The feed forward neural network with genetic algorithm for daily stock prediction

To cite this article: R Dipinto et al 2019 J. Phys.: Conf. Ser. 1217 012076

View the article online for updates and enhancements.
The feed forward neural network with genetic algorithm for daily stock prediction

R Dipinto¹, R Santoso¹, and A Prahutama¹

¹Department of Statistics, Diponegoro University
Jl. Prof. Soedharto, SH, Tembalang, Semarang 50275, Indonesia
E-mail: rukunsantoso25@gmail.com

Abstract. The power of Feed Forward Neural Network (FFNN) in conjunction with Genetic Algorithm (GA) was applied in this research to predict daily stock price. Finance time series data has a high complexity, so that the accurate prediction is hard to be gained by standard model. Machine learning becomes the new prediction tool which is often used because of its adaptive properties. Neural Network (NN) is one of the machine learning which able to complete inference tasks such as prediction, especially in large data sets. FFNN is one of the NN models that has simple network architecture. In the standard optimization method, the initial weights is randomly selected to desire the optimum solution. But it usually raises the problem of unsteady estimation. The GA optimization method was applied in this research to overcome this lack. GA optimizes any function effectively and seeks a global optimum solution efficiently. GA implementation on the FFNN was aimed to obtain optimum weights that minimizing the error. The daily stock price prediction of PT. Adhi Karya Tbk had RMSE of training and testing data at 51.2531 and 44.8706 respectively. This result was equivalent with MAPE values at 1.5714% and 1.5501%.

1. Introduction

The standard method for non-seasonal time series prediction is ARIMA (Autoregressives Integrated Moving Average) model. The operational of ARIMA is restricted by some assumption such as data stationerity and the residuals must be normal identic and independent distributed. In some finance data, such as stock price data, this restriction was causing difficulties on implementation stage. A free assumption method seems to be needed to overcome the problem.

Neural Network (NN) is an information processing system that has characteristics similar to the neural network of living things [5]. NN consists of a number of information processing elements called neurons. Neurons are arranged in layers and have patterns of interconnection within and between layers called network architecture. In general, NN architecture consists of several layers i.e. the input layer, hidden layer, and the output layer. Feed Forward Neural Network (FFNN) is one of the NN models that has a fairly simple network architecture with a hidden layer.

In general, FFNN is trained by backpropagation algorithms to get the network weights. Montana and Davis (1993) [7] have explained that the backpropagation algorithm can work well on simple training problems but its performance will decrease and converge to local optimum solutions when applied to data that has a large complexity. This problem can be overcome by Genetic Algorithm (GA) optimization method to obtain the global optimum solution. The searching of optimal solution point is refered on natural phenomena based on the idea of genetics (the study of inherited inheritance) i.e. Darwin's evolution theory of survival of the fittest. The mechanism of GA is utilized by natural
selection, the mechanism of crossing, mutation and others. A competition between individuals will always occur to maintain the survival. The Strong individuals are able to survive, while the weaks will extinct (Darwin's law). The defendable individuals will take a role in the regeneration process and generation improvement [1][8]. Imitatively this process, GA can be applied in the search for optimal solutions to problems in the real world [1]. The advantage of using GA is very clearly visible from its ability to find a good solution that is acceptable for high-dimensional problems. In its application, there are measures that must be determined in advance by the researcher, namely population size, probability of crossover, number of generations, probability of mutation. The choice of selection methods and crossover techniques is also very important [1]. GA is one of the most appropriate algorithms to solve complex optimization problems, which are difficult to do by conventional methods [9]. The data which support the model building are based on daily stock price of PT. Adhi Karya Tbk.

2. Literature Review

Time series data is a set of observation result which is ordered by time [11]. The goal of time series analysis is forecasting. This activity is processing the last condition to gain prediction in the future [4]. The good prediction must has a minimum deviation [11]. The good prediction method must accommodates the pattern of data. Makridakis et.al (1995) [6] group time series pattern into horizontal pattern, seasonal pattern, cyclic pattern and trend pattern. The long life fluctuation make a role in the cyclic pattern. An economic and business data, such as stock price, may perform a cyclic pattern. Wei (2006) [11] measure the relationship level of time series data over lag time by use of autocorrelation function (ACF) and partial autocorrelation function (PACF).

