Hydrol. Earth Syst. Sci. Discuss., 8, 6489–6532, 2011
www.hydrol-earth-syst-sci-discuss.net/8/6489/2011/
doi:10.5194/hessd-8-6489-2011

Hydrology and Earth System Sciences Discussions

Dynamic versus static neural network model for rainfall forecasting at Klang River Basin, Malaysia

A. El-Shafie[1], A. Noureldin[2], M. R. Taha[1], and A. Hussain[3]

[1]Civil and Structural Engineering Department, University Kebangsaan Malaysia, Kebangsaan, Malaysia
[2]Associate Professor, Electrical and Computer Engineering, Royal Military College, Kingston, Canada
[3]Electric, Electronics Systems Engineering Department, University Kebangsaan Malaysia, Kebangsaan, Malaysia

Received: 3 June 2011 – Accepted: 28 June 2011 – Published: 6 July 2011

Correspondence to: A. El-Shafie (elshafie@vlsi.eng.ukm.my)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

Rainfall is considered as one of the major component of the hydrological process, it takes significant part of evaluating drought and flooding events. Therefore, it is important to have accurate model for rainfall forecasting. Recently, several data-driven modeling approaches have been investigated to perform such forecasting task such as Multi-Layer Perceptron Neural Networks (MLP-NN). In fact, the rainfall time series modeling involves an important temporal dimension. On the other hand, the classical MLP-NN is a static and memoryless network architecture that is effective for complex nonlinear static mapping. This research focuses on investigating the potential of introducing a neural network that could address the temporal relationships of the rainfall series.

Two different static neural networks and one dynamic neural network namely; Multi-Layer Peceptron Neural network (MLP-NN), Radial Basis Function Neural Network (RBFNN) and Input Delay Neural Network (IDNN), respectively, have been examined in this study. Those models had been developed for two time horizon in monthly and weekly rainfall basis forecasting at Klang River, Malaysia. Data collected over 12 yr (1997–2008) on weekly basis and 22 yr (1987–2008) for monthly basis were used to develop and examine the performance of the proposed models. Comprehensive comparison analyses were carried out to evaluate the performance of the proposed static and dynamic neural network. Results showed that MLP-NN neural network model able to follow the similar trend of the actual rainfall, yet it still relatively poor. RBFNN model achieved better accuracy over the MLP-NN model. Moreover, the forecasting accuracy of the IDNN model outperformed during training and testing stage which prove a consistent level of accuracy with seen and unseen data. Furthermore, the IDNN significantly enhance the forecasting accuracy if compared with the other static neural network model as they could memorize the sequential or time varying patterns.

HESSD

8, 6489–6532, 2011

1 Introduction

Characteristic and amount of rainfall is not easily known until it occurs. As rainfall plays a crucial role on evaluation and management of drought and flood events, it is very important to be able to forecast rainfall before it occurs. However, in the past most of the methods used in rainfall forecasting are regression or auto-regression linear models which their ability is limited in dealing with natural phenomenon with non-linear trend (de Vos and Rientjes, 2005; Hung et al., 2009; Modarres, 2009). Time variations of rainfall rate have always been forecasted for actual use in advance of the daily activities. It is so important to mention that models for rainfall forecasting are fundamental tools in water resources studies, since they determine and provide the basis in establishing future reservoir water inflows. These predictions are of significance importance in the planning of water resources system, being responsible for the optimization of the system as a whole. This is why rainfall forecasting is a fundamental topic in many engineering applications like constructing dams, analysis and forecasting, planning and designing of reservoirs, hydro-power generation, irrigation, water management, controlling floods and others.

The rainfall forecasting problem has been traditionally tackled using linear techniques, such as AR, ARMAX, and Kalman filter, and also using nonlinear regression (see Cheng, 1994; Alvisi, et al., 2006; Abrahart and See, 2007; Bras and Rodriguez-Iturbe, 1985; Chiu, 1978; Box and Jenkins, 1970). Most of the forecasting methods consider one-day ahead forecast. For the rainfall a longer term forecast such as ten-day ahead or a month ahead is more of interest, though it is more difficult than the one-day ahead problem. In fact, there are several considerable drawbacks to the use of KF in rainfall forecasting application. These include: (1) the necessity of accurate stochastic modeling, which may not be possible in the case for rainfall; (2) the requirement for apriori information of the system measurement and develop covariance matrices for each new pattern, which could be challenging to accurately determine and

(3) the weak observability of some of temporal pattern states that may lead to un-stable estimates for the forecasted value (see Noureldin et al., 2007, 2011).

In this context, motivation for utilizing non-linear modeling approach based Artificial Intelligence (AI) modeling have received considerable attention from the hydrologists in the last two decades (Boucher et al., 2010; Vos and Rientjes, 2005). Lapedes and Farber (1987) conduct a simulated study and conclude that ANN can be used for modeling and forecasting nonlinear time-series. Recently, numerous ANN-based rainfall-runoff models have been proposed to forecast streamflow (Hsu et al., 1995; Thirumalaiah and Deo, 1998, 2000; Campolo et al., 1999; Sajikumar and Thandaveswara, 1999; Tokar and Johnson, 1999; Zealand et al., 1999) and reservoir inflow (Saad et al., 1996; Jain et al., 1999; Coulibaly et al., 2000a,b). In addition, neural networks and fuzzy logic have been used as effective modeling tools in different environmental processes such as wastewater treatment, water treatment and air pollution. Cinar et al. (2006) used an artificial neural network to predict the performance of a membrane bioreactor. They were able to estimate concentrations of chemical oxygen demand, phosphate, ammonia and nitrate. Altunkaynak et al. (2005a) used fuzzy logic modeling to forecast dissolved oxygen concentration. Altunkaynak et al. (2005b) compared the accuracy of fuzzy logic modeling and autoregressive integrated moving average (ARIMA) models in predicting water consumption in a city. They found that relative error rates for fuzzy logic and ARIMA were 2.5 and 2.8, respectively.

Recently, the authors developed several AI-based inflow forecasting architectures using Multi-Layer Perceptron Neural Networks (MLPNN), Radial Basis Function Neural Networks (RBFNN) and Adaptive Neuron-Fuzzy Inference Systems (ANFIS) at Aswan High Dam, Nile River, Egypt (Elshafie and Noureldin, 2011). The main idea behind all of these methods is to mimic the latest inflow pattern to forecast the inflow for 3 months ahead. The major drawback of such models is their inability to mimic the temporal inflow pattern trend during the model training stage procedure. Therefore, any of the existing AI-based models may not be capable of providing a reliable and accurate forecasting solution.

HESSD
8, 6489–6532, 2011

In this research, we aim at developing an AI-based rainfall forecasting model taking into consideration the temporal pattern trend and thus providing a better forecasting accuracy. Such technique combines the advantages of some of the existing models with the advantages of dynamic neural networks in representing the sequential process in the input data (the latest rainfall pattern). In this way, it should be possible for the proposed model to mimic the temporal pattern of the rainfall based on the current and some past. The proposed model will be carried out utilizing real rainfall records at Klang River, Malaysia. Finally, comprehensive comparative analyses are performed in order to examine the significance of utilizing the dynamic neural network over the classical static neural network methods.

2 Artificial neural network

Artificial Neural Network (ANN) is a parallel computing mathematical model for solving nonlinear time series problems. ANN can solve non-linear problems based on network architecture and activation transfer function that is employed. In this study, we evaluate two different static neural network methods, MLP-NN and RBFNN and one dynamic neural Network (IDNN). Hereafter, a brief explanation of all those neural network methods will be introduced.

2.1 Static neural network

2.1.1 Multi-layer perceptron

The network architecture of the MLP-NN is shown in Fig. 1. ANN architecture contains three types of layers. The layers are input layer, hidden layer and output layer. Each layer consists of one or more neurons. There are two types of neuron. First are passive neurons that relay data input as data output. Another is active neuron that computes data input using Activation Transfer Function (ATF) and produces an output. The most

Discussion Paper | Discussion Paper | Discussion Paper | Discussion Paper |

HESSD

8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

commonly use of ATF in the hidden and output neuron is sigmoid function (Zhang et al., 1998; Fernando et al., 2000). The input into an active neuron is a summation of previous neuron's output and its weight and the output is a computation of sigmoid function on the input. The process is shown in Fig. 1 and the equations for the input and output are:

$$\text{input} = \int_{1=0}^{n} (w_{ij}, x_{ij}) \text{ where } x_0 = 1\,, \tag{1}$$

$$\text{output} = \frac{1}{1 + e^{-k\text{input}}}\,, \tag{2}$$

where x is the output from previous neuron, w is the weight of the output and k is the slope steepness of the sigmoid function. Extra neuron x_0 is added in input layer and hidden layer with output value of 1. This is called bias and its function is to stabilize computed output between 0 and 1. It does not have link from previous neuron.

2.1.2 Radial basis function

The structure of a RBFNN consists of an input layer, one hidden layer and an output layer, see Fig. 2. The input layer connects the inputs to the network. The hidden layer applies a non-linear transformation from the input space to the hidden space. The output layer applies a linear transformation from the hidden space to the output space, see Mark (1996).

The radial basis functions $\phi_1, \phi_2, \ldots, \phi_N$ are known as hidden functions while $\{\phi_i(x)\}_{i=1}^{N}$ is called the hidden space. The number of basis functions (N) is typically less than the number of data points available for the input data set. Among several radial basis functions, the most commonly used is the Gaussian, which in its one-dimensional representation takes the following form:

$$\phi(x, \mu) = e^{-\frac{\|x-\mu\|^2}{2d^2}}\,, \tag{3}$$

HESSD
8, 6489–6532, 2011

where μ is the center of the Gaussian function (mean value of x) and d is the distance (radius) from the center of $\phi(x,\mu)$, which gives a measure of the spread of the Gaussian curve.

The hidden units use the radial basis function. If a Gaussian function is used, the output of each hidden unit depends on the distance of the input x from the center μ. During the training procedure, the center μ and the spread d are the parameters to be determined. It can be deduced from the Gaussian radial function that a hidden unit is more sensitive to data points near the centre. This sensitivity can be tuned (adjusted) by controlling the spread d. Figure 3 shows an example of a Gaussian radial function. It can be observed that the larger the spread, the less the sensitivity of the radial basis function to the input data. The number of radial basis functions inside the hidden layer depends on the complexity of the mapping to be modeled and not on the size of the data set, which is the case when utilizing multi-layer perceptron ANN (see Ripley, 1996; Bishop, 1996; Haykin, 1994).

