Research Article

Artificial Neural Network Modeling for Spatial and Temporal Variations of Pore-Water Pressure Responses to Rainfall

M. R. Mustafa,¹ R. B. Rezaur,² H. Rahardjo,³ M. H. Isa,¹ and A. Arif⁴

¹Civil Engineering Department, Universiti Teknologi Petronas, 31750 Tronoh, Perak Darul Ridzuan, Malaysia

²City of Moose Jaw, 228 Main Street North, Moose Jaw, SK, Canada S6H 3J8

³School of Civil and Environmental Engineering, Nanyang Technological University, Singapore 639798

⁴Engineering Physics Department, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia

Correspondence should be addressed to M. R. Mustafa; raza_geo@hotmail.com

Received 17 March 2014; Revised 28 June 2014; Accepted 8 August 2014

Academic Editor: Dimitrios Katsanos

Copyright © 2015 M. R. Mustafa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Knowledge of spatial and temporal variations of soil pore-water pressure in a slope is vital in hydrogeological and hillslope related processes (i.e., slope failure, slope stability analysis, etc.). Measurements of soil pore-water pressure data are challenging, expensive, time consuming, and difficult task. This paper evaluates the applicability of artificial neural network (ANN) technique for modeling soil pore-water pressure variations at multiple soil depths from the knowledge of rainfall patterns. A multilayer perceptron neural network model was constructed using Levenberg-Marquardt training algorithm for prediction of soil pore-water pressure variations. Time series records of rainfall and pore-water pressures at soil depth of 0.5 m were used to develop the ANN model. To investigate applicability of the model for prediction of spatial and temporal variations of pore-water pressure, the model was tested for the time series data of pore-water pressure at multiple soil depths (i.e., 0.5 m, 1.1 m, 1.7 m, 2.3 m, and 2.9 m). The performance of the ANN model was evaluated by root mean square error, mean absolute error, coefficient of correlation, and coefficient of efficiency. The results revealed that the ANN performed satisfactorily implying that the model can be used to examine the spatial and temporal behavior of time series of pore-water pressures with respect to multiple soil depths from knowledge of rainfall patterns and pore-water pressure with some antecedent conditions.

1. Introduction

Soil pore-water pressure is an important variable contributing to soil shear strength. In tropical regions unsaturated soil conditions above the groundwater table contribute to the stability of a slope due to the additional shear strength provided by the negative pore-water pressures of the unsaturated soil. However, the magnitude of the negative pore-water pressure is largely influenced by the climatic conditions (e.g., rainfall, evaporation, temperature, daily sunshine hours, etc.). During dry periods the soil undergoes drying due to evaporation and transpiration and as a result the pore-water pressure gradually becomes more negative over time. During wet periods the soil undergoes wetting due to rainfall and consequently the pore-water pressure becomes less negative or even positive. This leads to a decrease in the soil shear strength and may

eventually trigger a slope failure [1]. Time series of soil pore-water pressures in response to climatic conditions, therefore, exhibits highly dynamic, nonlinear, and complex behavior [2].

Knowledge of the variation in soil pore-water pressure is of prime importance in hydrogeological studies like slope stability analyses, seepage analyses, engineered slope design, and studies related to hillslope hydrological responses [1, 3, 4]. However, field instrumentation to get knowledge of pore-water pressure in a slope is challenging and difficult task. Usually, pore-water pressure information needed for such purposes either is obtained from previous field measurements made on a different site or is measured through a field instrumentation program. Such practices pose some concerns, like appropriateness of the use of pore-water pressure data from a different site whose climatic and geographic

settings might be different from the site under consideration, the time and cost involved in collecting fresh data on pore-water pressure through a field instrumentation program.

Recently, few studies have been performed to predict soil pore-water pressure using radial basis function (RBF) [5] and multilayer perceptron (MLP) [6, 7] neural networks. Mustafa et al. [5] used time series of pore-water pressure from one slope during training stage and successfully tested the RBF model for the time series data of pore-water pressure at different slope. Mustafa et al. [6] investigated the effect of antecedent values rainfall and pore-water pressure data on the prediction of pore-water pressure while using scaled conjugate training algorithm in MLP neural networks. Later on, in 2013, Mustafa et al. [7] investigated the effect of various training algorithms on the performance of pore-water pressure training algorithms. They compared the performance of four different training algorithms and suggested that Levenberg-Marquardt (LM) is advantageous due to its fast convergence and automated ability to adjust learning rate. All the above mentioned studies showed that artificial neural network (ANN) techniques have robust ability to predict soil pore-water pressure variations in a slope. However, according to authors' knowledge, no study has been performed to capture spatial and temporal behavior of soil pore-water pressure data at multiple soil depths in a slope.

Numerical modeling for prediction or estimation of hydrogeological variables is common in practice and recently has been applied to solve different hydrogeological related issues. Applications of ANN to problems related to hydrogeological studies are also recent [8]. It has been applied with success to hydrogeological related problems such as forecasting, for example, river flow [9, 10], water level [8, 11], river flood [12], and runoff from rainfall [13], prediction, for example, river suspended sediment [14–16], hourly and daily stream flows [17], river flow [18], runoff [19], peak discharge, and time to peak [20, 21], water quality [22, 23], and modeling and simulation, for example, stage-discharge relationship [24], rainfall-runoff process [25–28], and sheet sediment transport [29].

However, it appears that the aspect of prediction of time series of pore-water pressure variations in a slope at multiple soil depths using ANN has not been examined. Owing to the ability of ANNs to deal with time series data with nonregular cyclic variations, ANN seems to be an ideal choice for developing a pore-water pressure prediction model at multiple soil depths in a slope. The objectives of this study are therefore to (i) examine the applicability of ANN in modeling spatial and temporal changes of the time series of field measured pore-water pressures in a slope at multiple soil depths and (ii) to identify an appropriate ANN structure and parameters for modeling the pore-water pressures behavior.

2. Artificial Neural Network Theory

ANNs with different architecture [30] have found application for the prediction and forecasting of hydrologic variables. In this study a multilayer perceptron (MLP) feed forward (FF) neural network structured with Levenberg-Marquardt

training algorithm was established on the premise that MLP-FF network has been widely used. It has been found to be ideal for approximating function with a finite number of discontinued subjects that the learning is sufficient [31]. The Levenberg-Marquardt training algorithm provides second order training speed, is a trust region based method, and is considered the most efficient for training median sized ANN [32]. Also by making only a change in the value of the learning rate (μ) the LM training algorithm can be converted to gradient descent backpropagation or Quasi-Newton training algorithm. Mustafa et al. [7] performed a thorough comparison between the performances of Levenberg-Marquardt, scaled conjugate gradient, gradient decent, and gradient decent with momentum training algorithms for prediction of soil pore-water pressure and suggested LM is advantageous.

