



AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com



## Multivariate Inputs on a MIMO Neuro-Fuzzy structure with LMA training. A study case: Indonesian Banking Stock Market

F. Pasila, R. Lim, M. Santoso

Electrical Engineering Department, Petra Christian University, Surabaya, 60236, INDONESIA.

### ARTICLE INFO

#### Article history:

Received 20 November 2013

Received in revised form 24

January 2014

Accepted 29 January 2014

Available online 5 April 2014

#### Key words:

MIMO neuro-fuzzy; accelerated  
Levenberg-Marquardt algorithm;  
multivariate inputs, fundamental  
quantities; technical indicator;

### ABSTRACT

The paper describes the design and implementation of the multivariate inputs of multi-input-multi-output neuro-fuzzy with Levenberg-Marquardt algorithm training (MIMO neuro-fuzzy with accelerated LMA) to forecast stock market of Indonesian Banking. The accelerated LMA is efficient in the sense that it can bring the performance index of the network, such as the root mean squared error (RMSE), down to the desired error goal, more efficiently than the standard Levenberg-Marquardt algorithm. The MIMO neuro-fuzzy method is a hybrid intelligent system which combines the human-like reasoning style of fuzzy systems with the learning ability of neural nets. The main advantages of a MIMO neuro-fuzzy system are: it interprets IF-THEN rules from input-output relations and focuses on accuracy of the output network and offers efficient time consumption for on-line computation. The proposed architectures of this paper are a MIMO-neuro-fuzzy structure with multivariate input such as fundamental quantities as inputs network (High, Low, Open and Close) and a MIMO-neuro-fuzzy structure with other multivariate inputs, which is a combination inputs between two fundamental quantities (High and Low) and two inputs from technical indicator Exponential Moving Average (EMA High and EMA Low). Both proposed learning procedures, which are using accelerated LMA with optimal training parameters with at least one million iterations with different 16 membership functions, employ 12% of the input-output correspondences from the known input-output dataset. For experimental database, both structures are trained using the seven-year period (training data from 2 Oct 2006 to 28 Sept 2012) and tested using two-weeks period of the stock price index (prediction data from 1 Oct 2012 to 16 Oct 2012) and the proposed models are evaluated with a performance indicator, root mean squared error (RMSE) for mid-term forecasting application. The simulation results show that the MIMO-neuro-fuzzy structure with combination of fundamental quantities and technical indicators has better performance (RMSE) for two-weeks forecast.

© 2014 AENSI Publisher All rights reserved.

**To Cite This Article:** F. Pasila, R. Lim, M. Santoso, Multivariate Inputs on a MIMO Neuro-Fuzzy structure with LMA training. A study case: Indonesian Banking Stock Market. *Aust. J. Basic & Appl. Sci.*, 8(4): 476-481, 2014

### INTRODUCTION

As the arising of artificial intelligence algorithms in recent years, many researchers have applied soft computing algorithms in time-series model for financial forecasting. Some examples of the time-series forecasts can be seen below: Research of Kimoto, Asakawa, Yoda, and Takeoka(1990) developed a prediction system for stock market by using neural network; Nikolopoulos and Fellrath(1994) have combined genetic algorithms (GAs) and neural network (NN) to develop a hybrid expert system for investment decisions; Kim and Han (2000) proposed an approach based on genetic algorithms to feature discretization and the determination of connection weights for artificial neural networks (ANNs) to forecast the stock price index; Huarng and Yu (2006) applied a backpropagation neural network to establish fuzzy relationships in fuzzy time series for forecasting stock price; Moreover, Roh(2007) has combined neural network and time series model for forecasting the volatility of stock market index. In addition, another forecast strategy is done by Wei (2011), by proposing an adaptive-network-based fuzzy inference system model which uses multi-technical indicators and fundamental quantities, to predict stock price trends.

Even though most of the proposed solution schemes are formally very elegant and quite effective in forecasting the stock index, the problem of limited forecast (short-term forecasting) still remain.

