

Research Article

Development of Artificial Neural-Network-Based Models for the Simulation of Spring Discharge

M. Mohan Raju,¹ R. K. Srivastava,¹ Dinesh C. S. Bisht,² H. C. Sharma,¹ and Anil Kumar¹

¹Department of Irrigation and Drainage Engineering, College of Technology, G. B. Pant University of Agriculture and Technology, Uttarakhand, Pantnagar-263 145, India

²Department of Applied Sciences and Humanities, ITM University, Gurgaon 122017, India

Correspondence should be addressed to Dinesh C. S. Bisht, dcsbisht@gmail.com

Received 3 January 2011; Accepted 2 May 2011

Academic Editor: Martin Pelikan

Copyright © 2011 M. Mohan Raju 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.

The present study demonstrates the application of artificial neural networks (ANNs) in predicting the weekly spring discharge. The study was based on the weekly spring discharge from a spring located near Ranichauri in Tehri Garhwal district of Uttarakhand, India. Five models were developed for predicting the spring discharge based on a weekly interval using rainfall, evaporation, temperature with a specified lag time. All models were developed both with one and two hidden layers. Each model was developed with many trials by selecting different network architectures and different number of hidden neurons; finally a best predicting model presented against each developed model. The models were trained with three different algorithms, that is, quick-propagation algorithm, batch backpropagation algorithm, and Levenberg-Marquardt algorithm using weekly data from 1999 to 2005. A best model for the simulation was selected from the three presented algorithms using the statistical criteria such as correlation coefficient (R), determination coefficient, or Nash Sutcliffe's efficiency (DC). Finally, optimized number of neurons were considered for the best model. Training and testing results revealed that the models were predicting the weekly spring discharge satisfactorily. Based on these criteria, ANN-based model results in better agreement for the computation of spring discharge. LMR models were also developed in the study, and they also gave good results, but, when compared with the ANN methodology, ANN resulted in better optimized values.

1. Introduction

The process of discharge simulation from a spring is a very complex, highly nonlinear phenomenon having temporal and spatial variability. The weekly spring discharge modeling has a vital role in better management of the water resources management. Many models such as black box, conceptual, and physically-based models have been developed especially for rainfall, runoff, and sediment process. On the other hand, very few models are available for accurate estimation of discharge of spring, and, in many situations, simple tools such as linear theoretical models or black box models have been used with advantage. However, these models fail to represent the nonlinear process such as rainfall, runoff, and sediment yield [3]. The successful application of artificial intelligence techniques such as ANNs has added a new

dimension to model such complex systems in recent years to solve various problems in hydrologic/hydraulic engineering and water resources engineering [1, 2]. A new dimensional development came in existence in the recent years for better prediction and model development, namely, hybrid modeling, such as ANFIS, ANN-Fuzzy logic, and ANN-GA [3]. Unlike mathematical models that require precise knowledge of all contributing variable, a trained artificial neural network can estimate process behavior. It is proven fact that neural nets have a strong generalization ability, which means that, once they have been properly trained, they are able to provide accurate results even for cases they have never seen before [4, 5].

The artificial neural network, a soft computing tool, basically a black-box model and has its own limitations [6]. The main advantage of the ANN approach over traditional

methods is that it does not require the complex nature of the underlying process under consideration to be explicitly described in a mathematical form [7]. Other advantages of ANN over conventional models are discussed in [8]. Mathematically, an ANN can be treated as a universal approximator having an ability to learn from examples without explicit physics [1, 2, 9]. Such a model is easy to develop and yields satisfactory results when applied to complex systems which are poorly defined or implicitly understood. These are more tolerant to variable, incomplete, or ambiguous input data. The applications of ANNs in water engineering include the modeling rainfall-runoff-sediment process, forecasting of rainfall-runoff-sediment yield, river flow forecasting, reservoir inflow modeling and operation, setting up river-stage-discharge relationship, water-supply-system optimization, evapotranspiration process, draught forecasting, groundwater quality prediction, and groundwater remediation [7, 10–23].

