FORECASTING KUALA LUMPUR COMPOSITE INDEX: EVIDENCE OF THE ARTIFICIAL NEURAL NETWORK AND ARIMA

Raditya Sukmana & Mahmud Iwan Solihin
Academic Staff 0f Department of Economics Airlangga University &
Department of Mecatronics Engineering, International Islamic University Malaysia

ABSTRACT

The aim of this paper is to use, compare, and analyze two forecasting technique: namely Auto Regressive Integrated Moving Average(ARIMA) and Artificial Neural Network(ANN) using Kuala Lumpur Composite Index(KLCI) in Malaysia. Artificial Neural Network is used because of its popularity of capturing the volatility patterns in nonlinear time series while ARIMA used since it is a standard method in the forecasting tool. Daily data of Kuala Lumpur Composite Index from 4 January 1999 to 26 September 2005 is used. ANN training with "early stopping" technique is investigated. We found that the deviation or error showed in the ANN technique is much less than that in ARIMA. Hence ANN can be used as a good forecaster engine for univariate time series model. It can predict nonlinear time series using the pattern of the past data. The proposed technique may help government, decision makers and planners especially in Malaysia.

Keyword : *Auto Regressive Integrated Moving Average(ARIMA), artificial neural network(ANN) and Kuala Lumpur Composite Index(KLCI)*

1. INTRODUCTION

Discovering trends and patterns in banking and financial data has been put in highly attention by businesses and academicians. The detection of trends and patterns has combined statistical and econometric modeling. The mathematical models related to the forecasting may fail to predict the patterns in the dynamic of economic. However new methods of forecasting is growing very fast recently.

Recent methods include those using artificial neural networks (or often called as neural networks) has attracted forecaster to predict the trends and patters. In fact, many forecasters dealing with financial matters are using neural network method. Neural networks can be used to forecast stock market, foreign exchange trading, bond yield and commodity future trading.

Various attempts of economic time series prediction have been done ending with saying that neural network-based forecasting outperform the conventional model [1]. Many researchers have also suggested that neural network can serve as alternative and novel tools in business, e.g., in forecasting financial time series [2]. Neural network has many advantages over conventional methods of analysis. First, they have the ability to analyze complex patterns quickly and with a high degree of accuracy. Second, neural network makes no assumptions about the nature of the data distribution.

In dealing with the autocorrelation analysis, we often use ARIMA model (Box Jenkins) as a statistical approaches for forecasting. This analysis mainly detects the autocorrelation between the previous observations of time series. This method is very efficient to forecast linear time series. The problem of ARIMA approach arises when the time series is of increasing variance or when the time series represents nonlinear processes [3]. Kolarik [3] has conducted time series forecasting using neural network. He compared the result with the well known forecasting techniques, ARIMA. His result showed that forecasting error using neural network was less compared to that of ARIMA.

The structure of this paper is as follows: after introduction in Chapter 1, it follows with the theory of ARIMA (chapter 2). The major part of this paper will discuss on the theory of ANN and its uses in the forecasting technique (chapter 3 and 4). Data and Methodology are in the next chapter. Result and conclusion will be in the chapter 6 and 7.

2. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

ARIMA is also called Box Jenkins methodology. The emphasis of this forecasting technique is not on constructing single equation or simultaneous-equation models but on analyzing the probabilistic, or stochastic, properties of economic time series on their own under the philosophy "*let the data speak for themselves*" (Gujarati,1995). The difference between normal regression and this model is that in normal regression Y_t is explained by the k regressors $X_1,X_2,X_3,\ldots,X_k$. In this ARIMA model, Y_t may be explained by past, or lagged, value of Y itself and stochastic terms. Therefore this Box Jenkins Methodology is sometimes called a-theoretic models because they cannot be derived from any economic theory-and economic theories are often the basis of simultaneous equation models (Gujarati,1995).

In the time series regression, one important aspect which regard to the data is that data should be stationary. Forcing the non-stationary data into the equation would result into a spurious regression. It is a dubious regression in the sense that superficially the result look good but on further probing they look suspect. However, many researchers who are dealing a lot with time series data may aware that many of the time series data is not stationary. To solve this problem, the data should be generated into first difference. If the data is integrated of order one I(1) its first difference is I(0), that is, stationary. In the same way, if a time series data is I(2), it means that it stationary in the second difference or its second difference is I(0). Therefore, in general, if a time series is I(d), after differencing it d times we obtain an I(0) series.

