# Adaptive Multilayer Perceptron Model for Hourly Streamflow Hydrograph

*[1]Nor Irwan Ahmat Nor, [2]Amir Hashim Mohd. Kassim*
*Department of Water Resources and Environmental Engineering*
*Faculty of Civil and Environmental Engineering, Kolej Universiti Teknologi Tun Hussein Onn,*
*86400 Parit Raja, Batu Pahat, Johor, Malaysia*
*E-mail: [1]nisfan@kuittho.edu.my, [2]amir@kuittho.edu.my*

*[3]Sobri Harun*
*Department of Hydraulic and Hydrology*
*Faculty of Civil Engineering, Universiti Teknologi Malaysia,*
*81310 UTM Skudai, Johor, Malaysia*
*E-mail: [3]sobri@fka.utm.my*

## Abstract

The modelling of hydraulic and hydrological processes is important in view of the many uses of water resources such as hydropower generation, irrigation, water supply, and flood control. There are many previous works using the artificial neural network (ANN) method for modelling various complex non-linear relationships of hydrologic processes. The ANN is well known as a flexible mathematical structure and has the ability to generalize patterns in imprecise or noisy and ambiguous input and output data sets. In this study, the multi-layer feedforward neural network is applied in the context of rainfall-runoff modelling on the hourly data of selected catchment. The methodology is assessed using multilayer perceptron (MLP) to predict hourly runoff as a function of hourly rainfall for the Sungai Bekok catchment (Johor, Malaysia). Further, the results are compared between ANN and HEC-HMS approach model. It has been found that the ANN models show a good generalization of rainfall-runoff relationship and is better than HEC-HMS model.

## Keywords

Artificial Neural Network (ANN); Multilayer perceptron(MLP), Rainfall-Runoff Modelling, HEC-HMS

## Introduction

Determining the relationship between rainfall and runoff for watersheds is one of the most important problems faced by hydrologists and engineers. The problems are their attempt to provide rational answers to problems that arise by issues if non-linearity of physical processes, uncertainty in parameter estimates, the situation of an ungauged catchments, etc. The runoff is critical to many activities such as designing flood protection works for urban areas and agricultural land and assessing how much water may be extracted from a river for planning, design and management of water supply, irrigation, drainage system, etc. It is necessary for the investigation of complex system in cities, where a huge amount of data is needed and utilized.

Many approaches have evolved over last few decades in hydrological modelling and forecasting. They are deterministic as well as stochastic in nature, and include conceptual and statistical methods. MIKE11-NAM, HEC-HMS, RORB, MIKE-SHE, SWMM, etc. are the conceptual and physically based hydrological models. The methods used in this commercial model are classical methods and can be used as comparison and guidance for further research development. These models generally applied quite similar methodology. For example, losses rate, initial loses, transfer function, transform rate, lag time, deficit, and constant. These methods require many parameters for calibration and sometimes still cannot fit well in calibration processes. So that empiricism plays an important role in modelling studies in the new era. The current trend seems to be to model the data rather than the physical process.

The rapid increase in the capacity of modern computers has opened up a new world of methodologies for mathematical modelling. These researches focus on the application of new approach to solve problem in hydrology. The natural behaviour of hydrological processes is appropriate for the application of ANN method. The Artificial Neural Network (ANN) technique is proposed as a new improvement to reduce problems of data collection and calibration processes. An ANN can be defined as 'a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain' [1]. An attractive feature of ANN is their ability to extract the relation between the inputs and outputs of a process, without the physics being explicitly provided to them, and even if the data is noisy and contaminated with errors. The ANN models have been used successfully to model complex non-linear input-output relationships in an extremely interdisciplinary field. This new technique also shows a better performance and the time taken in modelling process is much shorter than the commercial models. The ANN method has been proven to be potentially useful tools in hydrological modelling such as for rainfall-runoff modeling processes [2, 3, 4, 5]; flow prediction [6, 7]; water quality predictions [8]; operation of reservoir system [9, 10]; and groundwater reclamation problems, [11]. This study employs the multilayer perceptron (MLP) method to model the event-based rainfall-runoff relationship. The objectives of this study are to examine and evaluate how successful ANN has been utilized in rainfall-runoff modelling.

## Study Area

The modelling work is carried out using the 10 years of rainfall and runoff records of Sungai Bekok (Johor, Malaysia) as shown in Fig. 1. The Sungai Bekok is a natural catchment with a size of 350 km$^2$. It is located on the southwestern part of Johor, latitude 02$^0$ 07' 15" and longitude 103$^0$ 02' 30". Fig. 1 also illustrates the location of the raingauges and water level gauging stations.