2.2 Dynamic neural network

2.2.1 Motivation

The rainfall forecasting model used by KF and or ARMA is a linearized one. In addition the extension of those methods to include stochastic pattern of the rainfall is also linearized in the form of 1st order difference equations. The non-linear and the non stationary parts of the rainfall pattern are not modeled for KF or ARMA, thus deteriorating the forecasting accuracy. This leads to relatively poor forecasting for the rainfall. The non-linear complex AI – based modeling capabilities is therefore suggested in this study. The benefit of utilizing AI-methods over the conventional method (KF and ARMA) that none of the above drawbacks could be found while utilizing the AI-methods. Furthermore, the advantage of utilizing the proposed IDNN over the other AI-methods that IDNN method is performing a temporal processing that gives the model complete

HESSD
8, 6489–6532, 2011

information about the temporal relationship of the input pattern, which is the main challenge in studying the rainfall pattern that incorporate major temporal dimension.

2.2.2 Input delay neural network

Dynamic networks are generally more powerful than static networks (although they may be somewhat more difficult to train) (Ripley, 1996; Bishop, 1996). Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. In fact, in order to predict temporal patterns, an ANN requires two distinct components: a memory and an associator. The memory holds the relevant past information, and the associator uses the memory to predict future events. In this case the associator is simply a static MLPNN network, and the memory is generated by a time delay unit (or shift register) that constitutes the tapped delay line (Ripley, 1996; Bishop, 1996). In fact, the MLPNN or other static neural network type model do not perform temporal processing since the vector space input encoding gives the model no information about the temporal relationship of the inputs. Traditional MLPNN is a static and memoryless network that is effective for complex non-linear static mapping. In fact, rainfall forecasting is a procedure where previous states of rainfall values have to be seriously considered. Apparently, rainfall process modeling involves a major temporal dimension and in the ANNs context there are efficient methods to represent and process such models (Haykin, 1994).

Figure 4 shows the general architecture of an Input Delay Neural Network (IDNN) in addition to zooming on the internal structure of a single neuron. The case shown in Fig. 4 considers a tapped delay line that involves the p most recent inputs. In this example, we show three delay elements represented by the operator d. For a case of p delay elements and an input variable $x(t)$, the network processes $x(t)$, $x(t-1)$, $x(t-2)$, ..., and $x(t-p)$, where p is known as the tapped delay line memory length (Haykin, 1994). Therefore, the input signal $S_i(t)$ to the neuron i (Fig. 1) is given as:

HESSD
8, 6489–6532, 2011

$$S_{\mathrm{i}}(t) = \sum_{k=0}^{p} w_i(k)x(t-k) + b_i \tag{4}$$

where $w_i(k)$ is the synaptic weight for neuron i, and b_i is its bias. Then the output of this neuron (U_i) is obtained by processing $S_{\mathrm{i}}(t)$ by the non-linear activation function $G(.)$, chosen as a sigmoid activation function of neuron i.

$$U_i = G\left(\sum_{k=0}^{p} w_i(k)x(t-k) + b_i\right) \tag{5}$$

$$G(S_{\mathrm{i}}(t)) = \frac{1}{1+e^{-S_{\mathrm{i}}(t)}} \tag{6}$$

The output of the IDNN, assuming that it has one output neuron j, a single hidden layer with m hidden neurons, and one input variable as shown in Fig. 4, is given by

$$y_j(t) = F\left(\sum_{i=1}^{m} w_{ji}U_i + \alpha_j\right) \tag{7}$$

where $F(.)$ is the transfer activation function of the output neuron j (which can be chosen to be a sigmoid or a linear function), α_j is its bias and w_{ji} is the weight between the neurons of the hidden layer and the neuron of the output layer.

During the update procedure, we use a second-order back-propagation variation; namely the Levenberg-Marquardt back-propagation (LMBP). The network training process is performed by providing input-output data to the network, which targets minimizing the error function by optimizing the network weights. LMBP uses the second derivative of the error matrix (**E**) to update the weights of the network in a recursive fashion (Haykin, 1994).

HESSD
8, 6489–6532, 2011

3 Study area and data collection

The Klang River Basin is located on the west coast of Peninsular Malaysia and encompasses the Federal Territory of Kuala Lumpur, parts of Gombak, Hulu Langat, Klang, and Petaling districts in Selangore Stats, and the municipal areas of Ampang Jaya, Petaling Jaya, and Shah Alam. Klang is geographically located at latitude (3.233°) 3° 13′ 58″ N of the Equator and longitude (101.75°) 101° 45′ 0″ E of the Prime Meridian on the Map of Kuala Lumpur. The study area location map has been shown in Fig. 5.

The Klang River originates in the mountainous area about 25 km northeast of Kuala Lumpur. It is joined by 11 major tributaries while passing through the Federal Territory and the area downstream of Kuala Lumpur, before joining the Straits of Malacca at port Klang. The Klang River has a total lengh of about 120 km. The basin is 1290 km^2, about 35 % of which has been developed for residental, commercial, industrial, and instituionsl use. The upper catchments of the Klang River and its tributaries – the Gombak and Batu Rivers – are covered with well maintained forests. However, the lower reaches of the basin, with extensive urban land development activities, are major contributors of sediment load and flood peaks (Tan, 2009; Hiew, 1996; Gibson and Dodge, 1983).

It is also characterized by uniform high temperature, high relative humidity, heavy rainfall and little wind. All information and data that are available about Klang River were based on Klang gates dam data. For this study, the data used were from year 1986 to 2008 (monthly basis) and between 1997 and 2008 (weekly basis). The available data for catchment is divided into two groups: training set (calibration) and a testing set (validation). The rainfall data statistics have been investigated, including maximum, minimum and mean averages. The average annual rainfall depth in the study area is about 2400 mm. The highest rainfall occurs in the month of April and November with a mean of 280 mm. The lowest rainfall occurs in the month of June with a mean of 115 mm. The rainfall data on monthly and weekly basis is shown in Figs. 6 and 7, respectively (Tan, 2009; Hiew, 1996; Gibson and Dodge, 1983). For monthly rainfall

HESSD
8, 6489–6532, 2011

forecasting, total monthly rainfall data is 261 which containing 237 samples was used for training and another 24 samples was used to test the generalization ability of the networks. Whereas for weekly forecasting, total weekly rainfall data is 621 which containing 571 samples was used for training and the rest containing 50 samples was used to test the generalization ability of the networks.

One of the steps of data pre-processing is data normalization. The need to make harmony and balance between network data range and activation function used causes the data to be normal in activation function range. Sigmoid logarithm function is used for all layers. By considering Sigmoid, it can be seen that the range is between 0 and 1, so data must be normalized between 0, 1 (Eq. 8). The following formula was used:

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{8}$$

where x is actual data and $x_{\min}$ is minimum value of original series and $x_{\max}$ is maximum value of original series.

4 Methodology

Most neural network approaches to the problem of forecasting use a multilayer network trained using the back-propagation algorithm. Consider a time series $x(1),...,x(t)$, where it is required to forecast the value of $x(t+1)$. The inputs to the multilayer network are typically chosen as the previous k values $x(t-k+1),...,x(t)$ and the output will be the forecast. The network is trained and tested on sufficiently training and testing sets that are extracted from the historical time series. In addition to previous time series values, one can utilize as inputs the values or forecasts of other time series (or external variables) that have a correlated or causal relationship with the series to be forecasted. For our rainfall forecasting problem such time series could be the temperature and relative humidity at the river basin. For the majority of forecasting problems such external inputs are not available or are difficult to obtain. As is the

HESSD
8, 6489–6532, 2011

case with many neural-network applications, preprocessing the inputs and the outputs can improve the results significantly. Input and output preprocessing means extracting features from the inputs and transforming the target outputs in a way that makes it easier for the network to extract useful information from the inputs and associate it with the required outputs. Preprocessing is considered an "art", and there are no set rules to choose it. Even some very intuitively appropriate transformations may turn out of no value when checking the actual results. For our case the main inputs are the previous time series values. We have used normalization as a preprocessing of the inputs (as described Sect. 3).

4.1 Model structure

Generally, formation of an appropriate architecture of a neural network for a particular application is an essential step and issue since the network topology directly affects not only its computational complexity and its generalization capability but also the accuracy level. Different theoretical and experimental studies have shown that larger-than-necessary networks tend to over-fit the training samples and thus have poor generalization performance and low accuracy level for the unseen data, while too-small networks (that is, with very few hidden neurons) will have difficulty learning the training data. Currently there is no established methodology for selecting the appropriate network architecture prior to training. Therefore, we resort to the trial-and-error method commonly used for network design. In addition, the performance goal (mean square error MSE) for the model during the training stage was forced to be 10^{-4}, thus the neural network is guaranteed to hedge over-fitting the training data.

One more important step in the model implementation, especially in the multivariate ANN forecasting context is the selection of appropriate input variables, since it provides the basic information about the system considered. In current study, the available data is the historical rainfall records, thus, different input pattern in terms of the length of the previous rainfall records (window size = w) have been examined. Five window sizes (w = 1, 2, 3, 4 and 5) were considered in this study. Searching for the best window

HESSD
8, 6489–6532, 2011

size is evaluated via two statistical indexes during training to determine the relative importance of each window size on the model accuracy level and generalization. Details about the model performance evaluation will be described in the following section.

A common method is to consider a sliding (or moving) window of input sequences. This approach has been widely used with the standard MLP-NN. In this case, a fixed number of past items of information are selected and introduced to the input layer of the network. For instance, if it is required to model the rainfall pattern based on the input at the present time instant and the past two samples (window size = 3), the ML-PNN input layer should have three input neurons see Fig. 8. Therefore, the network is provided with a static memory that is specifically expert knowledge-dependent, which is consider as major limitation of the MLP-NN, particularly and the implementation does not have a temporal dimension. The time line index for the proposed model process is presented in Fig. 9. The upper part of the Fig. 9 shows the process while utilizing the static neural network (MLP-NN and RBFNN) model.