3. ARTIFICIAL NEURAL NETWORK

A neural network is a computational technique that benefits from techniques similar to ones employed in the human brain. Its very basic concept was introduced in 1940s. It is designed to mimic the ability of the human brain to process data and information and comprehend patterns. It imitates the structure and operations of the three dimensional lattice of network among brain cells (nodes or neurons, and hence the term "neural").

A neural network can also be defined as a processing system consisting of a large number of simple, highly interconnected processing element (neurons) in an architecture inspired by the structure of the human brain [4]. A neural network is characterized by its architecture, learning algorithm and activations function. The architecture describes the connections between the neurons. It consists of, generally, an input layer, an output layer and adequate number of hidden layers in between. The layers in the network are interconnected by links that associated with weight that dictate the effect on the information passing through them. These weights are updated by learning algorithm. The activation function relates the output of a neuron to its input. The most commonly used activation functions are: threshold, linear, and sigmoid function. Figure 1 shows a single artificial neuron, where w_{ji} is the weight values and f is activation fucntion.

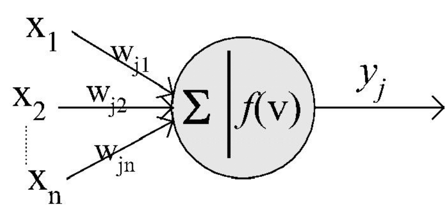


Figure 1. A single artificial neuron.

Table 1 shows the typical activation function used for multilayer neural network. The choice of activation function from one layer to another may be different.

Table 1
Activation functions

Type of activation function	
Linear	$f(v)=v$
Threshold	$f(v)=\begin{cases}1 \ if \ v\geq 0\\ 0 \ if \ v<0\end{cases}$
Log sigmoid	$f(v)=\frac{1}{1+e^{-v}}$
Tan sigmoid	$f(v)=\frac{1-e^{-2v}}{1+e^{2v}}$

The learning process of the neural network can be likened to the way a child learns to recognize patterns, shapes and sounds, and discerns among them. For example, the child has to be exposed to a number of examples of a particular type of tree for her to be able to recognize that type of tree latter on. In addition, the child has to be exposed to different types of trees for her to be able to differentiate among trees.

The human brain has the uncanny ability to recognize and comprehend various patterns. The neural network is extremely primitive in this aspect. The network's strength, however, is in its ability to comprehend and discern subtle patterns in a large number of variables at a time without being stifled by detail. It can also carry out multiple operations simultaneously. Not only can it identify patterns in a few variables, it also can detect correlations in hundreds of variables. It is this feature of the network that is particularly suitable in analyzing relationships between a large number of market variables. The networks can learn from experience. They can cope with "fuzzy" patterns – patterns that are difficult to reduce to precise rules. They can also be retrained and thus can adapt to changing market behavior.

The network holds particular promise for econometric applications. Multilayer feed forward neural networks as a common type of neural network with appropriate parameters are capable of approximating a large number of diverse functions arbitrarily well [5]. Even when a data set is noisy or has irrelevant inputs, the networks can learn important features of the data. Inputs that may appear irrelevant may in fact contain useful information. The promise of neural networks lies in their ability to learn patterns in a complex signal.

Recently, there are at least two most popular neural networks types in various applications, namely multilayer feed forward network, as mentioned before, and radial basis function network. Multilayer feed forward network is used in this paper. This type of neural network is shown in Figure 2.

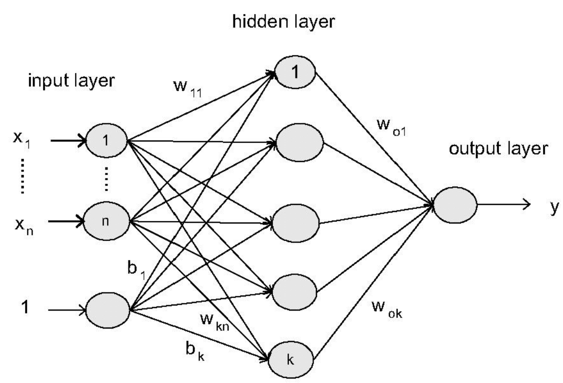


Figure 2. A Typical multilayer feedforward neural network.