2. Artificial Neural Networks

Artificial neural networks are highly simplified mathematical models of biological neural networks having the ability to learn and provide meaningful solutions to the problems with high-level complexity and nonlinearity. The ANN approach is faster compared to its conventional techniques, robust in noisy environments, and can solve a wide range of problems. Due to these advantages, ANNs have been used in numerous real-time applications. The most commonly used neural networks in hydrology being three layered and four layered having input layer, where the input is fed to the network, hidden layer(s) where the data is processed, and output layer where the output will be presented Figure 1.

The processing elements in each layer are called neurons or nodes. The information flow and processing in the network is from the input layer to the hidden layer and from the hidden layer to the output layer. The number of neurons and hidden layers in the network is problem dependent and is decided by the trial and error method. A synaptic weight is assigned to each link to represent the relative connection strength of two nodes at both ends in predicting the input-output relationship. The output, y_j of any node j , is given as

$$y_j = f \left(\sum_{i=1}^m W_i X_i + b_j \right), \quad (1)$$

where X_i is the input received at node j , W_i is the input connection pathway weight, m is the total number inputs to node j , and b_j is the node threshold. Function f is called an activation function which determines the response of a node to the total input signal that is received. The commonly used activation function is sigmoid transfer function and is given by

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2)$$

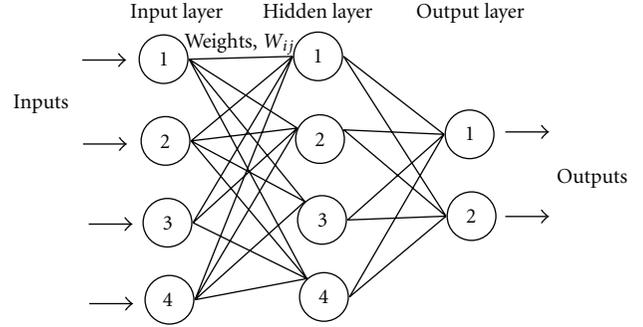


FIGURE 1: Three-layered feed-forward artificial neural network configuration.

Sigmoid function is continuous and differentiable everywhere, and a nonlinear process can be mapped with it [1]. Backpropagation algorithm is mostly used in training of the feed-forward neural networks [2]. In this algorithm, each input pattern of the training dataset is passed through network from the input layer to the output layer. The network generated output is compared with the target output, and the error is computed as

$$E = \sum_P \sum_p (y_i - t_i)^2, \quad (3)$$

where t_i is the component of the desired output/target T , y_i is the ANN generated output, p is the number of output nodes, and P is the number training patterns. This error will be propagated backward through each node, and correspondingly the connection weights will be updated.

3. Study Area and Data

Study area is located near the College of Forestry and Hill Agriculture, Hill Campus Ranichauri of the G. B. Pant University of Agriculture and Technology, Pantnagar, in the *Tehri Garhwal* district of Uttarakhand state in India, as shown in Figure 2. The campus was established at *Ranichauri* in the *Tehri Garhwal* district of Uttarakhand state, with an MoU between the Government of India and the Government of Uttar Pradesh in 1972 (formally inaugurated in 1976). The main objective in the establishment of the college lies in generating and dissemination of cutting-edge technologies with state-of-the-art inputs in the areas of hill agriculture, horticulture, forestry, and allied areas specifically for hill-agroclimatic conditions. It also carries out programmes of human resource development, multidisciplinary researches, and technology transfer suitable for hilly areas. The hill campus has altitudinal expansion from 1600 to 2200 meters above sea level, overseeing the majestic snow-capped peaks of Himalayas. The state of Uttarakhand state in India comprises of two subregions, *Kumaon* and *Garhwal*, having 13 districts. The entire region is mountainous, except a few valleys and *tarai area* (area just adjacent to the foot hills of Himalayas), in