Refer to Beasley [1] and Denuth [3], the GA works on a population which constitutes randomly generated solutions. Each member of the set that represents a solution is called an individual or chromosome. A chromosome contains a number of genes, which encode information stored on the chromosome. A chromosome breeds through repeated iterations called generations. In each generation, the resulting chromosomes will be evaluated using a measurement called fitness. The value of fitness is characterized by an objective function which will be optimized. The new generation is formed appropriate with the fitness value through crossover and mutation process. After through over some generations, the solution will converge to the best chromosome [9]. Each component or step of the GA structure has many variations of proposed method. For instance, the initialization component which aims to generate randomly a number of individuals as the initial population is running through certain method. Population size depends on the problem to be solved and the type of genetic operator that be implemented. One gene represents a parameter that will be estimated which contributes in optimizing objective function.

In the evolution rule, individuals which have high fitness value will be able to survive, while individuals with low fitness value will die [9]. It is possible that the high fitness value individual will be lost in the selection process, because the process is done randomly. In order to keep the individual that has high fitness value not lost during evolution, it is necessary to make one or several good chromosome copies in the population so that they are maintained for the next generation. This procedure is known as elitism [1]. The Linear Fitness Ranking (LFR) is another strategy which is scaling the fitness values obtained from individual evaluations. The LFR is done by equation (1) to avoid the tendency to converge on a local optimum solution with new fitness value which has a larger range coverage.

\[ f(h)_i = f(h)_{\text{max}} - (f(h)_{\text{max}} - f(h)_{\text{min}}) \left( \frac{R(h)_i - 1}{N - 1} \right) \]  

(1)

Equation (1) produces the interval value of fitness on \([f(h)_{\text{min}}, f(h)_{\text{max}}]\) [1].

Refer to Pearl [8] and Beasley [1], the superior individuals which are existed in a population need to be chosen to do crossover process in order to produce new better individuals. This choice aims to give a higher chance to the individuals with higher fitness value to do reproduction process. There are several parental selection methods, one of which is Roulette Wheel Selection. In this method,
individuals are mapped in a line segment regularly so that each individual segment is the same size as the fitness size. A random number will be generated and individuals who lie on the same segment with this random number area selected. This process is repeated until a number of expected individuals are obtained [9].

| Chromosome | Fitness Value | Percentage (%) | Interval        |
|------------|---------------|----------------|----------------|
| K1         | 1             | 25             | [0, 0.25]       |
| K2         | 2             | 50             | (0.25, 0.75)    |
| K3         | 0.5           | 12.5           | (0.75, 0.875)   |
| K4         | 0.5           | 12.5           | (0.875, 1)      |
| Total      |               | 4              |                |

Table 1 is an example of using the Roulette Wheel Selection method. The cumulative interval is an accumulation of ranges resulting from the percentage of the fitness value of each individual segment, with the aim of being the acceptance interval of the selection process. Suppose the random number generation gives a value of 0.2, then the K1 chromosome is chosen, but if its value is 0.9 then the K4 chromosome is chosen as the parent. There are two genetic operators, namely crossover and mutation. Crossover aims to increase string diversity in one population by crossing strings obtained from previous reproduction. Crossover process is performed on every individual with probability value of \( p_c \), which is determined randomly in the range of [0,1]. That is, crossover occurs only if the resulted random number less than determined value. Mutation is a process to change the value of one or several genes in a chromosome. The mutation process occurs when the random number generated is less than the probability of \( p_m \). If it happens then the gene is changed to its inverse value (in binary encoding, 0 is changed to 1 and 1 is changed to 0).

Haykin [5] states that Artificial Neural Network (ANN) or Neural Network (NN) is a machine designed to model the workings of the human brain in performing certain functions or tasks. This machine has the ability to store knowledge based on experience and make knowledge that has become useful. NN has two stages of information processing, which is the stage of training and the stage of testing. Samarasinghe [10] explains that the training phase begins by incorporating learning patterns (training data) into the network. By using these patterns, the network will modify the weights that are connecting between neurons. While the testing phase is carried out on an input pattern that has not been previously trained by using the weight of the results of the training stage. It is expected that the weights training results which produces minimum error will also give a minimum error in the testing phase. In NN there are neurons that are arranged in layers that have a pattern of interconnections within and between layers called network architectures. NN consists of an input layer, hidden layer and the output layer. Feed Forward Neural Network (FFNN) is an NN model that has a fairly simple network architecture with one hidden layer and can be applied to time series data predictions. In FFNN modeling for time series data, input models are past data \((x_{t-1}, x_{t-2}, ..., x_{t-p})\) and the target is current data \(x_t\) [10]. The general form of the FFNN model is formulated in equation (2).

