Comparison of ANN Back Propagation Techniques in Modelling Network Traffic Activities

Conference Paper · October 2014

Haviluddin Haviluddin
Universitas Mulawarman

UIN Sultan Syarif Kasim Riau
Fakultas Sains dan Teknologi

ICoSTechS 2014
International Conference on Science and Technology for Sustainability

Proceeding

The International Conference on Science and Technology for Sustainability

Supported by :

IEEE INDONESIA SECTION

BKS PTN Barat

www.icostechs.org

1st International Conference on Science and Technology for Sustainability
Proceeding, Volume 1, October 2014
ISSN 2356-542X

Comparison of ANN Back Propagation Techniques in Modelling Network Traffic Activities

Haviluddin
Faculty of Mathematics and Natural Science
Dept. of Computer Science, Universitas Mulawarman
Indonesia
haviluddin@gmail.com, haviluddin@unmul.ac.id

Rayner Alfred
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Malaysia
ralfred121@gmail.com, ralfred@ums.edu.my

Abstract – **In this paper, we demonstrated a method used for forecasting the daily network traffic activities by using artificial neural network (ANN) with back propagation (BP) algorithms. We used the inputs and outputs of data from network traffic to identify ANN-BP models and algorithms, and we studied the performance of seventeen BP algorithms. The results using the 17 BP algorithms that include R2, MSE, MAPE and MAD were obtained with (2-12-1) network structure. Then, we compared the results using MAPE and accuracies values. The results of the comparison shows that from the seventeen BP algorithms were tested, there are some BP algorithms that generate high efficiency and accuracy of predicting the network traffic activities. Based on the results obtained, Levenberg-Marquardt, Bayesian Regularization, Fletcher-Powell Conjugate Gradient, Gradient Descent, Gradient Descent with Adaptive Learning Rate, Batch Training with Weight and Bias Learning Rules, and Sequential Order Weight/Bias Training algorithms are found to be very good for forecasting network traffic activities.**

Keywords – component; network traffic; ANN; back propagation; R2, MSE, MAPE, MAD

I. INTRODUCTION

At present, educational institutions are required to provide the infrastructure to support teaching and learning activities, and one of which is the provision of internet services. Furthermore, the ICT unit as managers are required to be able to provide internet service as well. Therefore, management of internet usage is very necessary to provide optimal internet data services.

The motivation of this study is to look at the characteristics and predict the use of the internet traffic for teaching and learning process at the University. One approach is to do with preventive control to predict future use of the internet traffic. Therefore, we need a measurement tool to predict accurately and effectively using internet data traffic has a characteristic fluctuating or not stationary.

A network traffic data normally shows network activities that indicate the periods of time when the network is heavily utilized or not being fully utilized. Therefore, network traffic data should be trained as a forecasting because its characterized is always changing along with time changing. The network traffic data knowledge should be used to provide high-quality services and design optimization. A modeling and forecasting is a one of the best phenomenological to solved that problems, planning construct and internet services evaluation in the future.

The purpose of this paper is to model and forecast the internet network traffic activities time by using an artificial neural network (ANN) with back propagation (BP) algorithm. The organization of this paper is arranged as follows. Section 2 discusses the theoretical basis relevant to the work and the techniques used to perform ANN-BP algorithm. Section 3 presents the BP algorithm. Section 4 presents the experimental design and results obtained from the analyzed series, and Section 5 concludes this paper with some recommendations on future works.

II. Related Works

This section focuses on survey that investigates the work that has been done on time series forecasting using ANN with BP algorithm.