On the other hand, the IDNN model with a sliding window input sequence is shown in lower part of Fig. 9. In this study, one and two time-step input delay sequences will be considered. The second-order delay effect will be considered by training the IDNN model to experience, in the input layer, the previous one time-step sample in addition to the present rainfall record. Moreover, the higher-order error can be considered by having two and three time-step delay inputs. In Sect. 5.1, the impact of using one and two input delay elements will be demonstrated and discussed.

It should be noted here that a static neural network with four input pattern (window size = 4) is not similar to a dynamic neural network with three input pattern (window size = 3) with one-time-step input delay. This is due to that the dynamic neural network incorporates the associator (network weights and bias) procedures at the current time step and memorizes the past information (network weights and bias) from the previous time step, while the static neural network only handle certain time step with longer input pattern.

HESSD
8, 6489–6532, 2011

Recall that our collected rainfall data spanned the period from 1986 to 2008 on monthly basis and from 1997 to 2008 on weekly basis, then, our forecasting model algorithm established for these two time series horizon.

4.2 Training algorithm

The neurons in the network architecture are interconnected between the layers. These interconnections represent flow of computation in the ANN. The computation starts from input neurons where data input are received and then propagates to hidden neurons and further to output neuron. Neuron in the output layer produces model output. The computational process that is described above is called feed forward computation. If number of neurons and layers are established, the only unknown parameter in the computation is the weights. The process of data training determines the weights. Data training is a process of feeding sample historical data to the input and output of the network model so that the network model can simulate the sample data. The data training process involved feed-forward and back-propagation computation cycles. The back-propagation computation is an adjustment of output and hidden neuron's weights based on gradient descent method. These weights are normally initialized with random values to speed up the data training process to solution.

For optimization purposes, we use a second-order back-propagation variation – namely the Levenberg-Marquardt back-propagation (LMBP) – for the IDNN training. This method uses the approximate Hessian matrix in the weight update procedure as follows:

$$\Delta W_{\mu} = -[\mathbf{H} + \mu I]^{-1} \mathbf{J}^{t} r \tag{9}$$

where r = residual error, μ = variable small scalar that controls the learning process, $\mathbf{J} = \Delta E$ = Jacobian matrix, E = cost function, and $\mathbf{H} = \mathbf{J}^{t}\mathbf{J}$ denotes the approximate Hessian matrix. In practice, this method has been found effective in finding better optima than standard back-propagation and the conjugate gradient descent method (Hagan and

HESSD
8, 6489–6532, 2011

Menhaj, 1994). A detailed description of this algorithm is proposed by Masters (1995). Here the LMBP algorithm is also used to train the all proposed network models.

Upon successful of data training, data forecasting can be made to new data input. To evaluate forecasting performance, validation data are feed only to the input of network model where single feed-forward computation computes the data. The output of the computation is the model output. Several performance measures are applied to model outputs and observed outputs from validation dataset to determine the accuracy and reliability of the network model developed.

4.3 Model performance criteria

To compare and evaluate the effectiveness of rainfall forecasting model applied at Klang Gates Dam, models are assessing on the basis of important performance measures. Although, all models achieved a MSE of less than 10^{-4} during the training process, it is important to examine the model performance utilizing different input sequences and pattern. Consequently, statistical analysis for the model output in the testing session utilizing the inflow data for the period between 1998 and 2003 was carried out in order to evaluate the model performances. To analyze the fittingness of forecasted inflow with the natural inflow during the testing period, two statistical measures were used to examine the goodness to fit of the proposed models methods to the testing data. These measures include the RMSE (Root Mean Square Error) and the maximum relative error (RE) to examine the relative accuracy of both models for each inflow event as represented by Eqs. (7) and (8).

$$\mathrm{RMSE} = \left(\frac{1}{N} \sum_{1}^{N} (R_f - R_m)^2 \right)^{0.5} \tag{10}$$

$$\mathrm{MaxRE} = \max \sum_{n=1}^{n} \left(\frac{R_m - R_f}{R_m} \right) \cdot 100 \tag{11}$$

HESSD
8, 6489–6532, 2011

where, R_f is the Rainfall at Klang River, R_m is the actual rainfall and N is the number of the months/weeks.

In addition, as the forecasting accuracy of the peak and low inflow events is of particular interest of the reservoir operation, it is important to evaluate the model performance considering these inflow events. In order to assess the model performance during these events, another two error criteria are also utilized Peak Flow Criteria (PFC) and Low Flow Criteria (LFC), which can be computed by Eqs. (9) and (10)

$$\mathrm{PFC} = \frac{\left(\sum_{i=1}^{T_\mathrm{p}} (R_m - R_f)^2 \cdot (R_m)^2\right)^{0.25}}{\left(\sum_{i=1}^{T_\mathrm{p}} (R_m)^2\right)^{0.5}}, \tag{12}$$

$$\mathrm{LFC} = \frac{\left(\sum_{i=1}^{T_i} (R_m - R_f)^2 \cdot (R_m)^2\right)^{0.25}}{\left(\sum_{i=1}^{T_i} (R_m)^2\right)^{0.5}}, \tag{13}$$

where T_p = number of peak rainfall greater than one-third of the mean peak rainfall observed; T_i = number of low rainfall lower than one-third of the mean low rainfall observed. Coulibaly (2001) reported that both PFC and LFC provide better performance indicators for assessment of the forecasting model performance for the extreme rainfall events. As the model can provide low PFC or LFC as the model represents better fit.

One more index is examined for the proposed model which is evaluating the consequences of the rainfall. In fact, the model could provide relatively good fit in terms of the error values; however, the forecasting values might not follow the consequences changes of the rainfall pattern values. For example, in case the difference between

HESSD
8, 6489–6532, 2011

two consequence actual values of rainfall is positive $[(R_m(t+1) - R_m(t) > 0]$, while if examined using the forecasted value is negative $[R_f(t+1) - R_m(t) < 0]$, even the model might provide forecasted value with relatively small error value, it is considerable drawback of the model performance and shows mismatching with the real rainfall pattern consequences.

All the development made on this study was implemented using MATLAB computer-aided design software (Mathworks, Natick, MA). The Neural Network toolbox of MATLAB was utilized and the code was set up to include all the above procedures.

5 Results and discussions

The forecasting model architecture described in Sect. 4 is applied on monthly and weekly rainfall data at Klang River, Malaysia. In fact, the procedure of the study began with utilizing the MLP-NN method searching for best model configuration in term of input pattern (window size), then use this window size in the other methods. This way makes sure that the comparative analysis between all proposed methods is adequately performed.

All networks successfully achieved the target MSE of 10^{-4}. For example, the training curve utilizing the MLP-NN method for the weekly data is demonstrated in Fig. 10 showing convergence to the target MSE after 563 iterations. This section is organized to present the results for each method individually followed by evaluation of optimal model based on the model criterion indexes presented in Sect. 4.3.

5.1 Forecasting utilizing MLP-NN

Several trails and error in order to search for the optimal MLP-NN architecture have been carried out. One and two hidden layers and number of neurons ranging between one to ten, different transfer function (tan sigmoid, log sigmoid, linear) and finally different window sizes ($w = 1$ to $w = 5$) have been examined in order to attain the optimal model configuration. In fact, the procedure is performed by implementing all trails for number of hidden layer, number of neuron in each layer and the type of transfer function, while keep the window size unchangeable. Then, repeat all the trials again at the other window size. Such procedure was applied for both weekly and monthly time horizon as the objective for this study.

In this context, the model configuration that provides best performance in terms of lower maximum relative error and RMSE while training procedure is selected. For the weekly basis horizon, the optimal model configuration is achieved when the window size = 4 (number of neurons in the input layer), two hidden layers with number of neuron equal to 8 and 5, respectively, and log sigmoid transfer function between input layer to hidden layer #1 and from hidden layer #1 and #2 and linear transfer function between hidden layer #2 and the output layer. On the other hand, the optimal architecture for the monthly basis horizon is attained when window size = 3 and one hidden layer with 7 neurons. The transfer functions are tan sigmoid and linear between input layer and hidden layer #1 and from hidden layer #1 to output layer, respectively.

Figure 11 shows the performance of monthly and weekly rainfall forecasting using MLP-NN model. Figure 11a shows the RE for the monthly basis forecasting data used for training, it could be depicted that the maximum RE is 25 % while the RMSE is 55.6 mm. Whereas, the performance for the unseen data during the testing stage is about 65 % as the maximum RE and RMSE equal to 79.89 mm, see Fig. 11b. It should be noticed here, that RE during the testing is almost 3 times that one experienced during the training stage.

On the other hand, Fig. 11c shows the MLP-NN model while examining the data on weekly basis during the training. It could be observed that, although the model provides maximum RE at 50 % during training which is relatively high if compared with the case for monthly basis, the performance of the model during the testing stage as shown in Fig. 11d is also within the same range (except one odd case at week #17, RE equal to 80 %), which is not the case for the model on monthly basis. Such observation shows that the model for weekly basis provides higher consistent level over the monthly basis,

HESSD
8, 6489–6532, 2011

it might be due the fact that the model for weekly basis incorporates large data records for training that allow the model to mimic several pattern and able to provide same level of accuracy during testing. Such observation could be confirmed when examine the RMSE during the training and testing stages which are 37.2 mm and 43.5 mm, respectively.

5.2 Forecasting utilizing RBFNN model

Keeping in mind that the optimal window size achieved based on MLP-NN model, 3 and 4 for monthly and weekly, respectively. It should be noticed here that the architecture of RBFNN network is quite simple if compared with MLP-NN. Once the window size (input pattern) was resolved, adjustment of spread of the RBFNN model configuration is (as described in Sect. 2.1.2) the only parameter to be obtained. The spread (step size) is achieved by trial and error as well. The optimal values of the spread were found to be equal 0.07 for monthly and 0.03 for weekly model.

Figure 12 illustrates the accomplished results for the monthly and weekly rainfall forecasting using the RBFNN model. For monthly basis, as demonstrated in Fig. 12a, the RE during training is slightly increased if compared with MLP-NN, however, the RE level is improved for the testing data Fig. 12b. It could be observed that the maximum RE is within ± 40 %, which means considerable improvement over the MLP-NN model. In addition, the RMSE is slightly improved to be 68.7 mm which is almost 90 % of similar value when using MLP-NN.