\[
x_t = \psi_o \{ w_{bo} + \sum_{j=1}^{H} w_{jo} \psi_j (w_{bj} + \sum_{i=1}^{p} w_{ij} x_{t-i}) \}
\] (2)

Where

- \( \psi_o \) : Activation function output layer
- \( \psi_j \) : Activation function hidden layer
- \( w_{ij} \) : The weight of \( i^{th} \) neuron on the input layer towards the \( j^{th} \) neuron on the hidden layer
- \( w_{bj} \) : The weight of bias on the input layer towards the \( j^{th} \) neuron on the hidden layer
- \( w_{bo} \) : The weight of bias on the hidden layer towards the output layer
The training network is a procedure or sequence of integrated steps to modify the values of weights and bias in order to get the appropriate values of weights and biases that would allow them to produce the desired network output. If the error in network output is very small, then it can be said that the appropriate weight and bias values have been obtained and the network has achieved good performance [3]. The Performance of network is characterized by RMSE (Root Mean Square Error) or MAPE (Mean Absolute Percentage Error). These instruments measure the closeness between output and target. Model with smaller RMSE/MAPE value will gives more accurate prediction [11]. The RMSE is formulated by equation (3) and the MAPE is in equation (4).

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (x_t - \hat{x}_t)^2}$$

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left( \frac{x_t - \hat{x}_t}{x_t} \right) \times 100\%$$

The interpretation of the MAPE calculation according to Chen et al. [2] is as follows:
- MAPE <10% the model is very good to be used as a prediction model
- 10% - 20% the model is good to be used as prediction models
- 20% - 50% the model still able to be used as prediction models
- MAPE >50% the model cannot be used as prediction models.

3. Research Methods

The set data which is used in this research represents the daily closing price of PT Adhi Karya Tbk stock in the period of October 6, 2015 to February 27, 2018. This is consisting of 582 points secondary data which available in Yahoo Finance's site (www.finance.yahoo.com).

The architecture of neural network model that be implemented in this research is the Feed Forward Neural Network (FFNN). The FFNN architecture for predicting daily stock price consists of one unit of input layer, one unit of hidden layer and one unit of output layer. Furthermore the hidden layer is equipped by some neurons. The number of neurons in the hidden layer is determined by an empirical formula. The weights of neurons are optimized by Genetic Algorithm (GA) method. The objective of GA method refers to minimize the RMSE criterion [1] [11]. The GA method manages the population size, selection technique, crossover technique, and maximum generation level. The Monte Carlo tenet is applied to keep of consistent model.

In addition, the FFNN architecture with different input patterns and different size of training and testing data also be performed. The FFNN architecture which has smallest average RMSE value is chosen as the best prediction model. The MAPE value of the best model is calculated to ascertain whether the model is suitable to be used as a prediction model. The main software used in this research is Mathworks MATLAB R2017a. The supporting software are Microsoft Office Excel 2016 and MINITAB 17.

4. Result and Discussion

The FFNN model is built as in equation (2) with the hidden layer’s activation function ($\psi_j$) is sigmoid logistic and the output layer’s activation function ($\psi_o$) is identity function (purelin). The number of neurons in the hidden layer is determined by using an empirical formula [1]. According to Makridakis et al. [6]), the selection of inputs is based on lags that have significant PACF values (Figure 1). The over fitting process is performed by addition of input lags. Five kinds are selected, these are lag-1, compound of lag-1 and lag-4, compound of lag-1, lag-4 and lag-24, compound of lag-1, lag-4, lag-24 and lag-35 and compound of lag-1, lag-4, lag-24, lag-35 and lag-41.