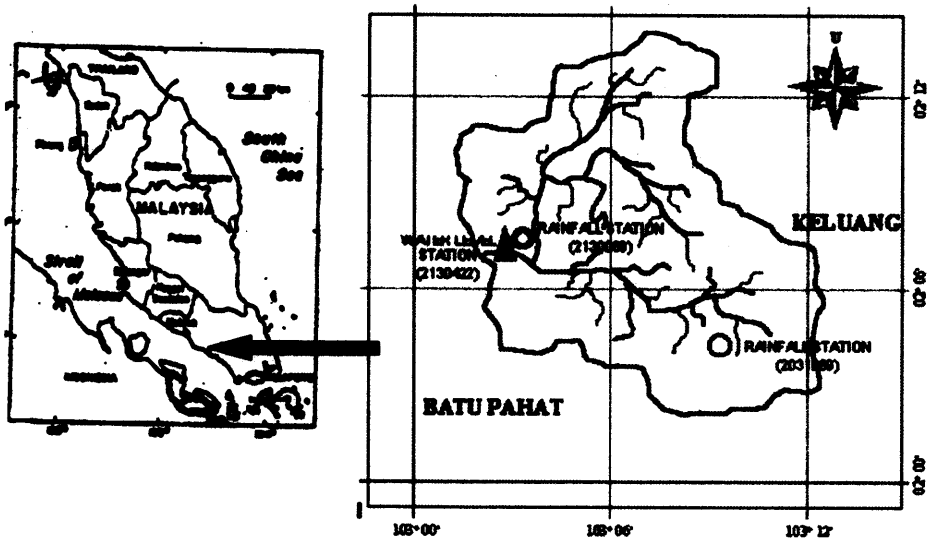



**Fig. 1** : Sungai Bekok catchment area

## Artificial Neural Network (ANN)

The MLP consists of three layers: the input layer, where the data are introduced to the network; the hidden layer, where the data are processed (that can be one or more) and the output layer, where the results for given inputs are produced. The architecture of an ANN is designed by weights between neurons, a transfer function that controls the generation of output in a neuron, and learning laws that define the relative importance of weights for input to a neuron [12]. It will process the information in a way that it is previously trained, to generate satisfactory results. Neural network can learn from experience, generalize from previous examples to new ones, and abstract essential characteristics from inputs containing irrelevant data [13]. The main control parameters of ANN model are interneuron connection strengths also known as weights and the biases. [5] reported that the most commonly used

activation function in the backpropagation network is the hyperbolic-tangent (tansig) functions. It is a continuous transfer functions that accomplished the modification of the network weights. This function is nonlinearity, differentiable, which is antisymmetric with respect to the origin and for which the amplitude of the output lies between –1 and +1. The transfer function also introduces a nonlinearity that further enhances the network's ability to model complex functions. Its functional form determines the nonlinear response of a node to the total input signal it receives to produce a continuous value. Normally, the output layer is chosen a linear activation function.

**Training of ANN**

There are two types of training approaches to training ANN namely supervised learning and unsupervised learning. Supervised learning is the most common type of learning used in ANN. Adapting the values of the weight and thresholds by presenting the input and output data is known as learning or training. Training is the actual process of adjusting weight factors based on trial-and-error. The training of a MLP is usually performed with the backpropagation algorithm, which is designed to minimize the mean square error between the actual output and the desired output. Back-propagation algorithm is the most popular algorithm for the supervised training of multilayer perceptrons to adjust the interconnection weights, [3, 5, 13]. [1] reported that, today it is estimated that 80% of all applications utilize this backpropagation algorithm in one form or another. It is a gradient (derivative) technique that are simple to compute locally, and it performs stochastic gradient descent in weight space (for pattern by pattern updating of synaptic weights). The back-propagation algorithm involves two steps. The first step is a forward pass, in which the effect of the input is passed forward through the network to reach the output layer. After the error is computed, a second step starts backward through the network. The errors at the output layer are propagated back toward the input layer with the weight being modified. In any training algorithm, the objective is to reduce the error $E$ that is defined as,

$$E = \frac{1}{p}\sum_{P=1}^{P} E_p \tag{1}$$

where $P$ is the total number of training patterns; and $E_p$ is the error for training pattern, $p$

$$E_p = \frac{1}{2}\sum_{k=0}^{N}(y_k - t_k)^2 \tag{2}$$

where $N$ is the total number of output nodes; $y_k$ is the network output at the $k$ th output node and $t_k$ is corresponding ANN output at the $k$ th output node.

The architecture of a typical neuron is shown in Fig. 2. Input layer is the previous rainfall data and the output layer constitute the runoff data. Each layer is made up of several nodes, and layers are interconnected by sets of correlation weights. Each input node unit ($i=1,...,m$) in input layer broadcasts the input signal to the hidden layer. Each hidden node ($j=1,...,n$) sums its weighted input signals,

$$z\_in_j = w_{0j} + \sum_{i=1}^{m} x_i w_{ij} \tag{3}$$

applies its activation function to compute its output signal from the input data as

$$z_j = f(z\_in_j) \tag{4}$$

and sends this signal to all units in the hidden layer. Note that $w_{ij}$ is the weight between the input layer and the hidden layer, $w_{0j}$ is the weight for the bias; and $x_i$ is the input rainfall signal. In this study, a sigmoid function used is tansig as proposed by [5]. This function is continuous, differentiable everywhere, monotonically increasing, and it is the most commonly used function in the backpropagation networks. The tansig activation function will process the signal that passes from each node by,

$$f(z\_in_j) = \frac{2}{1+e^{-2z\_in_j}} - 1 \tag{5}$$

Then, from the second layer, the signal is transmitted to the third layer. The output unit ($k=1$) sums its weighted input signals as,