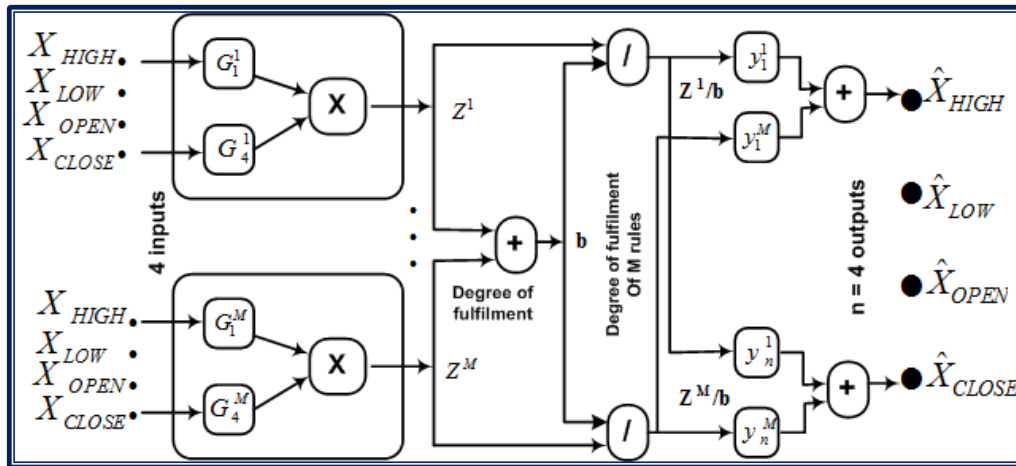
This paper proposes a Neuro-Fuzzy forecasting model for mid-term forecasting of the Indonesian Banking stock price. The two proposed model utilizes raw fundamental stock value (Open, High, Low, Close price) and combination of fundamental and technical indicator (High, Low, EMA\_High and EMA\_Low) as input factors of

**Corresponding Author:** F. Pasila, Electrical Engineering Department, Petra Christian University, Surabaya 60236, INDONESIA.  
E-mail: felix@petra.ac.id

the neuro-fuzzy type MIMO that can offer forecast and trend of stock market for two weeks. To test the performance of both proposed model, this paper employs a seven-year period of the several banking stock market such as BBKA, BBNI, BBRI and BMRI as experimental dataset.

**A MIMO Neuro-Fuzzy System For Modeling And Forecasting:**

A neuro-fuzzy network with an improved training algorithm for MIMO case was developed by Palit and Popovic(2005) and used for forecasting of electrical load data.



**Fig. 1:** The multivariate Architecture of the MIMO – feedforward Neuro-Fuzzy (NF) Network, type Takagi-Sugeno (TS)

The first proposed model of NF network as shown in Fig. 1 is based on Gaussian functions. It uses TS-fuzzy rule, product inference, and weighted average defuzzification. The nodes in the first layer compute the degree of membership of the fundamental input values (High, Low, Open and Close) in the antecedent fuzzy sets. The product nodes (x) denote the antecedent conjunction operator and the output of this node is the corresponding to degree of fulfillment or firing strength of the rule. The division (/), together with summation nodes (+), join to make the normalized degree of fulfillment ( $z^l / b$ ) of the corresponding rules, which after multiplication with the corresponding TS rule consequent ( $y_j^l$ ), is used as input to the last summation part (+) at the defuzzified output value, which, being crisp, is directly compatible with the next forecast data (High, Low, Open and Close).

The second model is also based on Fig. 1 but with combination between fundamental quantities and technical indicator in the input and output sides (multivariate inputs). The input  $X_{OPEN}$  and  $X_{CLOSE}$  will be replaced by input  $X_{EMA\_HIGH}$  and  $X_{EMA\_LOW}$ . The same changes also should be done in the output side.

**Neural Network Representation Of Fuzzy System (FS):**

Neuro-fuzzy representation of the FS is based on inference TS-type which has been explained clearly by Palit (p.153, p.233). There are two important steps in this representation: calculating of the degree of fulfillment and normalized degree of fulfillment. The FS considered here for constructing neuro-fuzzy structures is based on TS-type fuzzy model with Gaussian membership functions. It uses product inference rules and a weighted average defuzzifier defines as:

The corresponding  $l^{th}$  rule from the above FS can be written as

$$R_l = \text{If } x_1 \text{ is } G_1^l \text{ and } x_2 \text{ is } G_2^l \text{ and } \dots x_m \text{ is } G_m^l$$

$$\text{then } y_j^l = W_{oj}^l + W_{1j}^l x_1 + \dots + W_{mj}^l x_m \tag{1}$$

Where,  $x_i$  with  $i=1,2,\dots,m$ ; and  $f_j$  with  $j=1,2,\dots,n$ ; are the  $m$  system inputs and  $n$  system outputs, and  $G_i^l$  with  $i=1,2,\dots,m$ ; and  $l=1,2,\dots,M$ ; are the Gaussian membership functions of the form (1) with the corresponding mean and variance parameters  $c_i^l$  and  $\tau_i^l$  respectively and with  $y_j^l$  as the output consequent of the  $l^{th}$  rule.