The choice of training and testing data size determine the balance of RMSE values resulted in each part of data set. Training process is carried out using Genetic Algorithm (GA) optimization with six times repetitions for some kinds of FFNN architectures based on input lags and size of testing and training data. The GA algorithm is performed in some predefined condition, i.e. determines population size of 50, determines probability of crossover $p_c$=0.8, performs single point crossover technique, determines probability of mutation $p_m$=0.01, performs roulette wheel selection method, determines working at interval of [-1,1], determines maximum generations of 3000 and determines tolerance function 1e-9.
Table 2 shows the compositions of training and testing data size for some compounds of input lag which are selected refer to PACF value. Furthermore, the results model for every compound of input lag are compared each other, to gain the best prediction model.

| Overfitting | Input lags | 60/40 | 75/25 | 80/20 | Total data target |
|-------------|------------|-------|-------|-------|-------------------|
|             |            |       |       |       | Training | Testing | Training | Testing | Training | Testing |     |
| 1           | 1          | 349   | 232   | 436   | 145      | 465      | 116      | 581     |          |        |     |
| 2           | 1, 4,      | 347   | 231   | 433   | 145      | 462      | 116      | 578     |          |        |     |
| 3           | 1, 4, 24   | 335   | 223   | 418   | 140      | 446      | 112      | 558     |          |        |     |
| 4           | 1, 4, 24, 35 | 328   | 219   | 410   | 137      | 438      | 109      | 547     |          |        |     |
| 5           | 1, 4, 24, 35, 41 | 325   | 216   | 406   | 135      | 433      | 108      | 541     |          |        |     |

By use of the empirical formula, the number of neurons in the hidden layer is determined by the number of inputs ($N_i$) and the number of outputs ($N_o$). Because of the number of outputs is one ($N_o = 1$), then the formula of neuron numbers is $2N_i + 1$. The numbers of weights which must be estimated are formulated by $H(p + 2) + 1$ where $H$ is the number of neurons in the hidden layer and $p$ is the number of input lags. Five types of FFNN architecture are performed (look at Table 3) and then be compared to determine the best model refer to the smallest RMSE.

Table 3. Design of FFNN architecture

| Input lag | $N_i$ | $N_o$ | $2N_i + N_o$ | $H(p + 2) + 1$ |
|-----------|-------|-------|--------------|----------------|
| 1         | 1     | 1     | 3            | 10             |
| 1, 4,     | 2     | 1     | 5            | 21             |
| 1, 4, 24  | 3     | 1     | 7            | 36             |
| 1, 4, 24, 35 | 4     | 1     | 9            | 55             |
| 1, 4, 24, 35, 41 | 5     | 1     | 11           | 78             |

Table 4 and Table 5 show the results of the RMSE value for training and testing data respectively. The optimization method which is used for FFNN architecture is Genetic Algorithm (GA).
Table 4. The RMSE calculation for training data

| Proportion Data (%) | Input lag | 1 | 2 | 3 | 4 | 5 | 6 | Mean |
|---------------------|-----------|---|---|---|---|---|---|------|
| 60/40               | 1, 4,     | 83.0615 | 83.2040 | 82.7621 | 82.7332 | 82.7485 | 83.6878 | 83.0615 |
|                     | 1, 4, 24  | 50.7471 | 51.0683 | 52.0102 | 51.2424 | 50.7859 | 50.9332 | 51.1315 |
|                     | 1, 4, 24, 35 | 49.7897 | 49.8980 | 50.4070 | 49.8547 | 49.7738 | 49.9046 | 49.9380 |
|                     | 1, 4, 24, 35, 41 | 50.2591 | 50.2885 | 50.6299 | 50.6138 | 50.1717 | 50.3924 | 50.4259 |
|                     | 1, 4, 24, 35, 41 | 51.3136 | 50.3887 | 50.5037 | 50.3203 | 50.3212 | 50.3156 | 50.5605 |
| 75/25               | 1, 4,     | 79.7703 | 79.8832 | 79.7542 | 80.1917 | 79.7938 | 79.7823 | 79.8626 |
|                     | 1, 4, 24  | 51.1134 | 51.1823 | 51.0998 | 51.4929 | 51.0442 | 51.1465 | 51.1799 |
|                     | 1, 4, 24, 35 | 51.5418 | 51.0918 | 51.6799 | 51.6799 | 51.1862 | 51.4248 | 51.2852 |
|                     | 1, 4, 24, 35, 41 | 50.5322 | 50.8267 | 50.9206 | 51.4888 | 51.0176 | 51.2196 | 51.0009 |
| 80/20               | 1, 4,     | 81.1076 | 80.2496 | 80.2000 | 80.5208 | 80.6684 | 80.4890 | 80.5392 |
|                     | 1, 4, 24  | 50.1853 | 50.1677 | 50.3001 | 50.1100 | 50.3545 | 50.0881 | 50.2009 |
|                     | 1, 4, 24, 35 | 49.9710 | 49.7669 | 49.2452 | 49.3842 | 49.6055 | 49.4947 | 49.5779 |
|                     | 1, 4, 24, 35, 41 | 50.1463 | 51.2497 | 50.0014 | 50.2515 | 49.9489 | 50.2662 | 50.3107 |
|                     | 1, 4, 24, 35, 41 | 50.1658 | 50.2283 | 50.0056 | 50.5086 | 49.5172 | 50.7553 | 50.1968 |