2.1. Network Architecture. Artificial neural networks structure consists of a set of data processing elements called nodes (or neurons) arranged in layers. The first and last layers are named as input and output layers, respectively, whereas the layers in between are known as hidden layers. The layers are interconnected via nodes of each layer through weighted interconnections. The number of nodes in the input layer depends on the number of inputs to the network and the number of nodes in the output layer is restricted to the number of outputs expected from the network. The number of hidden layers and the number of nodes in the hidden layer depend on the complexity of the problem. The architecture of an ANN is designed by weights between the nodes, activation function, and learning laws [33].

The architecture of a MLP network is explained here in the context of a generalized case, where a training set $\{(x_1, z_1), \dots, (x_p, z_p), \dots, (x_p, z_p)\}$ consisting of P ordered pairs of vectors called the input and target patterns, respectively, is to be mapped through Levenberg-Marquardt training algorithm in a MLP network with L nodes in the input layer, N nodes in the output layer, and M nodes in a single hidden layer. A schematic representation of the MLP network architecture and activity levels generated by the nodes in different layers is shown in Figure 1.

The network shown in Figure 1 consists of three layers, namely, input, hidden, and output. Subscript denotes specific node from within a layer. Double subscript in connection weight denotes destination and source nodes, respectively. Each node in the hidden and output layer is composed of three units. The first unit denoted by the symbol Σ represents a linear combiner that adds products of weight's coefficients and input signal. The second unit denoted by the symbol f is called the neuron activation function and it realizes function. As the training algorithm makes use of the gradient of the error function to update connection weights and minimize the error function, the activation function to be used at each node of the network must be continuous and differentiable. The third unit (optional) denoted by the letter b is called bias. Bias provides a shift in the activation function and can have a value of either 0 or 1. Biases can be treated as another weight that is connected to a fictitious input of either 0 or 1. Bias values provide an additional degree of freedom to the network and assist in attaining the best

FIGURE 1: Schematic representation of a multilayer feed forward (MLP) network showing activity levels generated by the neurons in different layers.

weights during training stage. The connection weights are real numbers selected at random.

2.2. Functioning of ANN. There are two stages of ANN application, (i) training and (ii) testing. During training when an input pattern x_p from the training set $\{(x_1, z_1), \dots, (x_p, z_p), \dots, (x_P, z_P)\}$ is presented to the network, it produces an output y_p different in general from the target z_p . The objective of training is to make y_p and z_p identical for, $p = 1 \dots P$, pairs of training data by using an error minimizing rule called the learning algorithm. More precisely, the objective of the training is to minimize the global error function E of the network which is based on the squared difference between the network output and target for all training sets, defined as

$$E = \sum_{p=1}^P E_p = \sum_{p=1}^P \sum_{k=1}^N (z_{pk} - y_{pk})^2, \quad (1)$$

where y_{pk} is the output of the k th node in the output layer for the p th data pattern, E_p is the total output error from all the nodes in the output layer for the p th data pattern, and E is the global error for P pairs of training data.

The learning rule updates connection weights and finds the local minimum to minimize the error function E to a predefined tolerance. After minimizing the error function E for the training set, new unknown input patterns are presented to the network to produce outputs. This is called testing. The network must recognize whether a new input vector is similar to learned patterns and produce a similar output.

As earlier mentioned, there are several algorithms available for training a network, namely, (i) gradient descent backpropagation algorithm, (ii) gradient descent backpropagation with momentum algorithm, (iii) conjugate gradient backpropagation algorithm, (iv) Quasi-Newton algorithm, and (v) Levenberg-Marquardt training algorithm. However, the Levenberg-Marquardt training algorithm is receiving increased popularity because of the ease with which it can

be changed to gradient descent backpropagation or Quasi-Newton algorithm and also the learning can be made to adjust automatically once the increment and decrease in the learning rate are predefined.

MLP networks with Levenberg-Marquardt training algorithm involve five steps for training. These are (i) initialization of weights, biases, and presenting training pair of input and target data to the network, (ii) feed forward computation for network output error, (iii) computation of Jacobian matrix of network error and weight update using Levenberg-Marquardt learning rule, (iv) adjustment of learning parameter, and (v) repeating step (ii) to step (iv) for all data patterns until the error is minimized to a predefined tolerance.

2.2.1. Initialization of Network Training Parameters. In this step the connection weights and biases are initialized and input pattern x_p from the training set $\{(x_1, z_1), \dots, (x_p, z_p), \dots, (x_P, z_P)\}$ is presented to the network.

2.2.2. Feed Forward Computation for Network Output Error. During feed forward computation when a node receives weighted inputs from all the nodes in the previous layer they are summed up and the constant bias is added to compute the net input to the node. The net input is then converted to an activated value through the activation function to generate the output of the node. The output of a node is then passed to the neurons in the next layer. The same operation is repeated in all nodes of the subsequent layers until the output layer is reached and output of nodes in the output layer is generated. This completes the forward pass. The typical nodes, i ($1 \leq i \leq L$), j ($1 \leq j \leq M$), and k ($1 \leq k \leq N$), in the input, hidden, and output layer, respectively (Figure 1), can be used to explain the feed forward computations. Thus the net input n_j to the j th node in the hidden layer can be expressed as

$$n_j = \sum_{i=1}^L x_i w_{ji} + b_j. \quad (2)$$

The output y_j from the j th node in the hidden layer can be expressed as

$$y_j = a_j = f_j(n_j) = f_j\left(\sum_{i=1}^L x_i w_{ji} + b_j\right). \quad (3)$$

The net input n'_k to the k th node in the output layer can be expressed as

$$\begin{aligned} n'_k &= \sum_{j=1}^M y_j w_{kj} + b_k \\ &= \sum_{j=1}^M w_{kj} \left(f_j \left(\sum_{i=1}^L x_i w_{ji} + b_j \right) \right) + b_k. \end{aligned} \quad (4)$$

Thus the output y_k from the k th node in the output layer can be expressed as

$$\begin{aligned} y_k &= a'_k = f_k(n'_k) \\ &= f_k \left(\sum_{j=1}^M w_{kj} \left(f_j \left(\sum_{i=1}^L x_i w_{ji} + b_j \right) \right) + b_k \right). \end{aligned} \quad (5)$$