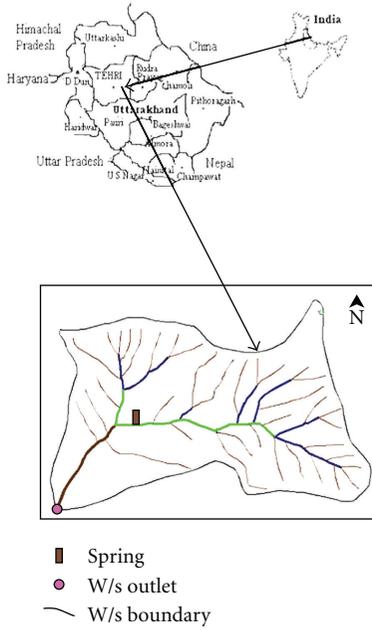


FIGURE 2: Location of the study area.

the *Nainital*, *Dehradun*, and *Hardwar* districts. On account of topography, the region has a peculiar geographical feature and is blessed by very good land and water resources. Though the whole economy of the region is based on agriculture, the total cultivable land is only 14 percent of the total geographical area of the region, with a good potential in conservation and utilization of natural resources. Especially the hill region is very rich in natural resources particularly in forests, minerals, and surface water.

The present watershed of the study area drains into the Henval river in the *Tehri Garhwal* district of the Uttarakhand state, covering an area of 871 ha (8.71 km²). It is located between 78°22'28'' and 78°24'57''E longitude and 30°17'19'' and 30°18'52''N latitude. The elevation varies from 960 to 2000 m above mean sea level (MSL). The study spring is located at 78°24'34''E and 30°18'47''N at the elevation of 1844 m above MSL in dense forest area. For the present study, the weekly data of spring discharge, rainfall, evaporation, temperature were collected for seven years from 1999 to 2005. Study area was tracked along with GPS, and the location (latitude and longitude) of the Hill campus spring was noted with the help of GPS. The same locations were marked on the map with GIS environment, which is shown in Figure 2. Spring layer was overlaid with DEM to get the height of natural springs above MSL. This spring had lesser catchment area as it was nearer to the watershed divide. This spring is located at 78°24'34''E and 30°18'47''N at the elevation of 1844 m above MSL in dense forest area.

4. Model Development and Methodology

The output from the model is spring discharge at the time step, Q_t . The spring discharge was mapped by considering

in addition the current value of discharge and the discharge at previous time steps. Therefore, in addition to Q_t , other variables such as Q_{t-1} , Q_{t-2} , R_{t-1} , R_{t-2} , E_{t-1} , E_{t-2} , and T_{t-1} , T_{t-2} were also considered for training of ANN in the present study are given in Table 1. In the development of one hidden layer architecture, 239 weeks for training, 55 weeks for testing, and 55 weeks for validation were taken, and, in two hidden layer architecture development, 234 weeks for training, 54 weeks for testing, and 54 weeks for validation were taken for the best possible model selection.

In the present study feed-forward Quick-propagation, Batch backpropagation, and Levenberg-Marquardt's ANN models have been used for the simulation of the spring discharge. The input-output datasets were first normalized considering the maximum value of the series, and the reducing the individual variables, in the range of 0 to 1 to avoid the saturation effect, may be possible by using sigmoid activation function.

The sigmoid function was the activation function used in the present study, and constant quick propagation coefficient 1.75 and learning rate 0.8 were selected by different hit and trial methods in the quick propagation algorithm for possible better optimization. The weights were updated after presenting each pattern from the learning dataset, rather than once per iteration. In the Batch backpropagation algorithm, a constant learning rate (η), 0.15 was used in the model development, and momentum rate (α) was used in the range of 0.6 to 0.9; moreover, local minima avoidance concept was used in the Levenberg-Marquardt algorithm.

The number of input nodes in the input layer was taken equal to the number of input variables. Since no guideline is yet available on the number of hidden nodes of the hidden layer (Vemuri, 1992) [9], these were initially tried from equal number of input nodes to double to that of input nodes (Hipel et al., 1994) [5]. However, corresponding to one output, only one node was taken in the output layer.

