ARTIFICIAL NEURAL NETWORKS (ANNS) FOR DAILY RAINFALL RUNOFF MODELLING

Kuok King Kuok and Nabil Bessaih[1]

[1]*Faculty of Engineering, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak*

Email: kkuok100@yahoo.com.sg and nabil.bessaieh@mf.gov.dz

ABSTRACT

Rainfall-runoff relationships are among the most complex hydrologic phenomena. Hydrologists have developed conceptual models to simulate runoff but these are composed of a large number of parameters and the interaction is highly complicated. ANN is an information-processing system composed of many nonlinear and densely interconnected neurons. ANN is able to extract the relation between the inputs and outputs of a process without the physics being provided to them. Natural behavior of hydrological processes is appropriate for the application of ANN in hydrology. Nowadays, ANNs are used to build rainfall-runoff models, estimate pier scour. Daily rainfall-runoff model for Sungai Bedup Basin, Sarawak was built using MLP, REC networks. Inputs used are antecedent rainfall, antecedent runoff and rainfall while output was the runoff. ANNs were trained using different training algorithms, learning rates, length of data and number of hidden neurons. All data was collected from DID Sarawak. Results were evaluated using Coefficient of Correlation (R) and Nash-Sutcliffe Coefficient (E^2). Results show that ANNs is able to simulate daily runoff with high accuracy (R=0.97). REC performs slightly better than MLP.

Keywords: *Artificial Neural Networks, Flood Forecasting, Rainfall-runoff Modeling*

1. INTRODUCTION

The rainfall-runoff relationship is the most complex hydrologic phenomena to comprehend due to the tremendous spatial and temporal variability of watershed characteristics and precipitation patterns. Conceptual models which formulate the physical process of rainfall-runoff are composed of a large number of parameters. The interaction of these parameters is highly complicated. The accuracy of a rainfall-runoff model is very subjective and dependent on the user's ability and understanding of the model.

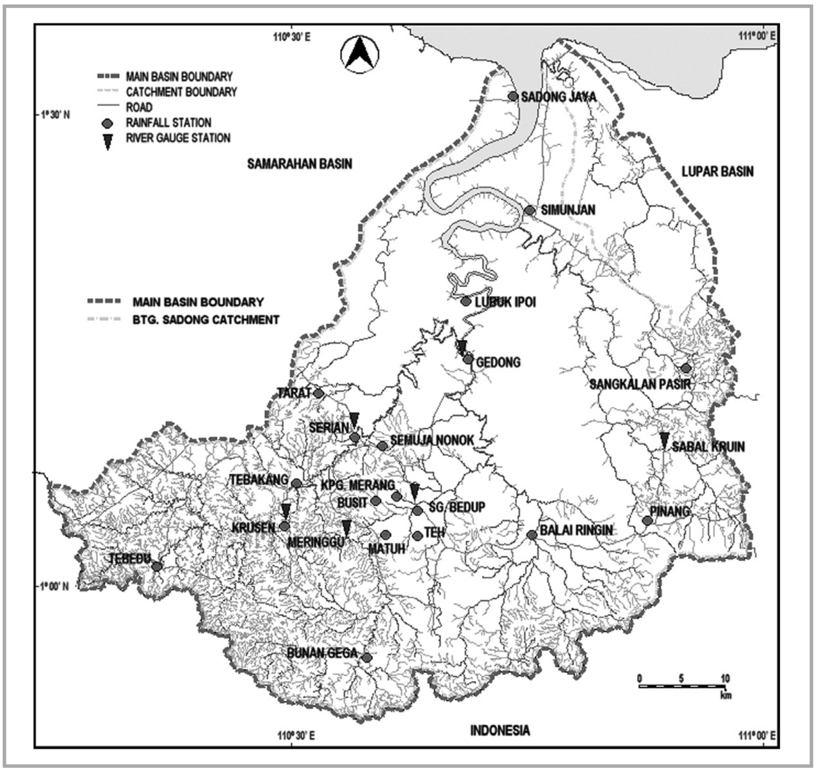

Figure 1: The Sadong basin main boundary

Artificial neural networks (ANNs) have been found to be a powerful tool for solving different problems in a variety of applications. It is an information-processing system composed of many nonlinear and densely interconnected processing elements or neurons. Each neuron is linked with its neighbors with an associated weight that represent information used by the net to solve a problem. Neurons arranged in groups called layers and operated in logical parallelism. Information is transmitted from one layer to others in serial operations. Three basic layers of ANNs are input layer, hidden layer and output layer.

Nowadays, ANNs are widely used as an efficient tool in different areas of water engineering. These include modeling of rainfall-runoff relationship [1; 2]; inflow estimation [3]; runoff analysis in humid forest catchment [4]; river flow prediction [5; 6]; setting up stage-discharge relations [7]; ungauged catchment flood prediction [8] and short term river flood forecasting [9].

In this study, multilayer perceptron (MLP) network, recurrent (REC) network were used to simulate daily runoff. The ANNs were trained and tested with different training algorithm, different length of training data, different number of hidden neurons, different learning rate and different number of antecedent days in order to select the best performance ANNs. The simulation work was carried out using Matlab 6.5 Software.

2. STUDY AREA

Sungai Bedup Basin (Figure 2), part of the Sadong Basin (Figure 1), located approximately about 80 km from Kuching City along Serian-Sri Aman road was selected for this study. This basin is located in the rural area of Samarahan Division. The area of the basin is approximately about 47.5km^2 The elevation varies from 8m to 686m above mean sea level. The vegetation cover is mainly of shrubs or low plant and forest. Sungai Bedup's

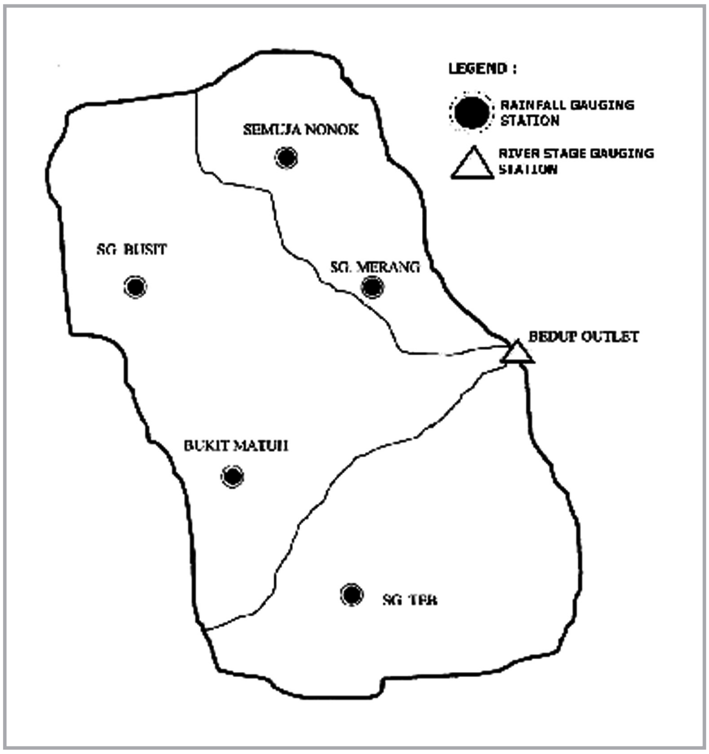

Figure 2: Sg Bedup Catchment

basin has a dendritic type channel system. The maximum stream length for the basin is approximately 10km, which is measured from the most remote area point of the stream to the basin outlet.

Rainfall is measured using five rainfall gauging stations namely Bukit Matuh (BM), Semuja Nonok (SN), Sungai Busit (SB), Sungai Merang (SM), Sungai Teb (ST). Water level is measured using one river stage gauging station at Sungai Bedup located at the outlet of the basin (Figure 2). Daily Rainfall data from the 5 rainfall stations are used as the input. The mean runoff data was converted from water level data through a rating curve.

3. NEURAL NETWORK ARCHITECTURES AND TRAINING ALGORITHM

Two networks architecture were used and they are MLP and REC. MLP network is the most popular training algorithm for ANNs [10]. This gradient descent technique will minimise the network error function [11]. REC network is a three-layer BP networks, with the addition of a feedback connection from the output of the hidden layer to its input. The feedback path makes it possible for REC networks to recognise and generate temporal and spatial patterns [10]. This makes REC networks useful in areas such as prediction where time plays an important role.

3.1 MLP NETWORK

The MLP network used is a two-layer feedforward network trained with backpropagation learning algorithm (Figure 3). The transfer function used in the hidden layer is tan-sigmoid (TANSIG) and linear transfer function (PURELIN) at the output layer [12]. The number of hidden neurons was determined through trial and error method. After trying various types of training algorithm at preliminary stages, three different variants of backpropagation algorithms were selected for further training and they are:

a) Scaled Conjugate Gradient (TRAINSCG). TRAINSCG was designed to avoid the time consuming line search, which produces generally faster convergence than the steepest descent directions used by the basic backpropagation [10].

b) Variable Learning Rate Backpropagation (TRAINGDX). TRAINGDX allows the learning rate to change during training process and attempt to keep the learning step size as large as possible while keeping learning stable [10]. This increases the learning rate without increases of large error.

c) Powell-Beale Restarts (TRAINCGB). TRAINCGB will restart if there is very little orthogonality left between the current gradient and the previous gradient and the search direction is reset to the negative of the gradient [10]. At each iteration, the step size is adjusted.

3.2 REC NETWORK

The type of REC networks used is Elman network (Figure 4). Elman networks are two-layer backpropagation networks with addition feedback connection from the output of the hidden layer to its input [9]. This feedback path allows Elman networks learn to recognise, generate temporal patterns and spatial patterns, stores values from the previous time step and use them in the current time step. Elman network used TANSIG transfer function in hidden (recurrent) layer and PURELIN neurons in output layer [11]. The training function used is TRAINGDX only. Trainscg and Traincgb were found to take too much time to train the ANN.