For weekly basis horizon model, if Fig. 12c shows the performance of the RBFNN during training. If the Fig. 12c is carefully examined, it could be observed that the pattern of RE is similar to RE pattern using MLP-NN model, but the RE value is relatively improved. Consequently, the performance for the RBFNN model in terms of RE is also enhanced when examined the testing data if compared with MLP-NN model, see Fig. 12d.

HESSD
8, 6489–6532, 2011

5.3 Performances of IDNN model

Similar procedures were applied for the rainfall forecasting utilizing the IDNN model. To investigate the effect of temporal dimension value of the rainfall at $(t+1)$ (the output of the IDNN module) on the present and past rainfall pattern inputs (the input to the IDNN module), we examined the performance of the IDNN model for using one time input delay element to the case of two input delay elements. In case of one time input delay, Fig. 13 shows the performance for monthly and weekly basis horizon. As can be depicted from Fig. 13a, that significant enhancements are taken place in the forecasting accuracy in terms of the relative error. The maximum RE does not go behind 12 % which is almost one-third of maximum RE experienced using MLP-NN and RBFNN. In addition, it is noticeable that a considerable improvement in the maximum RE (± 20 %) for testing data is achieved, as shown in Fig. 13b. Similar enhancement while applying the IDNN for the weekly basis data could be observed as shown in Fig. 13c,d.

On the other hand, in case of using two-time input delay, Table 1 shows the maximum RE and RMSE for both training and testing stages for monthly and weekly basis utilizing one and two input IDNN architecture. The results clearly show that utilizing two-time input delay elements has insignificant improvements to the model performance if compared to the one-time input delay IDNN architecture even worst, especially for weekly basis model. While the proposed IDNN-based model showed slight accuracy improvement when using two-time input delay elements instead of one for monthly basis model, the additional delay element significantly complicated the training procedure.

In the light of the results presented above, apparently, the highest RMSE exists at the MLP-NN model in monthly rainfall forecasting and it is equal to 79.89 mm whereas the smallest value of RMSE exists at the IDNN model in weekly rainfall forecasting model with only 7.3 mm. It could be remarked that, generally, the performance for the weekly rainfall forecasting is better than monthly rainfall forecasting. This is due to the inadequate of historical data records on monthly basis which is 261 records, while for weekly, historical 621 rainfall records, thus, the model could capture most of

HESSD
8, 6489–6532, 2011

temporal dimension of rainfall pattern and able to provide lower forecasting error. In addition, it is obvious that the optimal results received when using the IDNN method. Furthermore, with one-time step input delay the IDNN is sufficient to achieve significant level of accuracy. With respect to this observation, in Klang River Basin rainfall pattern, it might be the temporal dimension feature of rainfall is in second order level. However, it could be inadequate for other river basin that might require introducing higher order level.

For further assessment, the IDNN model with one-time step input delay is examined for the peak and low rainfall events, so that, the comparisons between the forecasted and actual rainfall values are visually presented using the 45° line and two deviation lines with ± 15 % deviation from the 45° line and demonstrate the low, average and peak rainfall ranges for both monthly and weekly basis as shown in Figs. 14 and 15, respectively. The scatter plot of forecasted rainfall based monthly as depicted in Fig. 14 is a little distant from the ideal line in case of the high rainfall while it is closer to the ideal line in the low inflow range. The same rainfall forecasting features for weekly basis could be observed from Fig. 15. Apparently, the model provides better accuracy for the low rainfall for either low or high rainfall seasons. This is due to the fact that the peak rainfall events for both low and high rainfall seasons have not experienced adequately during training period. In order to validate the previous analysis of the model performance in providing an accurate inflow forecasting for the peak and low inflow events, the PFC and LFC statistics as discussed above in Sect. 4.3 are presented in Table 2. As presented in Table 2, it can be observed that the developed model can perform the function of providing an accurate rainfall forecasting at Klang River for even for the extreme rainfall events with error does not go above 9.2 % of the actual rainfall.

Finally, the IDNN model with one-time step in the input delay was evaluated for its ability to model the rainfall consequences see Sect. 4.3. Table 3 shows the results for this evaluation index. The negative values means that the model failed in matching the rainfall consequences visa versa is the positive value. It could be observed that the IDNN model for weekly basis forecasting is outperformed the other methods.

HESSD
8, 6489–6532, 2011

6 Conclusions

This study is focusing on modeling the temporal dimension of the rainfall pattern in order to achieve better rainfall forecasting results. In this context, this study investigates three different AI-based static and dynamic methods. The proposed models implemented for offering rainfall forecasting model on Klang River Basin for monthly and weekly time horizon. The results reveal that the dynamic neural network namely IDNN could be suitable for modeling the temporal dimension of the rainfall pattern, thus, provides better forecasting accuracy. Based on this study, IDNN model with one-time step input delay for weekly basis rainfall forecasting achieved the optimal accuracy level. These results could also be applicable to other studies in other river basin with different time step input delay according to how far is the temporal dimension of rainfall pattern at this river basin. Therefore, it is suggested that additional studies employing be used to evaluate the dynamic neural network forecasting performance, especially, in the applications involve temporal dimension. The results of the present study also show that the proposed IDNN provides better accuracy for the extreme rainfall pattern events. In addition, the model was evaluated for matching the consequences of the rainfall, apparently; the model performance shall follow the rainfall consequences with significant level of accuracy due to the model could mimic the temporal rainfall pattern. For future research in applying AI-based model, it is highly recommended to find better and more reliable pre-processing method rather than using trial and error method that can figure out the best input window size. In addition, it is recommended also to study the temporal dimension order before establishing the IDNN model configuration, in order to find the optimal input-delay length in model architecture.

HESSD
8, 6489–6532, 2011

Acknowledgements. This research is supported by: (1) a research grants to the first author by Smart Engineering System, University Kebangsaan Malaysia; and eScience Fund project 01-01-02-SF0581, ministry of Science, Technology and Innovation, (MOSTI) (2) a research grant to the second author from the Natural Science and Engineering Research Council (NSERC) of Canada. The authors appreciate the support from Klang Gate Dam and Department of Drainage and Irrigation (DID), Malaysia for providing all the required data used for developing this research.

References

Abrahart, R. J. and See, L. M.: Neural network modelling of non-linear hydrological relationships, Hydrol. Earth Syst. Sci., 11, 1563–1579, doi:10.5194/hess-11-1563-2007, 2007.

Akhtar, M. K., Corzo, G. A., van Andel, S. J., and Jonoski, A.: River flow forecasting with artificial neural networks using satellite observed precipitation pre-processed with flow length and travel time information: case study of the Ganges river basin, Hydrol. Earth Syst. Sci., 13, 1607–1618, doi:10.5194/hess-13-1607-2009, 2009.

Altunkaynak, A., Özger, M., and Çakmakçı, M.: Fuzzy logic modeling of the dissolved oxygen fluctuations in Golden Horn, Ecol Model., 189(3–4), 436–446, 2005a.

Altunkaynak, A., Özger, M., and Çakmakçı, M.: Water consumption prediction of Istanbul City by using fuzzy logic approach, Water Resour. Manage., 19, 641–654, 2005b.

Alvisi, S., Mascellani, G., Franchini, M., and Bárdossy, A.: Water level forecasting through fuzzy logic and artificial neural network approaches, Hydrol. Earth Syst. Sci., 10, 1–17, doi:10.5194/hess-10-1-2006, 2006.

Anctil, F. and Lauzon, N.: Generalisation for neural networks through data sampling and training procedures, with applications to streamflow predictions, Hydrol. Earth Syst. Sci., 8, 940–958, doi:10.5194/hess-8-940-2004, 2004.

Awwad, H., Valdes, J., and Restrepo, P.: Streamflow forecasting for Han River Basin, Korea, J. Water Res. Pl.-ASCE, 120(5), 651–673, 1994.

Bishop, C. M.: Neural Networks for Pattern Recognition, 1st Edn., Oxford University Press, Oxford, UK, 1996.

HESSD

8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

Boucher, M.-A., Laliberté, J.-P., and Anctil, F.: An experiment on the evolution of an ensemble of neural networks for streamflow forecasting, Hydrol. Earth Syst. Sci., 14, 603–612, doi:10.5194/hess-14-603-2010, 2010.

Box, G. H. and Jenkins, G.: Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, 1970.

Bras, R. and Rodriguez-Iturbe, I.: Random Functions and Hydrology, Addison-Wesley, Reading, MA, 1985.

Campolo, M., Andreussi, P., and Soldati, A.: River flood forecasting with neural network model, Water Resour. Res., 35(4), 1191–1197, 1999.

Cheng, Y.: Evaluating an autoregressive model for stream flow forecasting, Proc. Hydraulic Eng. Conf., Hydraulic Engineering, New York, 1105–1109, 1994.

Chiu, C. L.: Applications of Kalman filter to hydrology, hydraulics, and water resources, Proceedings of AGU Chapman Conference, Pittsburgh, Department of Civil Engineering, University of Pittsburgh Press, 1978.

Cinar, O., Hasar, H., and Kinaci, C.: Modeling of submerged membrane bioreactor treating cheese whey wastewater by artificial neural network, J. Biotechnol., 123(2), 204–209, 2006.

Coulibaly, P., Anctil, F., and Bobe´e, B.: Daily reservoir inflow forecasting using artificial neural networks with stopped training approach, J. Hydrol., 230, 244–257, 2000a.

Coulibaly, P., Anctil, F., and Bobe´e, B.: Neural network-based long-term hydropower forecasting system, Comput.-Aided Civ. Inf., 15(5), 355–364, 2000b.

de Vos, N. J. and Rientjes, T. H. M.: Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation, Hydrol. Earth Syst. Sci., 9, 111–126, doi:10.5194/hess-9-111-2005, 2005.

Fernando, T. M. K. G., Maier, H. R., Dandy, G. C., and May, R. J.: Efficient selection of inputs for artificial neural network models, in: Proceedings of MODSIM 2005 International Congress on Modelling and Simulation, December 2005, edited by: Zerger, A. and Argent, R. M., Modelling and Simulation Society of Australia and New Zealand, 1806–1812, 2005.