$$x\_in_k = c_0^{(k)} + \sum_{j=1}^{n} z_j c_j^{(k)} \tag{6}$$

and applies its activation function to compute its output signal,

$$\vec{y}^{(k)} = f(x\_in_k) \tag{7}$$

where $c_j^{(k)}$ is the weight between the second layer and the third layer, and $c_0^{(k)}$ is the weight for the bias. The output node ($k=1$) receives a target pattern corresponding to the input training pattern, computes its error information term,

$$\delta_k = (t_k - \vec{y}^{(k)}) f'(x\_in_k) \tag{8}$$

calculates its weight correction (used to update $c_j^{(k)}$ later), and

$$\Delta c_j^{(k)} = \alpha \delta_k z_j \tag{9}$$

calculates its bias correction term (used to update $c_0^{(k)}$ later),

$$\Delta c_0^{(k)} = \alpha \delta_k \tag{10}$$

where $\alpha$ is learning rate; $t_k$ is the target neural network output; $\vec{y}^{(k)}$ is the neural network output as input variable; and $f'(x) = f(x)[1-f(x)]$. The error information is transferred from the output layer back to earlier layers to update weights and biases. This is known as the backpropagation of the output error to the input nodes to correct the weights. Each output node $\vec{y}_k$ ($k = 1,2,...,m$) updates its bias and weights ($j = 0,1,...,m$),

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij} \tag{11}$$

Each hidden node $z_j$ ($j = 1,2,...,n$) updates its bias and weights ($i = 0,1,...,m$). The process is terminated when this difference achieves a specified value. The training phase needs to produce an ANN that is both stable and convergent, to produce accurate input-output relations. After this, the network can be tested using data that have not been assigned during learning.

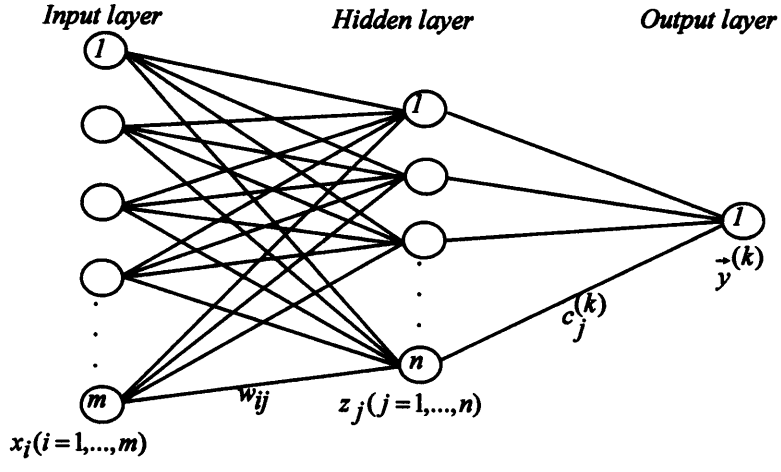



**Fig. 2**: Structure of a MLP model with one hidden layer

**Levenberg-Marquardt (LM) Algorithm**

In the current study, the Levenberg-Marquardt (LM) algorithm is used. The LM algorithm is an approximation to Newton's Method [14]. Newton's method is an alternative to the conjugate gradient methods for fast

optimisation and often converges faster than conjugate gradient methods (see [15]). The LM algorithm uses this approximation to the Hessian matrix in the following Newton's method,

$$w_{k+1} = w_k - H_k^{-1} g_k \tag{12}$$

$$= w_k - \left[J^T J + \mu I\right]_k^{-1} J^T e \tag{13}$$

$$w_{k+1} = w_k + \Delta w \tag{14}$$

where $w_k$ is a vector of current weights and biases, $g_k$ is the current gradient and $\Delta w = -H_k^{-1} g_k$. This equation is applied iteratively, with the computed value of $w_{k+1}$ being used repeatedly as the 'new' $w_k$. When the scalar $\mu$ is zero, this is just Newton's method, using the approximation Hessian matrix. When $\mu$ is large, this become gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, $\mu$ is decreased after each successful step and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

**Selection of the number of hidden layers and the number of hidden nodes**

Determination of structure of hidden layers and number of neurons or nodes is important in the multilayer perceptron modelling. There are no hard and fast rules for defining network parameters. [16] gave three simple guidelines to follow. The first is to use one hidden layer; second is to use very few hidden neurons; and the third is to train the model until we get the best optimal number of layers and nodes. It is important to note that backpropagation can be applied to an artificial neural network with any number of hidden layers [17]. There are no fixed rules about the number of neurons in the hidden layer. However, if this number is small, the network may not have sufficient degrees of freedom to learn the process correctly, and if the number is too high, the network will take a long time to get trained and may sometimes over fit the data [18]. The hidden layers enhance the network ability to model complex functions. A trial and error procedure is generally applied in selecting the number of hidden layers and in assigning the number of nodes to each of these layers. An appropriate number of neurons can be found by calibrating the network and evaluating its performance by increasing the number of hidden layer neurons in order to obtain high efficiency with as few neurons. If the hidden layer has too many neurons, then there are too many parameters to be estimated. [19] proposed that normally, neural networks were developed using 15, 30, 45, 60 and 100 hidden nodes. This procedure also investigates over the performance of neural network model with different number of hidden nodes.