Moreover, FS represents as a three layer MIMO feedforward network as shown in Fig. 1. Because of the implementation of the Takagi-Sugeno-type FS, this figure represents a Takagi-Sugeno-type of MIMO neuro-fuzzy network, where instead of the connection weights and the biases in training algorithm. We introduce

the mean  $c_i^l$  and the variance  $\tau_i^l$  parameters of Gaussian functions, along with  $W_{oj}^l, W_{ij}^l$  parameters from the rules consequent, as the equivalent adjustable parameters of TS-type network. If all the parameters for NF network are properly selected, then the FS can correctly approximate any nonlinear system based on given related data between four inputs and four outputs.

$$F_j = \sum_{i=1}^M y_j^l \cdot H^l \quad (2)$$

$$y_j^l = W_{0j}^l + W_{1j}^l x_1 + W_{2j}^l x_2 + \dots + W_{mj}^l x_m \quad (3)$$

$$H^l = (z^l / b), \text{ and } b = \sum_{i=1}^M Z^l \quad (4)$$

$$Z^l = \prod_{i=1}^m \exp \left( - \left( \frac{x_i - c_i^l}{\sigma_i^l} \right)^2 \right) \quad (5)$$

Prior to their use, NFTS and NFLUT models require the tuning of the parameters  $c_j^n, \sigma_j^n, w_{0i}^n, w_{ji}^n$  (for  $j = 1, 4; i = 1, 4$ ). Here, the number of parameters for the considered MPRs is 252 parameters. The values of these parameters are found by an optimized learning procedure. The learning procedure employs 12% of the input-output *XIO* correspondences from the known dataset.

#### Accelerated Levenberg-Marquardt Algorithm (LMA):

We introduce a function  $v(w)$  is meant to minimize with respect to the parameter vector  $w$  using Newton's method, the update of parameter vector  $w$  is defined as:

$$\Delta w = -[\nabla^2 V(w)]^{-1} \cdot \nabla V(w) \quad (6a)$$

$$w(k+1) = w(k) + \Delta w \quad (6b)$$

From equation (6a),  $\nabla^2 V(w)$  is the Hessian matrix and  $\nabla V(w)$  is the gradient of function  $v(w)$ . If the  $v(w)$  is taken to be SSE function as follows:

$$V(w) = 0.5 \cdot \sum_{r=1}^N e_r^2(w) \quad (7)$$

Then the gradient of  $v(w)$  and the hessian matrix of  $\nabla^2 V(w)$  are generally defined as:

$$\nabla V(w) = J^T(w) \cdot e_r(w) \quad (8a)$$

$$\nabla^2 V(w) = J^T(w) \cdot J(w) + \sum_{r=1}^N e_r(w) \cdot \nabla^2 e_r(w) \quad (8b)$$

where  $J(w)$  is the jacobian matrix, written as follows

$$J(w) = \begin{bmatrix} \frac{\partial e_1(w)}{\partial w_1} & \frac{\partial e_1(w)}{\partial w_2} & \dots & \frac{\partial e_1(w)}{\partial w_{N_A}} \\ \frac{\partial e_2(w)}{\partial w_1} & \frac{\partial e_2(w)}{\partial w_2} & \dots & \frac{\partial e_2(w)}{\partial w_{N_A}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(w)}{\partial w_1} & \frac{\partial e_N(w)}{\partial w_2} & \dots & \frac{\partial e_N(w)}{\partial w_{N_A}} \end{bmatrix} \quad (8c)$$

From (8c), it is seen that the dimension of the Jacobian matrix is  $(N \times N_A)$ , where  $N$  is the number of training models and  $N_A$  is the number of adjustable parameters in the network. For the Gauss-Newton method, the second term in (8b) is assumed to be zero. Consequently, the update equations according to (6a) will be:

$$\Delta w = -[J^T(w) \cdot J(w)]^{-1} \cdot J^T(w) \cdot e_r(w) \quad (9a)$$

Now let us see the LMA modifications of the Gauss-Newton method.

$$\Delta w = -[J^T(w) \cdot J(w) + \alpha \cdot I]^{-1} \cdot J^T(w) \cdot e_r(w) \quad (9b)$$

where dimension of  $I$  is the  $(N_A \times N_A)$  identity matrix, and the parameter  $\alpha$  is multiplied or divided by some factor whenever the iteration steps increase or decrease the value of  $v(w)$ .