Gibson, K. and Dodge, R. A.: Hydraulic Model Studies of Modification to Klang Gate Dam, Malaysia, Technical report, Engineering and Research Center, Denver, Colorado, 1983.

Haykin, S.: Neural Networks: Comprehensive Foundation, Prentice-Hall, Upper Saddle River, N. J., 1999.

Hiew, K. L: Flood mitigation and flood risk management in Malaysia, J. Floodplain Risk Manage., 11, 1205–1216, 1996.

HESSD

8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

Hsu, K. L., Gupta, H. V., and Sorooshian, S.: Artificial neural network modeling of rainfall-rainoff process, Water Resour. Res., 31(10), 2517–2530, 1995.
Hung, N. Q., Babel, M. S., Weesakul, S., and Tripathi, N. K.: An artificial neural network model for rainfall forecasting in Bangkok, Thailand, Hydrol. Earth Syst. Sci., 13, 1413–1425, doi:10.5194/hess-13-1413-2009, 2009.
Jain, S. K., Das, D., and Srivastava, D. K.: Application of ANN for reservoir inflow prediction and operation, J. Water Res. Pl.-ASCE, 125(5), 263–271, 1999.
Lapedes, A. and Farber, R.: Nonlinear Signal Processing using Neural Networks, Technical Report of Los Alamos National Laboratory, LAUR-87-2662, IEEE international conference on neural networks, San Diego, CA, USA, 1987.
Modarres, R.: Multi-criteria validation of artificial neural network rainfall-runoff modeling, Hydrol. Earth Syst. Sci., 13, 411–421, doi:10.5194/hess-13-411-2009, 2009.
Narendra, K. and Parthasarathy, K.: Identification and control of dynamical systems using neural networks, Trans. Neural Netw., 1, 4–27, 1990.
Noureldin, A., El-Shafie, A., and Taha, M. R.: Optimizing neuro-fuzzy modules for data fusion of vehicular navigation systems using temporal cross-validation, Eng. Appl. Artif. Intel., 20(1), 49–61, 2007.
Orr, M. J. L.: Introduction to Radial Basis Function Networks, University of Edinburgh, Scotland, 1996.
Ripley, B. D.: Pattern Recognition and Neural Networks, 1st Edition, Cambridge University Press, NY, USA, 1996.
Saad, M., Bigras, P., Turgeon, A., and Duquette, R.: Fuzzy learning decomposition for scheduling of hydroelectric power systems, Water Resour. Res., 32(1), 179–186, 1996.
Sajikumar, N. and Thandaveswara, B. S.: Non-linear rainfall runoff model using artificial neural network, J. Hydrol., 216, 32–35, 1999.
Siew, T.: Optimizing the operation release policy for Klang Gate Dam, Bachelor thesis, Faculty of Civil Engineering, Universiti Kebangsaan Malaysia, 2009.
Thirumalaiah, K. and Deo, M. C.: River stage forecasting using artificial neural networks, J. Hydrol. Eng.-ASCE, 3(1), 26–32, 1998.
Thirumalaiah, K. and Deo, M. C.: Hydrological forecasting using neural networks, J. Hydrol. Eng.-ASCE, 5(2), 180–189, 2000.
Tokar, A. S. and Johnson, P. A.: Rainfall-runoff modeling using artificial neural networks, J. Hydrol. Eng.-ASCE, 4(3), 232–239, 1999.

Weigend, A. S. and Gerschenfeld, N. A.: Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Reading, MA, 1994.
Weigend, A. S., Mangeas, M., and Srivastava, A.: Nonlinear gated experts for time series: discovering regimes and avoiding overfitting, Int. J. Neural Syst., 6, 373–399, 1995.
Xiong, Lihua, O'Connor, Kieran, M., and Guo, Shenglian: Comparison of three updating schemes using artificial neural network in flow forecasting, Hydrol. Earth Syst. Sci., 8, 247–255, doi:10.5194/hess-8-247-2004, 2004.
Zealand, C. M., Burn, D. H., and Simonovic, S. P.: Short term stream flow forecasting using artificial neural networks, J. Hydrol., 214, 32–48, 1999.

HESSD
8, 6489–6532, 2011

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

Table 1. RMSE and Maximum RE for monthly and weekly rainfall forecasting model utilizing IDNN.

Type of model	RMSE (mm)				Maximum RE %			
	Monthly		Weekly		Monthly		Weekly	
	Train	Test	Train	Test	Train	Test	Train	Test
IDNN One-time step	9.2	30.3	2.2	7.3	7.4	20.89	7.1	17.1
IDNN Two-time step	9.1	28.2	2.4	7.2	7.2	19.1	8.1	19.3

Table 2. IDNN performance based on the peak and low flow error criteria for monthly and weekly forecasting.

Year		PFC (%)	LFC (%)	Average (%)
Monthly	Train	4.20	1.80	3.00
	Test	6.15	3.07	4.61
Weekly	Train	8.50	2.40	5.45
	Test	9.20	6.30	7.75

HESSD
8, 6489–6532, 2011

Table 3. Models performance for matching the rainfall consequences.

Stage	Method	Monthly Positive	Monthly Negative	Percentage of corrected (%)	Weekly Positive	Weekly Negative	Percentage of corrected (%)
Train	MLP-NN	240	20	92	522	48	92
	RBFNN	248	12	95	528	42	93
	IDNN	235	7	90	553	17	97
Test	MLP-NN	15	9	63	38	12	76
	RBFNN	17	7	71	41	8	82
	IDNN	20	4	83	49	1	98

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

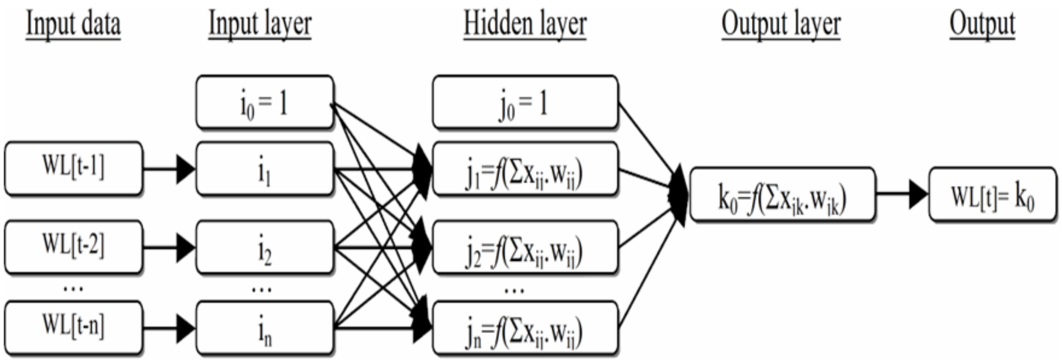

Fig. 1. The network model architecture.

HESSD
8, 6489–6532, 2011

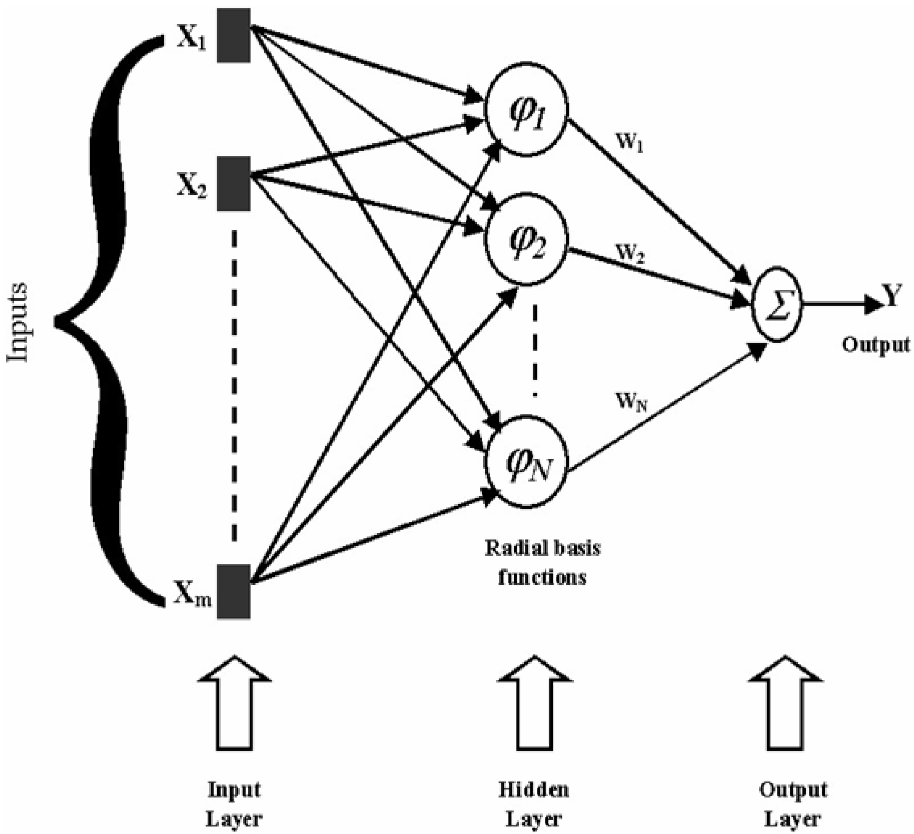

Fig. 2. Architecture of radial basis function neural network.

