International Journal of Information Technology and Knowledge Management
July-December 2008, Volume 1, No. 2, pp. 179-190

A HYBRID OF ARTIFICIAL NEURAL NETWORKS AND SARIMA MODELS FOR LOAD FORECASTING

Norizan Mohamed, Maizah Hura Ahmad,
Zuhaimy Ismail, & Khairil Anuar Arshad

Abstract

A Seasonal Autoregressive Integrated Moving Average (SARIMA) is a popular and well known model for forecasting seasonal time series data. Recently, we applied artificial neural networks (ANNs) in time series forecasting, but in seasonal series, artificial neural networks (ANNs) did not perform well. In this paper, a hybrid forecasting model, which combines the seasonal time series ARIMA (SARIMA) and the multilayer feedforward neural network is proposed to forecast time series with seasonality. The forecasting performances among these three models, i.e., the SARIMA model, the multilayer feedforward neural network model and the hybrid model are compared. Comparing the performances using the root mean squared error (RMSE), the mean absolute error (MAE) and mean absolute percentage error (MAPE), we find that the hybrid model outperforms both two single models.

Keywords: Load Forecasting; Seasonal Autoregressive Integrated Moving Average; Artificial Neural Networks; Multilayer Feed-forward Neural Network; Hybrid Model.

1. Introduction

Time series forecasting is an important area of forecasting in which past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. The model is then used to extrapolate the time series into the future. This modeling approach is particularly useful when little knowledge is available on the underlying data generating process or when there is no satisfactory explanatory model that relates the prediction variable to other explanatory variables.

One of the most important and widely used time series models is the autoregressive integrated moving average (ARIMA) model. An ARIMA process combines three different processes comprising an Autoregressive (AR) functions regressed on the past values of the process, moving average (MA) functions regressed on a purely random process with mean zero and variance σ^2 [5] and an integrated (I) part to make the data series stationary by differencing [5,16, 21].

The ARIMA model is extended in order to handle seasonal aspects of time series, and the general notation is $ARIMA(p, d, q)\ (P, D, Q)_s$ where (p, d, q) is the non-seasonal part of the model, (P, D, Q) is the seasonal part of the model and s is the seasonal length with abbreviated as seasonal ARIMA, SARIMA [13]. Seasonal ARIMA model building requires the specification of differencing orders (d, D) and the orders of both non-seasonal and seasonal autoregressive (AR) and moving average (MA) operators (p, q, P, Q) as well as the estimation of model parameters in the AR and MA (p, q) operator polynomials [21]. Seasonality is a periodic fluctuations of constant length caused by factors such as temperature, rainfall, month of the year, timing of holidays and corporate policies [13]. The accuracy of forecasting is very important, hence a time series need to be stationary before modeling and forecasting process take place [21]. Differencing method is one way to remove non-stationarity of seasonal and trend time series [13].

Recently, artificial neural networks (ANNs) have been extensively studied and received increasing attention in time series forecasting. ANN are composed of many nodes that operate in parallel, and communicate with each other through connecting synapses. The greatest advantage of a neural network is its ability to model a nonlinear process without priori assumptions on the nature of the process [2,11].

Much effort has been done to develop and improve time series forecasting. Lately a combination of two methods is believed as a promising technique to improve the forecasting performance. Taking into consideration that the data series composed of the linear component and the nonlinear component, we present a hybrid model which combines the linear model and the non linear model.

The remainder of this paper is organized as follows. In section 2, we present the Box-Jenkins seasonal ARIMA model, the multilayer feedforward neural network model, the hybrid model and forecasting performance. In section 3, details of the results are discussed. Finally in section 4 we give our conclusions.