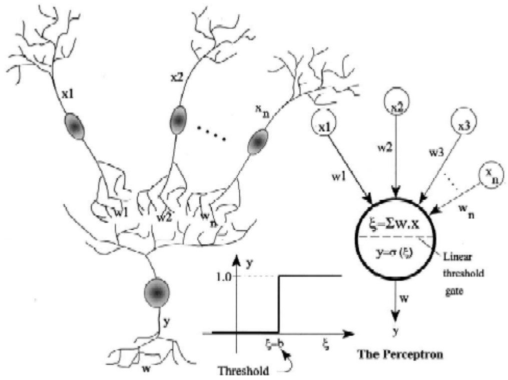
A. Artificial Neural Networks

An artificial neural networks (ANNs) is a computational models inspired in the natural neurons (Badiru & Cheung, 2002; Buchanan, 2005) and influenced by ideas from many disciplines (Basheer & Hajmeer, 2000). In fact, a simple of neural networks (NNs) was introduced by Warren McCulloch and Walter Pitts in 1943. They propose assigning weights in the network are set to perform simple logic function, called propagation function. Then, the propagation function results of weight are compared with the threshold functions, generated by the activation function. Then also, the combination of several simple neurons into a system will enhance the ability of neural computation.

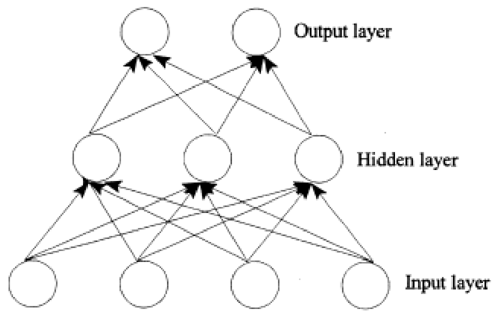
In 1958, Rosenblatt introduced and began to develop a network model called the perceptron, Fig. 1. (a). Then, in 1962 he developed into a multilayer perceptron (MLP) with the aim of overcoming the problem of weight optimization. Both of these training method is introduced to optimize the results of iteration, Fig. 1. (b) (Badiru & Cheung, 2002; Basheer & Hajmeer, 2000). The basic of perceptron neuron operation is expressed as

$$y = \begin{cases} 1, & \text{if} \sum_{i=1}^{n} w_i x_i \geq b, \\ 0, & \text{if} \sum_{i=1}^{n} w_i x_i < b, \end{cases} \tag{1}$$

with 1 indicating '*on'* and 0 indicating '*off'*.



(a)



(b)

Fig. 2. (a). single layer and (b). multilayer perceptron (Basheer & Hajmeer, 2000)

An artificial neural networks (ANNs) is a combination of two or more artificial neurons with its inputs, weights, transfer functions, and outputs bias. Where, the ANNs are trying to simulate the learning process of the human brain using a computer program that is able to resolve a number of the calculation process during the learning process with the assumptions, include (1) information processing lies in the number of components, called neurons, (2) propagate signals between one neuron to other neurons via the connecting line, (3) Each connecting line has a great weight and multiplying the value of the incoming signal, and (4) Each neuron applying the activation function for aggregating all inputs to determine the output signal. Then, to improve the performance, between input neurons and output neurons there are several layers called the hidden layer (Al Shamisi, Assi, & Hejase, 2011; Balkin & Ord, 2000; Basheer &

Hajmeer, 2000; Birdi, Aurora, & Arora, 2013; Buchanan, 2005; Krenker, Bešter, & Kos, 2011; Wagner, Michalewicz, Schellenberg, Chiriac, & Mohais, 2011).

B. Designing ANNs Models

In general, according to Al Shamisi (2011), designing ANNs model follows a number of systemic procedures. In general, there are five basics steps; (1) collecting data; collecting and preparing sample data is the first step in designing ANNs models, (2) preprocessing data; to train the ANNs more efficiently, (3) building the network; to design the number of hidden layers, neurons in each layer, transfer function in each layer, training function, weight/bias learning function and performance function, (4) training the network; during the training process, the weights are adjusted in order to make the actual outputs (predicated) close to the target (measured) outputs of the network, and (5) test performance of model; to test the performance of the developed model (Abhishek, Kumar, Ranjan, & Kumar, 2012; Al Shamisi et al., 2011).

In this work, multilayer perceptron (MLP) networks with learning by BP algorithm are used and describe briefly.

III. Back Propagation Algorithm

The back propagation algorithm (BP) was first introduced by Paul Werbos in 1974, then raised again by David Parker in 1982 and later popularized by Rumelhart and McCelland in 1986 (Basheer & Hajmeer, 2000). Then, the BP method is a part of MLP architecture with supervised learning method (Sermpinis, Dunis, Laws, & Stasinakis, 2012).