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

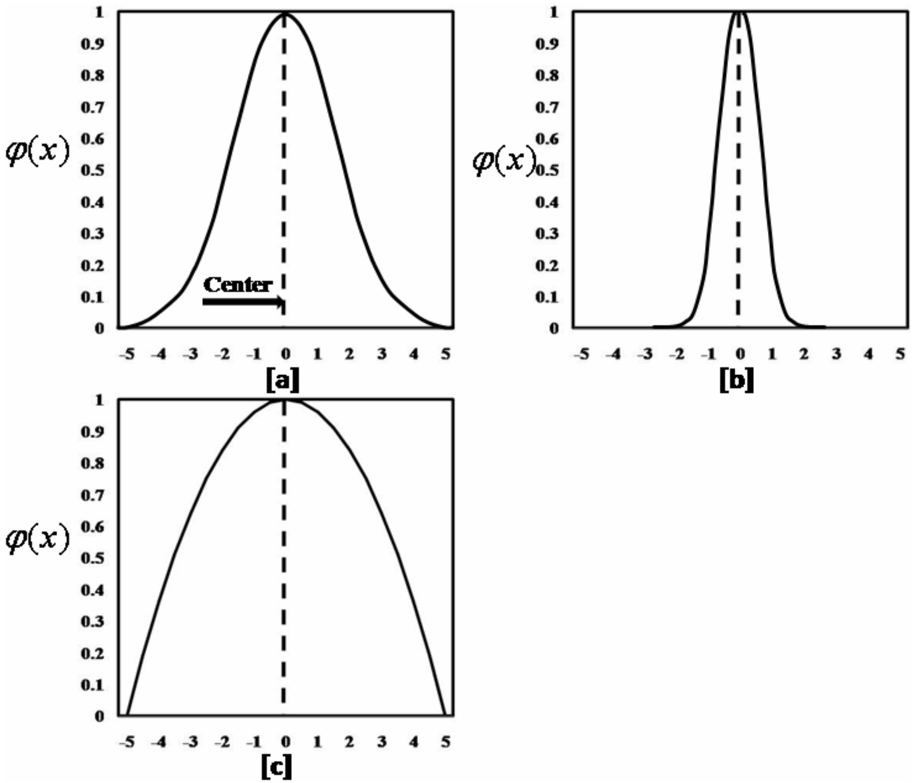

Fig. 3. Radial bases function with different levels of spread. **(a)** Normal spread, **(b)** small spread, **(c)** large spread.

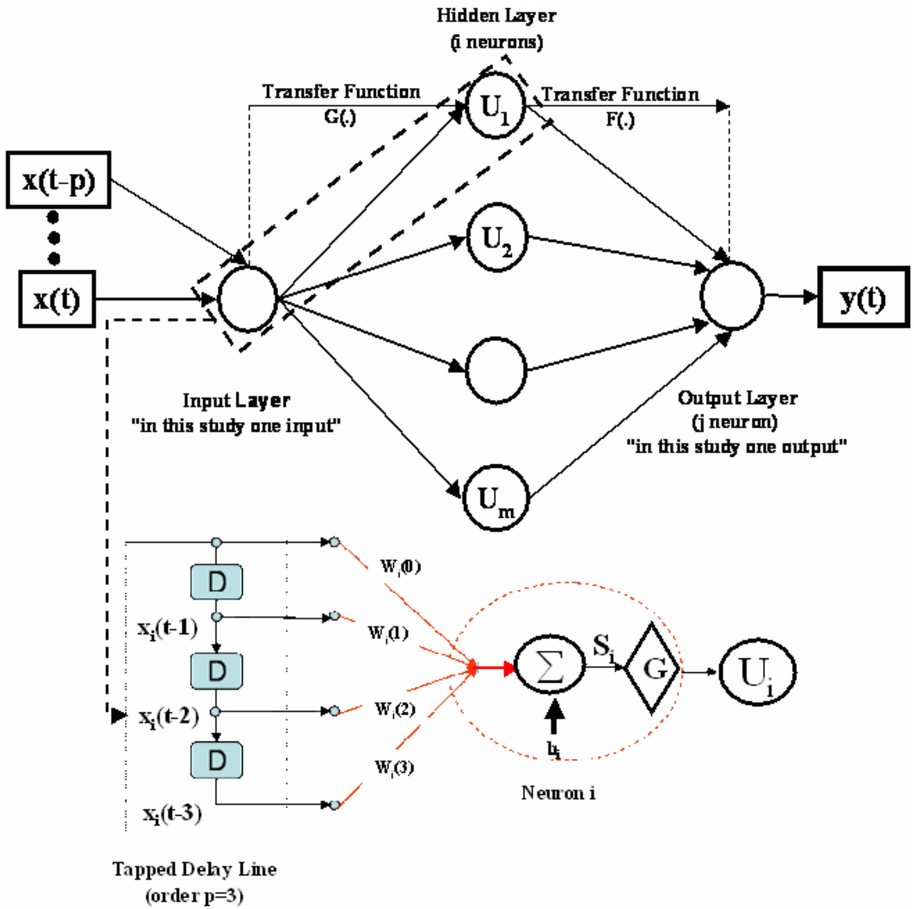

Fig. 4. Input delay neural network architecture and single neuron calculations.

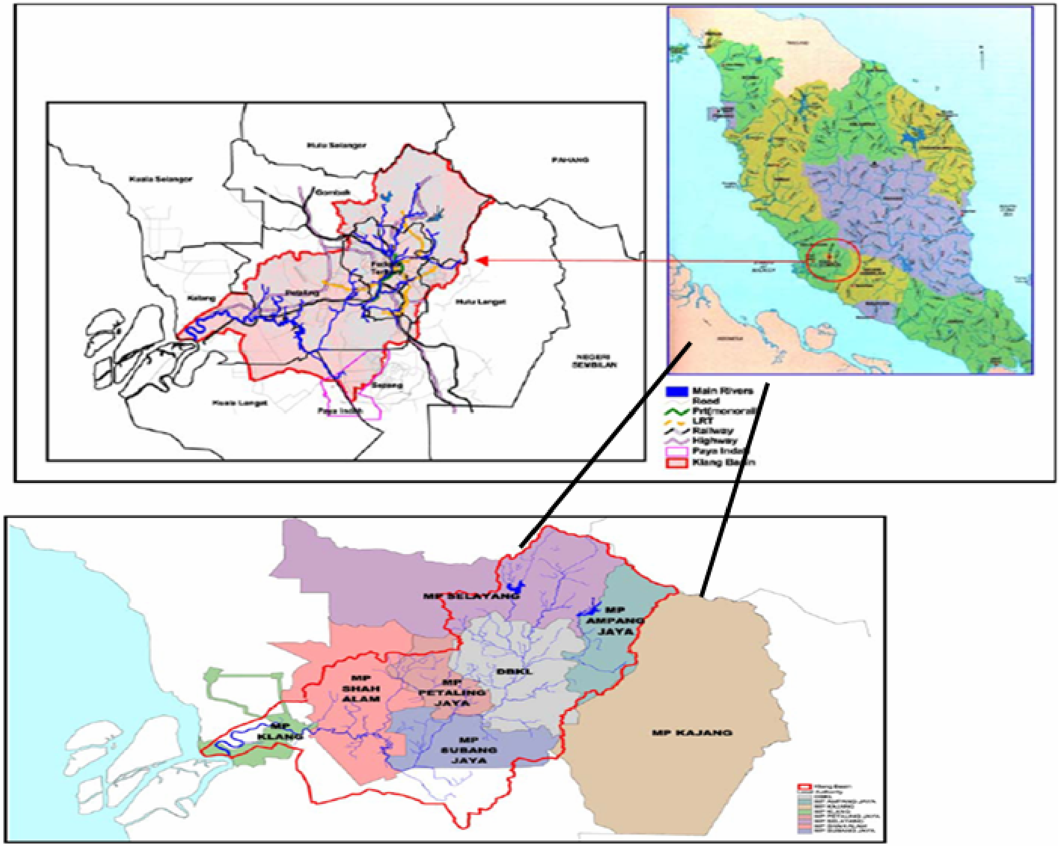

Fig. 5. Local authorities within Klang River Basin.

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

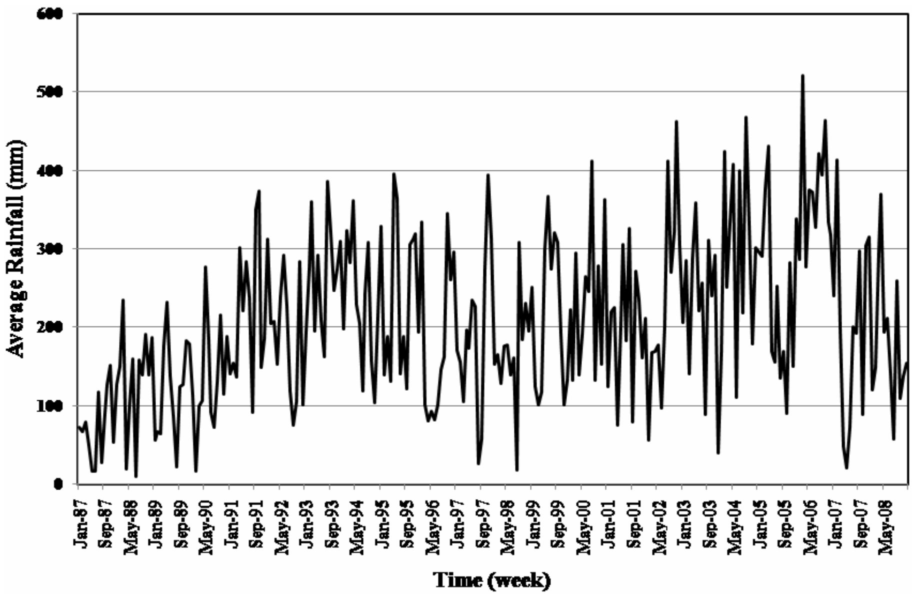

Fig. 6. Monthly actual rainfall records on Klang River on for period 1997–2008.

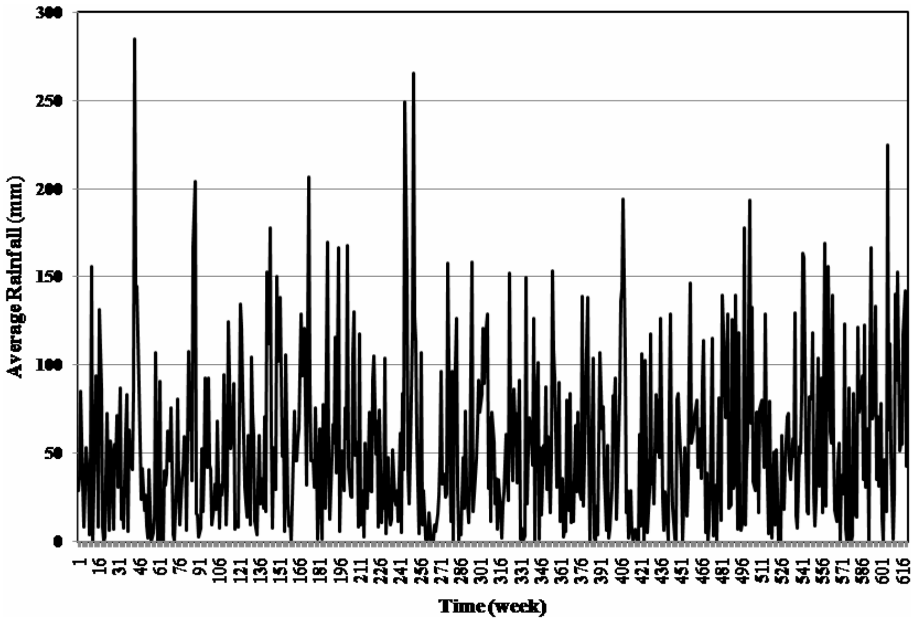

Fig. 7. Weekly actual rainfall records on Klang River on for period 1997–2008.