The output of node j in the hidden layer is given by

$$h_j = f\left(\sum_{i=1}^{n} w_{ji} . x_i + b_j\right)$$

where b_j is bias value and the output of the network by

$$y = \sum_{i=1}^{k} (w_{oi} . h_i)$$

The learning or training process of neural network is carried out using input-output data to update the weights and biases. One of the common technique used to train multilayer feedforward neural network is backpropagation algorithm [4,6]. The weight adjustment is done iteratively from the output layer backward to previous layers to achieve a minimum mean square error (MSE) between the network output and the desired output [4]. The number of iterations is often termed as epoch.

The backpropagation algorithm can be summarized as follows. Suppose a MFN is trained by gradient descent to approximate an unknown function, based on some training data consisting of pairs (*x,t*). The vector *x* represents a pattern of input to the network, and the vector *t* the corresponding target (desired output). The overall gradient with respect to the entire training set is just the sum of the gradients for each pattern; then it is described how to compute the gradient for just a single training pattern. Let's number the units, and denote the weight from unit j to unit i by w_{ij}.

1. **Definitions:**

o the **error** signal for unit j:	$\delta_j = -\partial E / \partial net_j$
o the (negative) **gradient** for weight w_{ij}:	$\Delta w_{ij} = -\partial E / \partial w_{ij}$
o the set of nodes **anterior** to unit i:	$A_i = \{j : \exists\, w_{ij}\}$
o the set of nodes **posterior** to unit j:	$P_j = \{i : \exists\, w_{ij}\}$

2. **The gradient.** The gradient is expanded into two factors by use of the chain rule:

$$\Delta w_{ij} = -\frac{\partial E}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}}$$

The first factor is the error of unit i. The second is

$$\frac{\partial net_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_{k \in A_i} w_{ik}\, y_k = y_j$$

Putting the two together, we get

$$\Delta w_{ij} = \delta_i y_j$$

To compute this gradient, we thus need to know the activity and the error for all relevant nodes in the network.

3. **Forward activaction.** The activity of the input units is determined by the network's external input **x**. For all other units, the activity is propagated forward:

$$y_i = f_i\Big(\sum_{j \in A_i} w_{ij}\, y_j\Big)$$

Note that before the activity of unit i can be calculated, the activity of all its anterior nodes (forming the set A_i) must be known. Since feedforward networks do not contain cycles, there is an ordering of nodes from input to output that respects this condition.

4. **Calculating output error.** Assuming that the sum-squared loss is used.

$$E = \frac{1}{2}\sum_{o}(t_o - y_o)^2$$

the error for output unit o is simply

$$\delta_o = t_o - y_o$$

5. **Error backpropagation.** For hidden units, the error is back propagated from the output nodes (hence the name of the algorithm). Again using the chain rule, we can expand the error of a hidden unit in terms of its posterior nodes:

$$\delta_j = -\sum_{i \in P_j} \frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial y_j}\frac{\partial y_j}{\partial net_j}$$

Of the three factors inside the sum, the first is just the error of node i. The second is

$$\frac{\partial net_i}{\partial y_j} = \frac{\partial}{\partial y_j}\sum_{k \in A_i} w_{ik}\, y_k = w_{ij}$$

while the third is the derivative of node j's activation function:

$$\frac{\partial y_j}{\partial net_j} = \frac{\partial f_j(net_j)}{\partial net_j} = f_j'(net_j)$$

For hidden units h that use the tanh activation function, we can make use of the special identity
tanh(u)' = 1 - tanh(u)^2, giving

$$f'_h(net_h) = 1 - y_h^2$$

Putting all the pieces together, yields

$$\delta_j = f'_j(net_j) \sum_{i \in P_j} \delta_i w_{ij}$$

Furthermore, the common steps in neural network learning or training process are as follows :

a. Input collection and preprocessing of the input data (if necessary). This includes the number of sampled data and normalization.
b. Determining the structure of the neural network. Including number of inputs, neurons, hidden layers and activation functions.
c. Training the neural network, using the input pattern and desired output.
d. Testing of the trained neural network.