In general, BP algorithm can be described as if a network gives an input as a train pattern, then straight away to hidden layer, then directed to outputs layer. Afterward, outputs layer gives a respond that is called network output. When, network output result is not same with output target, thus output should be back called is backward at hidden layer, then directed to neurons at inputs layer. The BP using the output error to change the weights in the backward direction. To obtain the error, step forward propagation must be done first. At the time the forward propagation is done, the neurons activated by using the activation function can be differentiated, such as *sigmoid*, *tansig*, or *purelin* functions.

Furthermore, there are three phases to the training of BP, covering the first phase is a feed forward; input patterns counted forward from the input layer to the output layer. The output of each unit of the hidden layer (z_j) in forward propagation using the specified activation function to generate network output (y_k). Then, the output of the network should be compared with the target to be achieved (tk) to obtain the error (t_k-y_k).

Then, the second phase is the phase of the backward; based on that error (t_k-y_k), calculate the factor δ_k (k=1,2,...,m) then using the error for the backward, started from line-related directly with the units in the output layer. The third phase is modified weight; to reduce errors that occur then all the factors δ and weights, calculated all modified lines simultaneously until the number of iterations has exceeded the maximum number of iterations specified. However, BP cannot assure how many epochs that must be passed to achieve the specified conditions.

sigmoid

$$y = f(x)\,\frac{1}{1+e^{-\sigma x}} \tag{2}$$

tansig

$$y = f(x)\,\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3}$$

purelin

$$y = f(x) = x \tag{4}$$

f can be a simple threshold function or a *sigmoid*, *tansig* or *purelin* functions.

The BP training algorithm described below

Step 0: Initiation of all weights

Step 1: If the termination condition is not fulfilled, do step 2-8

Step 2: For each pair of training data, do steps 3-8

Phase 1: Feed forward

Step 3: Each unit receives input signals and transmitted to the hidden unit above

Step 4: Calculate all the output in the hidden layer units z_j ($j = 1,2,\ldots,p$)

$$z\_net_j = v_{jo} + \sum_{i=1}^{n} \mathrm{x_i}\, v_{kj}$$

$$z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}}$$

Step 5: Calculate all the network output in unit output y_k ($k = 1,2,\ldots,m$)

$$y\_net_k = w_{ko} + \sum_{j=1}^{p} \mathrm{z_j}\, w_{kj}$$

$$y_k = f(y\_net_k) = \frac{1}{1 + e^{-y\_net_k}}$$

Phase 2: Back propagation

Step 6: Calculate factor δ output unit based on unit output error y_k ($k = 1,2,\ldots,m$)

$$\delta_k = (t_k - y_k)f'(y\text{-}net_k) = (t_k - y_k)y_k(1-y_k)$$

t_k = output target

δ = output unit that will be used in the layer underneath the weight change

Calculate weight change w_{kj}, with the learning rate α

$$\delta w_{ji} = \alpha\delta_k z_j,\ k = 1,2,\ldots,m;\ j = 0,1,\ldots,p$$

Step 7: Calculate factor δ unit hidden layer based on the error in each hidden layer unit

z_j ($j = 1,2,\ldots,p$)

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k\, w_{kj}$$

Factor δ hidden layer unit

$$\delta_j = \delta\_net_j\, f'(z\_net_j) = \delta\_net_j\, z_j\,(1-z_j)$$

Calculate weight change rate v_{ji}

$$\delta v_{ji} = \alpha\delta_k z_j,\ k = 1,2,\ldots,p;\ j = 0,1,\ldots,n$$

Phase 3: Weight modification

Step 8: Calculate the weight of all the changes that led to the output unit

$$w_{kj(new)} = w_{kj(old)} + \delta w_{ji};\ (k = 1,2,\ldots,p;\ j = 0,1,\ldots,n)$$

Weight changes that led to the hidden layer units

$$v_{kj(new)} = v_{kj(old)} + \delta v_{ji}; (j = 1,2,...,p; w = 0,1,...,n)$$

3.1. Back propagation architecture

The BP architecture has several neurons that exist in one or more hidden layers. Fig. 2, described that v_{ji} is the weight of the line from the input unit x_i to the hidden layer units z_j (v_{n0} is a weight line which connects the bias on the input units to the hidden layer units z_j). w_{kj} is the weight of the hidden layer units z_j to the output unit y_k (w_{k0} is the weight of the bias in the hidden layer to the output unit z_k).

