2014 Makassar International Conference on Electrical Engineering and Informatics (MICEEI)
Makassar Golden Hotel, Makassar, South Sulawesi, Indonesia
26-30 November 2014

Comparing performance of Backpropagation and RBF neural network models for predicting daily network traffic

Purnawansyah
Dept. of Informatics Engineering
Faculty of Computer Science
Universitas Muslim Indonesia
purnawansyah@gmail.com

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

***Abstract*—The predicting daily network traffic usage is a very important issue in the service activities of the university. This paper present techniques based on the development of backpropagation (BP) and radial basis function (RBF) neural network models, for modelling and predicting the daily network traffic at Universitas Mulawarman, East Kalimantan, Indonesia. The experiment results indicate that a strong agreement between model predictions and observed values, since MSE is below 0.005. When performance indices are compared, the RBFNN-based model is a more accurate predictor with MSE value is 0.00407999, MAPE is 0.03701870, and MAD is 0.06885187 than the BPNN model. Therefore, the smallest MSE value indicates a good method for accuracy, while RBF finding illustrates proposed best model to analyze daily network traffic.**

Keywords*—*BP, RBF, MSE, network traffic

I. INTRODUCTION

Currently, universal prediction method especially statistical methods that has been widely used as a simple regression analysis (SRA) method, decomposition, exponential smoothing (ES) method, and autoregressive integrated moving average (ARIMA), in which are these methods are very well implemented in some predictions, but it still has some obstacles. These methods are very well used to predict the linear data, but the results are less accurate when applied to data that are non-linear, also cannot be applied to the prediction that uses many factors [1, 2].

Hence, modeling using neural network (NN) models with various algorithms to be an alternative that can provide a better analysis of the results and it is effective for forecasting [3], in which, the method is able to work well the non-linear time-series data [1, 4-6].

Therefore, in this study, two models, BPNN and RBFNN have been developed and compared in predicting daily network traffic data from the ICT Center server of Universitas Mulawarman, in which has non-linear characteristics. Furthermore, the results of this analysis and predictions will be used for the management to manage the network traffic, particularly the use of the Internet, such as the amount of bandwidth distribution. This paper is consists of four sections. Section 1 is the motivation to do the writing of the article. Next, the methodology and techniques is discussed in Section 2. Section 3 presents the experimental results and discussion, and finally Section 4 describes the research summaries and conclusion.

II. METHODOLOGY

An artificial neural networks (ANNs) is a computational models inspired in the natural neurons [7, 8] and influenced by ideas from many disciplines [9]. A simple of neural networks 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.

A. BPNN structure

The BPNN is the abbreviation of backpropagation neural network. The BPNN system is a kind of sustenance forward neural framework which structures a bit of MLP structural building with oversaw learning methodology. Its improvement is centered on the limit unpleasant speculation theory. The BPNN framework was at first displayed by Paul Werbos in 1974, then raised again by David Parker in 1982 and later progressive by Rumelhart and McCelland in 1986. With everything taken into account, BP framework can be depicted as if a framework gives a data as a train plan, then straight away to disguised layer, then directed to yields layer. A brief time later, yields layer gives a respond that is called framework yield. Exactly when, framework yield result is not same with yield target, hence yield should be back called is backward at disguised layer, then guided to neurons at inputs layer.

The BPNN is a three-layer feed-forward neural network, which includes an input layer, a hidden layer and an output layer with linear neurons [9-12]. The typical structure of BPNN is shown in Fig. 1.

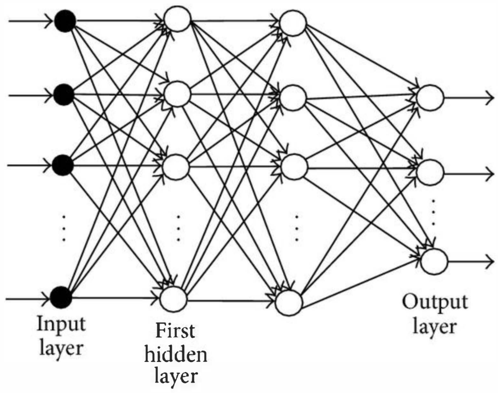

Fig. 1. BP neural network structure

The backpropagation algorithm consists of two phases: the forward phase where the activations are propagated from the input to the output layer, and the backward phase, where the error between the observed actual and the requested nominal value in the output layer is propagated backwards in order to modify the weights and bias values.