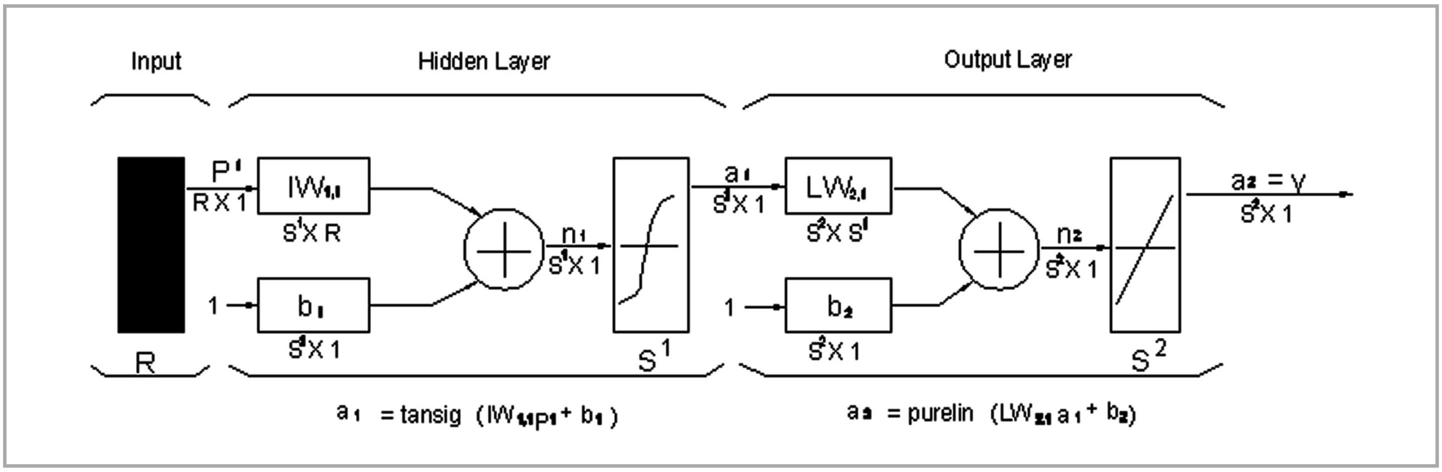

Figure 3: MLP Network Architecture

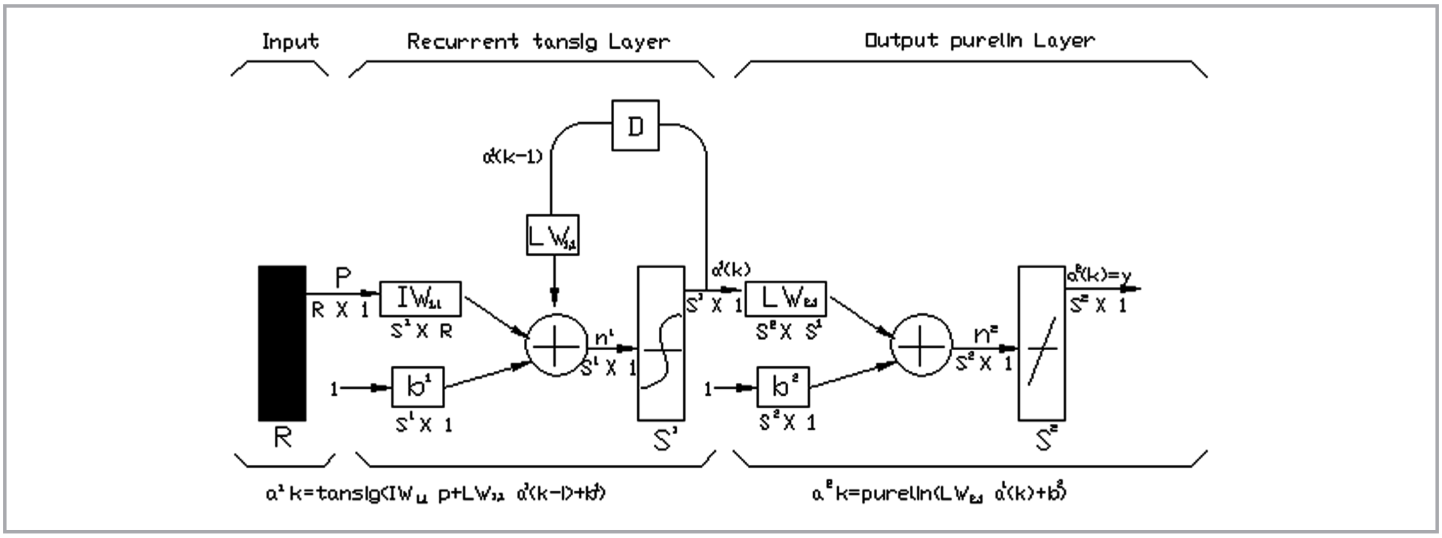

Figure 4: REC network architecture

Elman networks must have enough hidden neurons to divide the input space in a useful way. It will perform better when there are more hidden neurons than actually required. When fewer neurons are used, the ability of Elman network will be reduced in finding the most appropriate weights for hidden neurons since the error gradient is approximated [10].

4 MODEL DEVELOPMENT

Daily rainfall-runoff data were obtained from DID Hydrology Year Book for the year 1996, 1997, 1998 and 1999. Year 1999 is the most recent rainfall runoff data available at the time when this study was carried out. The ANNs was trained with 6, 12, 18, 24 and 27 months of length data. Trained neural networks were tested for 3 months of data (OCT 1999, NOV 1999 and DEC 1999). The performance of the network was evaluated using the coefficient of correlation (R) and Nash-Sutcliffe coefficient (E2) [13]. The formulas of these two coefficients are given in Table 1.

Table 1: Statistics for Model Comparison

Concept	Name	Formula
Coefficient of Correlation	R	$\dfrac{\sum(obs - obs)\,(pred - pred)}{\sqrt{\sum(obs - obs)^2 \sum(pred - pred)^2}}$
Nash-Sutcliffe Coefficient	E^2	$E^2 = 1 - \dfrac{\sum_i^j (obs - pred)^2}{\sum_i^j (obs - obs)^2}$

Note: obs = observed value, pred = predicted value, obs = mean observed values, pred = mean predicted values and N = number of values.

Input data used for daily rainfall-runoff models are antecedent total daily precipitation {P(t-1)………P(t-n)}, the total rainfall of the current day from each raingauge {P(t)}, antecedent daily mean discharges from the outlet gauge{Q(t-1)………Q(t-n)}. The output is the simulated mean daily runoff {Q(t)}. Equation 1 gives the equation of this nonlinear model

Q(t)={P(t),P(t-1),P(t-2), P(t-3)…P(t-n),Q(t-1),Q(t-2),Q(t-3)… Q(t-n)} (1)

The sequences of input arrangement are in order since the time is important.

Five models developed to investigate the number of antecedent events needed to obtain optimal results for daily runoff forecasting. These models are:

a) *Q(t)={P(t), P(t-1), Q(t-1)}* ------------------------------------- D1
b) *Q(t)={P(t), P(t-1), P(t-2), Q(t-1), Q(t-2)}* --------------------D2
c) *Q(t)={P(t), P(t-1), P(t-2), P(t-3), Q(t-1), Q(t-2), Q(t-3)}* -- D3
d) *Q(t)={P(t), P(t-1), P(t-2), P(t-3), P(t-4), Q(t-1), Q(t-2), Q(t-3), Q(t-4)}* --- D4
e) *Q(t)={P(t), P(t-1), P(t-2), P(t-3), P(t-4), P(t-5), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5)}* --------------------------------- D5

The models developed for daily runoff simulation will be trained and tested using MLP and REC networks respectively.

5 TRAINING AND TESTING OF THE MODEL

The models developed were trained and tested using TRAINSCG, TRAINGDX and TRAINCGB learning algorithm; 6, 12, 18, 24 and 27 months of training data; 100, 150, 200 and 250 number of neurons in the hidden layer; learning rate value of 0.2, 0.4, 0.6 and 0.8 respectively.

6. RESULTS
6.1 MLP NETWORK
6.1.1EFFECT OF DIFFERENT TYPES OF TRAINING ALGORITHM

Reasonable results were obtained using the three training algorithms. Optimum performance was obtained using TRAINSCG (Table 2). However, TRAINGDX and TRAINCGB require less training time compared to TRAINSCG. Comparison between simulated and measured runoff for MLPD4 using these three algorithms is shown in Figure 5.