## Application of HEC-HMS Model

Hydrologic Modelling System (HEC-HMS) program was developed by a team of Hydrologic Engineering Center, US Army Corps of Engineers, lead by the Director, Darryl Davis [14]. The program features a completely integrated work environment including a database, data entry utilities, computation engine, and results reporting tools. HEC-HMS was run with the previous hourly rainfall-runoff data in order to provide hourly prediction of runoff entering selected catchments. For Sungai Bekok catchment, the models used are Initial-Constant infiltration/loss parameterisation, the Clark hydrograph transformation routine, and a recession base flow component. The initial loss and initial flow are treated as initial conditions and vary from simulation to simulation. Calibration parameters for the HEC-HMS model for Sungai Bekok are shown in Table 1.

**Table 1: Calibration Coefficients of Sungai Bekok catchment**

| Model parameter | Calibrated value |
|---|---|
| Constant Rate (mm/hr) | 3 |
| Imperviousness (%) | 16 |
| Time of Concentration (hr) | 18.25 |
| Storage Coefficient (hr) | 18 |
| Recession Constant | 0.98 |
| Threshold Flow (cumecs) | 1 |

## Model Application

The rainfall-runoff model is required to ascertain the relationship between rainfall and runoff. In actual fact the relationship of rainfall-runoff is known to be highly non-linear and complex. Many of today's software tools are well-designed and well-suited for designing, building, and testing many of these future applications. However, there are also many areas where today's tools are lacking the features and functions needed to build these applications effectively [20]. Various well-known currently available rainfall-runoff models such as HEC-HMS, MIKE-11, SWMM, etc have been successfully applied in many problems and watersheds. However, the existing popular rainfall-runoff models can be detected as not flexible and they require many parameters for calibration. The spatial and temporal rainfall patterns and the variability of watershed characteristics create more complex hydrologic phenomena. The steps involved in the identification of a dynamic model of a system are selection of input-output data suitable for calibration and verification; selection of a model structure and estimation of its parameters; and validation of the identified models [3]. The selection of training data that represents the characteristics of a watershed and meteorological patterns is extremely important in modelling [21].

Input variable is selected to describe the physical phenomena of the rainfall-runoff process, in order to forecast runoff. To accomplish this, the network is trained with a large number of input-output pairs of data. Records of 10 years of hourly rainfall-runoff series of Sungai Bekok catchment (1991-2000) are used. A good quality input-output pairs of data sets was selected to develop and evaluate the neural network models. The neural network was trained under two sets of conditions. In this study, 55 sets of data have been selected from the records. The first 50 sets of data are used for model calibration (training), and the remaining 5 sets of data are used for model verification (testing). It is anticipated that increasing the number of training data in the training phase, with no change in neural network structure, will improve performance on the training and testing phase. Thus, it depends on providing an adequate number and quality of training data.

In this particular study, the structure of ANN model is designed based on a trial and error procedure to find the appropriate number of time-delayed input variables to the model. [3, 5, 7] treat the rainfall as directly related to runoff at the present time $t$ by using the following equation,

$$y(t) = f\{x(t), x(t-1), x(t-2),\ldots x(t-n), y(t-1), y(t-n)\} \quad (15)$$

This model treat the rainfall as directly related to runoff at the present time $t$. The goodness-of-fit statistics are computed for both training and testing for each ANN architecture. At the first step, the rainfall at time $t$ was added to the model. The goodness-of-fit statistics for the present model were computed for training and testing procedures. Then rainfall at time $(t-1)$ was added as an additional input variable to the model, and the goodness-of-fit statistics were computed. This procedure is repeated by adding rainfall at previous time periods as input variable until there is no significant change in model training and testing accuracy. After the first step was completed, another input variable; the runoff at previous time periods, $(t-2)$ is added to the best-fit model obtained from the first step. Then, the goodness-of-fit statistics for the present model were computed for training and testing procedures. This procedure is repeated by adding runoff at previous time periods as input variable until there is no significant change in model training and testing accuracy. The optimal number of input nodes for Sungai Bekok catchment is determined as follow:

$$y(t) = f\{x(t), x(t-1), x(t-2), x(t-3), x(t-4), x(t-5), y(t-1)\} \quad (16)$$

## Model Performance Criteria

The MLP model is designed to simulate the rainfall-runoff processes of watersheds systems. Because there was no definitive test to evaluate the success of each model, a multi-criteria assessment was carried out. Basically, the performance of model will be evaluated based on the comparison between the computed output and actual data. The prediction of each model is evaluated using the correlation of coefficient ($R^2$), root mean square error (RMSE), relative root mean square error (RRMSE), and mean absolute percentage error (MAPE). A RMSE is one

of the most commonly used performance measures in hydrological modeling. The other is to try to fill some of the gaps left by considering only RMSE. Formulas for calculating $R^2$, RMSE, RRMSE, and MAPE are given as follows:

$$R^2 = \frac{\sum_{t=1}^{n}\left[\left(Q_{o(t)} - \overline{Q}_{o(t)}\right)\left(Q_{s(t)} - \overline{Q}_{s(t)}\right)\right]}{\left[\sum_{t=1}^{n}\left(Q_{o(t)} - \overline{Q}_{o(t)}\right)^2 \sum_{t=1}^{n}\left(Q_{s(t)} - \overline{Q}_{s(t)}\right)^2\right]^{\frac{1}{2}}} \tag{17}$$