Based-on the BPNN algorithm model, there are four steps involved in building a forecasting algorithm which consists 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 (R2) [13-15]. Furthermore, 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 ($j = 1,2,...,p$)

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

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

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

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

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

Phase 2: Back propagation

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

$$\delta k = (tk - yk)f'(y\text{-}netk) = (tk - yk)yk(1-yk)$$

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

Calculate weight change wkj, with the learning rate α

$$\delta wji = \alpha\delta k\ zj,\ k = 1,2,\ldots,m;\ j = 0,1,\ldots,p$$

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

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

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

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}

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

Phase 3: Weight modification

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

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

Weight changes that led to the hidden layer units

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

B. *RBFNN structure*

The RBFNN is the abbreviation of radial basis function neural network which is based on the function approximation theory. The distance between weight vector and threshold vector is used to independent variable of the transfer function of the network. Hence, the distance through the product of input vector and weighted matrix's row vector is obtained.

The RBFNN emerged as a variant of NN in late 80's is a kind of feed-forward neural network (FFNN), which includes an input layer, a hidden layer, and an output layer. In general, RBF process is non-linear neurons from the input layer to the hidden layer, whereas from the hidden layer to the output layer is linear neurons radial-based activation function, commonly Gaussian function. Hence, RBFNN has a unique training algorithm consists of supervised and unsupervised as well The architecture of RBFNN as shown in Fig 2. Then, the RBFNN equation is

ISBN : 978-1-4799-6726-1

$$Y = \sum_{j=1}^{m} W_{jm} \cdot \varphi$$

Where: $Y = output\ value$; $\varphi = hidden\ layer\ value$; $W = weights\ (0-1)$.

The algorithm of RBF to analyze within time series data characteristics is:

1. Initialization of the network.
2. Determining the input signal to hidden layer, and find D_{ij} is a distance data i to j where $i, j = 1, 2, \dots, Q$

$$D_{ij} = \sqrt{\sum_{k=1}^{R} (p_{ik} - p_{jk})^2}$$

3. Find $a1$ is a result activation from distance data multiply bias.

$$a1_{ij} = e^{-(b1 * D_{ij})^2} \text{ x } b1 = \frac{\sqrt{-\ln(0.5)}}{spread}$$

4. Find weight and bias layers, $w2_k$ and $b2_k$, in each $k = 1, 2, \dots, S$

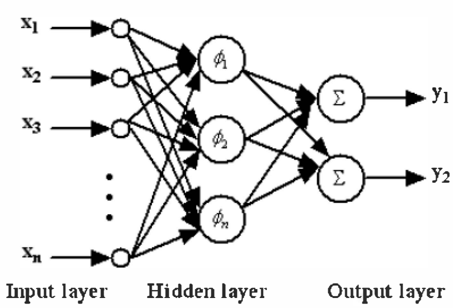

Fig. 2. RBF neural network structure [16]

C. Predicition performance model

The data were collected from 21 – 24 June 2014 (192 samples series data). Then, each network traffic data was captured by the CACTI software. The daily network traffic data was analyzed using MATLAB R2013b. Furthermore, in this experiment we used the mean of square error (MSE), mean of absolute percentage error (MAPE), and mean of absolute deviation (MAD) to engaged, then comparing the predicted output with the desired output between BP and RBF.

III. Results and Discussion

In this experiment, the training data were selected from 21 – 24 June 2014 (192 samples series data) was captured. Based on NNs rules, the data were divided into training and testing data then, the datasets consist of 144 (90%) samples for data training and 48 (10%) samples for data testing or five neurons, P = [p(t-5),p(t-4),p(t-3),p(t-2),p(t-1)], and the number of output neurons is one, p'(t), as shown in table 1.