Table 5. The RMSE calculation for testing data

| Proportion Data (%) | Input lag | 1 | 2 | 3 | 4 | 5 | 6 | Mean |
|---------------------|-----------|---|---|---|---|---|---|------|
| 60/40               | 1, 4,     | 90.4090 | 88.5532 | 89.4976 | 89.9578 | 88.9239 | 89.1231 | 89.2441 |
|                     | 1, 4, 24  | 48.0836 | 48.1064 | 49.5705 | 48.4431 | 47.7480 | 47.9102 | 48.3103 |
|                     | 1, 4, 24, 35 | 130.715 | 71.2556 | 52.1241 | 58.8742 | 52.5040 | 85.6175 | 75.1818 |
|                     | 1, 4, 24, 35, 41 | 48.5862 | 48.1685 | 49.4368 | 49.5263 | 48.3406 | 49.2563 | 48.8858 |
|                     | 1, 4, 24, 35, 41 | 50.6120 | 49.6181 | 50.0247 | 49.5697 | 48.5294 | 49.6795 | 49.6722 |
| 75/25               | 1, 4,     | 91.6712 | 91.7527 | 91.5507 | 90.5950 | 92.9998 | 93.0976 | 92.0053 |
|                     | 1, 4, 24  | 44.6683 | 44.6748 | 44.7887 | 45.1825 | 45.1096 | 44.7001 | 44.8540 |
|                     | 1, 4, 24, 35 | 56.9901 | 46.7174 | 46.3789 | 51.5453 | 48.1772 | 51.4384 | 50.2079 |
|                     | 1, 4, 24, 35, 41 | 46.9995 | 45.2732 | 47.2624 | 45.5590 | 47.0755 | 45.0116 | 46.1969 |
|                     | 1, 4, 24, 35, 41 | 43.8500 | 44.6145 | 44.2441 | 45.7319 | 45.8017 | 45.4792 | 44.9536 |
| 80/20               | 1, 4,     | 92.7741 | 89.9011 | 88.4557 | 87.7543 | 87.8577 | 87.7922 | 89.0892 |
|                     | 1, 4, 24  | 47.7071 | 47.3690 | 47.7194 | 47.5781 | 47.8458 | 47.7335 | 47.6588 |
|                     | 1, 4, 24, 35 | 147.682 | 53.6581 | 135.989 | 68.3002 | 100.815 | 69.8567 | 96.1000 |
|                     | 1, 4, 24, 35 | 48.9167 | 52.6221 | 48.8217 | 48.8527 | 48.5632 | 49.3272 | 49.5173 |
|                     | 1, 4, 24, 35, 41 | 49.8089 | 48.7138 | 48.1701 | 50.7810 | 49.7593 | 51.2944 | 49.4546 |

Table 5 shows that the lowest RMSE in the testing data is occurred when the input layer consists of $x_{t-1}$ and $x_{t-4}$ lags. It means that the best FFNN architecture consists of 2 input lags as be shown in Figure 2. Figure 3 shows that the result of GA iteration process in the training data is convergent after 3000th generation was reached. This condition is indicated by the fitness value which tends to constant. The minimum fitness value after six times running the model was 0.0356603. In addition, the average of fitness value is 0.0357054. Based on this best solution, 21 weights are obtained to be used in the prediction model. The estimated value of optimum weights is presented in Table 6. The best prediction model for the daily stock price data of PT. Adhi Karya Tbk can be written as equation (5).