The error e_k for the k th node in the output layer can be expressed as

$$e_k = (z_k - y_k)^2, \quad (6)$$

where z_k and y_k are the output and target for the k th node in the output layer. The total error from all the nodes in the output layer for the p th pattern can be expressed as

$$e_p = \sum_{k=1}^N (z_{pk} - y_{pk})^2 = \sum_{k=1}^N e_{pk}. \quad (7)$$

The global error when all the training patterns are introduced can be expressed as

$$\begin{aligned} E &= \sum_{p=1}^P e_p = \sum_{p=1}^P \sum_{k=1}^N (z_{pk} - y_{pk})^2 \\ &= \sum_{p=1}^P \sum_{k=1}^N e_{pk}. \end{aligned} \quad (8)$$

2.2.3. Computation of Jacobian Matrix Using Levenberg-Marquardt Algorithm. It is obvious (from (5)) that the output of each node in the output layer is a function of the connection weights. To minimize the error function it is necessary to have

$$\nabla E(w) = \frac{\partial E(w)}{\partial w_i} = 0 \quad \text{for all the weights } i. \quad (9)$$

When the gradient of the network error function $\nabla E(w_t)$ is expanded using Taylor's series around the current time $t + 1$ it gives

$$\nabla E(w_{t+1}) = \nabla E(w_t) + [w_{t+1} - w_t] \nabla^2 E(w_t) + \dots \quad (10)$$

Neglecting higher order terms and solving for the minimum w , by setting the left hand side of (10) to zero, the weight update rule for Newton's method is simplified and leads to

$$w_{t+1} = w_t - [\nabla^2 E(w_t)]^{-1} \nabla E(w_t). \quad (11)$$

Since, from (8), $E(w) = \sum_{p=1}^P e_p(w)$, it can be shown that

$$\nabla E(w_t) = \mathbf{J}(w_t)^T \mathbf{e}(w_t), \quad (12)$$

$$\nabla^2 E(w) = \mathbf{J}(w)^T \mathbf{J}(w) + \sum_{i=1}^P e_p \nabla \nabla e_p(w). \quad (13)$$

$\mathbf{J}(w_t)$ is known as the Jacobian matrix of $\mathbf{e}$ pertaining w calculated at previous iteration, $\mathbf{J}(w_t)^T$ is the transpose of the Jacobian matrix calculated at previous iteration, and $\mathbf{e}(w_t)$ is the matrix of error $\mathbf{e}(w_t) = [z_1 - y_1 \quad \dots \quad z_k - y_k \quad \dots \quad z_N - y_N \quad \dots \quad z_P - y_P]^T$ evaluated at previous iteration. In Gauss-Newton approach it is assumed that $\sum_{i=1}^P e_p \nabla \nabla e_p(w) = 0$. Therefore (13) reduces to

$$\nabla^2 E(w) \approx \mathbf{J}(w)^T \mathbf{J}(w) = \mathbf{H}. \quad (14)$$

Substitution of (12) and (14) in (11) results in

$$w_{(t+1)} = w_t - [\mathbf{J}(w_t)^T \mathbf{J}(w_t)]^{-1} \mathbf{J}(w_t)^T \mathbf{e}(w_t), \quad (15)$$

where $\mathbf{H}$ is the Hessian matrix. Equation (15) suggests that the Hessian matrix $\mathbf{H}$ can be obtained from the Jacobian matrix as $\mathbf{J}^T \mathbf{J}$ without direct calculation for the Hessian matrix. In fact, (15) is the weight update equation in Gauss-Newton's algorithm. When a scalar parameter μ and an identity matrix $\mathbf{I}$ are introduced to (15) it becomes

$$w_{(t+1)} = w_t - [\mathbf{J}^T(w_t) \mathbf{J}(w_t) + \mu_t \mathbf{I}]^{-1} \mathbf{J}^T(w_t) \mathbf{e}(w_t), \quad (16)$$

where w_{t+1} is the new connection weight, w_t is the connection weight at the previous iteration, $\mathbf{J}(w_t)$ is the Jacobian matrix of $\mathbf{e}$ with respect to w evaluated at previous iteration, $\mathbf{J}^T(w_t)$ is the transpose of the Jacobian matrix evaluated at previous iteration, $\mathbf{e}(w_t)$ is the matrix of error $\mathbf{e}(w_t) = [z_1 - y_1 \quad \dots \quad z_k - y_k \quad \dots \quad z_N - y_N \quad \dots \quad z_P - y_P]^T$ evaluated at previous iteration, and $\mathbf{I}$ is an identity matrix.

Equation (16) is the weight update equation in Levenberg-Marquardt training algorithm. The parameter μ is a scalar that controls the behavior of the algorithm and is called the learning rate. For $\mu = 0$ the Levenberg-Marquardt algorithm becomes Quasi-Newton algorithm with the use of appropriate Hessian matrix. For very large value of μ the Levenberg-Marquardt algorithm becomes gradient descent

backpropagation algorithm. The Jacobian matrix of networks errors can be written as

$$\mathbf{J}(w) = \begin{bmatrix} \frac{\partial e_1}{\partial w_{11}} & \dots & \frac{\partial e_1}{\partial w_{ji}} & \dots & \frac{\partial e_1}{\partial w_{ML}} & \dots & \frac{\partial e_1}{\partial w_{kj}} & \dots & \frac{\partial e_1}{\partial w_{NM}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_k}{\partial w_{11}} & \dots & \frac{\partial e_k}{\partial w_{ji}} & \dots & \frac{\partial e_k}{\partial w_{ML}} & \dots & \frac{\partial e_k}{\partial w_{kj}} & \dots & \frac{\partial e_k}{\partial w_{NM}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_N}{\partial w_{11}} & \dots & \frac{\partial e_N}{\partial w_{ji}} & \dots & \frac{\partial e_N}{\partial w_{ML}} & \dots & \frac{\partial e_N}{\partial w_{kj}} & \dots & \frac{\partial e_N}{\partial w_{NM}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_p}{\partial w_{11}} & \dots & \frac{\partial e_p}{\partial w_{ji}} & \dots & \frac{\partial e_p}{\partial w_{ML}} & \dots & \frac{\partial e_p}{\partial w_{kj}} & \dots & \frac{\partial e_p}{\partial w_{NM}} \end{bmatrix}, \quad (17)$$

where L , M , and N are the number of neurons in the input, hidden, and output layers, $e_1, e_2, \dots, e_p$ are the errors, and w_{ij} are the connection weights.

2.2.4. Adjustment of Learning Parameters. Levenberg-Marquardt training algorithm updates the weights by using (16). The Levenberg-Marquardt training algorithm can be established to find learning rate automatically at each iteration by introducing μ , μ_{inc} , and μ_{dec} parameters. For the LM training algorithm evaluated in this study $\mu = 0.001$, $\mu_{inc} = 10$, and $\mu_{dec} = 0.01$ were used.