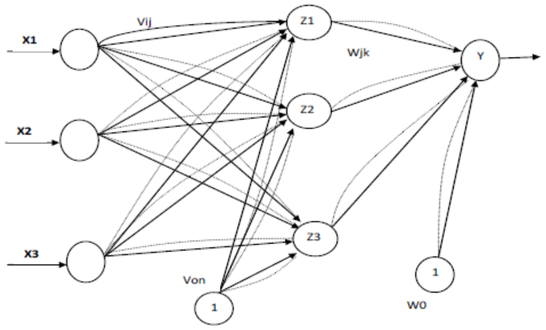


Fig. 2. Sample of back propagation architecture

In this research, the 17 BP training algorithms were tested in order to obtain the most appropriate algorithm for the training process. The algorithms include: Levenberg-Marquardt, Bayesian Regularization, BFGS Quasi-Newton, Resilient back propagation, Scaled Conjugate Gradient, Conjugate Gradient with Powell-Beale Restarts, Fletcher-Powell Conjugate Gradient, Polak-Ribiére Conjugate Gradient, One Step Secant, Gradient Descent with Momentum & Adaptive Learning Rate, Gradient Descent with Momentum, Gradient Descent, Gradient Descent with Adaptive Learning Rate, Random Order Weight/Bias Training, Batch Training with Weight and Bias Learning Rules, Cyclical Order Weight/Bias Training, Sequential Order Weight/Bias Training Learning Rules.

3.2. Back propagation training

Based-on the ANN-BP algorithm model, there are four steps to build forecasting purpose, consist of (1) collecting data; collecting and preparing sample data, (2) data normalization; to train the ANNs more efficiently, (3) training and testing data; to train and test the performance of the model, and (4) comparing the predicted output with the desired output; using statistical analysis, e.g. sum of square error (SSE), mean of square error (MSE), mean of percentage error (MPE), mean of absolute percentage error (MAPE), mean of absolute deviation (MAD), determination (R), and coefficient of determination (R^2) (Abhishek et al., 2012; Al Shamisi et al., 2011; Ticknor, 2013).

3.3. Data specification

In this research, each network traffic data was captured by the CACTI software. The network traffic has two graphic areas, indicated by green and blue colors. The green color indicates the *inbound* traffic and the blue color indicates the *outbound* traffic. The *Inbound* traffic shows data that comes from outside (e.g., a computer) into the network. On the other hand, the *outbound* traffic shows data that goes out from the network. In this study, four days daily network traffic data from 21 – 24 June 2013 (192 data series samples) was captured. The datasets consist of 144 (90%) samples for data training and 48 (10%) samples for data testing. The inputs and target to the neural network forecaster are addition inbound and outbound series; input1 series of (48-144), input2 series of (1-97) and target series of (49-145).

IV. Results And Discussion

This research presents the best achieved results for both the ANN-BP algorithm function. Table 1 show the computed values of R^2, MSE, MAPE and MAD with (2-12-1) network structure, Fig. 3. Then, the comparative analysis is based on the speed of convergence to the benchmarks when the number of iterations required for convergence training, and based on the level of accuracy with MAPE value benchmarks for case modeling network traffic daily activities, Table 1.

There are several algorithms that have the best MAPE values, that include Levenberg-Marquardt (0.089), Bayesian Regularization (0.085), Fletcher-

Powell Conjugate Gradient (0.090), Gradient Descent (0.095), Gradient Descent with Adaptive Learning Rate (0.095), Batch Training with Weight and Bias Learning Rules (0.066), and Sequential Order Weight/Bias Training (0.067) algorithms.

Then, seven algorithms have better accuracies to predict the daily activities of network traffic which include the Levenberg-Marquardt (0.969%), Bayesian Regularization (0.925%), Fletcher-Powell Conjugate Gradient (0.922%), Gradient Descent (0.939%), Gradient Descent with Adaptive Learning Rate (0.926%), Batch Training with Weight and Bias Learning Rules (0.913%), and Sequential Order of Weight/Bias Training (0.907%).