2. METHODOLOGY

2.1 Box-Jenkins Seasonal ARIMA (SARIMA) Model

In practice, many time series contain a seasonal periodic component, which repeats every s observation. To deal with seasonality, the ARIMA model is generalized hence a general multiplicative seasonal ARIMA (SARIMA) model is defined [1] which follows the ARIMA general procedure represented as follows:

$$\varphi p\,(B)\Phi P\,(B^s\,)W_t = \theta q\,(B)\Theta Q\,(B^s)a_t \quad (1)$$

with

$$\phi_p\,(B) = 1\ \phi_1\,B - \phi_2\,B^2 - \ldots - \phi_p\,B^p$$
$$\Phi_P\,(B) = 1 - \Phi_s\,B^s - \Phi_{2s}\,B^{2s} - \ldots - \Phi_{Ps}\,B^{Ps},$$
$$\theta_q\,(B) = 1 - \theta_1\,B - \theta_2\,B^2 - \ldots - \theta_q B^q,$$
$$\Theta_Q\,(B) = 1 - \Theta_1\,B^s - \Theta_2 B^{2s} - \ldots - \Theta_{Qs}\,B^{Qs}$$

where B denotes the backward shift operator, a_t denotes a purely random process and

$$W_t = (1 - B)^d (1 - B^s)^D X_t \tag{2}$$

denotes the differenced series. $(1 - B)^d$ and $(1 - B^s)^D$ are the seasonal and nonseasonal differencing operators, respectively [13]. If the integer D is not zero, then seasonal differencing is involved. The above model is called a SARIMA model of order $(p, d, q)(P, D, Q)_s$. The differenced series W_t is formed from the original series X_t by appropriate differencing to remove non-stationary terms. If d is non-zero, then there is a simple differencing to remove trend, while seasonal differencing $(1 - B^s)^D$ may be used to remove seasonality. In practice, the values of d and D are usually zero or one and rarely two. In practice D is rarely more than one and P and Q are typically less than three [3].

SARIMA modeling procedure involves an iterative five-stage process:

a) Preparation of data including transformations and differencing;
b) Identification of the *SARIMA* $(p, d, q)(P, D, Q)_s$ structure;
c) Estimation of the unknown parameters;
d) Checking the adequacy of fitted model by performing ACF and PACF on model residuals;
e) Forecast future outcome based on the known data.

2.2 Artificial Neural Networks

Neural networks are essentially a nonlinear modeling approach that provides a fairly accurate universal approximation to any functions. The flexibility of its mathematical structure is capable in identifying the complex and non-linear relationships between inputs and outputs [9]. Appropriate neural network architectures can be trained to predict the future values of the dependent variables [10]. Predefined network architecture such as connectivity and node transfer functions, and selection of training algorithm such as back-propagation to learn the weights and biases, are applied in most applications [9]. If the network paradigm and parameters are appropriately designed, these can result in satisfactory performance[12].

A well known neural model is the multi-layer feed-forward neural network (MFNN), which consists of an input layer, one or several hidden layers and an output layer. The neurons in the feed-forward neural network, are generally grouped into layers. Signals flow from the input layer to the output layer via unidirectional connections, the neurons being connected from one layer to the next, but not within the same layer [14]. The multi-layer feed-forward neural network (MFNN) represents the performance for nonlinear prediction of time series [7, 8].

2.2.1 Multilayer Feedforward Neural Network

An essential factor of successes of the neural networks depends on the training network. Among the several learning algorithms available, back-propagation has been the most popular and most widely implemented learning algorithm of all neural networks paradigms [2,4]. Among the advantages of back-propagation (BP) is its ability to store numbers of patterns that exceed its built- in vector dimensionality [2].

Basically, the BP training algorithm with three-layer feed-forward architecture means that, the network has an input layer, one hidden layer and an output layer. Single hidden layer feedforward network is the most widely used for neural network forecasting [22]. More hidden layer can be used but three layers are sufficient to enable this type of network to model any deterministic process within reasonable limit [19].

Neural network function sends the vector $(x_1, \ldots, X_N)$ in R^N to the vector $(y_1, \ldots, y_M)$ in R^M. Thus, the feedforward network can be represented as [19]:

$$y + F(x) \tag{3}$$

where $x = (x_1, \ldots, x_N)$ $x = (x_1, \ldots, x_N)$ and $y = (y_1, \ldots, y_M)$. The action of this function is determined in a specific way. For a network with N input nodes, H hidden layer nodes and M output nodes, the values y_k are given by:

$$y_k = g\left(\sum_{j=1}^{H} w^{\circ}{}_{jk} h_j\right), k = 1, \cdots, M \tag{4}$$