5. Development of Linear Multiple Regression Models

In developing Linear Multiple Regression (LMR) models, the spring discharge at time t , Q_t , is used as a criterion variable, and rain fall, evaporation, temperature, and spring discharge in the past are used as predictor variables. The LMR models can be represented as follows:

$$Q_t = \beta_0 + \beta_1 Q_{t-1} + \beta_2 Q_{t-2} + \beta_3 R_t + \beta_4 R_{t-1} + \beta_5 R_{t-2} + \beta_6 E_t + \beta_7 E_{t-1} + \beta_8 E_{t-2} + \beta_9 T_t + \beta_{10} T_{t-1} + \beta_{11} T_{t-2}, \quad (4)$$

where β_i s represent the regression coefficients to be determined, R_t s represent the rainfall, E_t s represent the evaporation, T_t s represent the temperature, Q_t s represent the spring discharge, and t is the index representing time.

TABLE 1: Selection of the best model from the batch backpropagation algorithm.

Model inputs	No. of inputs	Algorithm	Architecture	No. of hidden layer	Training		Testing		Validation	
					R	DC	R	DC	R	DC
R_t, E_t, T_t	3		[3-4-1]	One	0.671	-0.235	0.600	-0.133	0.594	-0.883
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7		[7-2-1]	One	0.979	0.956	0.979	0.941	0.986	0.961
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11		[11-28-1]	One	0.959	0.907	0.921	0.809	0.953	0.893
$Q_{t-1}, Q_{t-2}, Q_{t-3}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, T_t, T_{t-1}, T_{t-2}, T_{t-3}$	15		[15-21-1]	One	0.943	0.851	0.945	0.865	0.967	0.928
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, E_{t-4}, T_t, T_{t-1}, T_{t-2}, T_{t-3}, T_{t-4}$	19	Batch backpropagation	[19-10-1]	One	0.985	0.969	0.958	0.901	0.989	0.976
R_t, E_t, T_t	3		[3-2-2-1]	Two	0.623	-0.575	0.623	-1.378	0.554	-1.322
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7		[7-6-5-1]	Two	0.982	0.960	0.970	0.935	0.975	0.944
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11		[11-4-3-1]	Two	0.988	0.975	0.981	0.960	0.967	0.907
$Q_{t-1}, Q_{t-2}, Q_{t-3}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, T_t, T_{t-1}, T_{t-2}, T_{t-3}$	15		[15-7-3-1]	Two	0.980	0.958	0.960	0.901	0.985	0.963
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, E_{t-4}, T_t, T_{t-1}, T_{t-2}, T_{t-3}, T_{t-4}$	19		[19-9-4-1]	Two	0.986	0.971	0.957	0.907	0.983	0.966

The developed regression models' description is as follows.

Model: LMR-1:

$$Q_t = 373.56 + 0.91Q_{t-1} + 107.46R_t + 248.80R_{t-1} - 541.81E_t - 3102.35E_{t-1} + 1163.65T_t - 558.72T_{t-1}. \quad (5)$$

Model: LMR-2:

$$Q_t = 3870.08 + 1.14Q_{t-1} - 0.24Q_{t-2} + 88.84R_t + 205.11R_{t-1} + 35.56R_{t-2} - 1097.86E_t - 563.58E_{t-1} - 2935.29E_{t-2} + 992.81T_t - 289.59T_{t-1} - 2.05T_{t-2}. \quad (6)$$

The graphical representation along with the corresponding scattered plots of developed LMR models are shown in Figures 6 and 7.

6. Results and Discussion

It can be seen from Table 1 that correlation coefficient (R) is very high, that is, more than 93%, for all the ANN models during the training, testing, and validation phases, except ANN1, ANN6. There is no significant decrease in R values during validation when compared with the training phase. The best performed models in R statistic were ANN2 among one hidden layer models and ANN8 among two hidden layer models. The R values for ANN2 are 0.990, 0.970, and 0.983 during training, testing, and validation, respectively. The increase in R values during validation indicates the good generalization capability of the ANN model. The root

TABLE 2: Selection of the best model from the quick-propagation algorithm.