Meanwhile, there are seven other algorithms have also produced good accuracy values, but the MAPE values are greater than the other algorithms. These MAPE values are computed using the BFGS Quasi-Newton (0119), Resilient Back Propagation (0102), Scaled Conjugate Gradient (0108), Conjugate Gradient with Powell/Beale Restarts (0110), Polak-Ribiére Conjugate Gradient (0107), One Step Secant (0614), Gradient Descent with Momentum and Adaptive Learning Rate (0107), Gradient Descent with Momentum (0101), Random Order Weight/Bias Training (0103), and Cyclical Order Weight/Bias Training (0104).

Therefore, the seven algorithms have a large MAPE values cannot be used as a model for the prediction of daily network traffic activities.

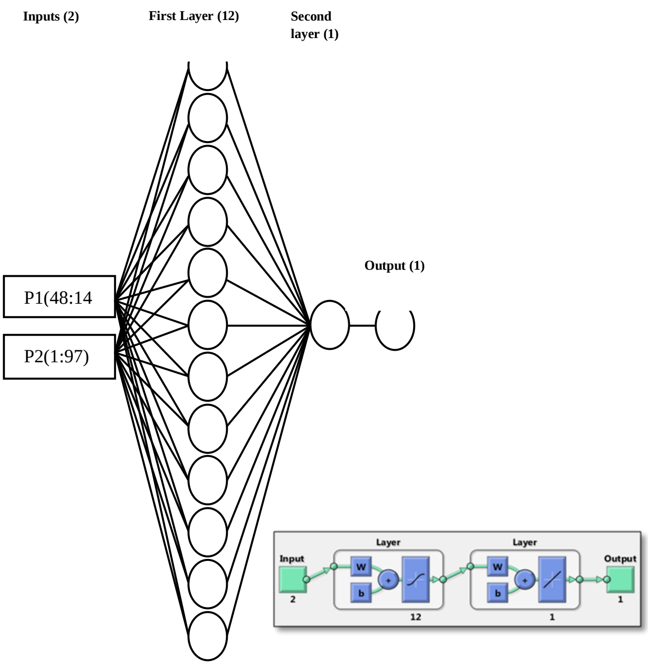
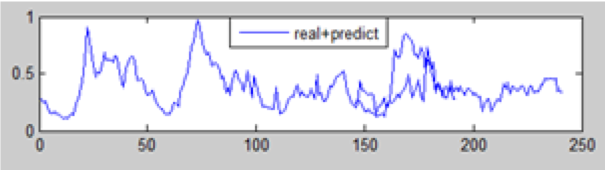
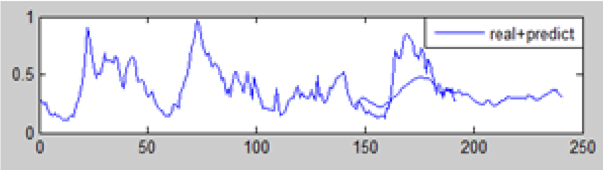


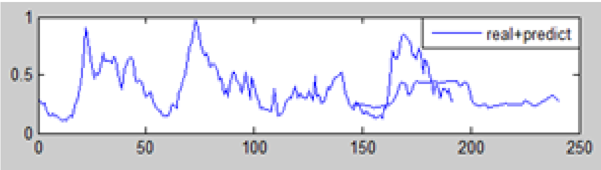
Fig. 3. Neural network model for traffic forecasting