Here $w^{\circ}{}_{jk}$ is an output "weight" from hidden node j to output node k, and g is a function. The values of the hidden layer nodes h_j $j = 1, \ldots, H$ are given by:

$$h_j = \sigma\left(\sum_{i=1}^{n} w_{ij}^{1} x_i + w_j^T\right), \quad j = 1, \cdots, H \tag{5}$$

Here, w_{ij}^{1} is the input "weight" from input node i to hidden node j, w_j^T, is a threshold "weight" from an input node which has the constant value 1, to hidden node j, x_i is the value at input node i and σ is called "sigmoid" function given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

The function σ is called the "activation function" of the neural network. The function g may be the same as activation function or may be a different function. The action of the feedforward network is determined by the architecture and the values of the weights.

The network architecture are composed of the numbers of input, hidden and output nodes. The numbers of input and output nodes are determined by the application and in fact, fixed. The number of hidden nodes, however is adjustable and usually estimated by a trial and error approach [6,19]. The weights, in network are adjusted by comparing the actual response with the target response in such a way to minimize the error. The process of adjusting these weight values in order to obtain a desired network performance is known as "training" the network. The training process requires a set of examples of proper network inputs and target outputs. During the training, the weights and biases of the network are iteratively adjusted to minimize the network performance function. The network is said to "learn" as the weight values are being modified to achieve the training goal. The weights are a set of parameters, that determine the behavior of a particular function.

Neural Network modeling procedure involves an iterative five-stage process:

(a) Normalize the three partition data;

(b) Decide the architecture and parameter;

(c) Initialize weights and biases randomly and save for next training process;

(d) Choose the network with minimum error;

(e) Forecast future outcome.

2.3 Hybrid Model

Seasonal ARIMA (SARIMA) and artificial neural network models have been extensively studied and have been used as time series forecasting method. The seasonal ARIMA model has been shown to give good forecast for seasonal time series especially for short-term periods with a large amount (at least 50 and preferably 100 or more) of historical data [17], where as the neural networks have shown their ability to model a nonlinear process without priori assumptions on the nature of the process [2,11]. In a real data problem, the difficulty in forecasting often arises due to the data characteristics. To overcome this difficulty, a hybrid model which combines both linear and nonlinear capabilities that employ seasonal ARIMA model and artificial neural network model can be a good strategy to practice[17, 20].

A hybrid model is considered to be composed of a linear and a nonlinear component and can be represented as[20]:

$$y_t = L_t + N_t \tag{7}$$

where L_t denotes the linear component and N_t denotes the nonlinear component. Both of these two components have to be estimated from the data. Firstly, to forecast the linear component, seasonal ARIMA model is fitted to the data series. Hence, the residuals from the linear model, seasonal ARIMA model is assumed to contain only the nonlinear associations. Let r_t be the residual at time t of the linear model, then,

$$r_t = y_t - \hat{L}_t \tag{8}$$

where $\hat{L}_t$ is the forecast value at time t from the linear model. Nonlinear relationships can be discovered by modeling residuals using artificial neural network model as follows:

$$r_t = f(r_{t-1}, r_{t-2}, \ldots, r_{t-n}) + \varepsilon_t \tag{9}$$

where f is a nonlinear function, n is the number of input nodes and ε_t is the random error. Let $\hat{N}_t$ be the forecast value at time t from artificial neural network model, then the combine forecast will be

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \tag{10}$$

Hence, the hybrid method involves a two-stage process:

(a) Forecast the linear part using seasonal ARIMA;

(b) Forecast the nonlinear part (the residuals from seasonal ARIMA) using the neural network model.

2.4 Forecasting Performance

The accuracy measures namely, the root mean squared error (RMSE), the mean absolute error (MAE) and the mean absolute percentage error (MAPE) are chosen to evaluate the overall performance of a model fitting, which are given by the equations [13,15]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{n}} \tag{11}$$

$$MAE = \frac{\sum_{i=1}^{n} |x_i - \hat{x}_i|}{n} \tag{12}$$

$$MAPE = \frac{\sum_{i=1}^{n} \frac{|x_i - \hat{x}_i|}{x_i}}{n} \times 100 \tag{13}$$

where x_i and $\hat{x}_i$ are the actual values and the predicted values respectively while n is the number of predicted values.