Model inputs	No. of inputs	Algorithm	Architecture	No. of hidden layer	Training		Testing		Validation	
					R	DC	R	DC	R	DC
R_t, E_t, T_t	3		[3-2-1]	One	0.710	0.039	0.582	-0.179	0.586	-0.626
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7		[7-7-1]	One	0.990	0.979	0.970	0.920	0.983	0.964
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11		[11-14-1]	One	0.984	0.967	0.981	0.963	0.965	0.922
$Q_{t-1}, Q_{t-2}, Q_{t-3}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, T_t, T_{t-1}, T_{t-2}, T_{t-3}$	15		[15-15-1]	One	0.996	0.993	0.977	0.952	0.974	0.944
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, E_{t-4}, T_t, T_{t-1}, T_{t-2}, T_{t-3}, T_{t-4}$	19	Quick propagation	[19-30-1]	One	0.997	0.995	0.931	0.854	0.970	0.936
R_t, E_t, T_t	3		[3-2-2-1]	Two	0.626	-0.499	0.611	-0.187	0.646	-0.233
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7		[7-4-5-1]	Two	0.988	0.975	0.975	0.941	0.985	0.970
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11		[11-6-5-1]	Two	0.994	0.998	0.983	0.962	0.990	0.960
$Q_{t-1}, Q_{t-2}, Q_{t-3}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, T_t, T_{t-1}, T_{t-2}, T_{t-3}$	15		[15-11-14-1]	Two	0.998	0.996	0.979	0.948	0.979	0.950
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, E_{t-4}, T_t, T_{t-1}, T_{t-2}, T_{t-3}, T_{t-4}$	19		[19-10-13-1]	Two	0.998	0.960	0.990	0.978	0.978	0.965

mean square error for ANN2 is 7764.502 lit/week and for ANN8 5472.556 lit/week, which are significantly low and indicate the good generalization capability of the model. In the determination coefficient statistic, all the ANN models except ANN1 and ANN6 perform well. The DC values of the models are fairly high, that is, more than 92%, except during the training, testing, and validation period. In DC statistic, also the models ANN2 and ANN8 models perform the best. The DC values for ANN2 are 0.979, 0.920, and 0.964, and for ANN8 are 0.998, 0.962, and 0.960 during training, testing, and validation. There is not much decrease in DC values during validation; this also indicates the good-generalization capability of the model.

Hence, ANN2 and ANN8 are the best performance models for the study spring. Finally, the model ANN8 was selected on the basis of the overall performance for the spring discharge simulation for the current spring in

the quick-propagation algorithm. It can also be seen from Tables 2 and 3 that the correlation coefficient (R) is very high, that is, around 95%, for most of the ANN models during the training, testing, and validation phases, except ANN1, ANN6 in both the batch backpropagation algorithm and Levenberg-Marquardt algorithm. There is no significant decrease in R values during validation when compared with the training phase. The best performed models in R statistic were ANN5 among one hidden layer models and ANN10 among two hidden layer models in Batch backpropagation algorithm, and, in Levenberg-Marquardt algorithm, ANN3 among one hidden layer and ANN10 among two hidden layers were the best performed models. The R values of Batch backpropagation algorithm ANN models for ANN5 are 0.985, 0.958, and 0.989, for ANN10 are 0.986, 0.957, and 0.983 during training, testing, and validation; respectively, the R values of Levenberg-Marquardt algorithm ANN

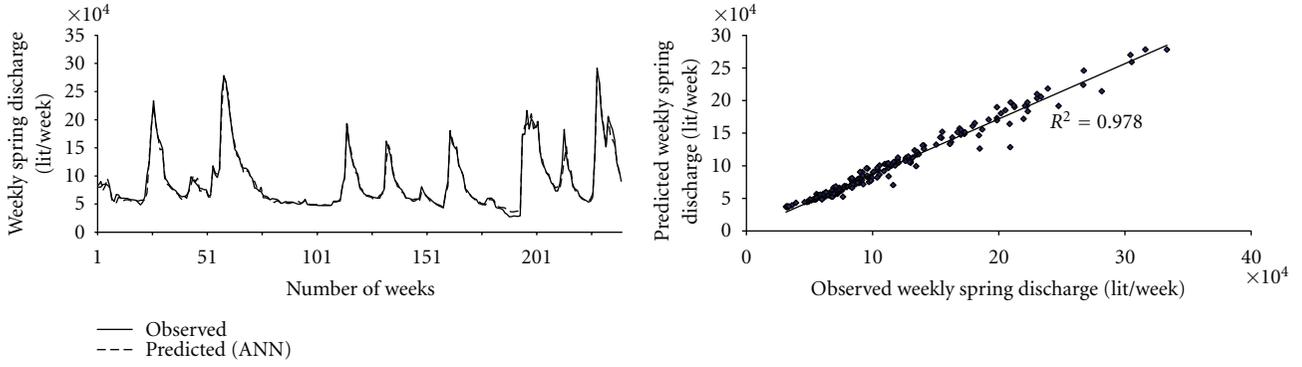
TABLE 3: Selection of the best model from the Levenberg-Marquardt algorithm.