2.2.5. Repeating Steps (ii) to (iv) until Achieving the Goal. Since during the forward pass cycle information is passed in the forward direction through the neurons in the hidden layer(s) until it reaches the neurons in the output layer where outputs are generated and errors are computed, such networks derive the name feed forward (FF) network and since errors are minimized through updating the connection weights by a rule where errors are propagated in the backward direction the learning law derives the name backpropagation algorithm.

A complete representation of all the training data (input and target data) is known as epoch. Epochs are repeated until the network reaches a predefined goal. The learning rate defines the size of the changes that are made to the weights and biases at each epoch. Generally, smaller value of learning rate increases the number of epochs and slows down the network convergence but produced better accuracy. Conversely, large value of learning rate leads the network to fast convergence but with less accuracy.

3. Methodology

3.1. Data Source. The data used in this study are synchronized measurement on time series of rainfall and pore-water pressure records from 0.5 m, 1.1 m, 1.7 m, 2.3 m, and 2.9 m soil depths, of 4 hr resolution during dry periods (no rainfall) and of 10 min resolution during rainfall (wet period). Schematic arrangement of field instrumentation and location of site slopes are shown in Figure 2. A trace of the time series of pore-water at 0.5 m soil depth responses to climatic changes is shown in Figure 3. The data were collected through a field instrumentation program of a residual soil slope in Yishun, Singapore [34]. The entire monitoring program ranged over

FIGURE 2: Schematic diagram of typical arrangement of instrumentation and location of the study slopes adapted from Mustafa et al., 2012.

FIGURE 3: A trace of time series of rainfall and pore-water pressures at 0.5 m soil depth showing pore-water pressure responses to climatic changes. (Data ranges used for training and testing of the ANN model are shown in figure inset.)

a period of three years and included time series of pore-water pressure and rainfall measurements at 4 different slopes in 2 major geological formations (sedimentary Jurong formation and Bukit Timah granitic formation) in Singapore [34]. Pore-water pressure measurements were made at 0.5, 1.1, 1.7, 2.3, and 2.9 m soil depths. The data were primarily collected with a view to understand rainfall-induced slope failure mechanism

TABLE 1: Normal statistics of data used for training and validating the ANN model (soil depth = 0.5 m).

Statistics*	Model training data		Model testing data	
	Rainfall	Pore-water pressure	Rainfall	Pore-water pressure
<i>N</i>	2995	2995	1000	1000
Max.	44.75	4.85	38.50	5.54
Min.	0.00	-55.90	0.00	-49.63
μ	0.45	-3.10	0.47	-2.83
σ	1.87	9.69	2.66	9.51
σ^2	3.50	93.88	7.09	90.44
CV (%)	411.33	-312.48	572.24	-336.27

* *N* = number of observations; Max. = maximum; Min. = minimum; σ = standard deviation; μ = sample mean; σ^2 = variance; CV = coefficient of variation.

and hydrological responses of slopes under tropical climate [1–4, 35]. The pore-water pressure data from 0.5 m soil depth was used during training the ANN model on the premise that they were the closest to the ground surface and therefore likely to show more dynamic behavior to climatic changes than at greater soil depths, which is deemed necessary to examine the potential of ANN in predicting the dynamic behavior of pore-water pressure at multiple soil depths.

3.2. Data Selection for Training and Testing. Achievement in nonlinear complex behavior identification using ANN depends largely on the selection of training data which represents the complexity and nonlinear trend to be learnt by the ANN during training [36]. Therefore, it is necessary to select an appropriate division of input data for the training and testing stage of the network. The available time series of pore-water pressure and rainfall data at 0.5 m depth was divided into two sets; about 70% of the available data (from October 12, 1998, to July 9, 1999, 9 months) was used for training and validation and the remaining 30% (from July 10, 1999, to November 20, 1999, 4 months) was used for testing the network (Figure 3). The training data was selected on the premise to include maximum variations of the pore-water pressure pattern and incorporate the highest and lowest values of the available data set. The testing of pore-water pressure at multiple soil depths (1.1, 1.7, 2.3, and 2.9 m) was also performed in the same period, that is, from July 10, 1999, to November 20, 1999, 4 months. The most recent data was used for testing to evaluate the performance of the model in predicting pore-water pressures. The statistical characteristic of the data used for training and testing the ANN model is shown in Table 1. The time series of rainfall and pore-water pressure data were scaled in accordance with the limits of the activation function use in hidden neurons. The data were normalized in the range of -1 to 1.

3.3. Selection of Input and Output Variable and Dimensions. Selection of appropriate model input is extremely important as the input variables would contain complex autocorrelation structure of the process to be mapped by the ANN. Furthermore, the number of input variables will dictate the number of nodes in the input layer and hence the ANN architecture. While too few or inappropriate input variables may lead to failure of the network to map the underlying function,

too many input variables may lead to a large and inefficient network in the form of redundancies in the connection weights of the network. Therefore, it is important to choose the appropriate input variables. The choice of input variables is usually based on a priori knowledge of causal variables in conjunction with inspection of time series plots of potential inputs and outputs.

In the case of time series of pore-water pressure, it is recognized that pore-water pressure changes are caused by rainfall, evaporation, soil properties, and soil depth as well as antecedent rainfall and antecedent pore-water pressure condition of the soil [1, 2, 4]. In the absence of data on other variables, in this study it is presumed that pore-water pressure is a function of rainfall, antecedent rainfall, and antecedent pore-water pressures. However, how many antecedent rainfall and pore-water pressure measurement events are to be included as input variable needs to be carefully evaluated. Previous studies [1, 2, 4] on evaluating the effect of antecedent rainfall on slope stability indicated that a 5-day antecedent rainfall is needed to lead to the worst pore-water pressure conditions (zero or positive pore-water pressure) in the slope.

To identify the appropriate number and type of input variables, many options including only rainfall, rainfall and some antecedent rainfall, and rainfall and pore-water pressure with antecedent conditions were examined. However, rainfall and pore-water pressure with antecedent conditions were found to be appropriate. To find appropriate number of antecedent values the autocorrelation between pore-water pressure data and cross-correlation between pore-water pressure and rainfall were tested. Influence of antecedent rainfall on pore-water pressure changes was established from an analysis of cross-correlation between pore-water pressure and rainfall data. Influence of antecedent pore-water pressures was established from an analysis of both autocorrelation of pore-water pressure and cross-correlation between pore-water pressure and rainfall data.

The cross-correlation function (CCF) between pore-water pressure and rainfall is shown in Figure 4(a). Positive correlation values (rising limbs of CCF) in Figure 4(a) represent that positive cross-correlation values (rising limbs of CCF) are associated with rainfall (wetting of soil) while negative cross-correlation values (falling limbs of CCF) are not associated with rainfall (no rainfall). Examination of the plot of time

FIGURE 4: (a) Cross-correlation between pore-water pressure and rainfall. (b) Autocorrelation of pore-water pressures.

series of pore-water pressure and rainfall (Figure 3) suggested that changes in pore-water pressure occur both during wet period (due to rainfall) and during dry period (no rainfall). The changes in pore-water pressures during positive cross-correlation were due to the effect of rainfall while changes in pore-water pressures during negative cross-correlation were a consequence of drying of the soil due to evaporation and transpiration.