3. RESULTS

The performances of the SARIMA model, the neural network model and the hybrid model are compared using a half hourly daily demand ("load") measured in Megawatts (MW). This four months data from September 01, 2005 to December 31, 2005 are gathered from Tenaga Nasional Berhad (TNB), Malaysia and are illustrated in Figure 1. Sections 3.1, 3.2 and 3.3 report the results obtained from the SARIMA model, the multilayer feedforward neural network model and the hybrid model respectively.

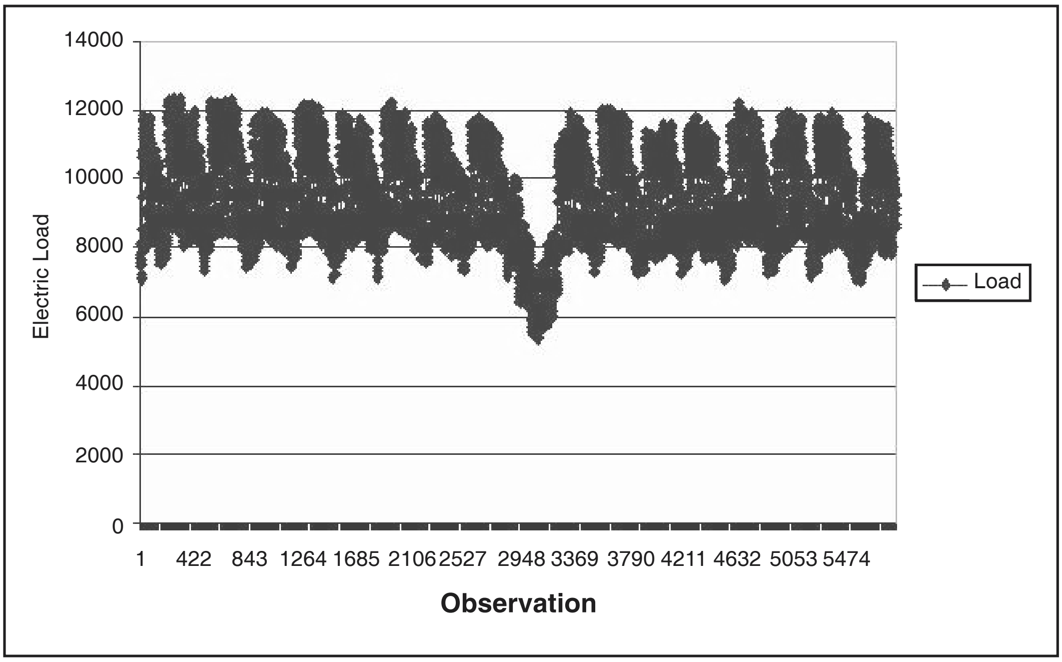

Figure 1: A Half Hourly Load from September 01, 2005 to December 31, 2005

3.1 Sarima Model

Following the rule of transformation, proposed by Wei, (2006), where $\lambda = 0$, a four months half hourly data was pre-processed using natural logarithm. Then the first-order regular differencing and the difference of seasonal at lag 48 are performed in order to stabilize the variance and remove the growth trend and seasonality. MINITAB statistical package is used to formulate the SARIMA model. ACF and PACF are performed on residuals of formulated model in order to confirm the adequacy of the formulated model. The best derived model is SARIMA (2, 1, 2) $(2, 10, 2)_{48}$ and the equation is presented as follows:

$$(1-0.4338B-0.1297B^2)(1-1.1292B^{48}+0.1301B^{96})(1-B)Y_t = (1-0.4034B+0.0977B^2)(1-0.9913B^{48}+0.0502B^{96})e_t \quad (14)$$

Table 1, reports the results of SARIMA (2, 1, 2) (2, 0, 2)$_{48}$, where RMSE is the root mean squared error, MAE is the mean absolute error and MAPE is the mean absolute percentage. The comparison between the actual values and the forecast values for this model is given in Figure 2.