$$\hat{x}_t = 0.0526 + \frac{0.06424}{1 + \exp(-(-0.2590 + 0.9999x_{t-1} + 0.0649x_{t-4}))} + \frac{0.6744}{1 + \exp(-(-0.1730 + 0.9963x_{t-1} + 0.2958x_{t-4}))} + \frac{0.9143}{1 + \exp(-(-0.2992 + 0.9991x_{t-1} + 0.0338x_{t-4}))} + \frac{0.5605}{1 + \exp(-(-0.0982 + 0.9999x_{t-1} + 0.0649x_{t-4}))}$$
Furthermore, equation (5) is used to make prediction so that the accuracy of prediction model can be tested. Figure 4 constitutes a visual comparison between targets and network output, either in the training or testing data. Meanwhile, MAPE criterion is used to perform formally comparison.

\[
\frac{-0.5522}{1 + \exp\left(-\left(0.8351 - 0.9990x_{t-1} - 0.1876x_{t-4}\right)\right)} + \frac{-0.8843}{1 + \exp\left(-\left(0.9256 - 0.9977x_{t-1} - 0.0627x_{t-4}\right)\right)}
\]

(5)

**Figure 2.** The best architecture of model

**Figure 3.** Plot performance of GA for two-inputs architecture
Table 6. The estimation value of network weights

| $\hat{w}_{bj}$ | $\hat{w}_{ij}$ | $\hat{w}_{bo}$ | $\hat{w}_{jo}$ |
|----------------|----------------|----------------|----------------|
| $\hat{w}_{b1} = -0.2590$ | $\hat{w}_{11} = 0.9999$ | $\hat{w}_{21} = 0.0649$ | $\hat{w}_{2j} = 0.0526$ |
| $\hat{w}_{b2} = -0.1730$ | $\hat{w}_{12} = 0.9963$ | $\hat{w}_{22} = 0.2958$ | $\hat{w}_{2j} = 0.6744$ |
| $\hat{w}_{b3} = -0.2992$ | $\hat{w}_{13} = 0.9991$ | $\hat{w}_{23} = 0.0338$ | $\hat{w}_{2j} = 0.9143$ |
| $\hat{w}_{b4} = 0.8351$ | $\hat{w}_{14} = -0.9990$ | $\hat{w}_{24} = -0.1876$ | $\hat{w}_{2j} = -0.5522$ |
| $\hat{w}_{b5} = 0.9256$ | $\hat{w}_{15} = -0.9977$ | $\hat{w}_{25} = -0.0627$ | $\hat{w}_{2j} = -0.8843$ |

Figure 3. The comparison of target and output

5. Conclusion

The weights estimation of the FFNN model using GA optimization method is able to produce a predictive model that minimizes error. The prediction model obtained is able to produce output that approaches and follows the target pattern well. The best model obtained is in the very good category to be used as a prediction model because the MAPE training and testing values are less than 10%. The average MAPE value of six times repetition is 1.5714% for training and 1.5501% for testing. GA has strength in the stability of estimation results even though the initialization of weights is done randomly. The stability of the estimation results is indicated by the error variance of the repetition process which is practically small enough to provide a consistent error calculation result.

References

[1] Beasley J E and Chu P C 1996 European Journal of Operational Research 392.
[2] Chen R J C, Peter B and Frederick W C 2007 Journal of Hospitality & Tourism Research.
[3] Denuth H and Beale M 2002 Neural Network Toolbox: For Use with MATLAB (Natick M A: The MathWorks, Inc.).
[4] Frechtling D D 2001 Forecasting Tourism Demand: Methods and Strategies (Oxford: Elsevier).
[5] Haykin S 1994 Neural Networks: A Comprehensive Foundation. (New York: Macmillan Publishing Company).
[6] Makridakis S, Wheelwright S C and Hyndman R J 1998 Forecasting: Methods and Applications Third edition (New York: John Wiley and Sons).
[7] Montana D J and Davis L 1993 Training Feedforward Neural Networks Using Genetic Algorithms (10 Mouton St. Cambridge, MA: BBN Systems and Technologies Corp.).
[8] Pearl J 1983 Heuristics: Intelligent Search Strategies for Computer Problem Solving (New York: Addison-Wesley).
[9] Russell S J and Norvig P Norvig 2003 Artificial Intelligence: A Modern Approach 2nd Edition, (New York: Prentice-Hall International Inc.).
[10] Samarasinghe S 2006 Neural Network for Applied Science and Engineering (New York: Auerbach Pub.).
[11] Wei W W S 2006 Time Series Analysis: Univariate and Multivariate Methods 2nd Ed. (Canada: Addison-Wesley Publishing Company Inc.).