TABLE I. The Daily Network Data After Normalized in 21-24 June 2014

Group	input neurons P = [p(t-5),p(t-4),p(t-3),p(t-2),p(t-1)]						output neurons T
		p(t-5)	p(t-4)	p(t-3)	p(t-2)	p(t-1)	p'(t)
Train Group	1	0.262	0.231	0.237	0.201	0.154	0.139
	2	0.231	0.237	0.201	0.154	0.139	0.164
	3	0.237	0.201	0.154	0.139	0.164	0.145
	4	0.201	0.154	0.139	0.164	0.145	0.136
	5	0.154	0.139	0.164	0.145	0.136	0.117
	.....	.....	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....	.....	.....
	140	0.490	0.446	0.322	0.284	0.232	0.213
	141	0.446	0.322	0.284	0.232	0.213	0.187
	142	0.322	0.284	0.232	0.213	0.187	0.251
	143	0.284	0.232	0.213	0.187	0.251	0.246
	144	0.232	0.213	0.187	0.251	0.246	0.211
Test Group	145	0.213	0.187	0.251	0.246	0.211	0.162
	146	0.187	0.251	0.246	0.211	0.162	0.163
	147	0.251	0.246	0.211	0.162	0.163	0.180
	148	0.246	0.211	0.162	0.163	0.180	0.149
	149	0.211	0.162	0.163	0.180	0.149	0.141
	.....	.....	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....	.....	.....
	188	0.352	0.322	0.359	0.259	0.253	0.262
	189	0.322	0.359	0.259	0.253	0.262	0.231
	190	0.359	0.259	0.253	0.262	0.231	0.237
	191	0.259	0.253	0.262	0.231	0.237	0.201
	192	0.253	0.262	0.231	0.237	0.201	0.154

The first analysis, daily network traffic data were tested using RBFNN technique. Then, creating a precise neural network by *newrb (P, T, error_goal, spread)* function, which is this function creates RBF structure, automatically selected the number of hidden layer and made the error to 0. For the error goal values are 0.001. Spread is the density of basis function, then spread value of 100 was settled.

In the second analysis, daily network traffic data were tested using BPNN technique. Then, creating a precise BPNN by `newff (PR,[S1 S2...SN1],[TF1,TF2...TFN1],BTF,BLF,PF)` function. In this experiment, the BPNN architecture that has been used two-hidden layers (4-50-25-1). Then, the activation function from input to hidden layers were *tansig* and *logsig*, and from hidden layers to output was *purelin*, then used gradient descent with momentum and adaptive learning rate (*traingdx*) algorithms. Then, *epoch* 10000, *goal* 0.001, *learning rate* 0.1 with *momentum* 0.8 were settled.

In this study, to compare the predicted output with the desired output, MSE, MAPE and MAD was predefined, as shown in table 2.

ISBN : 978-1-4799-6726-1

TABLE II. PERFOMANCE RECORDS TESTING FOR MODELS

Model	MSE	MAPE	MAD
RBF	0.00407999	0.02124013	0.0520818
BPNN	0.00814160	0.03701870	0.06885187

IV. SUMMARY AND CONCLUSION

In this paper, the analysis using BP and RBF techniques to achieve the model of daily network traffic have been conducted in the ICT Center Universitas Mulawarman, East Kalimantan. According to Fig. 2, the results of BP analysis show that MSE value is 0.00814160, MAPE is 0.03701870, and MAD is 0.06885187. In addition, the results of RBF shows that for MSE value is 0.00407999, MAPE is 0.03701870, and MAD is 0.06885187.

Indicator test result of data is the smallest error value, where value indicating an error testing is the best model [16]. Therefore, the determination of the best model is determined by selecting the smallest value of testing error. Based on the results, RBF has the smallest value of testing error. Thus, the test results of RBF are considered closer to the actual value. In other words, the RBF model illustrates the proposed best model to predict daily network traffic. Therefore, one of the planned future works is to combine the Back propagation method with a genetic algorithm (GA) in order to optimize the prediction accuracy.

