful in many other cases. This is because the forecasts converge to the mean of the series after three or four forecast values. The accuracy of forecasts obtained by fitting time series models to economic data, such as stock price indices, received a great attention by many economists. In the present study, we applied neural networks (ANN) methods on forecasting the daily data of Al-Quds Index of the Palestinian Stock Exchange Market. ANN proved to be useful as an alternative method to Box-Jenkins approach, particularly in the case of nonlinear data and for forecasting many future values. The forecasting capabilities of ANN are studied and compared with traditional ARIMA models through fitting both ANN and ARIMA models to forecast the logarithms of the daily data of Al-Quds Index in 3 years, and the results of ANN proved to be more efficient than those of the ARIMA models.

Key words: ARIMA models, nonlinear time series, Artificial Neural Networks, Forecasting, backpropagation, Al-Quds Index.
1: Introduction
1.1 Rationale
Forecasting future observations based on historical data, received specialists’ attention in many fields. Several techniques have been developed to address this issue to predict future behavior of a particular phenomenon. Forecasting is one of the main statistical techniques used in economic planning. Economic processes are composed of complex operations, such as inflation, stock market returns and stock indices. They are usually difficult to predict in a timely manner. One source of difficulty 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 autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), and integrated ARMA time series models were commonly used. This is because the resulting models are easy to understand and to interpret.

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 and Jenkins; 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 produces 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 Historical 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 Maaoumi 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 (1970) 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.

As we have seen through the above literature review, there are many studies conducted comparison between ANN and other traditional methods, especially ARIMA in terms of their performances.

2: Artificial Neural Networks

2.1 Introduction

The most important feature of the human brain is its ability to learn from the past, according to a complex system of sending and receiving electrical pulses between neurons. This fact has prompted many researchers and led to the establishment of the cognitive sciences, known as artificial intelligence and building the 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 fig (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. It determines the threshold to activate the system. If the input is greater than the threshold in neurons are activated and given the value 1 and 0 otherwise.

The first law of learning put by Hebb (1949) was based on simultaneous combinations of neurons capable of strengthening the connection between them. 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).

Werbos (1974) presented a new style of learning that is backpropagation. However, few reports published on this method at the time where estimation errors back to the hidden layer. The backpropagation method has been used successfully in many applications such as playing table, and handwriting recognition, filtration processes, control system, economic forecasting and financial, etc (Kabundu, 2002). The process of estimating the parameters is the most important step to build the 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
162

Mahmoud K. Okasha and Assem A. Yassen,

to the data. Data will be divided into two groups, training group and test group. Training set used to estimate the weights in the model. Thus, the learning process is an important stage in the development of neural network models. The test set, usually consists of 10% to 30% of the total data set, to assess the ability to generalize from the learning process.

2.2 The Back-Propagation Learning Algorithm

The procedure which uses the gradient descent learning technique for multilayer feedforward ANN is known as Back-Propagation, or generalized delta rule as stated by Rumelhart and McClelland (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](image)

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:

\[ E_t = T_t - Y_t \]  

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 absolute error function can be stated as:

\[ MAE = \frac{1}{N} \sum_{t=1}^{N} |T_t - Y_t| \]  

However the mean squared error (MSE), it is expressed as:

\[ MSE = \frac{1}{N} \sum_{t=1}^{N} (T_t - Y_t)^2 \]  

The network is most commonly trained using the MSE error function. But in NN modeling this differs slightly from equation (2.3) where the network objective function is:
\[ E = \frac{1}{2} \sum_{t=1}^{N} (T_t - Y_t)^2 \] (2.4)

The constant \( \frac{1}{2} \) 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 (2.4) the constant \( \frac{1}{2} \) disappears, while if (2.3) 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.