$$RMSE = \left[\frac{1}{n}\sum_{t=1}^{n}\left(Q_{o(t)} - Q_{s(t)}\right)^2\right]^{\frac{1}{2}} \tag{18}$$

$$RRMSE = \left[\frac{1}{n}\sum_{t=1}^{n}\left[\frac{\left(Q_{o(t)} - Q_{s(t)}\right)}{Q_{o(t)}}\right]^2\right]^{\frac{1}{2}} \tag{19}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Q_{o(t)-Q_{s(t)}}}{Q_{o(t)}}\right| \times 100\% \tag{20}$$

where, $Q_{O(t)}$ and $Q_{S(t)}$ are the observed and simulated values of output; $n$ is the number of observations or time periods over which the errors are simulated. The value of $R^2$ of 90% indicates a very satisfactory model performance while a value in the range 80-90% indicates a fairly good model. Values of $R^2$ in the range 60-80% would indicate unsatisfactory model fit [22]. Generally, RMSE and RRMSE formulas evaluate the models based on a comparison of the estimated errors between the actual observations and the fitted model. A model with the minimum error is considered the best choice. [23] assigned that the MAPE around 30% is considered a reasonable prediction. Further, the analysis will be considered very accurate when the MAPE is in the range of 5% to 10%.

**Results and Discussion**

Results of ANN modelling are shown in Table 2a and 2b. Meanwhile results of HEC-HMS modelling are shown in Table 2c. Figures 3(a)-(f) and figures 4(a)-(f) illustrate the graphical results of 3-layer and 4-layer of MLP model respectively during training and testing. The measures of performance of each model are indicated by correlation of coefficient ($R^2$), root mean square error (RMSE), relative root mean square error (RRMSE), and mean absolute percentage error (MAPE).

The Sungai Bekok is a fully natural catchment. To evaluate the performance of the model, records of 10 years of hourly rainfall-runoff data of Sungai Bekok catchment are used. A large number of training data sets is required to perform successful training. There are, 50 selected data sets used for training task and 5 selected sets of data used as the prediction set. The numbers of input nodes considered for MLP is 7 nodes for Sungai Bekok catchment. For the neural network training process, the best hidden nodes are chosen based on the minimum root mean square error (RMSE) computed for the training data.

Most values of $R^2$ approach 1.0. This outcome indicates that the MLP model consistently show a good performance in rainfall-runoff modeling. [22] reported that when $R^2$ more than 90%, the model is is very satisfactory. It is fairly good with $R^2$ in the range of 80% to 90%. [23] decided on model accuracy based on MAPE value. The prediction model considered reasonable with MAPE below 30% and very accurate with MAPE less than 10%. Results of modelling for Sungai Bekok with MAPE less than 10% can be considered as very accurate. Results of HEC-HMS with MAPE less than 30% are considered as reasonable prediction.

Meanwhile, $R^2$ is between 20% to 60% and this condition shows that the model performance is unsatisfactory model fit.

**Table 2a**: Results of 3 Layer MLP models for Sungai Bekok catchment

| MODEL Data Set | Model Structure | No. of Parameter | Correlation of Coefficient ($R^2$) | RMSE (cumecs) | RRMSE | MAPE (%) |
|---|---|---|---|---|---|---|
| MLP TRAINING | 7-6-1* | 61 | 0.9927 | 0.0477 | 0.0105 | 0.1891 |
| MLP-TEST Set 1 | 7-6-1* | 61 | 0.9976 | 0.0082 | 0.0016 | 0.1077 |
| MLP-TEST Set 2 | 7-6-1* | 61 | 0.9968 | 0.0091 | 0.0018 | 0.1171 |
| MLP-TEST Set 3 | 7-6-1* | 61 | 0.9896 | 0.0066 | 0.0013 | 0.0879 |
| MLP-TEST Set 4 | 7-6-1* | 61 | 0.9875 | 0.0082 | 0.0019 | 0.1422 |
| MLP-TEST Set 5 | 7-6-1* | 61 | 0.9861 | 0.0033 | 0.0008 | 0.0694 |

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

**Table 2b**: Results of 4 Layer MLP models for Sungai Bekok catchment

| MODEL Data Set | Model Structure | No. of Parameter | Correlation of Coefficient ($R^2$) | RMSE (cumecs) | RRMSE | MAPE (%) |
|---|---|---|---|---|---|---|
| MLP TRAINING | 7-6-8-1* | 119 | 0.9927 | 0.0477 | 0.0105 | 0.1901 |
| MLP-TEST Set 1 | 7-6-8-1* | 119 | 0.9927 | 0.0088 | 0.0017 | 0.1125 |
| MLP-TEST Set 2 | 7-6-8-1* | 119 | 0.9976 | 0.0076 | 0.0015 | 0.0972 |
| MLP-TEST Set 3 | 7-6-8-1* | 119 | 0.9923 | 0.0056 | 0.0011 | 0.0750 |
| MLP-TEST Set 4 | 7-6-8-1* | 119 | 0.9908 | 0.0077 | 0.0018 | 0.1346 |
| MLP-TEST Set 5 | 7-6-8-1* | 119 | 0.9745 | 0.0036 | 0.0009 | 0.0785 |