Model inputs	No. of inputs	Algorithm	Architecture	No. of hidden layer	Training		Testing		Validation	
					R	DC	R	DC	R	DC
R_t, E_t, T_t	3		[3-8-1]	One	0.606	-0.957	0.491	-1.892	0.623	-1.449
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7		[7-3-1]	One	0.995	0.990	0.966	0.932	0.961	0.909
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11		[11-10-1]	One	0.990	0.979	0.956	0.907	0.989	0.978
$Q_{t-1}, Q_{t-2}, Q_{t-3}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, T_t, T_{t-1}, T_{t-2}, T_{t-3}$	15		[15-10-1]	One	0.983	0.960	0.950	0.894	0.971	0.940
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, E_{t-4}, T_t, T_{t-1}, T_{t-2}, T_{t-3}, T_{t-4}$	19	Levenberg-Marquardt	[19-13-1]	One	0.982	0.956	0.989	0.969	0.975	0.928
R_t, E_t, T_t	3		[3-2-2-1]	Two	0.528	-210441.73	0.521	-168494.734	0.392	-214100.83
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7		[7-3-2-1]	Two	0.985	0.967	0.970	0.935	0.168	0.934
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11		[11-3-2-1]	Two	0.986	0.972	0.983	0.961	0.957	0.914
$Q_{t-1}, Q_{t-2}, Q_{t-3}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, T_t, T_{t-1}, T_{t-2}, T_{t-3}$	15		[15-3-3-1]	Two	0.982	0.941	0.93	0.89	0.986	0.91
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, R_t, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-4}, E_t, E_{t-1}, E_{t-2}, E_{t-3}, E_{t-4}, T_t, T_{t-1}, T_{t-2}, T_{t-3}, T_{t-4}$	19		[19-4-3-1]	Two	0.998	0.997	0.962	0.925	0.900	0.937

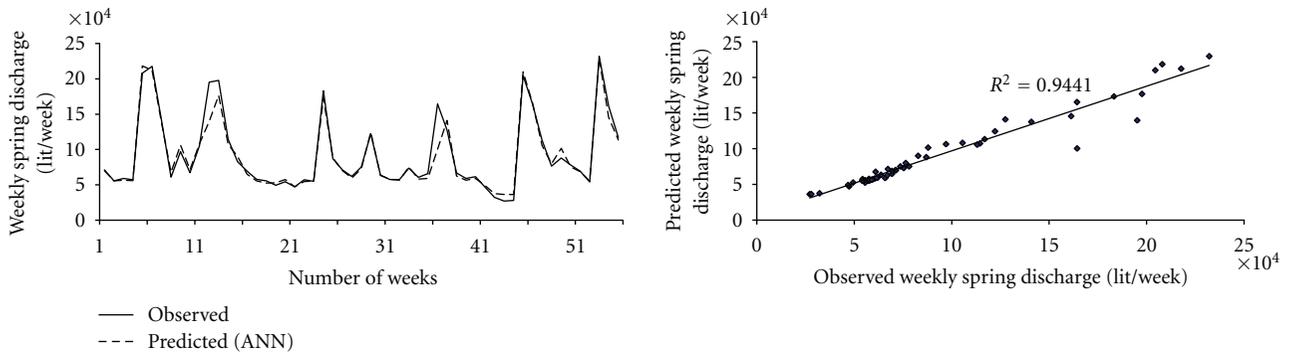
models, for ANN3 are 0.990, 0.956, and 0.989, for ANN10 are 0.998, 0.962, and 0.900 during training, testing and validation, respectively. The increase in R values during validation indicates the good generalization capability of the ANN model. In the determination coefficient statistic, all the ANN models except ANN1 and ANN6 perform well for the both batch backpropagation and Levenberg-Marquardt's algorithms. The DC values of the models are fairly high, that is, more than 90%, in all most all the models during the training, testing, and validation period except ANN1 and ANN6. In DC statistic, also the models ANN5 and ANN10 in batch backpropagation algorithm, ANN3 and ANN10 in Levenberg-Marquardt algorithm perform the best. The DC values in batch backpropagation algorithm for ANN5 are 0.969, 0.901, and 0.976 and for ANN10 are 0.971,