HESSD
8, 6489–6532, 2011

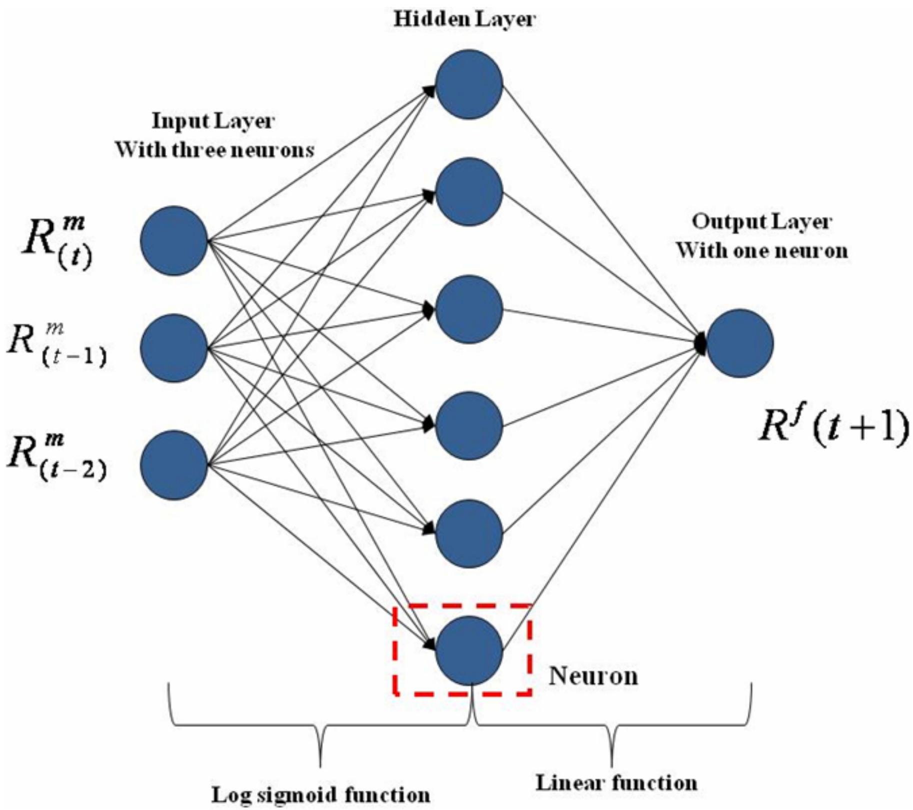

Fig. 8. Neural network model architechture utilized for rainfall forecasting.

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

Title Page
Abstract | Introduction
Conclusions | References
Tables | Figures
◀| |▶
◀ ▶
Back | Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

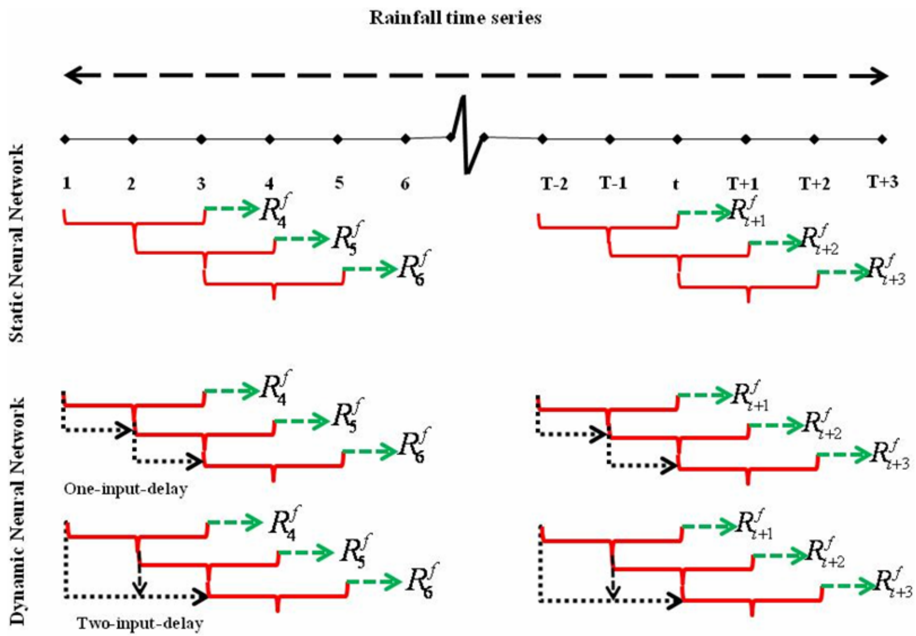

Fig. 9. Model time line index with sliding window method.

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

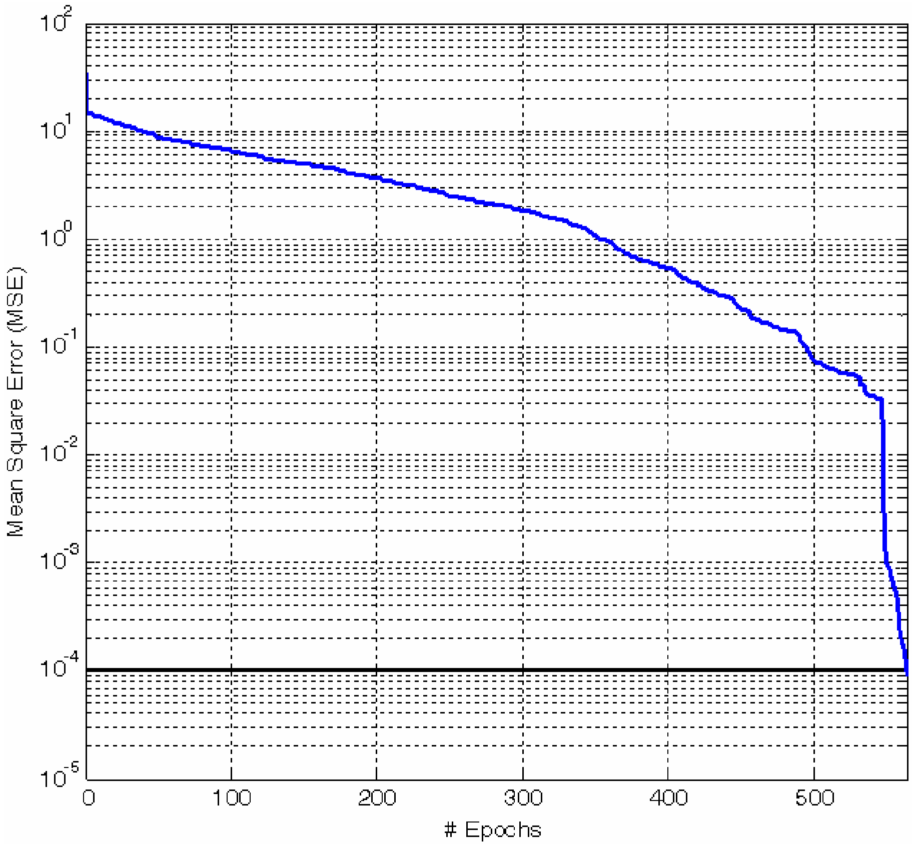

Fig. 10. Training curve for monthly basis using MLP-NN model.

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

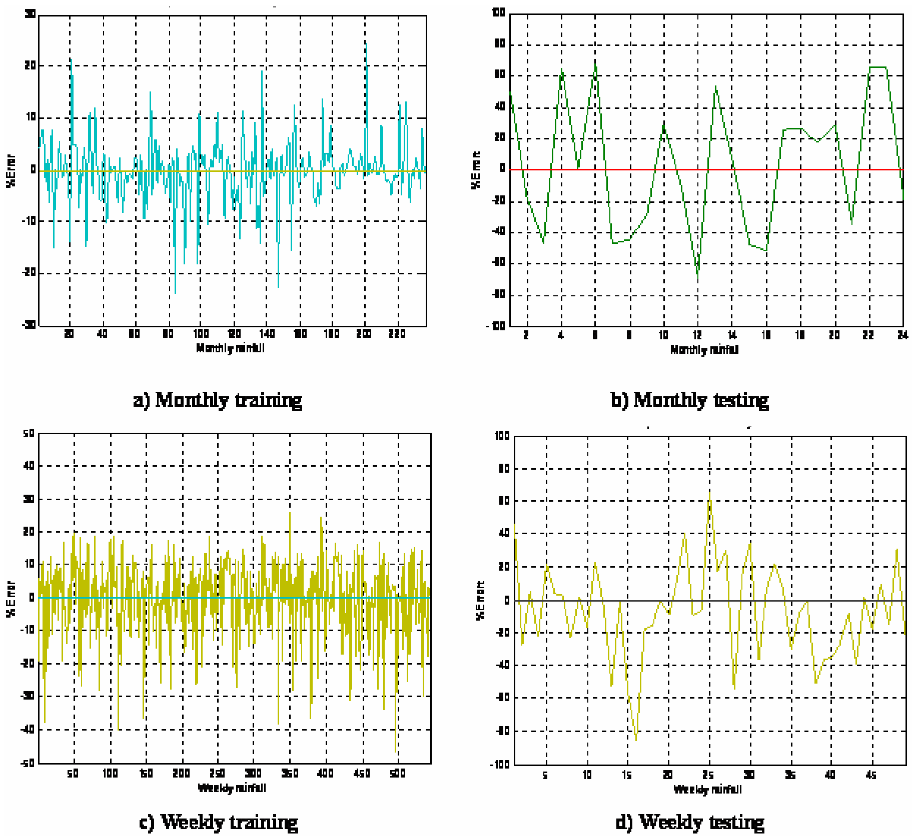

Fig. 11. Performance of the MLP-NN model for monthly and rainfall forecasting during training and testing stages.