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

**Table 2c** : Results of HEC-HMS model for Sungai Bekok catchment

| MODEL Data Set | No. of Parameter | Correlation of Coefficient ($R^2$) | RMSE (cumecs) | RRMSE | MAPE (%) |
|---|---|---|---|---|---|
| HEC TRAINING | 9 | 0.6097 | 0.5329 | 0.0945 | 7.2194 |
| HEC-TEST Set 1 | 9 | 0.6128 | 0.3508 | 0.0677 | 5.1404 |
| HEC-TEST Set 2 | 9 | 0.373 | 0.8719 | 0.1092 | 14.0590 |
| HEC-TEST Set 3 | 9 | 0.324 | 0.8794 | 0.1759 | 13.7849 |
| HEC-TEST Set 4 | 9 | 0.1518 | 1.1288 | 0.3922 | 16.8040 |
| HEC-TEST Set 5 | 9 | 0.2066 | 0.6248 | 0.0362 | 7.6862 |

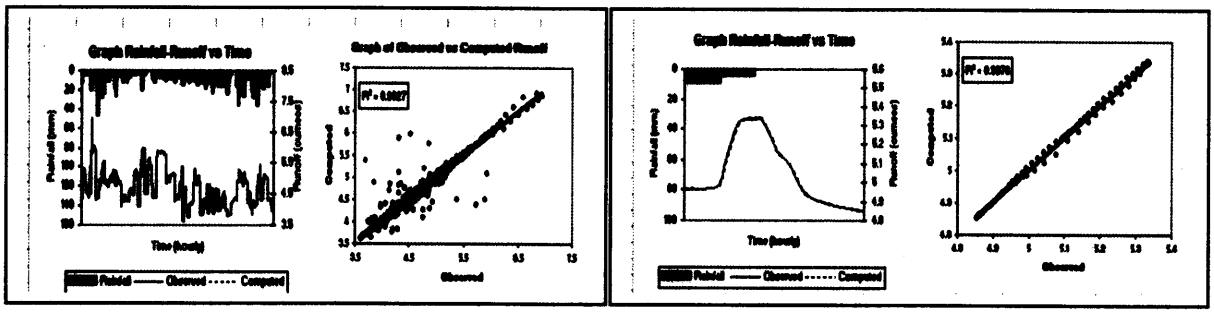
* cumecs-meter cubic second

During the training phase, the RMSE for Sungai Bekok is consistently less than 0.1 cumecs for the 3 layer (7-6-1) and 4 layer (7-6-8-1) model structures. The RRMSE also maintains at 0.0105 for both model structures. During testing, the RMSE (<0.01 cumecs) and RRMSE (<0.002) come close to zero. Obviously, the application of MLP method to model rainfall-runoff relationship of Sungai Bekok is very successful. The Sungai Bekok has an observed flow of between 3.90 cumecs to 5.4 cumecs and a catchment size of 350 $km^2$.
The neural network performances are influenced by the level of nonlinearity and the selection of training data, quality of the data, and the characteristics of the catchments area. Normally, for a large catchment size, the river flow is highly nonlinear and influenced by storage effect. In addition, the effect of spatial rainfall and control structures may contribute to the complexity of the system. Results of rainfall-runoff modeling indicate that application of MLP method is very accurate for Sungai Bekok catchment.

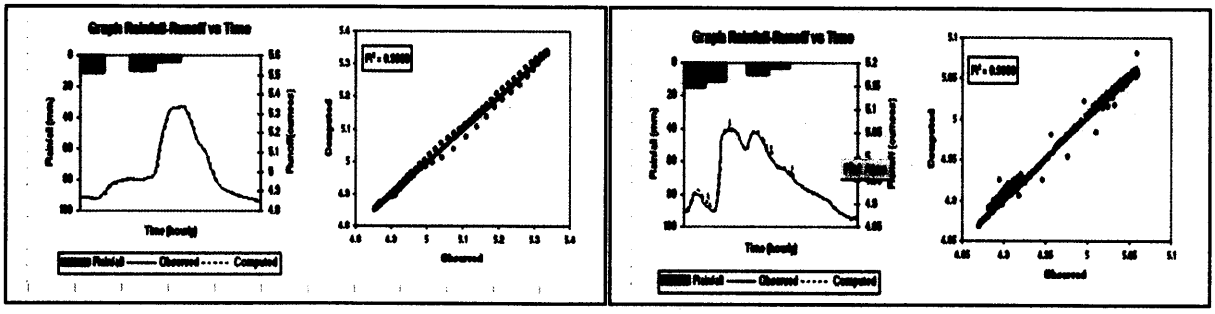
The training process is time consuming. If the architecture of the training algorithms is not suitable, it will affect the accuracy of predictions and a network's learning ability. The number of hidden nodes significantly influences the performance of a network and the time taken to train the model. The number of nodes in the hidden layer can be as small or large as required. It is related to the complexity of the system being modeled and to the resolution of the data fit. The number of nodes in the hidden layer was determined by trial and error for each case. If this number of hidden nodes is small, the network can suffer from under fit of the data and may not achieve the desired level of accuracy, while with too many nodes it will take a long time to be adequately trained and may some times over fit the data. Overall, by increasing the number of hidden layer and number of hidden nodes in the model, it will increase the complexity of the system, and it may slow down the calibration process without substantially improving the efficiency of the network.