Table 2: R Values of MLP Network with Different Training Algorithm

	TRAINSCG	TRAINGDX	TRAINCGB
MLPD1	0.955	0.915	0.952
MLPD2	0.968	0.966	0.958
MLPD3	0.939	0.950	0.958
MLPD4	0.969	0.928	0.938
MLPD5	0.860	0.909	0.899

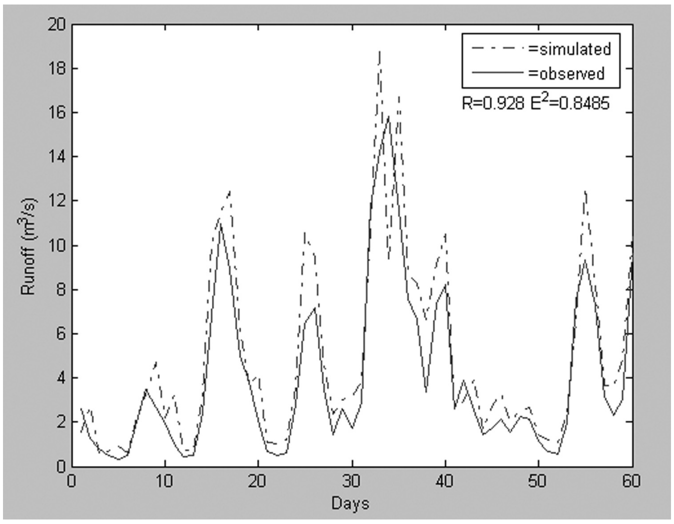

Figure 5a: FiMLPD4 Trained with TRAINSCGs

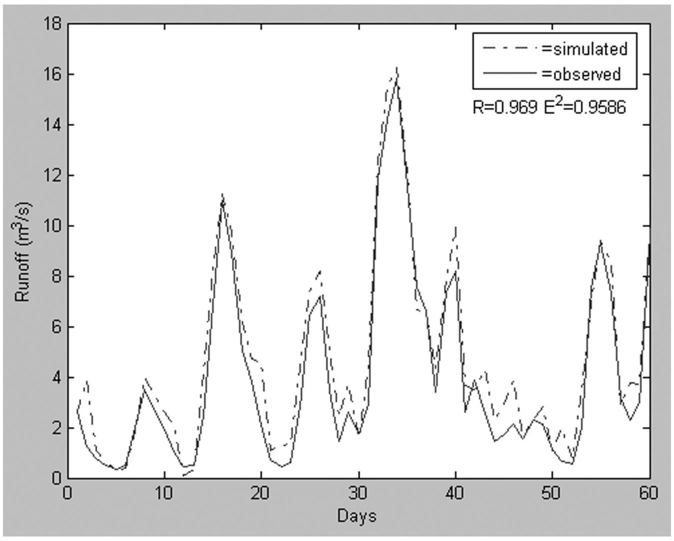

Figure 5b: MLPD4 Trained with TRAINGDX

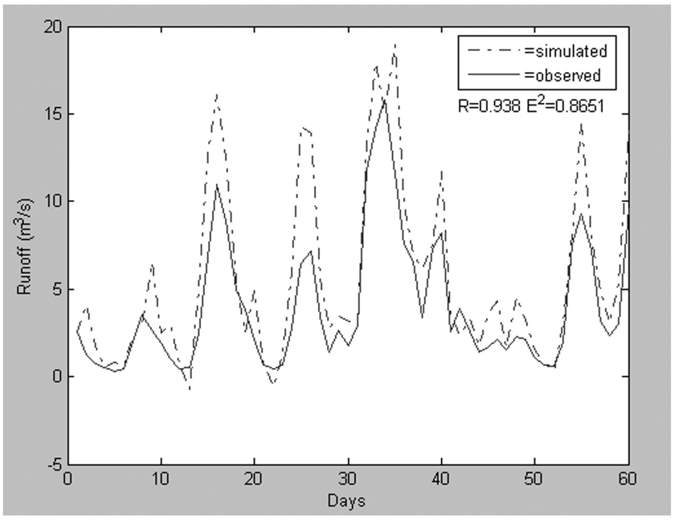

Figure 5c: MLPD4 Trained with TRAINCGB

Figure 5: Comparison between Simulated and Observed Runoff in Testing for MLPD4 Trained with TRAINSCG, TRAINGDX and TRAINCGB

6.1.2 EFFECT OF LENGTH OF TRAINING DATA

The Performance of MLP increased as the length of training increased (Table 3). A minimum of 12 months of training data is needed to obtain accurate results. If more data is available, than MLP network performs better as a more accurate determination of the synaptic weights is made by the ANN (Figure 6).

Table 3: Results for MLPD4 at Different Length of Training Data

Length of Training Data	R (Testing)	E^2 (Testing)
6 months	0.853	0.8851
12 months	0.936	0.8636
18 months	0.948	0.9222
24 months	0.952	0.9356
27 months	0.969	0.9586

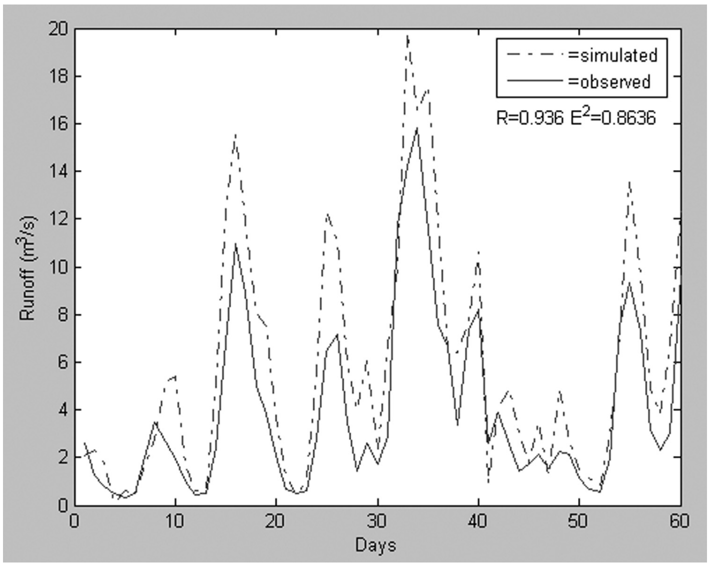

Figure 6a: Trained with 6 months training data

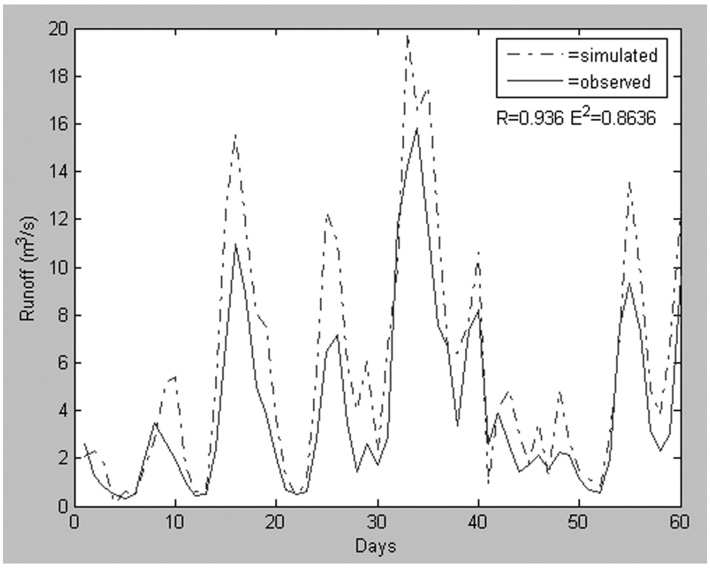

Figure 6b: Trained with 12 months training data

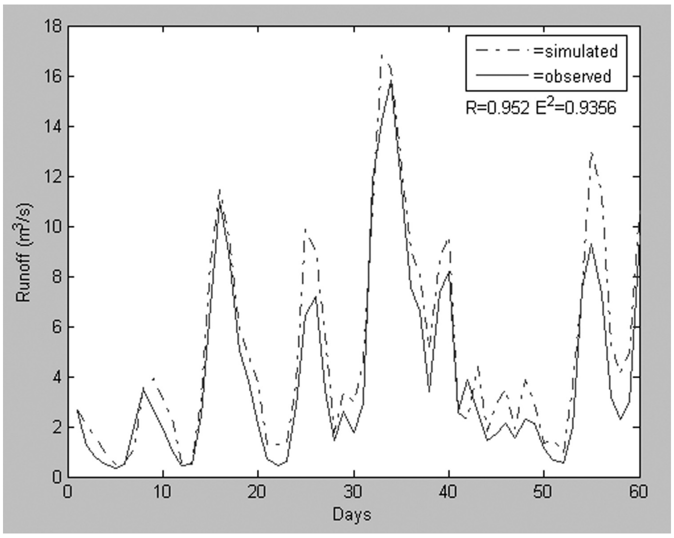

Figure 6c: Trained with 18 months training data

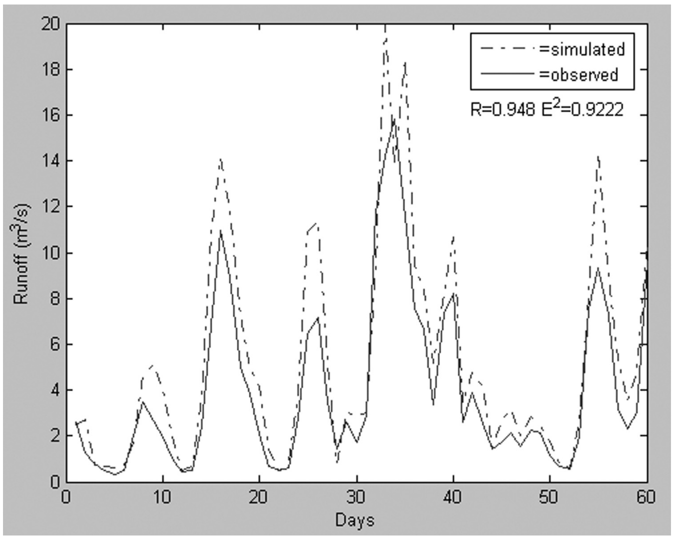

Figure 6d: Trained with 24 months training data

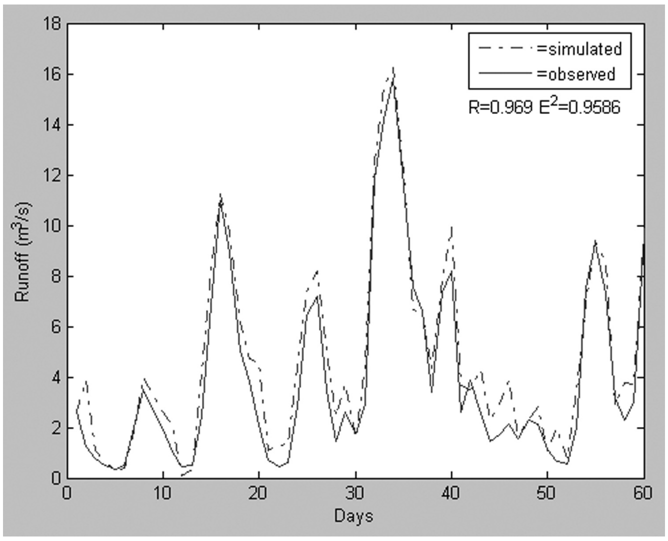

Figure 6e: Trained with 27 months training data

Figure 6: Comparison between Simulated and Measured Runoff for MLPD4 Trained with Different Length of Training Data

6.1.3 EFFECT OF NUMBER OF HIDDEN NEURONS DATA

The performance of MLP network increased with the increase of number of hidden nodes (Table 4 and Figure 7) and the best number of hidden neurons was found to be 250. However, training period was getting longer with the increase of number of hidden nodes.