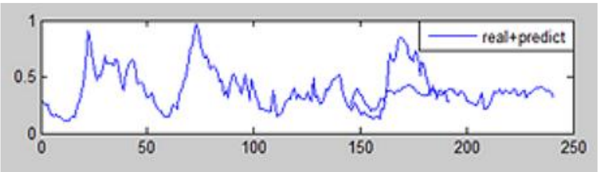
Levenberg-Marquardt algorithm



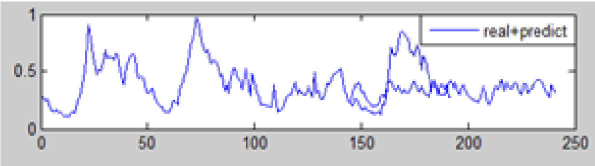
Bayesian Regularization algorithm



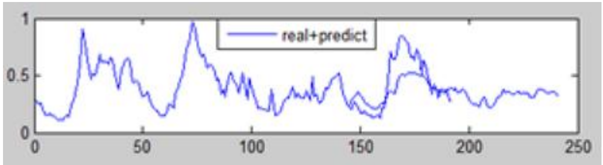
Fletcher-Powell Conjugate Gradient algorithm



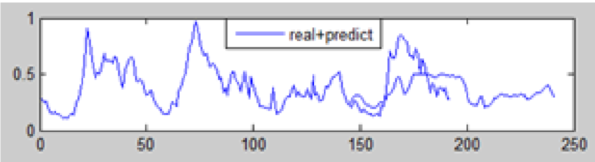
Gradient Descent



Gradient Descent with adaptive learning rate algorithm



Batch Training with Weight & Bias Learning Rules



Sequential order weight/bias training algorithm

Fig. 4. Plots comparison between measured data and predicted of seven ANN-BP algorithms

TABLE I. RESULT OF BP ALGORITHMS

No	**BP algorithms**	**Test Results**			
		R^2	*MSE*	*MAPE*	*MAD*
1	Levenberg-Marquardt	0.538	0.008	0.089	0.042
2	Bayesian Regularization	0.579	0.007	0.085	0.035
3	BFGS Quasi-Newton	0.445	0.014	0.119	0.051
4	Resilient back propagation	0.490	0.010	0.102	0.044
5	Scaled Conjugate Gradient	0.492	0.012	0.108	0.046
6	Conjugate Gradient with Powell-Beale Restarts	0.489	0.012	0.110	0.046
7	Fletcher-Powell Conjugate Gradient	0.555	0.008	0.090	0.039
8	Polak-Ribiére Conjugate Gradient	0.483	0.011	0.107	0.046
9	One Step Secant	0.223	0.377	0.614	0.284
10	Gradient Descent with momentum & adaptive learning rate	0.491	0.011	0.107	0.045
11	Gradient Descent with Momentum	0.516	0.010	0.101	0.041
12	Gradient Descent	0.526	0.009	0.095	0.040
13	Gradient Descent with adaptive learning rate	0.527	0.009	0.095	0.041
14	Random order weight/bias training	0.512	0.011	0.103	0.041
15	Batch training with weight & bias learning rules	0.628	0.004	0.066	0.030
16	Cyclical order weight/bias training	0.487	0.011	0.104	0.044
17	Sequential order weight/bias training	0.628	0.004	0.067	0.031

V. CONCLUSION AND FUTURE WORK

In this research, a models and forecasting network traffic using ANN-BP for the day-ahead is presented. We used the inputs, target and output data to describe ANN-BP, and the performance of training algorithm is accessed using R^2, MSE, MAPE and MAD with (2-12-1) network structure. Then, the comparison of test results is done by looking at the value of MAPE and accuracy. The seventeen BP algorithm were used to train the model of ANN. Then, some of the comparison BP algorithms demonstrated the efficiency and the accuracy of the seven algorithms, includes: Levenberg-Marquardt, Bayesian Regularization, Fletcher-Powell Conjugate Gradient, Gradient Descent, Gradient Descent with Adaptive Learning Rate, Batch Training with Weight and Bias Learning Rules, and Sequential Order Weight/Bias Training algorithms.

Furthermore, seven of the algorithms are selected for further investigation since they have large MAPE values such as Gradient Descent (0.095) and Gradient Descent with Adaptive Learning Rate (0.095) to be optimized by using a particular optimization method. Therefore, for future research, the Back propagation method will be optimized using a genetic algorithm in order to generate a higher and efficient accuracy in predicting the network traffic daily activities.