Here, the updated equation according to (6a)

$$w(k+1) = w(k) - [J^T(w) \cdot J(w) + \alpha \cdot I]^{-1} \cdot J^T(w) \cdot e_r(w) \quad (9c)$$

This is important to know that for large  $\alpha$ , the algorithm becomes the steepest descent algorithm with step size  $1/\alpha$ , and for small  $\alpha$ , it becomes the Gauss-Newton method.

Now, comes to the computation of Jacobian Matrices. The gradient  $\nabla V(w_{0j}^l) \equiv SSE$  can be written as

$$\nabla V(w_{0j}^l) \equiv (\partial S / \partial w_{0j}^l) = \{Z^l / b\} \cdot (X_j - d_j) \quad (10)$$

Where  $X_j$  and  $d_j$  are the actual output of the Takagi-Sugeno type MIMO and the corresponding desired output from matrix input-output training data. And then by comparing (8) to (9a), where the gradient  $\nabla V(w)$  is expressed with the transpose of the Jacobian matrix multiplied with the network's error vector,

$$\nabla V(w) = J^T(w) \cdot e_r(w) \quad (11)$$

then the Jacobian matrix, and the transpose of Jacobian matrix for the parameter  $W_{0j}^l$ ,  $W_{ij}^l$ ,  $c_i^l$  and  $\tau_i^l$  of the NF network can be written by

$$J^T(w_{0j}^l) = (Z^l / b) \quad (12a)$$

$$J(w_{0j}^l) \equiv [J^T(w_{0j}^l)]^T = [Z^l / b]^T \quad (12b)$$

$$J^T(w_{ij}^l) = (Z^l / b) \cdot x_i \quad (13a)$$

$$J(w_{ij}^l) \equiv [J^T(w_{ij}^l)]^T = [(Z^l / b) \cdot x_i]^T \quad (13b)$$

$$J^T(c_i^l) = \{2 \cdot D_{eqv} \cdot Z^l \cdot (x_i - c_i^l) / (\sigma_i^l)^2\} \quad (14a)$$

$$J(c_i^l) = [J^T(c_i^l)]^T = [2 \cdot D_{eqv} \cdot Z^l \cdot (x_i - c_i^l) / (\sigma_i^l)^2]^T \quad (14b)$$

$$J^T(\sigma_i^l) = \{2 \cdot D_{eqv} \cdot Z^l \cdot (x_i - c_i^l)^2 / (\sigma_i^l)^3\} \quad (15a)$$

$$J(\sigma_i^l) = [J^T(\sigma_i^l)]^T = [2 \cdot D_{eqv} \cdot Z^l \cdot (x_i - c_i^l)^2 / (\sigma_i^l)^3]^T \quad (15b)$$

Where

$$D_{eqv} = ((f_1 - X_1) + (f_2 - X_2) \cdot e_2 + \dots + (f_n - X_n) \cdot e_n) \cdot (e_{eqv}^p)^{-1} \quad (16)$$

isa matrix form using pseudo inverse and

$$e_{eqv}^p = \sqrt{(e_1^{p^2} + e_2^{p^2} + \dots + e_n^{p^2})} \quad (17)$$

is sum square error with  $p = 1, 2, 3, \dots, N$  training samples that explained in Palit (2005) clearly. After finishing all computation in the (12-15), then back to the Eq. (6.b) for updating the four matrix parameters. The updating procedure will stop after achieving the maximum iteration or founding the minimum error function.