Furthermore, considering the peak positive and negative cross-correlation values, the rising limb is associated with two lag observations and the falling limb is associated with five lag observations. It is perceived that rising limb indicates the increase in the pore-water pressure and the increase in pore-water pressure is because of the rainfall values, while falling limb or the decrease in the pore-water pressure indicates the dry period when there is no rainfall. Therefore, these analyses suggest that 2 antecedent rainfall values and 5 antecedent pore-water values would be ideal as model input. The autocorrelation function (ACF) of pore-water pressure is shown in Figure 4(b). It is apparent from Figure 4(b) that pore-water pressures are positively autocorrelated which implies that as the number of lag observations increases the autocorrelation between pore-water pressures decreases.

Thus it was concluded that pore-water pressure at the present time u_t can be represented as a function of the present and two antecedent rainfalls (r_t, r_{t-1}, r_{t-2}) and five antecedent pore-water pressures ($u_{t-1}, u_{t-2}, u_{t-3}, u_{t-4}, u_{t-5}$) such that

$$u_t = f(r_t, r_{t-1}, r_{t-2}, u_{t-1}, u_{t-2}, u_{t-3}, u_{t-4}, u_{t-5}). \quad (18)$$

Therefore, the appropriate number of inputs for the network was decided to be eight which includes five antecedent pore-water pressures and three rainfall values (1 present and 2 antecedents). Therefore, the total number of input neurons was selected to be eight.

One hidden layer was found to be sufficient to develop the required pore-water pressure prediction model. Thus, the complete representation of the model has been expressed by maximum of three layers, namely, input, hidden, and output. There is no defined rule for selecting the number of neurons

in the hidden layer. Some studies report the trial number of neurons in the hidden layer in their studies to follow the thumb rule, $N_H = N_I + 1$, [22] or $N_H = (2/3)(N_I + N_O)$ [37], where N_I , N_H , and N_O are the number of neurons in input, hidden, and output layers, respectively. In the present study, attempt to achieve good results by appropriate selection of hidden neurons using any rule of thumb was not found to be satisfactory. Therefore, the appropriate number of hidden neurons was established using trial and error procedure [30] and was found to be four. Generally, a large number of hidden neurons cause overfitting problem. Since, this study based on small number of hidden neurons (i.e., four); therefore, overfitting problem was not observed during training or testing stage. Since only pore-water pressure is sought from the model; therefore, output neuron is limited to one. Thus, an MLP network with architecture 8-4-1, representing 8 neurons in the input layer, 4 neurons in the hidden layer, and 1 neuron in the output layer, was used in this study.

3.4. Selection of Activation Function. There are a couple of activation functions which are used in neural network training like hyperbolic tangent, tangent sigmoid, linear, and so forth. All these functions have upper and lower bound limits depending upon their functions. Since the pore-water pressure data ranges between negative and positive values, to commensurate with the pore-water pressure ranges, the hyperbolic tangent sigmoid activation function (also known as hyperbolic tangent or tansig) whose upper and lower bound limits lie between -1 and $+1$ was used for neurons in the hidden layer and linear activation function (also known as purelin) was used for neurons in the output layer. The hyperbolic tangent sigmoid function can generate values between -1 and $+1$ and thus when used as an activation function at the hidden nodes, it causes all points on the solution surface to fall between these values [38]. Therefore, using hyperbolic tangent sigmoid type activation functions in the hidden layer and linear activation functions in the output layer provides advantage when it is necessary to extrapolate beyond the range of the training data [30].

TABLE 2: Performance statistics of the ANN model during training and testing (soil depth = 0.5 m).

Model stage	Data span	MAE	RMSE	CE	r^2
Training	12 Oct 1998 to 09 Jul 1999	0.38497	0.88358	0.99168	0.992
Testing	10 Jul 1999 to 20 Nov 1999	0.50933	1.46662	0.97619	0.977

3.5. Model Performance Evaluation. Performance statistics of hydrological prediction models are generally tested by adopting different efficiency criterion such as coefficient of determination, Nash-Sutcliffe efficiency, and relative efficiency criteria. These prediction models are also evaluated by using different error measures like mean absolute error (MAE), mean square error, sum of square error, root mean square error (RMSE), and so forth. In this study, the performance of the ANN model was evaluated by using three different standard statistical measures widely used in ANN modeling of hydrological events, namely, the root mean square error (see (19)), coefficient of efficiency (CE, see (20)), and mean absolute error (see (21)). Consider the following.

Root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (u_p - u_o)^2}. \quad (19)$$

Coefficient of efficiency (Nash and Sutcliffe, [39])

$$\text{CE} = 1 - \frac{\sum_{i=1}^n (u_o - u_p)^2}{\sum_{i=1}^n (u_o - \bar{u}_o)^2}. \quad (20)$$

Mean absolute error

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |u_p - u_o|, \quad (21)$$

where u_p and u_o are the predicted and observed values of pore-water pressures, respectively, $\bar{u}_o$ is the mean of observed pore-water pressures, and n is the number of pore-water pressure observations for which the error has been computed.

3.6. Stopping Criteria. For the application of the ANN model a program code was written using Matlab toolbox for ANN. Provision was made in the program code to stop the network training whenever the maximum number of given epochs is reached or the sum of mean square error $\leq$ goal (tolerance), whichever is satisfied first. A maximum number of 500 epochs and a goal of 0.001 were predefined in the program code.

4. Results and Discussion

Time series of soil pore-water pressure was trained using data from 0.5 m soil depth and the model was tested at various soil depths (i.e., 0.5, 1.1, 1.7, 2.3, and 2.9 m soil depths). Therefore, this section mainly consists of two parts. Initially, time series of pore-water pressure trained and tested with data of 0.5 m soil depth was demonstrated. Secondly, time series of pore-water pressure trained with data of 0.5 m soil depth but tested with multiple soil depths was illustrated.

FIGURE 5: Comparison between observed and trained time series of pore-water pressures obtained from training the ANN model.

4.1. Prediction of Pore-Water Pressure with Training and Testing at 0.5 m Soil Depth. The ANN model performance evaluated in terms of various performance measure statistics is shown in Table 2. The ANN model showed very small error during training (MAE = 0.3850, RMSE = 0.8836) and testing (MAE = 0.5093, RMSE = 1.4666) stages. The lower error measures suggest that the ANN model has predicted pore-water pressures very close to the observed pore-water pressures. The coefficient of efficiency CE and the coefficient of correlation r^2 are also very high for training (CE = 0.992, $r^2 = 0.992$) and testing (CE = 0.976, $r^2 = 0.977$).