Table 4: Results of MLPD4 at Different Number of Hidden Nodes

No. of Hidden Neurons	R (Testing)	E^2(Testing)
100	0.927	0.9251
150	0.952	0.9483
200	0.954	0.9552
250	0.969	0.9586

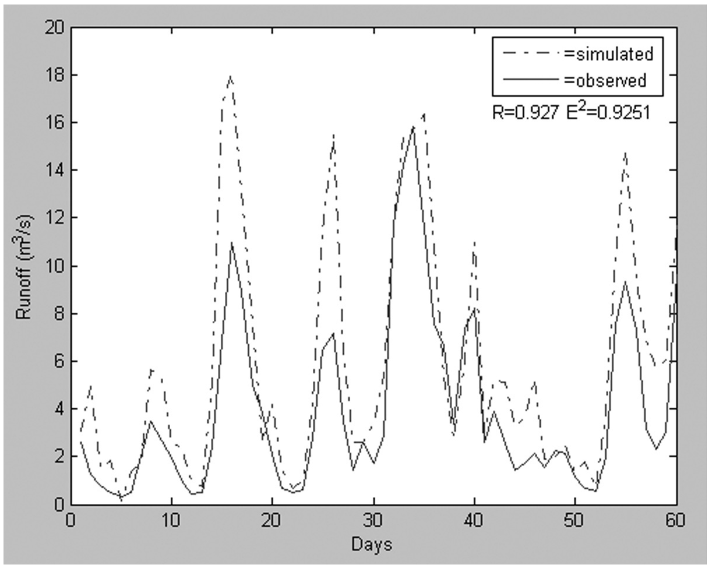

Figure 7a: Trained with 100 hidden neurons

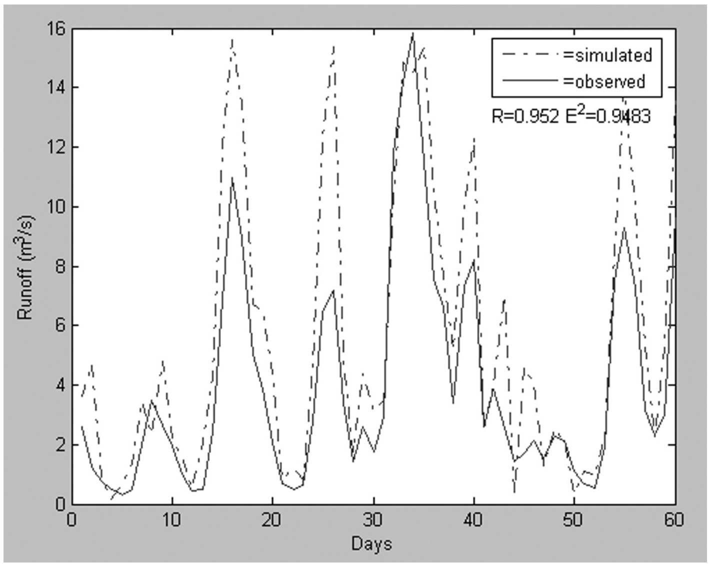

Figure 7b: Trained with 150 hidden neurons

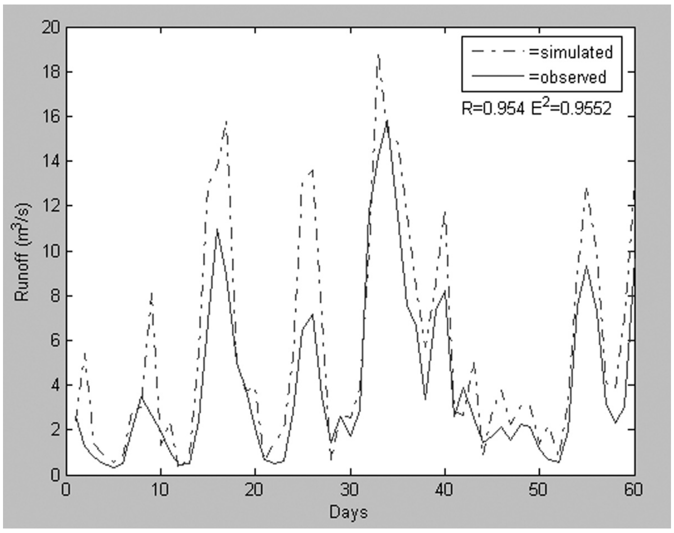

Figure 7c: Trained with 200 hidden neurons

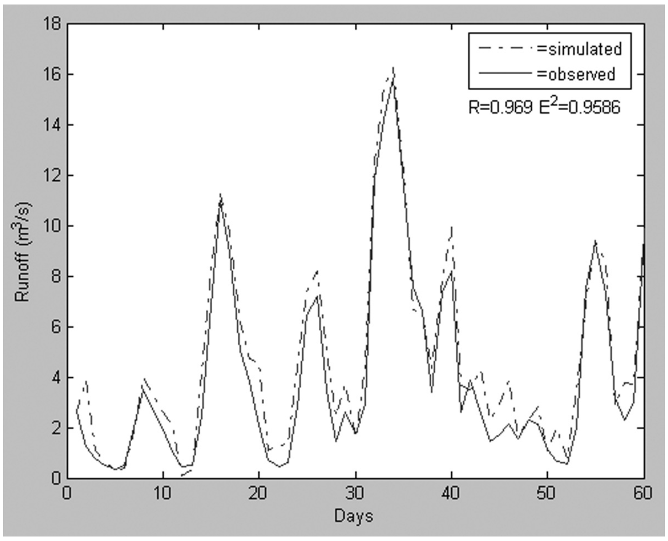

Figure 7d: Trained with 250 hidden neurons

Figure 7: Comparison between Simulated and Measured Runoff for MLPD4 Trained with Different Number of Hidden Neurons

Table 5: Results of MLPD4 at Different Learning Rate Value

Learning Rate	R (Testing)	E^2(Testing)
0.2	0.958	0.9429
0.4	0.968	0.9438
0.6	0.962	0.9468
0.8	0.969	0.9586

6.1.4 EFFECT OF LEARNING RATE VALUE

Table 5 show that although the learning rate was varied from 0.2 to 0.5 the performance of the ANN was affected. However, the learning rate had an effect on the training time, shorter learning rate increases the learning time. Therefore, it recommended to adopt a learning rate of 0.8.

Table 6: Results for MLP network at different number of antecedent days

No. of Antecedent Days	R (Testing)	E^2 (Testing)
MLPD1	0.955	0.9432
MLPD2	0.968	0.9508
MLPD3	0.939	0.9417
MLPD4	0.969	0.9586
MLPD5	0.860	0.8975

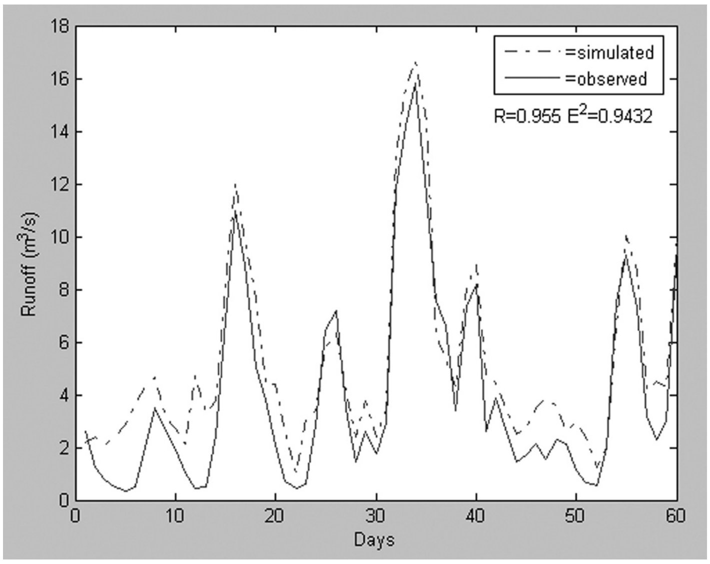

Figure 8a: Trained with 1 antecedent data

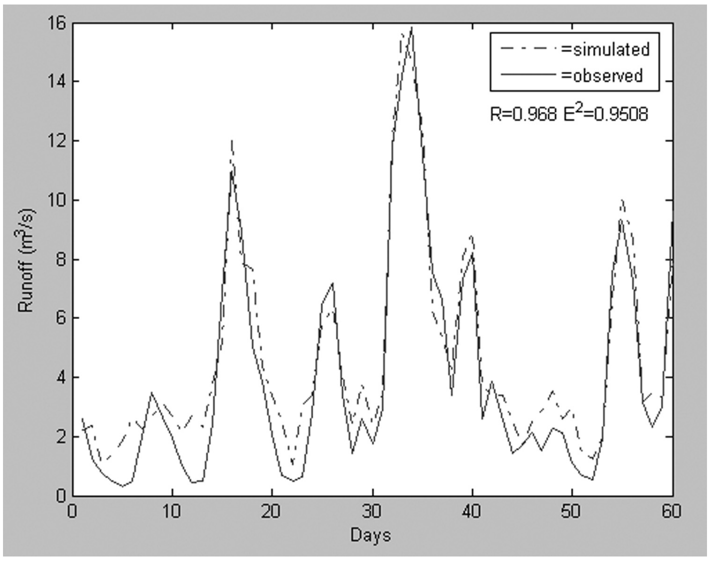

Figure 8b: Trained with 2 antecedent data

6.1.5 EFFECT OF ANTECEDENT DATA

As shown by Table 6 and Figure 8, increasing the number of antecedent data increases the accuracy of the results obtained. However, if the number of antecedent data is too high (more than 4 in this study), the performance of the ANN decreases. The optimum result was obtained using 4 days of antecedent data.