#### Proposed Input-Output Relation For Nf Model:

In case of mid-term forecasting model, a MIMO system will be used for training and forecasting. The first case uses 4 inputs and 4 outputs. The inputs for training and forecasting are previous and the latest daily-data of HIGH, LOW, OPEN and CLOSE of the stock market respectively, while the prediction or the output in the training and testing are the next data (with the interval data equals to one). Similar to the first case, the second case uses inputs and 4 outputs. The input data of the second model are HIGH, LOW, EMA HIGH and EMA LOW. The term EMA here is an Exponential Moving Average with ten times of period for each EMA HIGH and EMA LOW respectively. The related matrix between inputs and outputs of the MIMO Neuro-Fuzzy predictor of the first case should be arranged on XIO matrix, as shown below:

$$XIO_I = \begin{bmatrix} Day_H & Day_L & Day_O & Day_C & \rightarrow & \hat{Day}_{H+1} & \hat{Day}_{L+1} & \hat{Day}_{O+1} & \hat{Day}_{C+1} \\ \hat{Day}_{H+1} & \hat{Day}_{L+1} & \hat{Day}_{O+1} & \hat{Day}_{C+1} & \rightarrow & \hat{Day}_{H+2} & \hat{Day}_{L+2} & \hat{Day}_{O+2} & \hat{Day}_{C+2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \hat{Day}_{H+9} & \hat{Day}_{L+9} & \hat{Day}_{O+9} & \hat{Day}_{C+9} & \rightarrow & \hat{Day}_{H+10} & \hat{Day}_{L+10} & \hat{Day}_{O+10} & \hat{Day}_{C+10} \end{bmatrix} \quad (18a)$$

$$XIO_{II} = \begin{bmatrix} Day_H & Day_L & EMA_H & EMA_L & \rightarrow & Day_{H+1} & Day_{L+1} & EMA_{H+1} & EMA_{L+1} \\ Day_{H+1} & Day_{L+1} & EMA_{H+1} & EMA_{L+1} & \rightarrow & Day_{H+2} & Day_{L+2} & EMA_{H+2} & EMA_{L+2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Day_{H+9} & Day_{L+9} & EMA_{H+9} & EMA_{L+9} & \rightarrow & Day_{H+10} & Day_{L+10} & EMA_{H+10} & EMA_{L+10} \end{bmatrix} \tag{18b}$$

As shown in equation (18a), 4 input days (the last training data for inputs) are trained to produce 4 output days in NF network. Each input and output represents one daily set of data. Output from the first forecast  $Day_{H+1} \hat{Day}_{L+1} \hat{Day}_{O+1} \hat{Day}_{C+1}$  will replace the 2<sup>nd</sup> row of  $XIO_I$  as input for second prediction. After 10 loops of forecasting, all ten outputs (in the right side) are output prediction of the next day until the next ten days of the related stock market. The same procedure above is done for the second case, with the matrix  $XIO_{II}$  shown in (18b). So far, both matrix  $XIO$  are found optimal in the experimental procedures using optimal training parameters with at least one million iterations with different 16 membership functions (from M = 5 until M = 15).

**RESULT AND DISCUSSION**

**Table 1:** Mid-Term Performance of NF Case I and II for BBKA (Data Training 2 Oct 2006 – 28 Sept 2012)

No.	Date	Actual Data High Price	Prediction High Case I	Prediction High Case II	Error Prediction Case I (abs %)	Error Prediction Case II (abs %)
1	1 Oct 2012	7.950	7.945,8	7.958,2	0.05	0.01
2	2 Oct 2012	7.950	7.950,8	7.970,1	0.01	0.25
3	3 Oct 2012	7.950	7.962,8	7.974,2	0.16	0.30
4	4 Oct 2012	7.950	7.977,3	7.977,0	0.34	0,34
5	5 Oct 2012	7.950	7.992,7	8.013,8	0.54	0.80
6	8 Oct 2012	7.950	8.008,9	8.009,9	0.74	0.75
7	10 Oct 2012	8.000	8.025,8	8.036,0	0.32	0.45
8	12 Oct 2012	8.150	8.043,4	8.024,1	1.31	1.55
9	15 Oct 2012	8.100	8.061,6	8.063,7	0.47	0.45
10	16 Oct 2012	8.150	8.080,6	8.083,2	0.85	0.82
Average Error					0.48	0.58

**Table 2:** Average Mid-Term Performance of NF Case I and II for Indonesian Banking Stock Market (Data Training 2 Oct 2006 – 28 Sept 2012, Prediction 1 Oct 2012 – 16 Oct 2012)