4. ARTIFICIAL NEURAL NETWORK FOR TIME SERIES FORECASTING

The ability of neural network to discover nonlinear relationships in input data makes them ideal for modeling nonlinear time series. Therefore, neural network can be used to predict future values in a time series based on current and historical data. This can be considered as univariate modeling of time series. Figure 3 shows the typical network for time series forecasting.

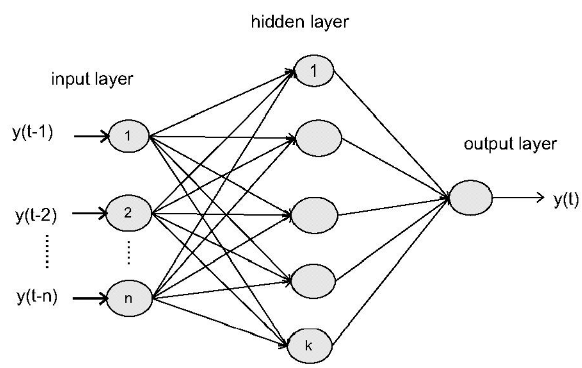


Figure 3. Multilayer feedforward network for time series forecasting

However, due to the tedious and time consuming of neural networks training, some of models reported in literature did not go through comprehensive experiments. The most time consuming process in neural network computation is determining the structure of the neural network. Unfortunately, the number of neurons, hidden layers and the types of activation functions can not be determined beforehand. It is usually found empirically. Even they have resulted poor prediction though they have successfully train the neural network model. This happens when the neural network has poor generalization capability or the data used is limited to small number. The neural network has memorized the training examples but it has not learned to generalize the new input pattern in prediction. This is because what so called *overfitting* occurs during the training. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large.

One method to improve generalization capability is called *early stopping.* In this technique the available data is divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned [7].

5. DATA AND METHODOLOGY

The number of available data of Kuala Lumpur Composite Index (KLCI) is 1655, and it is daily from 4 January 1999 up to 26 September 2005. The data can be downloaded online from yahoo.com (Yahoo.com is getting data from Kuala Lumpur Stock Exchange). The first 80% data, which is 1324 observations, is used for training and the last 20% data, which is 331 observation, is forecasted by ARIMA and the neural network. The data is previously normalized into between -1 and 1 (for ANN only). Figure 3 shows the data used with the partitions.
-ARIMA

As have been mentioning in the ARIMA theory above, the KLCI data used in this paper should be checked for its stationarity. To be able to check it, this paper uses Unit root test (shown in table 2)

Table 2

	P-Value			
	Level		First Difference	
	ADF	PP	ADF	PP
KLCI	0.467	0.4388	0.0000***	0.0000***

Notes:
-This Unit Root Is Using Akaike Info Criteria
-This Unit Root Is Using Trend and Intercept
-*** indicates significant in the 1% level

From Table 1, we can conclude that KLCI data is I(1), it means that in the level KLCI is not stationary whereas it is stationary in the first difference. Hence, this data can be used for further calculation.

Next step is to use KLCI data to ARIMA model. Notice that since the data is in the first difference then our data should all be generated in the first difference and use it in the regression. In this ARIMA model, KLCI will be regressed with its own lags. In order to get the parsimonious (means simple and accurate) regression, and how many lags used, it will follow General to Specific Method as proposed by Campos, Erricsson and Hendry (2005). Basically, it starts with any lags, in this paper we start with independent variables with lag 1 to lag 7 since it uses daily data. The General to Specific method is to delete the last lag which is not significant and re-regress again and continue with deleting the last lag which is not significant. Finally we will come up with all significant independent variables. This can be shown in the table 3 below:

Table 3

Dependent variable: DKLCI

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.119217	0.239942	0.496857	0.6194
DKLCI(-1)	0.161819	0.027136	5.963267	0
DKLCI(-5)	0.075018	0.027104	2.767752	0.0057

From Table 2 above we can imply it is a parsimonious equation that since the probability of two independent variables above (DKLCI(-1) and DKLCI(-5)) less than 1%. Hence the equation will be:

DKLCI=0.119217 +0.161819*DKLCI(-1) +0.075018*DKLCI(-5)..........................(3)

Notice that equation 1 above is in first difference, in order to make forecasting, equation 1 above should be degenerate into level, therefore, the equation will be:

KLCI-KLCI(-1) =0.119217 +0.161819*(KLCI(-1)-KLCI(-2)) +0.075018*((KLCI(-5)-KLCI(-6))...(4)

Since we need to make a prediction then KLCI(-1) should be put on the left side of the equation. Therefore the new equation will be:

KLCI =0.119217 + (((1.161819)*KLCI(-1))-0.161819*KLCI(-2)) +0.075018*((KLCI(-5)-KLCI(-6))...(5)

Equation 5 will be used to make a prediction.

-ANN

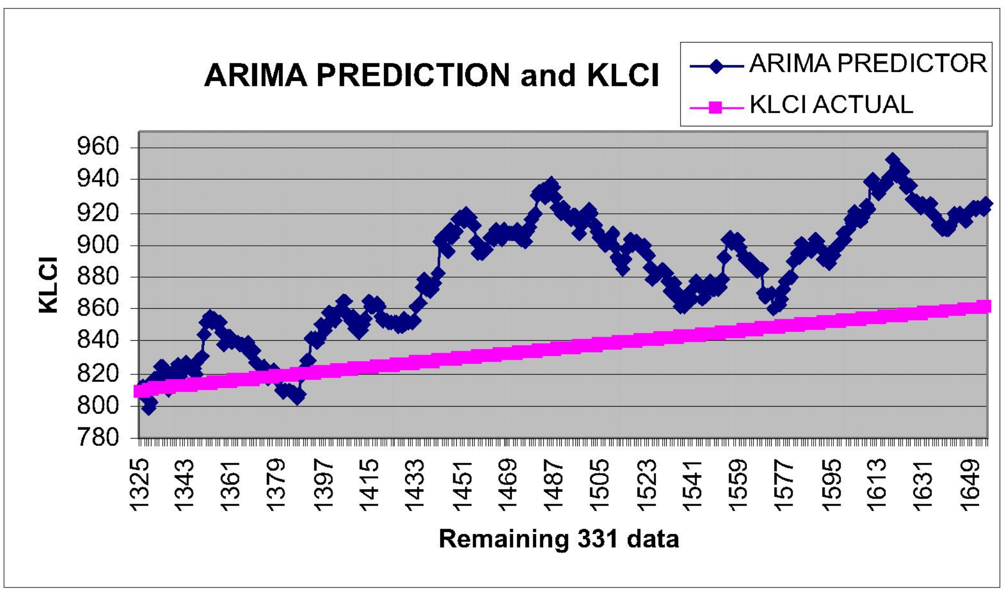
The forecasting is carried out as one step-ahead prediction based on historical data using time delay input referring to Figure 2. Multilayer feedback neural network with back propagation learning algorithm is used in this paper. In the time series forecasting using neural network, the data is divided into two sets. Typically, the first 75% is used to train the neural network and the rest is to test the obtained neural network forecaster. Since early stopping is applied to improve the prediction, thus the training data is divided further into two sets, one as training set and one as validation set for early stopping. This step is vital to build good generalization of the neural network. After one step ahead has been predicted, then the neural network is retrained with new input pattern to predict the next one step. This is carried out iteratively until 32 times prediction. The training was specified to achieve MSE of 10^{-5}.

The training phase has usually no much problem. The problem is when new input pattern fed to the neural network to predict the new value. This is where *early stopping* plays an important role here to improve generalization capability of the neural network.