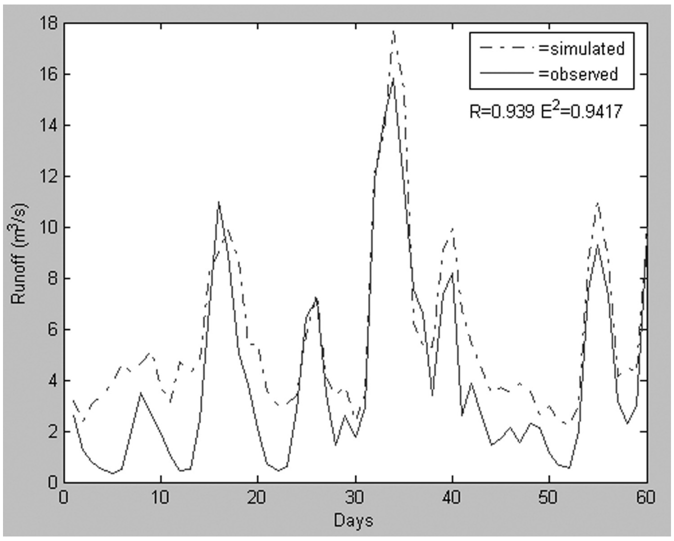

Figure 8c: Trained with 3 antecedent data

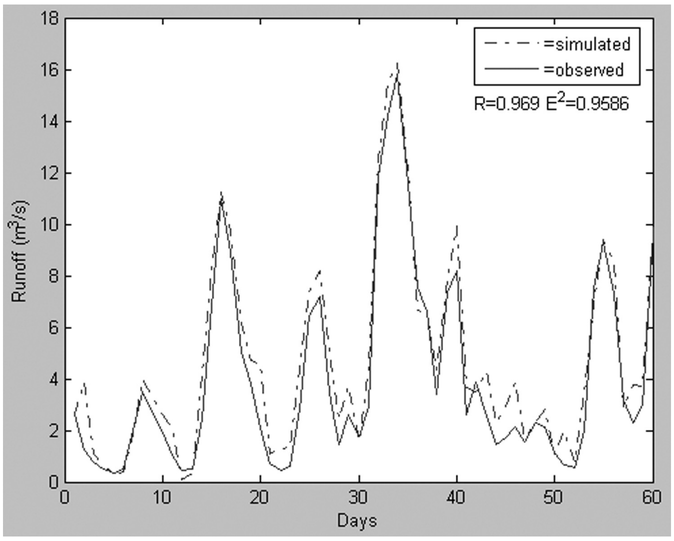

Figure 8d: Trained with 4 antecedent data

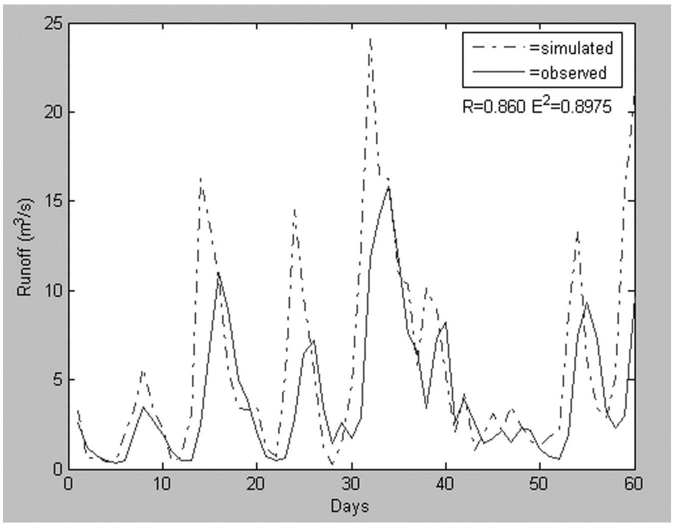

Figure 8e: Trained with 5 antecedent data

Figure 8: Comparison between Simulated and Measured Runoff in Testing for MLP Network with the Increase of Antecedent Days

Table 7: Results for RECD4 at different length of training data

Length of Training Data	R (Testing)	E^2 (Testing)
6 months	0.853	0.7885
12 months	0.969	0.8574
18 months	0.974	0.8858
24 months	0.979	0.9378
27 months	0.998	0.9958

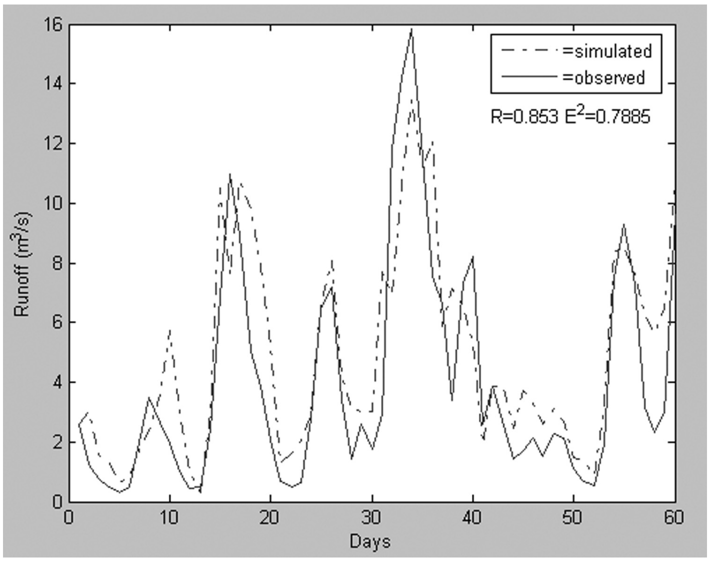

Figure 9a: Trained with 6 months training data

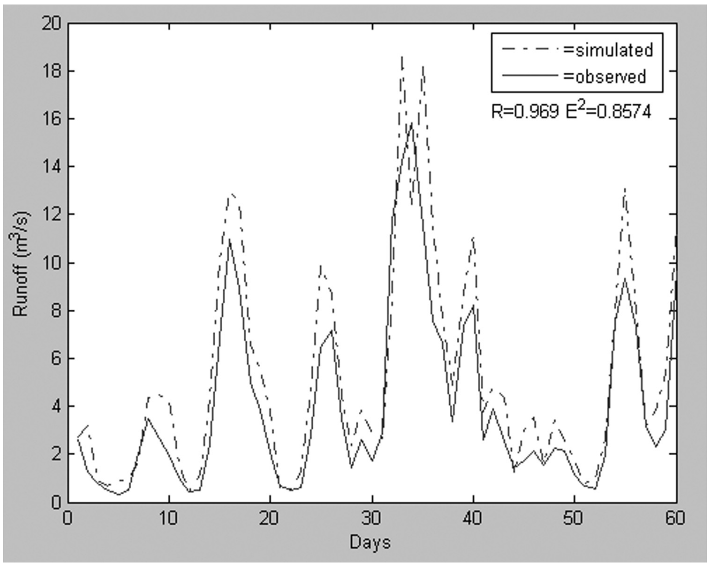

Figure 9b: Trained with 12 months training data

6.2 REC NETWORK
6.2.1 EFFECT OF LENGTH OF TRAINING DATA

Similarly with MLP networks the Performance of REC increased as the length of training increased. A minimum of 12 months of training data is needed to obtain accurate results (Table 7). If more data is available, than REC network performs better as the ANN makes a more accurate determination of the synaptic weights (Figure 9).