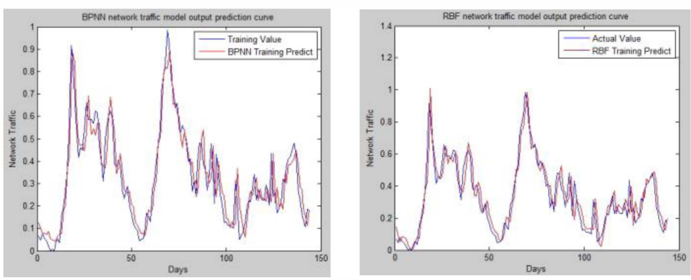

Fig. 3. BP and RBF neural network training curves

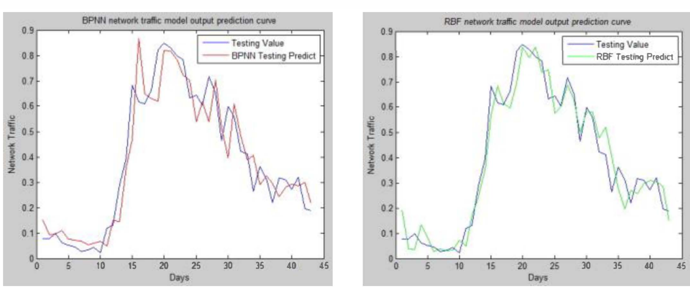

Fig. 4. BP and RBF neural network testing curves

REFERENCES

[1] O. Claveria, and S. Torra, "Forecasting tourism demand to Catalonia: Neural networks vs. time series models," *Economic Modelling,* vol. 36, no. 2014, pp. 220–228, 2014.

[2] Haviluddin, and R. Alfred, "Forecasting Network Activities Using ARIMA Method," *Journal of Advances in Computer Networks,* vol. 2, no. 3, September 2014, pp. 173-179, 2014.

[3] B. Majhi, M. Rout, and V. Baghel, "On the development and performance evaluation of a multiobjective GA-based RBF adaptive model for the prediction of stock indices," *Journal of King Saud University – Computer and Information Sciences*, no. 2014, 2014.

[4] G. Chen, K. Fu, Z. Liang *et al.*, "The genetic algorithm based back propagation neural network for MMP prediction in CO_2-EOR process," *Fuel,* vol. 126, no. 2014, pp. 202–212, 2014.

[5] A. U. Haque, P. Mandal, M. E. Kaye *et al.*, "A new strategy for predicting short-term wind speed using soft computing models," *Renewable and Sustainable Energy Reviews,* vol. 16, no. 2012, pp. 4563–4573, 2012.

[6] Y. Birdi, T. Aurora, and P. Arora, "Study of Artificial Neural Networks and Neural Implants," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 1, no. 4, 2013.

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

[8] A. B. Badiru, and J. Y. Cheung, "FUZZY ENGINEERING EXPERT SYSTEMS WITH NEURAL NETWORK APPLICATIONS," © 2002 by John Wiley & Sons, New York. All rights reserved., 2002.

[9] I. A. Basheer, and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods,* vol. 43, no. 2000, pp. 3-31, 2000.

[10] K. G. Upadhyay, A. K. Choudhary, and M. M. Tripathi, "Short-term wind speed forecasting using feed-forward back-propagation neural network," *International Journal of Engineering, Science and Technology,* vol. 3, No. 5, no. 2011, pp. 107-112, 2011.

[11] H. H. Örkcü, and H. Bal, "Comparing performances of backpropagation and genetic algorithms in the data classification," *Expert Systems with Applications,* vol. 38, no. 2011, pp. 3703–3709, 2011.

[12] G. Sermpinis, C. Dunis, J. Laws *et al.*, "Forecasting and trading the EUR/USD exchange rate with stochastic Neural Network combination and time-varying leverage," *Decision Support Systems,* vol. 54, no. 2012, pp. 316–329, 2012.

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

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

[15] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications,* vol. 40, no. 2013, pp. 5501–5506, 2013.

[16] J. Wu-Yu, and J. Yu, "Rainfall time series forecasting based on Modular RBF Neural Network model coupled with SSA and PLS," *Journal of Theoretical and Applied Computer Science,* vol. 6, no. 2, pp. 3-12, 2012.

ISBN : 978-1-4799-6726-1