No.	Initial Stock Market	Error Training RMSE Case I	Error Training RMSE Case II	Error Prediction LOW Case I (%)	Error Prediction LOW Case II (%)	Error Prediction HIGH Case I (%)	Error Prediction HIGH Case II (%)
1	BBKA	0.0083	0.0090	0.69	0.78	0.48	0.58
2	BBNI	0.0091	0.0069	0.71	0.54	0.59	0.45
3	BBRI	0.0153	0.0131	2.46	2.07	0.69	0.61
4	BMRI	0.0159	0.0135	2.63	2.28	0.88	0.76
Average Error		0.0122	0.0106	1.62	1.41	0.66	0.60

In case of *mid-term* forecasting model, a MIMO system proposed two model of NF which  $XIO$  matrix are written in (18a) and (18b). Table I shows the two-weeks daily High-price forecast of BBKA (Bank BCA) index with the prediction and actual data of High price and also error prediction for both cases. According to this Table, the performance result from first case in average is better than the second case. Compare to other stock indexes, such as BBNI (BANK BNI), BBRI (Bank BRI) and BMRI (Bank MANDIRI), the results of error performances of each stock show that in general, the second case is better than the first one. As presented in Table 2, the second case, which use combination fundamental quantities and technical indicators as input network has less average error compare to the first case (with input fundamental quantities only).

There are two calculations of error that presented in Table 2. First, calculating the RMSE train of each case for 7 years (year 2006 to year 2012). The RMSE is the root mean squared error of all output models. This is done in the training mode, like shown in the column 3 and 4, and it is calculated using RMSE equation (19):

$$RMSE_{train} = \sqrt{\frac{\sum_{r=1}^{N_{Train}} e_r^2}{N_{Train}}} \tag{19}$$

where  $e_r$  and  $N_{Train}$  are the error training and the number of data training respectively. Second, calculating the average error of LOW and HIGH of the stock price forecast. These values are determined in the testing mode, as presented in the column 5 to 8. The average errors are calculated by comparing the actual and prediction data of LOW and HIGH price for two-weeks forecast. For additional information, the presented error in these Tables is not the real error in the stock market trading. The real prediction value of each stock index must be changed to

the value that closed to the index price interval (index price interval always use the basic interval value such as 10, 25, 50, 100, etc.).

### **Conclusion:**

As conclusion, this paper presented: 1) two forecast models using NF structures type MIMO with different inputs (multivariate inputs) and outputs; 2) training method based on Levenberg-Marquardt training Algorithm (LMA); 3) four stock markets of Indonesian Banking which are BBKA, BBNI, BBRI and BMRI with 7 years data training and two weeks data testing. Comparison between the proposed NF models highlighted that: 1) the proposed NF models are both suitable for forecasting the stock market Indonesia Banking (two weeks in advanced); 2) the prediction of average error of HIGH price for all indexes have better accuracy; 3) the prediction of second case, both in LOW and HIGH price, has lower error prediction compared to the first case. It can be inferred that second case of NF structure, which combining the fundamental quantities (HIGH and LOW) with technical indicator inputs (EMA\_HIGH and EMA\_LOW), has better accuracy for mid-term forecasting of Indonesian Banking stock market compared to the previous case.

### **REFERENCES**

- Kimoto, T., K. Asakawa, M. Yoda and M. Takeoka, 1990. Stock market prediction system with modular neural network. Proceedings of the international joint conference on neural networks, San Diego, California, pp: 1-6.
- Nikolopoulos, C. and P. Fellrath, 1994. A hybrid expert system for investment advising. *Expert Systems*, 11(4): pp: 245-250
- Kim, K.J. and I. Han, 2000. Genetic algorithms approach to feature discretization in artificial neural networks for prediction of stock index. *Expert System with Application*, 19: 125-132.
- Huang, K.H. and T.H.K. Yu, 2006. The application of neural networks to forecast fuzzytime series. *Physica A*, 363(2): 481-491
- Roh, T.H., 2007. Forecasting the volatility of stock price index. *Journal Expert Systems with Applications*, 33(4): 916-922.
- Wei, L.Y., T.L. Chen and T.H. Ho, 2011. A hybrid model based on adaptive-network-based fuzzy inference system to forecast Taiwan stock market”, *Expert Syst.*, 38(11): 13625-13631.
- Palit, A.K. and D. Popovic, 2005. *Computational Intelligence in Time Series Forecasting, Theory and Engineering Applications*. Springer-Verlag, London, pp: 230-246.