### 2.3 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 \( \theta \) is given by:

\[ \nabla E(\theta) = \frac{\partial E}{\partial \theta} \] (2.5)

where \( \theta \) 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:

\[ \theta_{t+1} = \theta_t + \Delta(\theta)_t \] (2.6)

where: \( \theta_t \) are network weights of the \( t \)th iteration, \( \theta_{t+1} \) 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:

\[ \Delta(\theta)_t = -a\nabla E(\theta) \] (2.7)

where \( a \) is a positive constant called the learning rate. Return to the error function (2.4), where

\[ Y_t = f\left(\sum_{j=1}^{m} \sum_{i=1}^{n} X_{ij} w_{ij}\right) \quad \text{or simply} \quad Y_i = f(X_i, \theta_t) \] (2.8)

From (2.8), equation (2.5) can written as:

\[ \nabla E(\theta) = \frac{\partial E}{\partial \theta_j} = \frac{\partial E}{\partial w_{ij}} \] (2.9)

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:

\[ f(\mu) = \frac{1}{1+e^{-\mu}} \] (2.10)

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

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

$$= -f(u)(1-f_i(u))(T_i - f_i) \quad (2.12)$$

From equation (2.12), we get

$$\frac{\partial u}{\partial w_{ji}} = X_i \quad (2.13)$$

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 (2.4), we have:

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

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

$$= -(T_i - f_i(u)) \quad (2.14)$$

Similarly, the partial derivative of (2.10) with respect of $u$ is

$$\frac{\partial f(u)}{\partial u} = -f_i(u)(1 - f_i(u)) \quad (2.15)$$

Substituting (2.13), (2.14), and (2.15) into (2.12), we have:

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

$$= -f_i(u)(1 - f_i(u))(T_i - f_i(u))X_i \quad (2.16)$$

Equation (2.11) becomes

$$\nabla E(\theta) = \frac{\partial E}{\partial \theta_i} = \frac{\partial E}{\partial w_{ji}}$$

$$= -X_i f_i(u)(1 - f_i(u))(T_i - f_i(u)) \quad (2.17)$$

Substituting (2.17) into (2.9), and thereafter into (2.8), we get the number of iterations and $\Delta(\theta)_t$ of the learning process as follows:

$$\theta_{t+1} = \theta_t + a X_i f_i(u)(1 - f_i(u))(T_i - f_i(u)) \quad (2.18)$$
or
\[ \theta_{t+1} = \theta_t + a X_t (1 - f(X_t, \theta_t)) (T_t - f(X_t, \theta_t)) \]

The NN learning experience updates the \( \Delta(\theta) \) 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 (2.18) we conclude that the backpropagation method is based on three factors: the learning rate \( a \), the distance between the actual output and predicted output \( (T_t - f(X_t, \theta_t)) \), and the activation function \( f(X_t, \theta_t) \).

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 the style gradient descent, which is used heavily in the field of modeling biological systems and dynamics.

3: Data analysis using traditional Box-Jenkins approach:

3.1 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 (2011). The number of observations in the series is 760, representing daily scores in the period from the first of August 2007 until the end of August 2010. 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:

\[
\text{Al-Quds index} = \frac{\text{sum} \left( \text{number of shares subscribed} \times \text{the trading price} \right) \times 100}{\text{sum} \left( \text{number of shares subscribed} \times \text{primary price per share} \right)}
\]
3.2 Fitting ARIMA model to the data:
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 543.249, the median of the time series is estimated at 513.0850 and the standard deviation is estimated at 75.6577. 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 (3.1) 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.

Fig. 3.1: The logarithmic transformation of the Al-Quds Index time series

The autocorrelation function of Al-Quds index of PSE time series – as shown in figure 3.2 - indicates that all autocorrelations are "significantly different from zero", and the only pattern in the correlogram is the linear decrease associated with the increasing lag. Moreover, the sample Partial Autocorrelation Function is indeterminate, and cut-off after the second lag (figure 3.2). This indicates that we are dealing with a typical correlogram of a nonstationary time series. Therefore, this requires us to take the first difference of the natural logarithm of the original time series of of Al-Quds index of PSE in order to transform it to a stationary series. Figure
The Application of Artificial Neural Networks In Forecasting Economic Time Series

3.3 below exhibits the series of the first difference of the natural logarithm of the original Al-Quds index of PSE time series.

**Figure 3.2:** The correlogram of logarithm of the Al-Quds Index of PSE

![ACF and Partial ACF plots for the Al-Quds Index of PSE](image)

**Figure 3.3:** Graphical display of the 1st differences of the natural logarithm of Al-Quds index of PSE

![Graphical display of differences](image)
Looking at the correlogram of first difference of natural logarithm series in figure (3.4) above we can observe that the transformed series became a stationary one. Also we can test the stationarity of the transformed series through the KPSS test of Kwiatkowski et. al. (1992) which is a popular test in econometrics, as it is quite efficient and easy to be interpreted. The results of applying the KPSS test on the series of the first differences of the natural logarithm of the original Al-Quds index of PSE time series is that KPSS Level equals 0.1608, Truncation lag parameter is 6, and the p-value is approximately 0.1. This indicates that the KPSS Test for Level stationarity of our series is that the first difference of the natural logarithm of Al-Quds index of PSE is now a stationary.