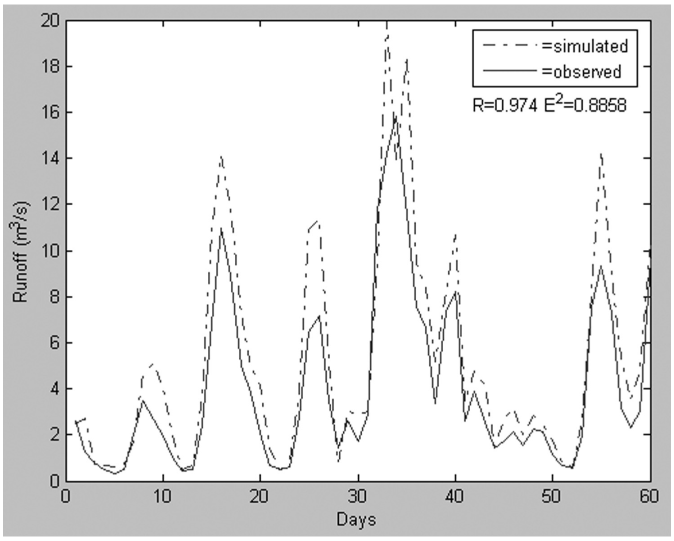

Figure 9c: Trained with 18 months Training Data

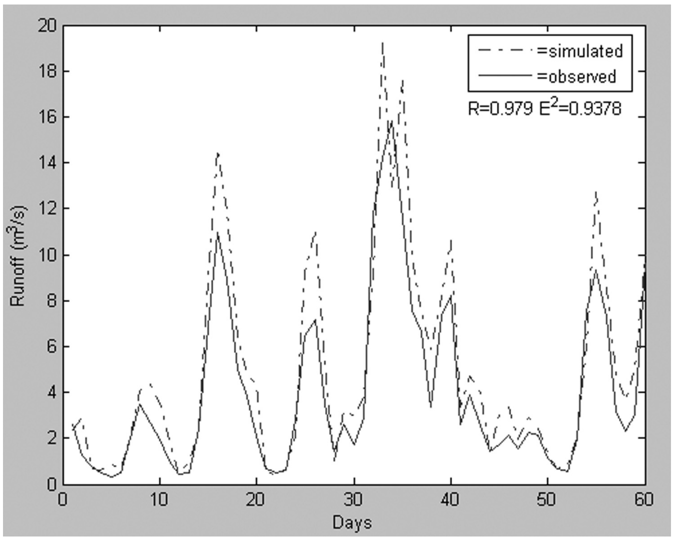

Figure 9d: Trained with 24 months Training Data

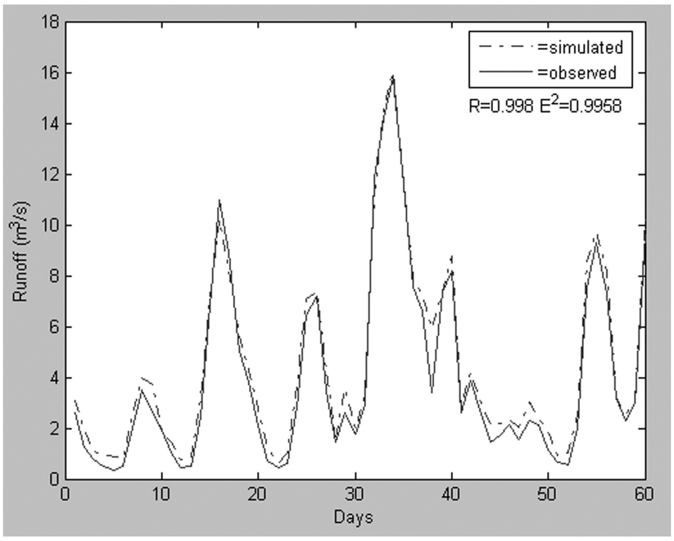

Figure 9e: Trained with 27 months Training Data

Figure 9: Comparison between Simulated and Measured Runoff for RECD4 Trained with Different Length of Training Data

6.2.2 EFFECT OF NUMBER OF HIDDEN NUERONS

As the number of hidden neurons increased, the performance of REC network increased too (Table 8 and Figure 10) and the optimum number of hidden neurons was found to be 250. Meanwhile, REC network also require longer training period with the increase of number of hidden neurons.

Table 8: Results of RECD4 at different number of hidden nodes

No. of Hidden-Nodes	R (Testing)	E^2 (Testing)
100	0.977	0.9093
150	0.976	0.9353
200	0.982	0.9766
250	0.998	0.9958

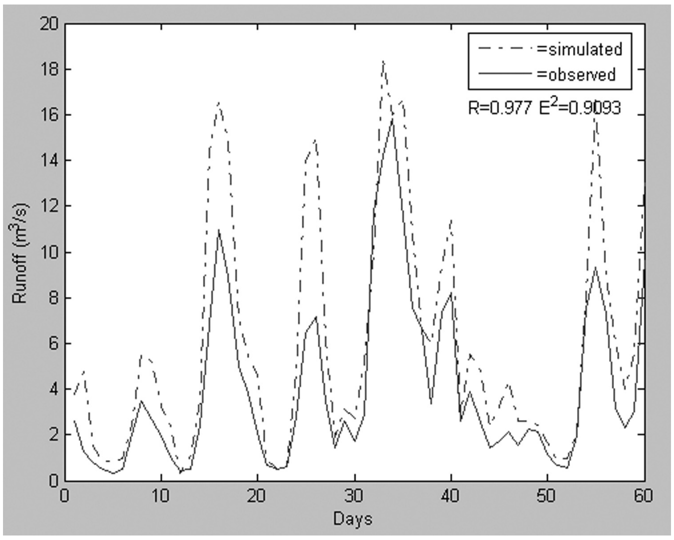

Figure 10a: Trained with 100 Hidden Neurons

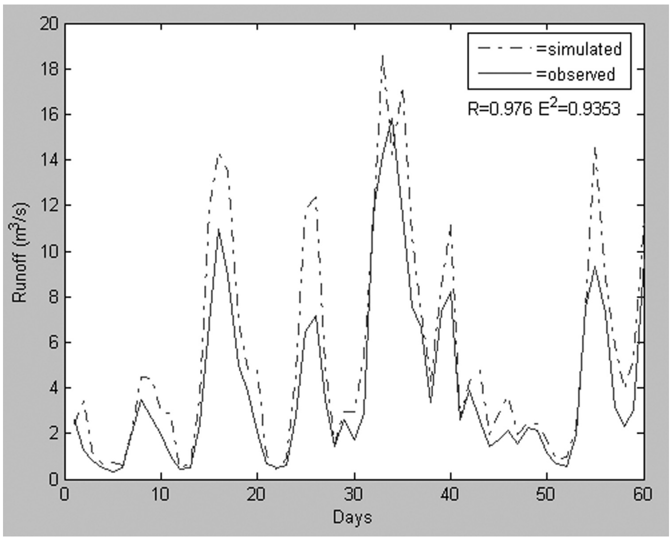

Figure 10b: Trained with 150 Hidden Neurons

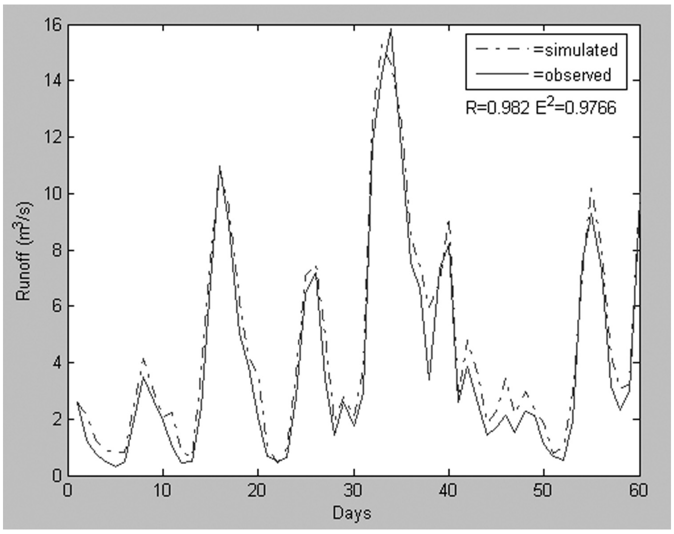

Figure 10c: Trained with 200 hidden neurons

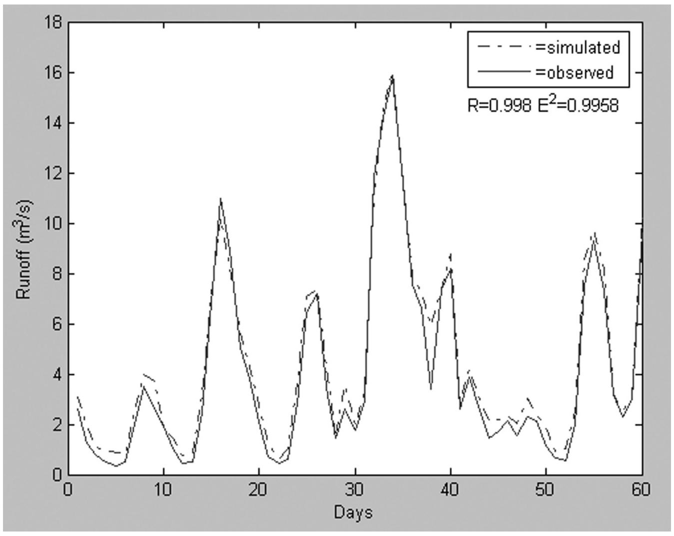

Figure 10d: Trained with 250 hidden neurons

Figure 10: Comparison between Simulated and Measured Runoff for RECD4 Trained with Different Number of Hidden Neurons

Table 9: Results of RECD4 at different learning rate value

Learning Rate	R (Testing)	E^2 (Testing)
0.2	0.996	0.9841
0.4	0.992	0.9718
0.6	0.985	0.9766
0.8	0.998	0.9958

6.2.3 EFFECT OF LEARNING RATE VALUES

The results show that the performance of REC network is not affected by the increase of learning rate value. Table 9 shows that there are no clear relationship between learning rate and the performance of REC network. However, training period decreased with the increased in learning rate. Hence, it is recommended to use learning rate value of 0.8 for REC network.

6.2.4 EFEECT OF ANTECEDENT DATA

The performance of REC network is kept on improving from RECD1 to RECD4. Similar to MLP network, REC network performed the best with 4 antecedent days (RECD4) as shown in Table 10 and Figure 11.

Table 10: Results for REC network at Different Number of Antecedent Days

No. of Antecedent Days	R (Testing)	E^2 (Testing)
RECD1	0.926	0.8996
RECD2	0.998	0.9886
RECD3	0.997	0.9920
RECD4	0.998	0.9958
RECD5	0.932	0.9525