A comparison between the time series of observed and predicted pore-water pressures obtained from the training and testing exercise of the ANN model is shown in Figures 5 and 6, respectively. Figures 5 and 6 show that, in both cases, during training and testing, the trends in the trained and predicted time series of pore-water pressures follow very closely the trends in time series of observed pore-water pressures. It is interesting to note that the model input variables were related to the wetting process of the soil only (rainfall) which causes the pore-water pressure to rise or change from negative to positive values (see Figure 6). The variables related to the drying process (no rainfall, evaporation, evapotranspiration, or temperature) of the soil which causes the pore-water pressure to decrease change from positive to negative values or change from negative to more negative values (see Figure 6) or any physical properties of the soil which also influence the response of pore-water pressure to climatic changes were not available to the model as input. However, the training process enabled the model to learn the trend in the data set and therefore the model, in addition to wet conditions, could also generalize the pore-water pressure

FIGURE 6: Comparison between observed and predicted time series of pore-water pressure obtained using the ANN model.

FIGURE 7: Comparison between observed and trained pore-water pressures obtained during training the ANN model.

responses during dry climatic conditions (no rainfall) from antecedent conditions (antecedent rainfall and antecedent pore-water pressures) which was provided to the network as additional model input parameters. Thus it appears from the results (Figures 5 and 6) that the ANN model developed in this study with its associated network architecture, training algorithm, and input data type and structure was appropriate and ideal for predicting the dynamics of pore-water pressure responses in a slope to climatic changes.

Considering each field observed pore-water pressure data is event based, a comparison between the field measured individual pore-water pressures and predicted pore-water pressures obtained during training and testing of the ANN model is made in Figures 7 and 8, respectively. The correlation between the observed and predicted pore-water pressures is evaluated using the coefficient of correlation, r^2 . The close to unity r^2 value ($r^2 = 0.992$ for training, Figure 7, and $r^2 = 0.977$ for testing, Figure 8) suggests nearly perfect agreement between the predicted and observed pore-water pressures

FIGURE 8: Comparison between observed and predicted pore-water pressures obtained using the ANN model.

FIGURE 9: Time and epochs required by the ANN model to reach desired accuracy.

with few outliers. The occasional few outlier values are not readily obvious in the time series plots of predicted and observed pore-water pressures shown in Figures 5 and 6. They are apparent in Figures 7 and 8. An examination of Figures 5 and 7 and Figures 6 and 8 suggests that these outlier values resulted from relatively under predictions during sudden and very rapid change (rise) in pore-water pressure responses to rainfall after a relatively dry period when the pore-water pressures are at highly negative values.

The performance of the model was also judged through an evaluation of the time and number of epochs required to produce outputs of desired accuracy (goal/tolerance). Figure 9 shows the time and number of epochs required by the ANN model to reach an accuracy of 0.001 within a predefined number of epochs of 500. Figure 9 shows that the network produced the predictions with 0.42 s computation time and with 7 epochs only. Furthermore, the MSE versus time and epoch plot (Figure 9) for the MLP-NN model also suggests that the network did not suffer from any local minima or convergence problems. The training algorithm converged to the solution sharply and rapidly. The initial network error

TABLE 3: Performance statistics of ANN at different soil depths during testing stage.

Soil depth (m)	Data mean (KPa)	MAE	RMSE	CE	R^2
0.5	-2.83	0.50	1.46	0.9766	0.9775
1.1	-4.44	0.51	1.12	0.9796	0.9800
1.7	-4.90	0.57	1.16	0.9838	0.9851
2.3	-1.26	0.60	1.20	0.9833	0.9848
2.9	4.36	1.71	2.24	0.9383	0.9799

which is of magnitude 0.8 is reduced to 0.001 within 0.42 s and with 7 epochs. These results further reinforce the fact that the network architecture adopted in this study for pore-water pressure prediction is not only robust but efficient as well.

The model however showed slightly poor performance during testing compared to the performance during training as is evident from slightly higher error and lower performance index measures during testing (Table 2, MAE = 0.510, RMSE = 1.467, CE = 0.976, and r^2 = 0.977 during testing against MAE = 0.385, RMSE = 0.884, CE = 0.992, and r^2 = 0.992 during training).

Similar trends in ANN model performance during training and testing were also observed in a number of other studies [40–42]. The slightly poor performance of the model during testing compared to the performance during training revealed from the performance statistics measures in this study could also be attributed to the higher variability inherent in the testing data set (CV rainfall = 572.24, CV pore-water pressure = -336.27 for testing data set as opposed to CV rainfall = 411.33, and CV pore-water pressure = -312.48 for training data set, Table 1).

The Nash and Sutcliffe [39] model coefficient of efficiency is commonly used to assess the predictive efficiency of hydrological models. The coefficient of efficiency (CE, (20)) can range from $-\infty$ to 1. Ideally, the closer the coefficient of efficiency is to 1, the more accurate the model prediction is. A coefficient of efficiency, CE = 1, indicates a perfect match between the predicted and observed data. CE = 0 indicates that the model predictions are as accurate as the mean of the observed data, whereas a coefficient of efficiency less than zero (CE < 0) indicates that the mean of the observed data is a better predictor than the model and occurs in situations when the residual variance (represented by the numerator in (20)) is larger than the data variance (represented by the denominator in (20)). The coefficient of efficiency for the ANN model evaluated in this study showed CE values which are very close to unity (training CE = 0.992, testing CE = 0.976) indicating excellent efficiency of the model and thereby suggesting that the learning algorithm chosen to predict pore-water pressure responses to climatic variations is appropriate. Furthermore, the higher values of model efficiency measure also suggest that the number and type of input variables derived from the cross-correlation and autocorrelation analyses among the dependent and independent variable were appropriate and therefore resulted in higher model efficiency measures.

4.2. Predictions with Training at 0.5 m Data but Testing at Multiple Depths. The successful prediction of soil pore-water pressure at the soil depth of 0.5 m encouraged testing

the ANN model at different soil depths. Thus, an attempt was made to train the ANN models with the LM training algorithm at the 0.5 m soil depth, but testing of the model was performed at different soil depths (0.5 m, 1.1 m, 1.7 m, 2.3 m, and 2.9 m). It was observed that the trained model in 0.5 m of soil is able to predict the soil pore-water pressure at different soil depths but with slightly different accuracy. The predicted pore-water pressures at all depths were very close to the observed pore-water pressure data. The summary of the performance statistics of the models at different soil depths during the testing stage of the model is shown in Table 3.