ACKNOWLEDGMENTS

I would like to express my gratitude to the "Beasiswa Kaltim Cemerlang" for providing me a scholarship in PhD Program at Universiti Malaysia Sabah (UMS) and the Committee of International Conference on Science and Technology for Sustainability October 22nd 2014, Batam, Riau Island, Indonesia" for selecting and accepting this paper.

REFERENCES

[1] Abhishek, K., Kumar, A., Ranjan, R., & Kumar, S. (2012). A Rainfall Prediction Model using Artificial Neural Network. *2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012)*.

[2] Al Shamisi, M. H., Assi, A. H., & Hejase, H. A. N. (2011). Using MATLAB to Develop Artificial Neural Network Models for Predicting Global Solar Radiation in Al Ain City – UAE. In A. Assi (Ed.), *Engineering Education and Research Using MATLAB*: InTech.

[3] Badiru, A. B., & Cheung, J. Y. (2002). *FUZZY ENGINEERING EXPERT SYSTEMS WITH NEURAL NETWORK APPLICATIONS*.

[4] Balkin, S. D., & Ord, J. K. (2000). Automatic neural network modeling for univariate time series. *International Journal of Forecasting, 16*(2000), 509–515.

[5] Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods, 43*(2000), 3-31.

[6] Birdi, Y., Aurora, T., & Arora, P. (2013). Study of Artificial Neural Networks and Neural Implants. *International Journal on Recent and Innovation Trends in Computing and Communication, 1*(4).

[7] Buchanan, B. G. (2005). A (Very) Brief History of Artificial Intelligence. *American Association for Artificial Intelligence*.

[8] Krenker, A., Bešter, J., & Kos, A. (2011). Introduction to the Artificial Neural Networks. In K. Suzuki, Prof. (Ed.), Artificial Neural Networks - Methodological Advances and Biomedical Applications (Vol. April, 2011, pp. 1-18): InTech.

[9] Sermpinis, G., Dunis, C., Laws, J., & Stasinakis, C. (2012). Forecasting and trading the EUR/USD exchange rate with stochastic Neural Network combination and time-varying leverage. *Decision Support Systems, 54*(2012), 316–329. doi: 10.1016/j.dss.2012.05.039

[10] Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications, 40*(2013), 5501–5506. doi: http://dx.doi.org/10.1016/j.eswa.2013.04.013

[11] Wagner, N., Michalewicz, Z., Schellenberg, S., Chiriac, C., & Mohais, A. (2011). Intelligent techniques for forecasting multiple time series in real-world systems. *International Journal of Intelligent Computing and Cybernetics, 4*(3, 2011), 284-310. doi: 10.1108/17563781111159996

Haviluddin was born in Loa Tebu, East Kalimantan, Indonesia. He graduated from STMIK WCD Samarinda in 2002 in the field of Management Information, and he completed a Master at Gadjah Mada University, Yogyakarta in 2005 in the field of Computer Science. He is also a Lecturer in the Faculty of Mathematics and Natural Science, University Mulawarman, East Kalimantan, Indonesia. Currently, he is pursuing his PhD in the field of Computer Science at the Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia. He is a member of the Institute of Electrical and Electronic Engineers (IEEE) and Institute of Advanced Engineering and Science (IAES) societies.

Rayner Alfred was born in Kota Kinabalu, Sabah. He completed a PhD in 2008 looking at intelligent techniques to model and optimize the complex, dynamic and distributed processes of knowledge discovery for structured and unstructured data. He holds a PhD degree in Computer Science from York University (United Kingdom), a Master degree in Computer Science from Western Michigan University, Kalamazoo (USA) and a Computer Science degree from Polytechnic University of Brooklyn, New York (USA).

Dr. Rayner leads and defines projects around knowledge discovery and information retrieval at Universiti Malaysia Sabah. One focus of Dr. Rayner's work is to build smarter mechanism that enables knowledge discovery in relational databases. His work addresses the challenges related to big data problem: How can we create and apply smarter collaborative knowledge discovery technologies that cope with the big data problem.

Dr. Rayner has authored and co-authored more than 75 journals/book chapters and conference papers, editorials, and served on the program and organizing committees of numerous national and international conferences and workshops. He is a member of the Institute of Electrical and Electronic Engineers (IEEE) and Association for Computing Machinery (ACM) societies.