Table 1
The RMSE, MAE and MAPE of the *SARI MA* (2, 1, 2)(2, 0, 2)$_{48}$ Model

	SARI MA (2, 1, 2)(2, 0, 2)$_{48}$
RMSE	117.2246
MAE	90.17704
MAPE	0.9774

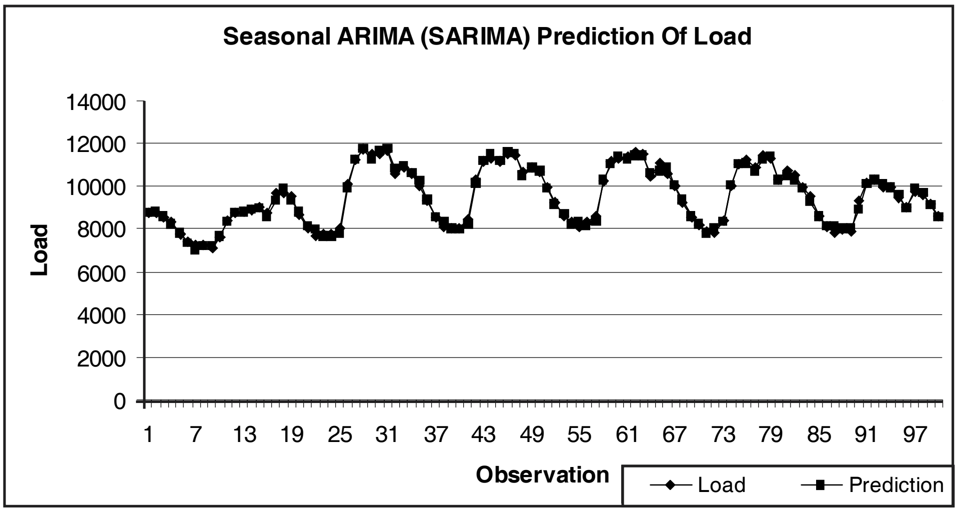

Figure 2: Seasonal ARIMA (SARIMA) Model Prediction of Load

3.2 Neural Network Model

The three-layer feed-forward architecture with BP learning algorithm and Levenberg Marquart as training algorithm is trained by the three partitioned normalized data within the range of [0,1] using MATLAB package. Before training the networks, the first task is to select the numbers of input nodes, hidden nodes and output nodes. In this study we used three input nodes, 32 hidden nodes and one output node. We used log sigmoid for both hidden and output layers. The training process involves three parts which are training, validation and testing. We initialized weights and biases randomly for the training part, then the weights and biases are saved for the next training process (validation part and testing part). Finally we chose the best-fit one to report its training, validation and testing results.

Results are shown in Table 2, where RMSE is the root mean squared error, MAE is the mean absolute error and MAPE is the mean absolute percentage error for training, validation and testing. Figure 3 gives the actual values and the forecast values for multilayer feedforward neural network model.

Table 2
The RMSE, MAE and MAPE of the Multilayer Feedforward Neural Network Model

	Multilayer Feedforward Neural Network Model		
	Training	*Validation*	*Testing*
RMSE	419.5622	487.9086	472.2600
MAE	322.03828	383.35343	358.32399
MAPE%	3.44143	4.07739	3.85983