## Conclusion

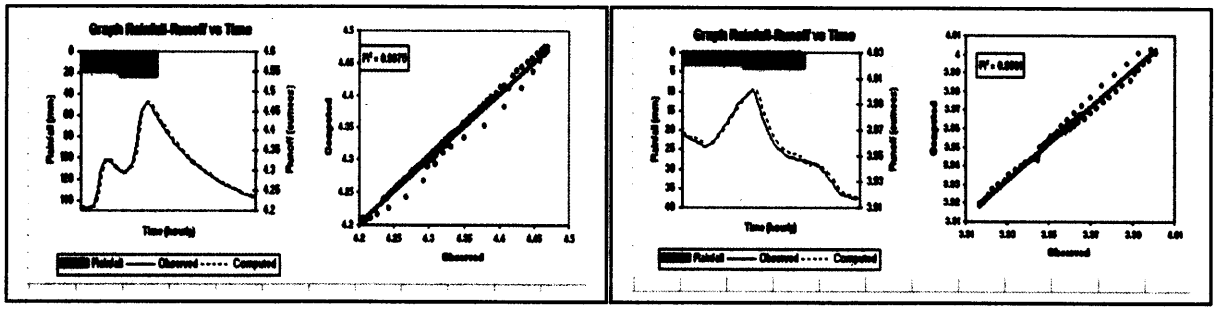
The study demonstrated that the neural network model based on MLP is suitable for modelling the rainfall-runoff relationship. It have the ability to learn spatial rainfall-runoff data from different locations. The MLP has been identified as a robust model in modeling the rainfall-runoff relationship. It can model accurately the storm hydrograph for single-storm and multiple-storm events. The predicted peak discharge and time to peak are in close agreement to the actual values. Obviously, the MLP application to model the hourly streamflow hydrograph was successful.



(a) (b)

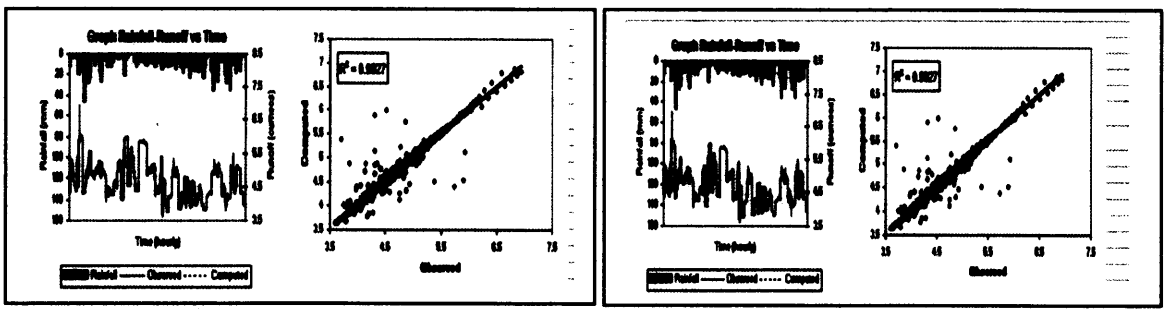


(c) (d)

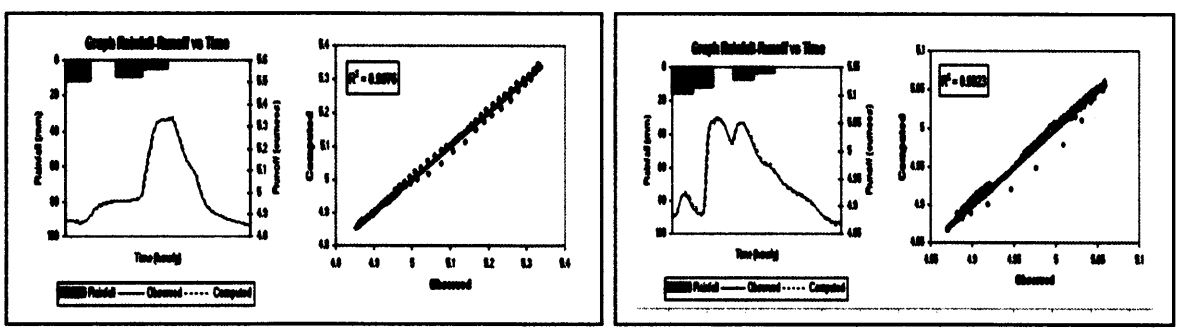


(e) (f)