6. RESULTS

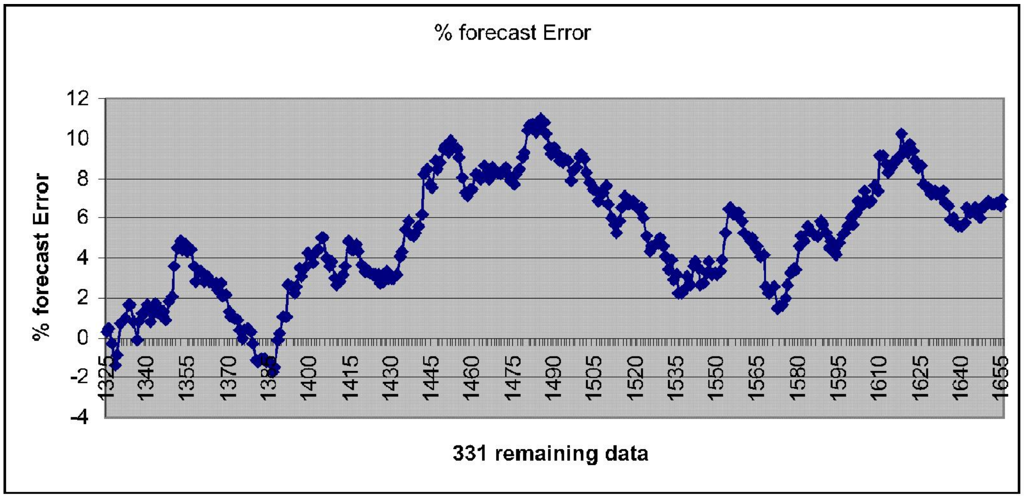
Having found the Forecasting equation[1] on ARIMA method, we then used it to forecast the remaining 20% of the data (331 observations). The predicted value with the data actual of KLCI can be seen in the following figure.

Figure 4



next, it is worth to look at the error made from the ARIMA prediction. From the graph 1 above it shows that although we have already uses a parsimonious equation, deviation from actual data seems to exist and remain large (see figure 5).

Figure 5



From graph 2 above, it shows that the band is from -2 to about 11%. It also shows that ARIMA is good predictor for a next few days, however ARIMA seems not be a good predictor for long term. The longer the term, more deviation would occur. From this result, we can conclude that what has been argued by Kolarok[3] that ARIMA suffers from increasing variance seems to be true or in another word, our result support the kolarik's argument.

The plots comparing the actual index and the forecasting result by ANN with *early stopping technique* is shown in Figures 7. The error of forecasting is also shown. From the plot, it can be seen that the ANN has shown a good capability for time series forecasting based on historical data of KLCI. The proposed method of ANN was able to forecast the pattern with error less than 2% although the data seems to be volatile.

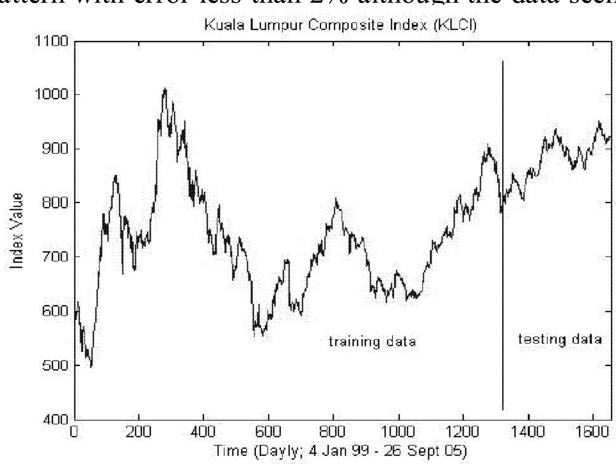


Figure 6. Daily Kuala Lumpur Composite Index (4 January 1999 – 26 September 2005)