Next, we need to fit the Box-Jenkins ARIMA model for the Al-Quds index of PSE time series. We identify the order $p$ and $q$ of the ARIMA model noting that we identified $d=1$. The correlogram of the first difference of the natural logarithm of Al-Quds index of PSE time series given in figure (3.4) above enables us to identify the values of these parameters. In this correlogram, it is possible to note a significant autocorrelation and partial autocorrelation at lag 1. There are several graphical tools to facilitate identifying the ARMA orders. Those include, the corner method (Becuin et al., 1980), the extended autocorrelation (EACF) method (Tsay and Tiao, 1984), and
the smallest canonical correlation method (Tsay and Tiao, 1985). 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 (3.1) below.

Table 3.1: The Theoretical Extended ACF (EACF)

<table>
<thead>
<tr>
<th>AR/MA</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the above table we can observe that an appropriate model for the series can be either ARIM(0,1,1), ARIM(1,1,2), or ARIM(2,1,2). Therefore we estimate the 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) as can be seen in table (3.2), since this model has the lowest Akaike Information Criterion, (AIC), and Bayesian Information Criterion, (BIC).

Table 3.2: The value for (AIC, AICc) for different ARIMA models

<table>
<thead>
<tr>
<th>ARIMA</th>
<th>AIC</th>
<th>AICc</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA(0,1,1)</td>
<td>-4367.22</td>
<td>-4367.19</td>
<td>-4353.32</td>
</tr>
<tr>
<td>ARIMA(2,1,2)</td>
<td>-4361.94</td>
<td>-4361.83</td>
<td>-4334.15</td>
</tr>
<tr>
<td>ARIMA(1,1,2)</td>
<td>-4363.91</td>
<td>-4363.83</td>
<td>-4340.75</td>
</tr>
</tbody>
</table>

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

\[
\hat{Y} = 0.2203 \varepsilon_{t-1} 
\]

with AIC = -4367.22 ; RMSE = 1.357019e-02, where \( Y_t \) denotes the differenced natural logarithm of Al-Quds index of PSE series.

Now, figure 3.5 below displays three diagnostic tools for the above fitted model.
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 appears to be a very good fit for the natural logarithms of Al-Quds index of PSE time series. Therefore the estimated ARIMA(0,1,1) model seems to be capturing the dependence structure of the natural logarithms of Al-Quds index of PSE time series quite well.

Figure 3.5: Diagnostic tests for the ARIMA(0,1,1) Model for Diff(LOGPAL)
Now the ARIMA(0,1,1) model has been fitted to the series of stock price index. Investigating the results of fitting ARIMA(0,1,1) model to the data, we can conclude that all that all coefficients are significant and the diagnostic model suggests that this model fits the data adequately.

Using the model in 3.1 above for forecasting we get the results in table(3.3) and figure(3.6) below. They illustrate the forecasted ten points of the time series compared with last ten actual values with 95% forecast limits. We note that the first three values only are close to the actual values and the rest revert to the mean of the series. Since the model does not contain many autocorrelations, the forecasts, quickly settle down to the mean of the series and the forecasting limits contain all of the actual values.

**Table 3.3:** Forecasting results of ARIMA model for LOGPAL time series

<table>
<thead>
<tr>
<th>year</th>
<th>actual</th>
<th>forecast</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010.162</td>
<td>2.68480</td>
<td>2.72123</td>
</tr>
<tr>
<td>2010.163</td>
<td>2.68466</td>
<td>2.68466</td>
</tr>
<tr>
<td>2010.164</td>
<td>2.68345</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.165</td>
<td>2.68460</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.166</td>
<td>2.68571</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.167</td>
<td>2.69127</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.168</td>
<td>2.69245</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.169</td>
<td>2.68935</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.170</td>
<td>2.69182</td>
<td>2.68356</td>
</tr>
<tr>
<td>2010.171</td>
<td>2.68851</td>
<td>2.68356</td>
</tr>
</tbody>
</table>