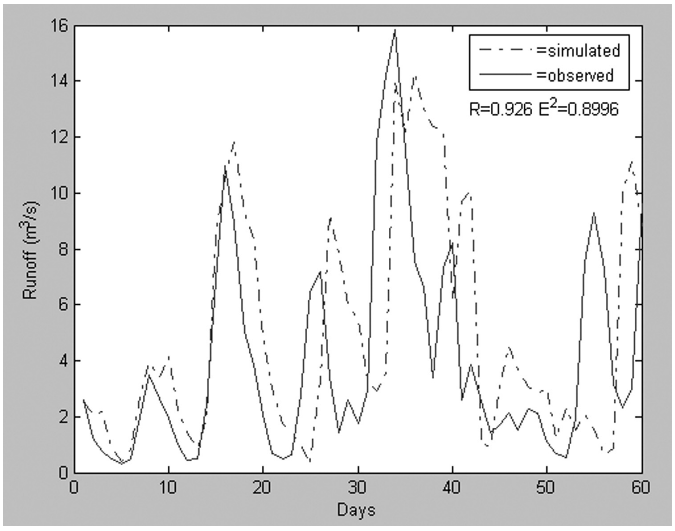

Figure 11a: Trained with 1 antecedent data

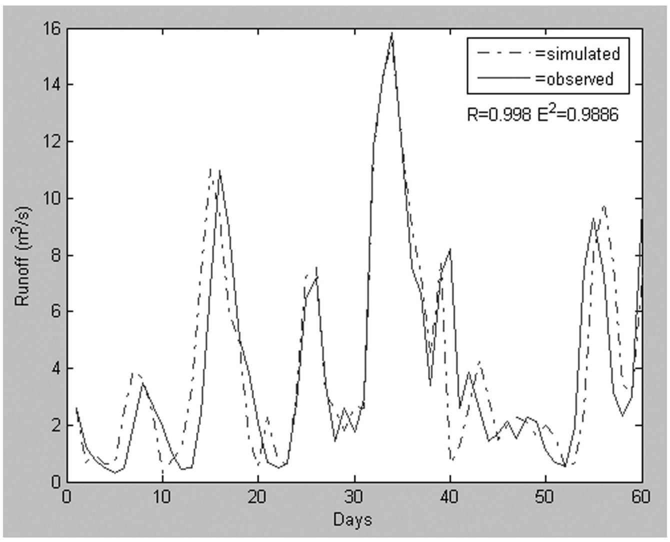

Figure 11b: Trained with 2 antecedent data

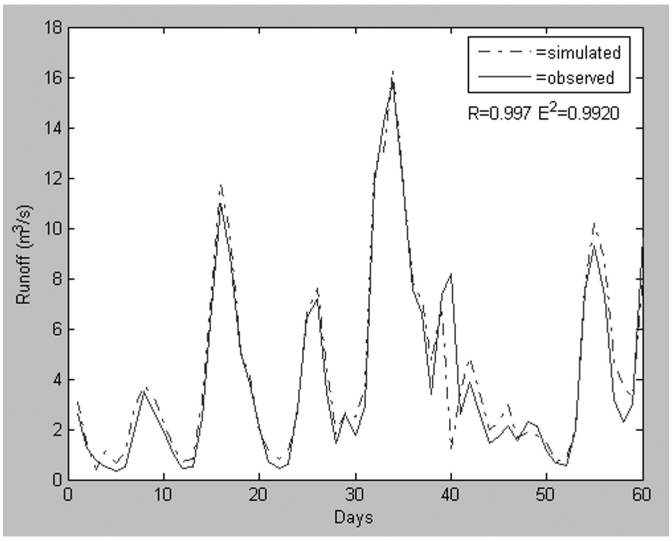

Figure 11c: Trained with 3 antecedent data

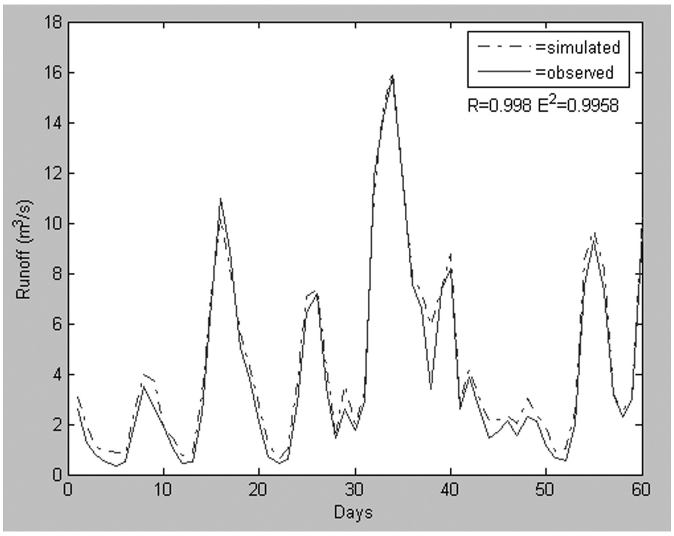

Figure 11d: Trained with 4 antecedent data

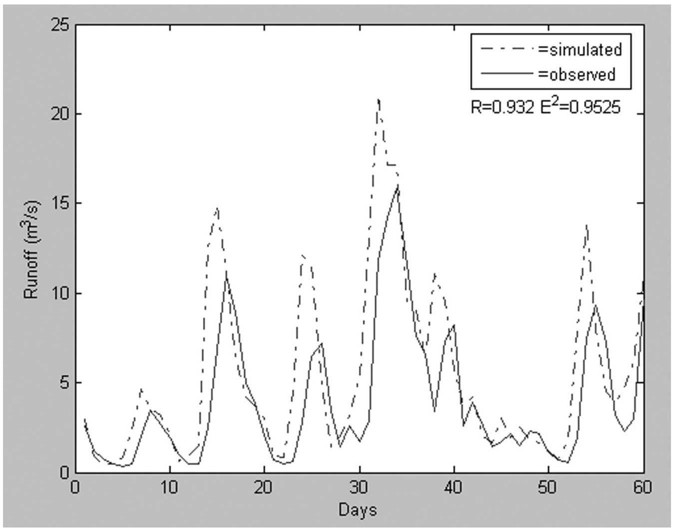

Figure 11e: Trained with 5 antecedent data

Figure 11: Comparison between Simulated and Measured Runoff for REC Trained with Different Number of Antecedent Days

6.3. COMPARISON OF THE TWO ANNS FOR DAILY RUNOFF

REC and MLP network have shown encouraging results in terms of simulating daily runoff (Figures 12 and 13). In comparison with MLP, the performance of REC was higher (R=0.998 for REC compared with R=0.969 for MLP, E^2=0.9958 for REC compared with E^2=0.9586 for MLP).

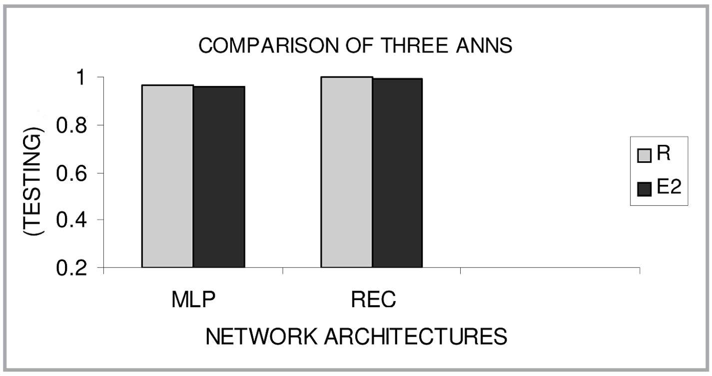

Figure 12: Comparison Between MLPD4, RECD4 for Daily Runoff

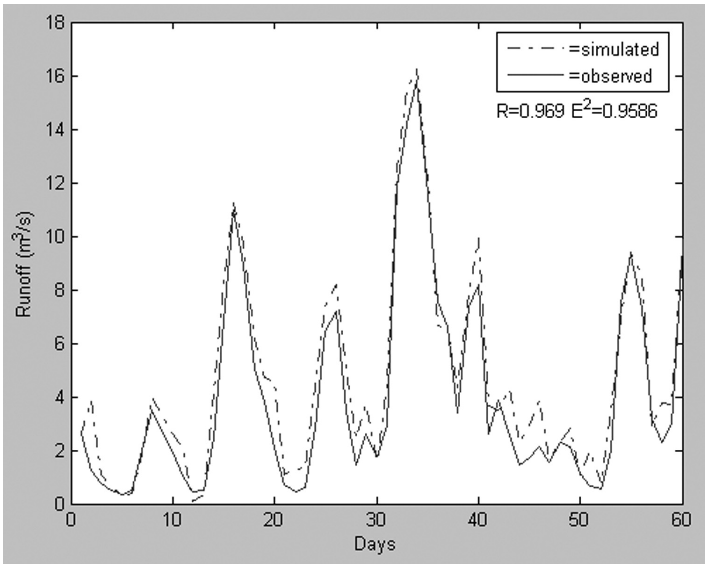

Figure 13a: MLPD4

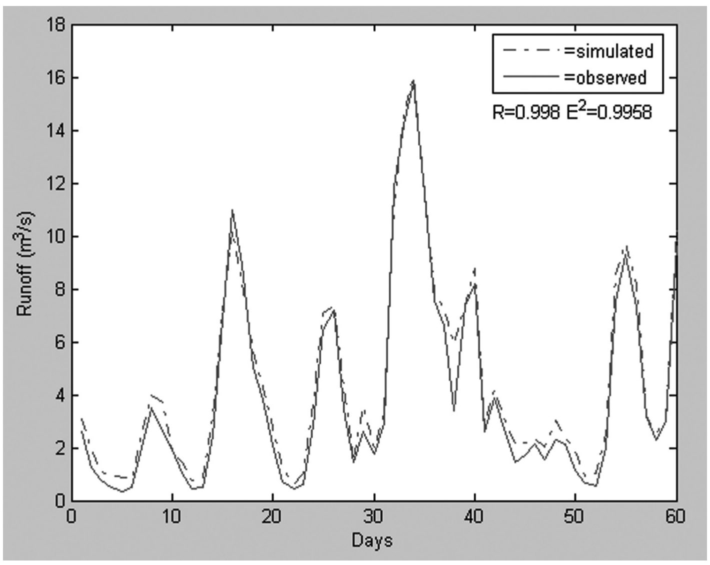

Figure 13b: RECD4

7. SIMULATION FOR LONGER PERIOD OF TIME

The optimal configurations of MLP and REC networks have given very good results when tested for three months. In this section these two ANNs will be tested to see how they will perform if they have to simulate runoff for 6 months (January to June 2003) or 1 year (July 2002 to June 2003). The results obtained are shown in Table 11 and Figure 14.

Table 11 shows that the results for the 6 months or 12 months simulation are satisfactory (R=0.945 and 0.937 respectively). It can also be seen that as the simulation time increased, the performance of both ANNs decreased. The Performance of the REC is better than MLP.