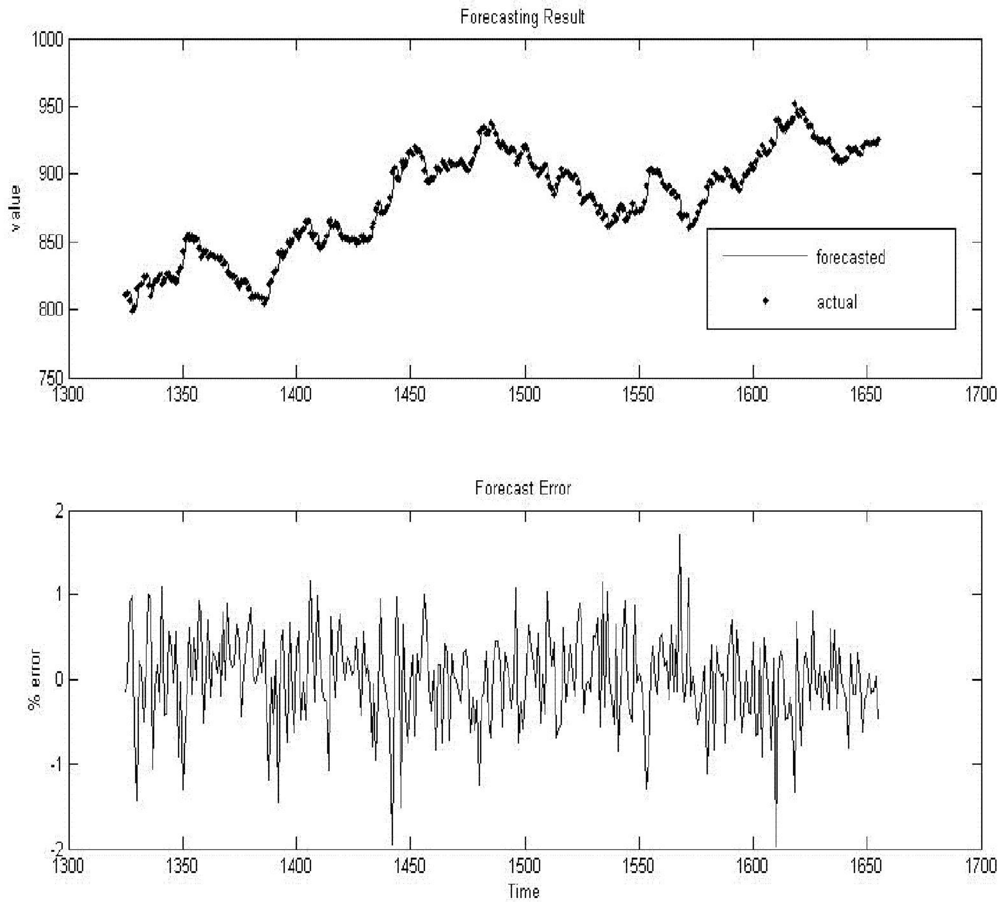


Figure 7. Forecasting result and percent error of forecasting

7. CONCLUSIONS

Forecasting is very important for decision maker to have an idea of what might occur in the future. We, in this paper, have tried to compare the classical forecasting tool namely ARIMA with an alternative method for nonlinear time series forecasting which is ANN. The ANN employs a univariate time series model where the historical data of KLCI is used to predict the coming value. One step-ahead prediction is carried out in this paper. Results show that error made by ARIMA is much bigger as compare with ANN. The error of forecasting is found less than 2% while in ARIMA the error forecasting up to 11%. Therefore, in general ANN with early stopping technique can be used as an alternative for forecasting .

REFERENCES

Hamid, S.A., *Primer On Using Neural Networks For Forecasting Market Variables*. Working Paper No. 2004-03, Southern New Hampshire University, 2004.

Wong, B. K, T A Bodnovich, Y Selvi, *Neural Network Applications in Business: a Review and Analysis of the Literature (1988-1995)*, Decision Support Systems, Vol 19, No. 4, 1998, pp. 301-320.

Kolarik, T., Rudorfer., G, *Time Series Forecasting Using Neural Network*, Time series and Neural Network, APL,1994.

Lin, C.T., and Lee, G.C.S., *Neural Fuzzy Systems, A Synergism To Intelligent System*, Prentice-Hall, 1996.

Daniel, C.P.C., Wei, N.C., Lek, T.H., *Using Neural Network to Forecast Foreign Exchange Rate*. Nanyang Technological University, 1999.

Kumar, S. *Neural networks: a classroom approach.* New Delhi: Tata-Mc. Graw Hill. (2004).

Neural Network Toolbox User's Guide. The MathWorks, Inc. 2001.

Lin, F., Yu, X.H., Gregor S and Irons R, *Time Series Forecasting with Neural Networks*, Complexity International Vol 2, April 1995.

White, H. (1989). *Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings*. Neural Networks 3, 535-549.

Gujarati D (1995). *Basic Econometrics*: Mc Graw Hill.

Campos,J. Erricsson,N. R. Hendry D.F," *General to Specific Modelling: An Overview and Selected Bibliography*. Board of Governors of the Federal Reserve System International Finance Discussion paper Number 838 August 2005