HESSD
8, 6489–6532, 2011

Title Page
Abstract | Introduction
Conclusions | References
Tables | Figures
◀◀ | ▶▶
◀ | ▶
Back | Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

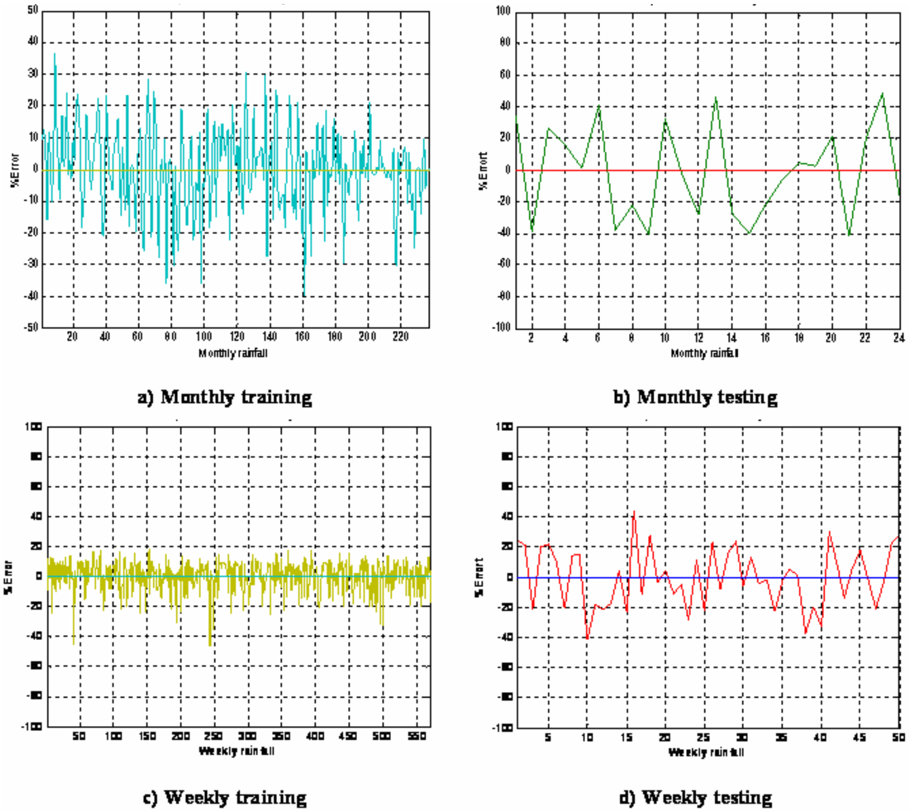

Fig. 12. Performance of the RBFNN model for monthly and rainfall forecasting during training and testing stages.

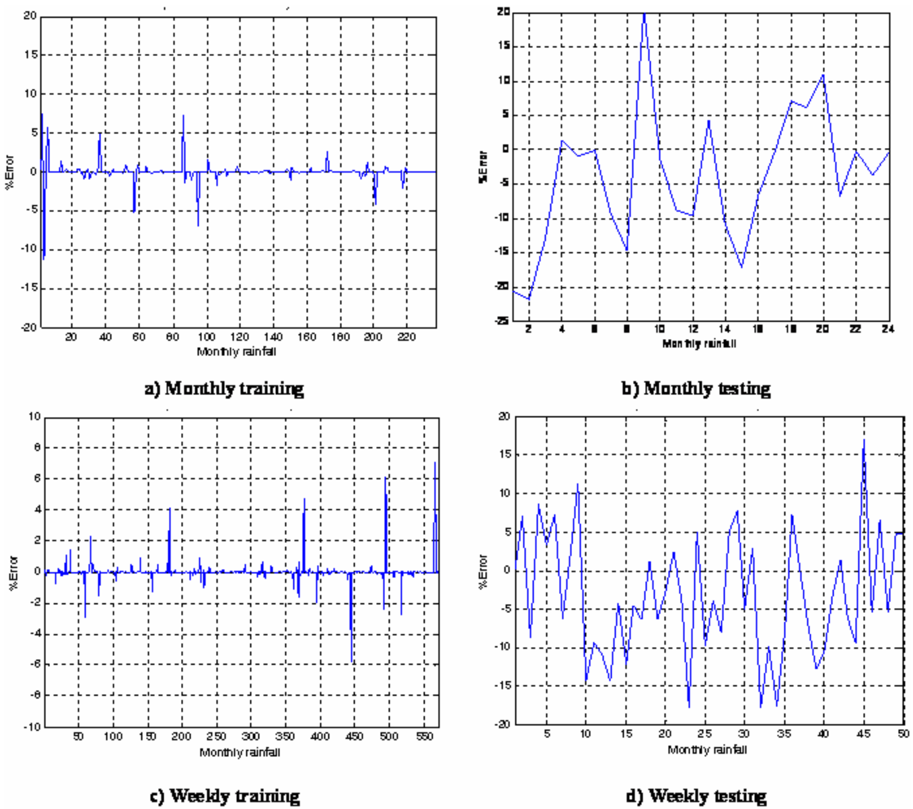

Fig. 13. Performance of the IDNN model for monthly and rainfall forecasting during training and testing stages.

HESSD
8, 6489–6532, 2011

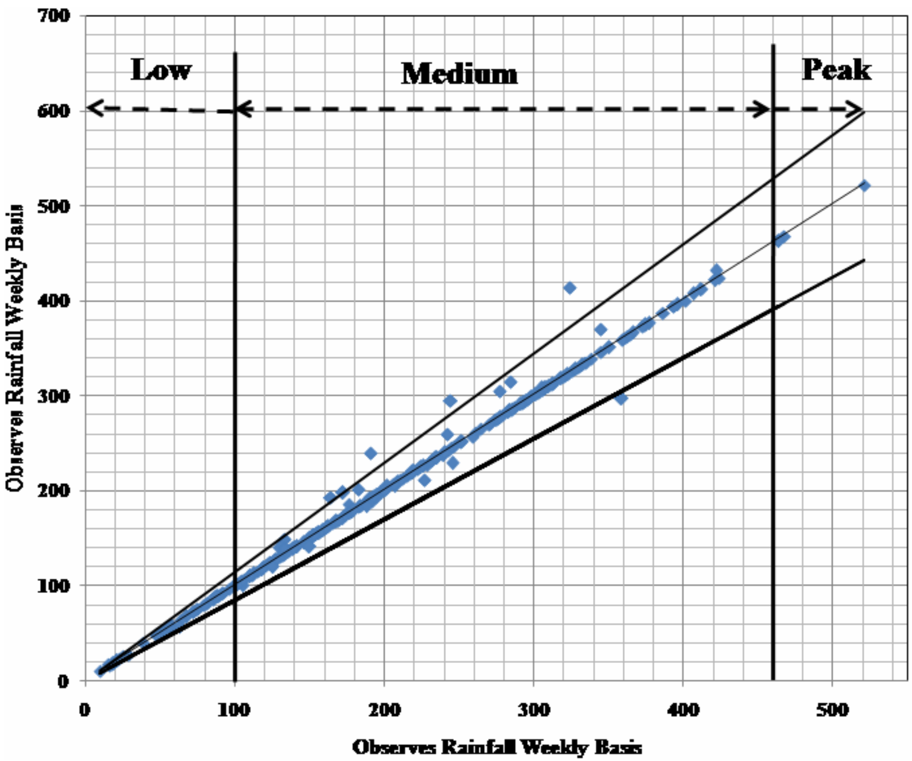

Fig. 14. Forecasted and actual rainfall on Klang River (weekly basis).

HESSD
8, 6489–6532, 2011

Dynamic versus static neural network model for rainfall forecasting, Malaysia

A. El-Shafie et al.

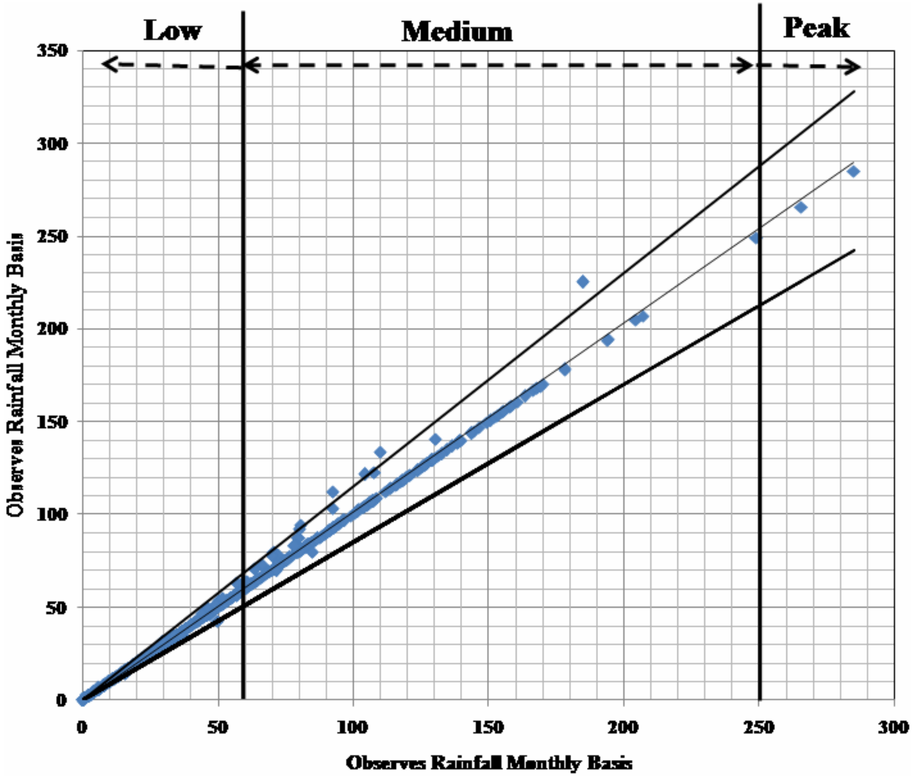

Fig. 15. Forecasted and actual rainfall on Klang River (monthly basis).

HESSD
8, 6489–6532, 2011