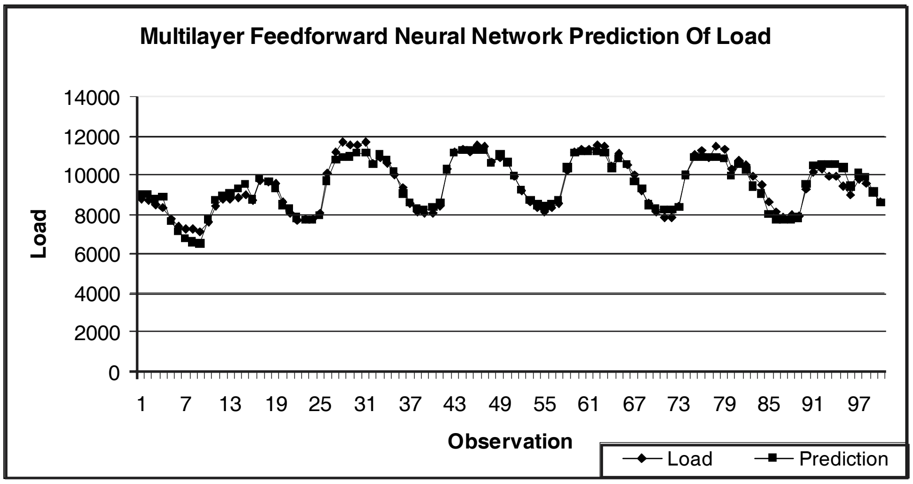

Figure 3 : Multilayer Feedforward Neural Network Model Prediction of Load

3.3 Hybrid Model

Following the procedure of SARIMA modeling, we forecast the linear part, then the residuals from seasonal ARIMA or the nonlinear part is forecasted using the neural network model procedure.

Results are shown in Table 3, where RMSE is the root mean squared error, MAE is the mean absolute error and MAPE is the mean absolute percentage error for training, validation and testing. Figure 4 gives the actual values and the forecast values for the hybrid model.

Table 3
The RMSE, MAE and MAPE of the Hybrid Model

		Hybrid Model	
	Training	*Validation*	*Testing*
RMSE	107.42865	111.01899	104.96537
MAE	82.32272	85.48085	82.21137
MAPE%	0.89642	0.92211	0.88441

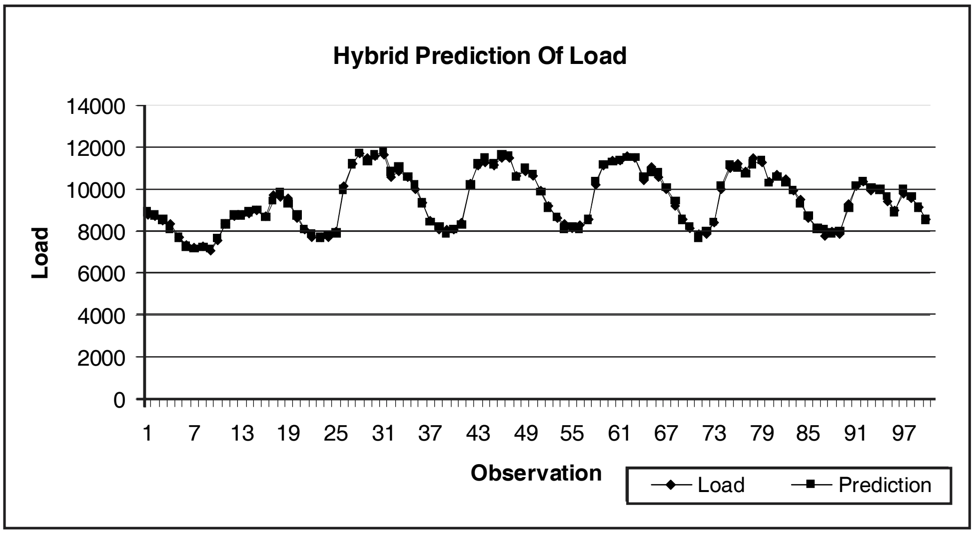

Figure 4: Hybrid Model Prediction of Load

4. CONCLUSIONS

In this paper, we proposed the hybrid model, which combines the seasonal time series ARIMA (SARIMA) and the multilayer feedforward neural network to forecast time series with seasonality. The root mean squared error (RMSE), the mean absolute error (MAE) and mean absolute percentage error (MAPE) for the hybrid model are the lowest. Thus, it is concluded that we can improve a single model forecasting performance by introducing a hybrid model.

References

[1] Box, G. E. P., Jenkins, G. M. & Reinsel, G. C., *Time Series Analysis: Forecasting and Control*. Prentice Hall, New Jersey, 1994.

[2] BuHamra, S., Smaoui, N., & Gabr, M., The Box-Jenkins Analysis and Neural Networks: Prediction and Time series Modeling, *Applied Mathematical Modeling*, **27**, (2003), 805–815.

[3] Brockwell, P. J., & Davis, R. A., Introduction to Time Series and Forecasting, Springer, New York, (1996).