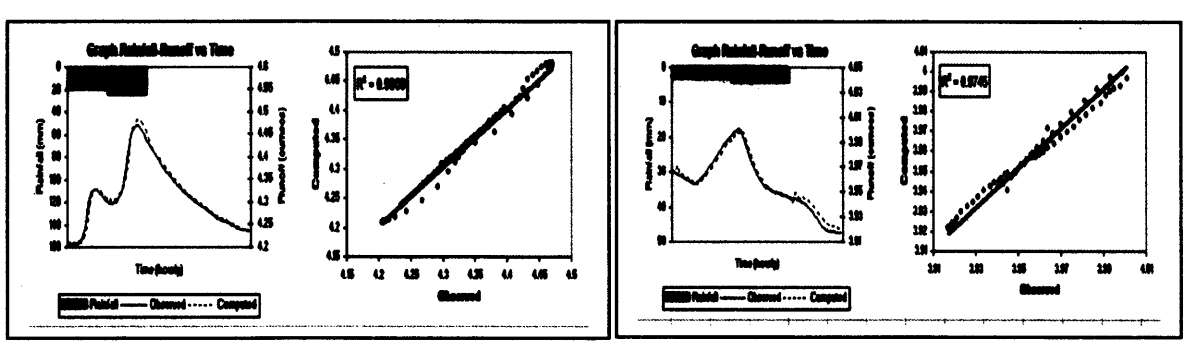
**Figures 3: (a) the graphical results of 3-layer MLP model during training; (b)-(f) the graphical results of 3-layer MLP model during testing**



(a) (b)



(c) (d)



(e) (f)

**Figures 4: (a) the graphical results of 4-layer MLP model during training; (b)-(f) the graphical results of 4-layer MLP model during testing**

## References


[1] L. H. Tsoukalas and R. E. Uhrig (1997), *Fuzzy and Neural Approaches in Engineering*, New York: John Wiley & Sons Inc.

[2] A. M. Buch, H. S. Mazumdar, and P. C. Pandey (1993), "A case study of runoff simulation of a Himalayan glacier basin", *Proc. Intern. Conf. on Neural Networks*, Vol. 1.

[3] K-L Hsu, H. V. Gupta and S. Sorooshian (1993), "Artificial Neural Network Modelling of the Rainfall-Runoff Process", *Water Res. Research.*, Vol. 29, No. 4.

[4] A. Shamseldin (1997), "Application of a Neural Network Technique To Rainfall-Runoff Modelling", *J. of Hydr.*, Vol. 199.

[5] A. S. Tokar and P. A. Johnson (1999), "Rainfall-Runoff Modelling Using Artificial Neural Networks", *J. of Hydr. Eng.*, Vol. 4, No. 3.

[6] N. Karunithi, W. J. Grenney, D. Whitley, and K. Bovee (1994), "Neural Networks for River Flow Prediction", *J. of Comp. in Civil Engrg.*, Vol. 8, No. 2.

[7] Y. B. Dibike and D. P. Solomatine (1999), "River Flow Forecasting Using Artificial Neural Networks", *J. of Physics and Chemistry of the Earth*, EGS1.1-0002.

[8] H. R. Maier and G. C. Dandy (1996), "The Use of Artificial Neural Networks for the Prediction of Water Quality Parameters", *Water Res. Research*, Vol. 32, No. 4.

[9] H. Raman and V. Chandramouli (1996), "Deriving a General Operating Policy for Reservoirs Using Neural Network", *J. of Water Res. Planning and Mngmt.*, Vol.122, No. 5.

[10] S. Harun (1999), *Forecasting and Simulation of Net Inflows for Reservoir Operation and Management*, Universiti Teknologi Malaysia: PhD thesis.

[11] S. Ranjithan and J. W. Eheart (1993), "Neural Network-based Screening for Groundwater Reclamation Under Uncertainty", *Water Res. Research*, Vol. 29, No. 3.

[12] M. Caudill (1987), "Neural Networks Primer, Part I", *AI Expert*, December.

[13] L. Fausett (1994), *Fundamentals of Neural Networks*, New Jersey: Prentice Hall, Englewood Cliffs.

[14] M. T. Hagen and M. B. Menhaj (1994), "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Transactions on Neural Networks*, Vol. 5, No. 6.

[15] H. Demuth and M. Beale (2001), *Neural Network Toolbox: For Use with Matlab*, The MathWorks, Inc. Natick, MA.

[16] T. Masters (1993), *Practical Neural Network Recipes in C++*, San Diego, California: Academic Press Inc.

[17] P. J. Werbos (1974), *Beyond Regression: New tools for prediction and analysis in the behavioural science*, PhD Thesis, Havard University, Cambridge, MA.

[18] N. Karunithi, W. J. Grenney, D. Whitley, and K. Bovee (1994), "Neural Networks for River Flow Prediction", *J. of Comp. in Civil Engrg.*, Vol. 8, No. 2.

[19] M. N. French, W. F. Krajewski and R. R. Cuykendall (1992), "Rainfall Forecasting In Space and Time Using a Neural Network", *J. of Hydr.*, Vol. 137, No. 1.

[20] A. I. Wasserman (2000), "Software Tools: Past, Present, and Future", *IEEE Transactions on Software Engineering*, Vol. 9, No. 3, pp. 3-6.

[21] P. O. Yapo, V. H. Gupta, and S. Sorooshian (1996), "Automatic Calibration of Conceptual Rainfall-Runoff Models: Sensitivity to Calibration Data", *J. of Hydr.*, Vol. 181.

[22] R. K. Kachroo (1986), *HOMS Workshop on River Flow Forecasting, Nanjing, China*, Unpublished Internal Report, Dept. of Engrg. Hydr., University College Galway, Ireland.

[23] D. Johnson and M. King (1988), *Basic Forecasting Techniques*, Butterworth & Co. (Publishers) Ltd., Great Britain.