The performance statistics of the ANN model for prediction of pore-water pressure at different soil depths indicate that every testing stage successfully predicted the time series of the pore-water pressure at all the soil depths (Table 3). The highest error (MAE = 1.71, RMSE = 2.24) and lowest coefficient of efficiency (CE = 0.9383) were observed at the 2.9 m soil depth. The error and coefficient of efficiency at 0.5 m (MAE = 0.50, RMSE = 1.46, and CE = 0.9766), 1.1 m (MAE = 0.51, RMSE = 1.12, and CE = 0.98), 1.7 m (MAE = 0.57, RMSE = 1.16, and CE = 0.9838), and 2.3 m (MAE = 0.60, RMSE = 1.20, and CE = 0.9833) are similar and apparent discrepancies are insignificant. However, the coefficient of determination (R^2) for all soil depths is nearly equal (close to 0.98) which shows a good agreement between the observed and predicted pore-water pressures with the perfect line of agreement. Obviously, it is clear from the results that the ANN model trained using the data of the pore-water pressure at the 0.5 m soil depth is able to predict soil pore-water pressures at multiple soil depths with a slight difference in accuracy.

The results were also found to be comparable with previous studies performed for prediction of pore-water pressure using neural networks [5–7]. Mustafa et al. [5] performed training using data of Yishun slope and testing at two slopes Yishun and CSE. The coefficient of determination during testing stages produced at Yishun was R^2 = 0.97 and R^2 = 0.99. Mustafa et al. [7] while investigating the performance of various training algorithm for prediction of soil pore-water pressure also produced coefficient of determination close to one (0.94, 0.97) and 0.98. Therefore, all the results produced in this study particularly the comparison made with previous studies in terms of coefficient of determination values at testing stages showed the competence of the presented model.

The results of the ANN model training and testing demonstrate that the ANN model learned the nonlinear behavior of pore-water pressure in response to wet (rainfall) and dry (no rainfall) climatic conditions and produced

reasonably good prediction results. The model successfully predicted all the time series of pore-water pressure at multiple soil depths. Further, the structure/architecture and the training parameters used for ANN modeling are appropriate to capture the pattern of pore-water pressure data. The results also reveal that LM training algorithm is able to adopt the nonlinear pattern within a very short time, ensured fast convergence, and reached the goal (0.001) through self-adjustment of the learning rate.

5. Conclusions

A multilayer perceptron neural network model with Levenberg-Marquardt training algorithm has been developed to predict spatial and temporal behavior of time series of pore-water pressure responses to rainfall. A network configuration appropriate for mapping the nonlinear behavior of pore-water pressure responses (at 0.5 m soil depth) to climatic condition was identified to be 8-4-1. The same configuration is also able to estimate soil pore-water pressure data at multiple soil depths. The study indicated that the LM training algorithm is suitable for application to problems associated with predictions of nonlinear and complex behavior such as pore-water pressure variation due to rainfall. Results of the study also indicated that it is necessary to account for antecedent pore-water pressure and rainfall in order to achieve predictions of pore-water pressure with fairly reasonable accuracy. The study also indicated that the appropriate number of inputs could be achieved from autocorrelation and crosscorrelation analyses rather than using a trial and error procedure. The Levenberg-Marquardt algorithm used for training the network showed high efficiency, less prediction error, and very fast convergence. An additional major benefit derived from the training algorithm is that it adjusted the learning rate automatically.

However, the study showed that inclusion of antecedent pore-water pressure values in the input structure is of prime importance while using the rainfall data only. However, introducing some other readily available parameters such as temperature, relative humidity, evaporation, soil properties, and soil depth into the input structure may help to completely eliminate the antecedent pore-water pressure values.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors gratefully thank the Nanyang Technological University for providing time series data of rainfall and pore-water pressure to conduct this research. The authors are also thankful to the Ministry of Education, Malaysia, for providing financial support under FRGS cost center 0153AB-161.

References

- [1] H. Rahardjo, E. C. Leong, and R. B. Rezaur, "Effect of antecedent rainfall on pore-water pressure distribution characteristics in residual soil slopes under tropical rainfall," *Hydrological Processes*, vol. 22, no. 4, pp. 506–523, 2008.
- [2] R. B. Rezaur, H. Rahardjo, E. C. Leong, and T. T. Lee, "Hydrologic behavior of residual soil slopes in Singapore," *Journal of Hydrologic Engineering*, vol. 8, no. 3, pp. 133–144, 2003.
- [3] H. Rahardjo, T. H. Ong, R. B. Rezaur, and E. C. Leong, "Factors controlling instability of homogeneous soil slopes under rainfall," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 133, no. 12, pp. 1532–1543, 2007.
- [4] F. Melinda, H. Rahardjo, K. K. Han, and E. C. Leong, "Shear strength of compacted soil under infiltration condition," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 130, no. 8, pp. 807–817, 2004.
- [5] M. R. Mustafa, R. B. Rezaur, H. Rahardjo, and M. H. Isa, "Prediction of pore-water pressure using radial basis function neural network," *Engineering Geology*, vol. 135–136, pp. 40–47, 2012.
- [6] M. R. U. Mustafa, R. R. Bhuiyan, M. H. Isa, S. Saiedi, and H. Rahardjo, "Effect of antecedent conditions on prediction of pore-water pressure using artificial neural networks," *Modern Applied Science*, vol. 6, no. 2, pp. 6–15, 2012.
- [7] M. R. Mustafa Ph.D., R. B. Rezaur, S. Saiedi, H. Rahardjo, and M. H. Isa, "Evaluation of MLP-ANN training algorithms for modeling soil pore-water pressure responses to rainfall," *Journal of Hydrologic Engineering*, vol. 18, no. 1, pp. 50–57, 2013.
- [8] S. Supharatid, "Application of a neural network model in establishing a stage-discharge relationship for a tidal river," *Hydrological Processes*, vol. 17, no. 15, pp. 3085–3099, 2003.
- [9] D. A. K. Fernando and A. Y. Shamseldin, "Investigation of internal functioning of the radial-basis-function neural network river flow forecasting models," *Journal of Hydrologic Engineering*, vol. 14, no. 3, pp. 286–292, 2009.
- [10] A. Bonafe, G. Galeati, and M. Sforan, "Neural network for daily mean flow forecasting," in *Hydrology and Engineering Software—V*, K. L. Blain and W. R. Katsifarakis, Eds., vol. 1, pp. 131–138, Computational Mechanics Publication Southampton, London, UK, 1994.
- [11] S.-Y. Liong, W.-H. Lim, and G. N. Paudyal, "River stage forecasting in Bangladesh: neural network approach," *Journal of Computing in Civil Engineering*, vol. 14, no. 1, pp. 1–8, 2000.
- [12] M. Campolo, P. Andreussi, and A. Soldati, "River flood forecasting with a neural network model," *Water Resources Research*, vol. 35, no. 4, pp. 1191–1197, 1999.
- [13] D. A. K. Fernando and A. W. Jayawardena, "Runoff forecasting using RBF networks with OLS algorithm," *Journal of Hydrologic Engineering*, vol. 3, no. 3, pp. 203–209, 1998.
- [14] M. R. Mustafa, R. B. Rezaur, and M. H. Isa, "A comparison of artificial neural networks for prediction of suspended sediment discharge in river—a case study in Malaysia," in *Proceedings of the International Conference on Environmental Science and Engineering (ICESE '2011)*, pp. 372–376, September 2011.
- [15] M. R. Mustafa, R. B. Rezaur, S. Saiedi, and M. H. Isa, "River suspended sediment prediction using various multilayer perceptron neural network training algorithms—a case study in Malaysia," *Water Resources Management*, vol. 26, no. 7, pp. 1879–1897, 2012.
- [16] M. R. U. Mustafa, M. H. Isa, and R. R. Bhuiyan, "Prediction of river suspended sediment load using radial basis function