Table 11: Results for the three ANNs Tested with 6 and 12 months data

	6 months		12 months	
	R	E^2	R	E^2
MLPD4	0.935	0.9431	0.923	0.9267
RECD4	0.945	0.9550	0.937	0.9353

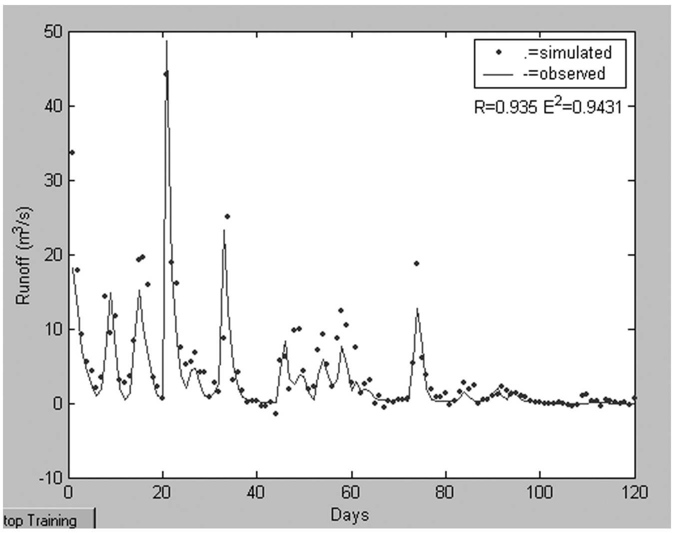

Figure 14a: MLPD4 Tested with 6 months data

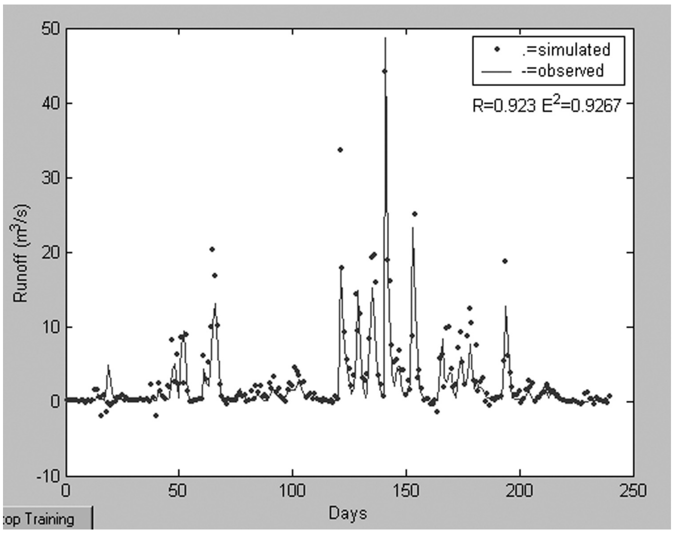

Figure 14b : MLPD4 Tested with 12 months data

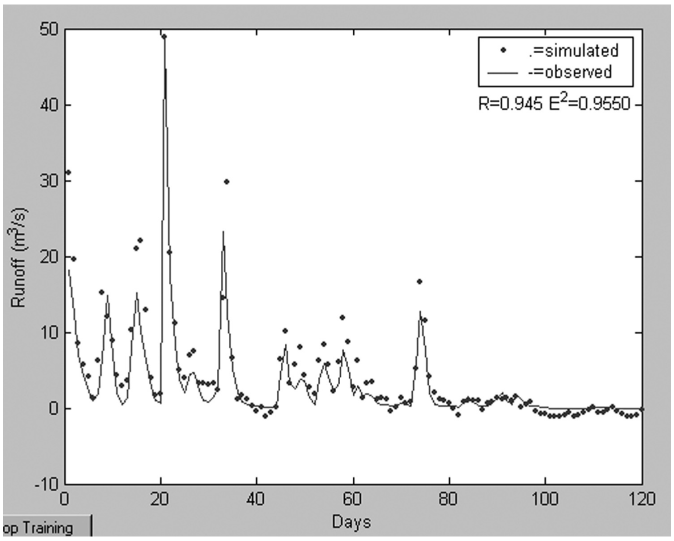

Figure 14c: RECD4 Tested with 6 months data

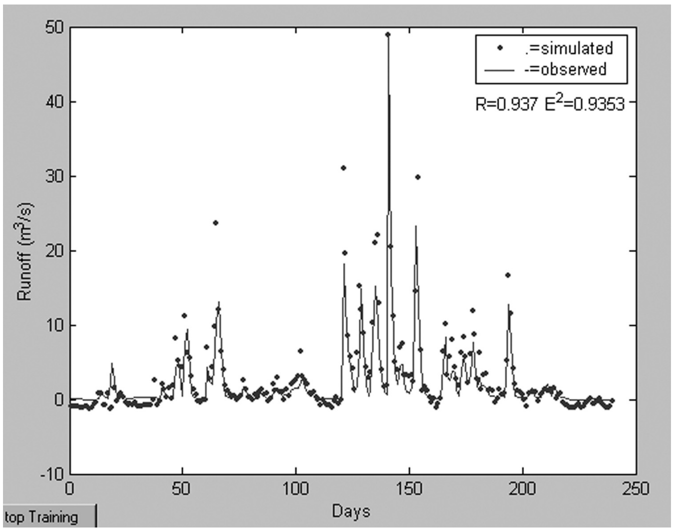

Figure 14d: RECD4 Tested with 12 months data

Figure 14: Performance of MLPD4, RECD4 Tested with 6 and 12 months of data

8. CONCLUSION

In this investigation, two types of ANNs architectures namely MLP and REC have shown that they can predict daily runoff accurately for Sungai Bedup Basin. The results have shown that the best neural network for daily runoff prediction is RECD4 as the R and E^2 values given are higher than MLPD4. Both MLP and REC networks required 4 antecedent data to perform best.

For three months simulation, it was found that the model needs at least 12 months of training. The performance of ANN increases with more training data. If 27 months were used, it was found that R=0.969 in the case of MLPD4 and 0.998 for RECD4. It was also shown that if 27 months of data were used for training, the ANN can simulate accurately for 6 months (R=0.935 for MLPD4 and R=0.945 for RECD4) and 12 months (R=0.923 for MLPD4 and R=0.937 for RECD4).

This study shows that it is not necessary to include the lag time as input. The ANNs tested did demonstrate the ability of ANNs to adapt to the respective lag time of each gauge through training. For catchment in tropical region, rainfall and runoff are sufficient as inputs to develop rainfall-runoff model. Inclusion of more parameters such as temperature, moisture content, evaporation will make the ANNs more complex, the learning time very long and this may decrease the performance of ANNs. ■

REFERENCES

[1] Bessaih, N., Mah, Y. S., Muhammad, S.M., Kuok, K.K., and Rosmina, A.B., *"Artificial Neural Networks for Daily Runoff Simulation"*, Faculty of Engineering, Universiti Malaysia Sarawak, 2003.

[2] Dastorani, M.T., and Wright, N.G., *"Artificial Neural Net work Based Real-time River Flow Prediction"*, School of Civil Engineering, University of Nottingham, Nottingham NG7 2RD, UK, 2001.

[3] Demuth, H., and Beale, M., Neural Network Toolbox - For Use With MATLAB, The Math Works, Inc, 2001.

[4] Elshorbagy, A., Simonovic , S.P., and Panu, U.S., *"Performance Evaluation of Artificial Neural Networks for Runoff Prediction"*, Journal of Hydrologic Engineering, 5(4): 424-427, 2000.

[5] Garcia-Bartual, R., *"Short Term River Flood Forecasting with Neural Networks"*, Universidad Politecnica de Valencia, Spain, 160-165, 2002.

[6] Gautam, M.R., Watanabe, K., and Saegusa, H., *"Runoff Analysis in Humid Forest Catchment with Artificial Neural Networks"*, Journal of Hydrology, 235: 117-136, 2000.

[7] Harun, S., Kassim, A.H., and Van, T.N., *"Inflow Estimation with Neural Networks"*, 10th Congress of The Asia and Pacific Division of the International Association for Hydraulic Research,150-155, 1996.

[8] Idris, A., MATLAB for Engineering Students, Prentice Hall, Pearson Education Malaysia Sdn. Bhd, 2000.

[9] Imrie, C.E., Durucan, S. and Korre, A., *"River Flow Prediction Using Artificial Neural Networks: Generaliza tion Beyond the Calibration Range"*, Journal of Hydrology, 233: 138-153, 2000.

[10] Jain, S.K., and Chalisgaonkar, C., *"Setting Up Stage-Discharge Relations Using ANN"*, Journal of Hydrologic Engineering, 5(4), 424-433, 2000.

[11] Palm, W.J., Introduction to MATLAB 6 for Engineers, Mc Graw-Hill Companies, Inc. United States of America, 2001.

[12] Wright, N.G., and Dastorani, M.T., *"Effects of River Basin Classification on Artificial Neural Networks Based Ungauged Catchment Flood Prediction"*, Proceeding of the 2001 International Symposium on Environmental Hydraulics, 2001.

[13] Wu, J.K., Neural Networks and Simulation Methods, Marcel Dekker, Inc. United States of America, 1994.

PROFILES

KUOK KING KUOK
After completion of first degree from University Technology of Malaysia in 1999, worked as a design engineer in a few private consulting firms for three years and was mainly involved in structural, civil and road design. Meanwhile, enrolled as a research student on part time basis in University Malaysia Sarawak and was graduated with Master of Engineering major in Hydrology in 2004. Currently, work as civil engineer with Department of Irrigation and Drainage Sarawak and enrolled as a postgraduate student on part time basis with University Technology of Malaysia.

NABIL BESSAIH
Nabil Bessaih graduated with Bachelor in Civil Engineering from University of Algeria. Then, he furthers his study in Salford University, UK and graduated with PhD. He worked as Associate Professor in UNIMAS for 10 years and returned to Algeria in 2005. Currently, he is working with one of the government agencies in Algeria.