[4] Chatfield, C., The Analysis of Time Series: An Introduction, Sixth Edition, Chapman & Hall, New York, (2004).

[5] Chen, W. H., Shih, J. Y. and Wu, S., Comparison of Support Vector Machines and Back Propagation Neural Networks in Forecasting the Six Major Asian Stock Markets, *Int. J. Electronic Finance*, **1**, (1), (2006), 49–67.

[6] Cichocki, A. and Unbehauen, R., Neural Networks for Optimization and Signal Processing. John Wiley and Sons, New York, (1993).

[7] Connor, J., and Atlas, L., Recurrent Neural Networks and Time Series Prediction, Proceeding IJCNN-91-Seattle, *International Joint Conference on Neural Networks*, (1991), 301–306.

[8] Connor, J. T., Martin, R. D., & Atlas, L. E., Recurrent Neural Networks and Robust Time Series Prediction, *IEEE Transactions on Neural Networks*, **5**, (2), (1994), 240–254.

[9] El-Telbany, M., & El-Karmi, F., Short-term Forecasting of Jordonian Electricity Demand using Particle Swarm Optimization, *Electric Power Systems Research*, (2007).

[10] Ho, S. L., Xie, M., and Goh, T. N., A Comparative Study of Neural Network and Box-Jenkins ARIMA Modeling in Time Series Prediction. *Computer & Industrial Engineering*, **42**, (2002), 371–375.

[11] Khoa, N. L. D., Sakakibara, K. and Nishikawa, I., Stock Price Forecasting using Back Propagation Neural Networks with Time and Profit Based Adjusted Weight Factors. *SICE-ICASE International Joint Conference*, (2006), 5484–5488.

[12] Kohzadi, N., Boyd, M. S., Kermanshi, B. and Kaastra, I., A Comparison of Artificial Neural Network and Time Series Model for Forecasting Commodity Prices, *Neurocomputing*, **10**, (1996), 169–181.

[13] Makridakis, S., Wheelwright, S. C and Hyndman, R. J., *Forecasting: Methods and Application*. John Willey and Sons, New York, 1998.

[14] Pham, T. D., and Liu, X., Neural Networks for Identification, *Prediction and Control*, Great Britain, (1995).

[15] Rabunal, J. R., and Dorado, J., Artificial Neural Networks in Real-Life Applications. Idea Group Publishing, London, (2006).

[16] Taskaya-Temizel, T., and Casey, M. C., A Comparative Study of Autoregressive Neural Network Hybrids. *Neural Networks* **18**, (2005), 781–789.

[17] Tseng, F. -M., Yu, H. -C., and Tzeng, G.-H., Combining Neural Network Model with Seasonal Time Series ARIMA Model, Technological Forecasting & Social Change, **69**, (2002), 71–87.

[18] Wei, W. W. S., *Time Series Analysis*, *Univariate and Multivariate Methods*, Second Edition. Pearson, Addison Wesley, New York, (2006).

[19] Welstead, S. T., Neural Network and Fuzzy Logic Applications in C/C++, John Willey and Sons, Canada, (1994).

[20] Zhang, G. P., Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model, *Neurocomputing*, **50**, (2003), 159–175.

[21] Zhang, G. P., and Qi, M., Neural Network Forecasting for Seasonal and Trend Time Series. *European Journal of Operation Research* **160**, (2005), 501–514.

[22] Zhang, G., Patuwo, B. E., and Hu, Y. H., Forecasting with Artificial Neural Network: The State of Art. *International Journal of Forecasting*, **14**, (1998), 35–62.

Norizan Mohamed
Mathematics Department
Faculty of Science and Technology
Universiti Malaysia Terengganu (UMT)
21030 Kuala Terengganu
Terengganu
Malaysia
E-mail: norizan@umt.edu.my

Maizah Hura Ahmad, Zuhaimy Ismail & Khairil Anuar Arshad
Department of Mathematics
Faculty of Science
Universiti Teknologi Malaysia
81310 UTM Skudai, Johor
Malaysia
E-mail: maizah@mel.fs.utm.my
zuhaimyi@yahoo.com
khairil@mel.fs.utm.my