- neural network—a case study in Malaysia,” in *Proceedings of the National Postgraduate Conference (NPC '11)*, pp. 1–4, 2011.
- [17] K. W. Kang, J. H. Kim, J. C. Park, and K. J. Ham, “Evaluation of hydrological forecasting system based on neural network model,” in *Proceedings of the 25th Congress of IAHR*, pp. 257–264, Delft, The Netherlands, 1993.
 - [18] N. Karunanithi, W. J. Grenney, D. Whitley, and K. Bovee, “Neural networks for river flow prediction,” *Journal of Computing in Civil Engineering*, vol. 8, no. 2, pp. 201–220, 1994.
 - [19] A. H. Halff, H. M. Halff, and M. Azmoodeh, “Predicting runoff from rainfall using neural networks,” in *Proceedings of the Symposium on Engineering Hydrology*, pp. 760–765, ASCE, New York, NY, USA, July 1993.
 - [20] J. Smith and R. N. Eli, “Neural-network models of rainfall-runoff process,” *Journal of Water Resources Planning & Management*, vol. 121, no. 6, pp. 499–508, 1995.
 - [21] L. Crespo and E. Mora, “Drought estimation with neural networks,” in *Advances in Engineering Software*, pp. 167–170, Elsevier, Whitstable, UK, 1993.
 - [22] D. B. May and M. Sivakumar, “Prediction of urban stormwater quality using artificial neural networks,” *Environmental Modelling & Software*, vol. 24, no. 2, pp. 296–302, 2009.
 - [23] H. R. Maier and G. C. Dandy, “The use of artificial neural networks for the prediction of water quality parameters,” *Water Resources Research*, vol. 32, no. 4, pp. 1013–1022, 1996.
 - [24] M. Tawfik, A. Ibrahim, and H. Fahmy, “Hysteresis sensitive neural network for modeling rating curves,” *Journal of Computing in Civil Engineering*, vol. 11, no. 3, pp. 206–211, 1997.
 - [25] Q. Ju, Z. Yu, Z. Hao, G. Ou, J. Zhao, and D. Liu, “Division-based rainfall-runoff simulations with BP neural networks and Xinanjiang model,” *Neurocomputing*, vol. 72, no. 13–15, pp. 2873–2883, 2009.
 - [26] A. W. Jayawardena and D. A. K. Fernando, “Use of radial basis function type artificial neural networks for runoff simulation,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 13, no. 2, pp. 91–99, 1998.
 - [27] A. Y. Shamseldin, “Application of a neural network technique to rainfall-runoff modelling,” *Journal of Hydrology*, vol. 199, no. 3–4, pp. 272–294, 1997.
 - [28] K.-L. Hsu, H. V. Gupta, and S. Sorooshian, “Artificial neural network modeling of the rainfall-runoff process,” *Water Resources Research*, vol. 31, no. 10, pp. 2517–2530, 1995.
 - [29] G. Tayfur, “Artificial neural networks for sheet sediment transport/application des réseaux de neurones artificiels pour le transport sédimentaire en nappe,” *Hydrological Sciences Journal*, vol. 47, pp. 879–892, 2002.
 - [30] H. R. Maier and G. C. Dandy, “Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications,” *Environmental Modelling and Software*, vol. 15, no. 1, pp. 101–124, 2000.
 - [31] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
 - [32] S. M. A. Burney, T. A. Jilani, and C. Ardil, “Levenberg-Marquardt algorithm for Karachi Stock Exchange share rates forecasting,” *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 3, pp. 171–176, 2005.
 - [33] M. Caudill, “Neural networks primer, part I,” *AI Expert*, vol. 2, pp. 46–52, 1987.
 - [34] H. Rahardjo, “Rainfall-induced slope failures,” Research Report NSTB 17/6/16, Nanyang Technological University, Singapore, 2000.
 - [35] R. B. Rezaur, H. Rahardjo, and E. C. Leong, “Spatial and temporal variability of pore-water pressures in residual soil slopes in a tropical climate,” *Earth Surface Processes and Landforms*, vol. 27, no. 3, pp. 317–338, 2002.
 - [36] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer, Berlin, Germany, 1996.
 - [37] M. C. Demirel, A. Venancio, and E. Kahya, “Flow forecast by SWAT model and ANN in Pracana basin, Portugal,” *Advances in Engineering Software*, vol. 40, no. 7, pp. 467–473, 2009.
 - [38] I. Flood and N. Kartam, “Neural networks in civil engineering. I: principles and understanding,” *Journal of Computing in Civil Engineering*, vol. 8, no. 2, pp. 131–148, 1994.
 - [39] J. E. Nash and J. V. Sutcliffe, “River flow forecasting through conceptual models part I—a discussion of principles,” *Journal of Hydrology*, vol. 10, no. 3, pp. 282–290, 1970.
 - [40] B. Unal, M. Mamak, G. Seckin, and M. Cobaner, “Comparison of an ANN approach with 1-D and 2-D methods for estimating discharge capacity of straight compound channels,” *Advances in Engineering Software*, vol. 41, no. 2, pp. 120–129, 2010.
 - [41] A. M. Melesse and X. Wang, “Multitemporal scale hydrograph prediction using artificial neural networks,” *Journal of the American Water Resources Association*, vol. 42, no. 6, pp. 1647–1657, 2006.
 - [42] D. F. Lekkas, C. Onof, M. J. Lee, and E. A. Baltas, “Application of artificial neural networks for flood forecasting,” *Global Nest*, vol. 6, pp. 205–211, 2004.