**Figure 3.6:** Actual, forecast and forecast limits for D(LOGPAL index)
Mahmoud K. Okasha and Assem A. Yassen,

4: Fitting the Artificial Neural Network model for the data:
For fitting ANN model for a time series data, as described in section 2 above, we may use the Matlab software. The Matlab commands 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:

\[ MSE = \frac{1}{N} \sum_{t=1}^{N} (T_t - Y_t)^2, \]  

(4.1)

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 method on our dataset, the number of observations to be used for training, should be the same as the number of observations used in estimating ARIMA model. Therefore, we increased the series by 85 observations to have an input string of 845 instead of 760 observations, (90% as a training set, and 10% as a test set for the goodness of the forecasts). In fitting ANN we 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 we 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.

**Figure 3.7:** Forecasting outputs of LOGPAL (ANN –Matlab)

Selection of hidden layers is not a straightforward technique. When the number of
hidden layer units is small, the correlation of the output and input cannot be studied properly and the errors increase. However, when the number of hidden layer units is inadequately large, the unrelated noise as well as the correlation of both input and output might be studied, and the error increases accordingly. Many methods to identify the number of hidden layer units exist, but there is no ideal solution for this problem (Kermanhahi, et al, 2002). Therefore, in our analysis we started with one hidden layer and gradually increased the number of hidden layers to a fifteen layers.

When then applied the Feedforward Backpropagation network for the stock price index data and starting with one unit in the hidden layer associated with one and two lags and different learning rates, the ninety subsequent results produced ninety networks. Figure (3.7), shows that the 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 of the ninety runs are shown in table (3.4) below.

**Table 3.4:** Lowest MinRMSE results of ANN model for LOGPAL time series.

<table>
<thead>
<tr>
<th>Learning Rate</th>
<th>1 unit in hidden layer</th>
<th>2 unit in hidden layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0048999</td>
<td>0.0048377</td>
</tr>
<tr>
<td>0.05</td>
<td>0.0048393</td>
<td>0.0048501</td>
</tr>
<tr>
<td>0.01</td>
<td>0.0048898</td>
<td>0.0048520</td>
</tr>
</tbody>
</table>

From table (3.4) we can see that the minimum RMSE of the natural logarithms of AI-
Mahmoud K. Okasha and Assem A. Yassen,

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.

Table 3.4: Forecasting results of ANN model for LOGPAL time series

<table>
<thead>
<tr>
<th>year</th>
<th>actual</th>
<th>ANN forecast</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010.162</td>
<td>2.68480</td>
<td>2.7013</td>
</tr>
<tr>
<td>2010.163</td>
<td>2.68466</td>
<td>2.7013</td>
</tr>
<tr>
<td>2010.164</td>
<td>2.68345</td>
<td>2.70098</td>
</tr>
<tr>
<td>2010.165</td>
<td>2.68460</td>
<td>2.70096</td>
</tr>
<tr>
<td>2010.166</td>
<td>2.68571</td>
<td>2.70096</td>
</tr>
<tr>
<td>2010.167</td>
<td>2.69127</td>
<td>2.70064</td>
</tr>
<tr>
<td>2010.168</td>
<td>2.69245</td>
<td>2.70065</td>
</tr>
<tr>
<td>2010.169</td>
<td>2.68935</td>
<td>2.70065</td>
</tr>
<tr>
<td>2010.170</td>
<td>2.69182</td>
<td>2.70033</td>
</tr>
<tr>
<td>2010.171</td>
<td>2.68851</td>
<td>2.70035</td>
</tr>
</tbody>
</table>

The results of applying both ARIMA and ANN methods are compared through the results and figures (3.6), (3.7) and (3.8), as well as tables (3.1), (3.2) (3.3) 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 0.0135702. 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.

5: Conclusion
Forecasting financial time series, such as indices and stock prices is a complex
The Application of Artificial Neural Networks In Forecasting Economic Time Series

The 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

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


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


[30] Zhang, G. P; (2003); "Time Series Forecasting Using A Hybrid ARIMA and Neural Network Model"; Neurocomputing, 50:159-175.
Mahmoud K. Okasha and Assem A. Yassen,