0.907, and 0.966 during training, testing, and validation, in Levenberg-Marquardt's algorithm for ANN3 are 0.979, 0.907, and 0.978 and for ANN10, 0.997, 0.925, and 0.937 during training, testing, and validation. There is not much decrease in DC values during validation; this also indicates the good generalization capability of the model. Hence, ANN5 and ANN10 are the best performance models in batch backpropagation algorithm, and ANN3, ANN10 are the best performance models in Levenberg-Marquardt's algorithm out all the models presented for the study spring. Finally, the model ANN5 (batch backpropagation algorithm) and ANN3 (Levenberg-Marquardt algorithm) were selected on the basis of the overall performance as best models Figures 4 and 5.

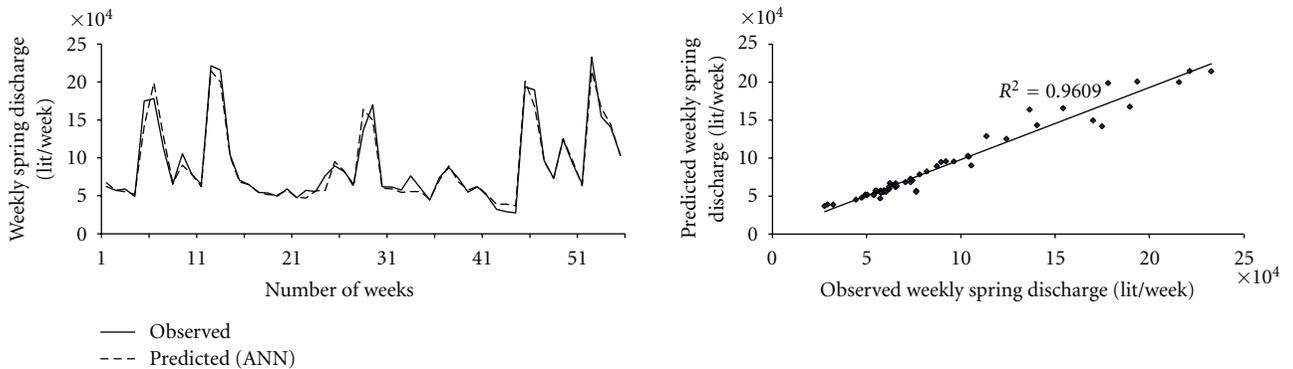
The aim of the current study is to present the representative algorithm, with a particular ANN model for the



(a) Training



(b) Testing



(c) Validation

FIGURE 3: Observed and predicted (ANN2) weekly spring discharge—quick-propagation algorithm.

spring discharge simulation for the current spring; in this line of context, all the presented models from the three algorithm are representative for the current spring discharge simulation, but the performance of the model ANN8 with quick-propagation algorithm is quiet good amongst all the best presented models. Hence, finally, the ANN8 model with

quick-propagation algorithm is selected for the simulation of spring discharge in the study.

The comparative plots presented for the observed and estimated spring discharges and their corresponding scatter plots for the best representative model for the study location during training, testing, and validation in Figures 3, 4, and 5.

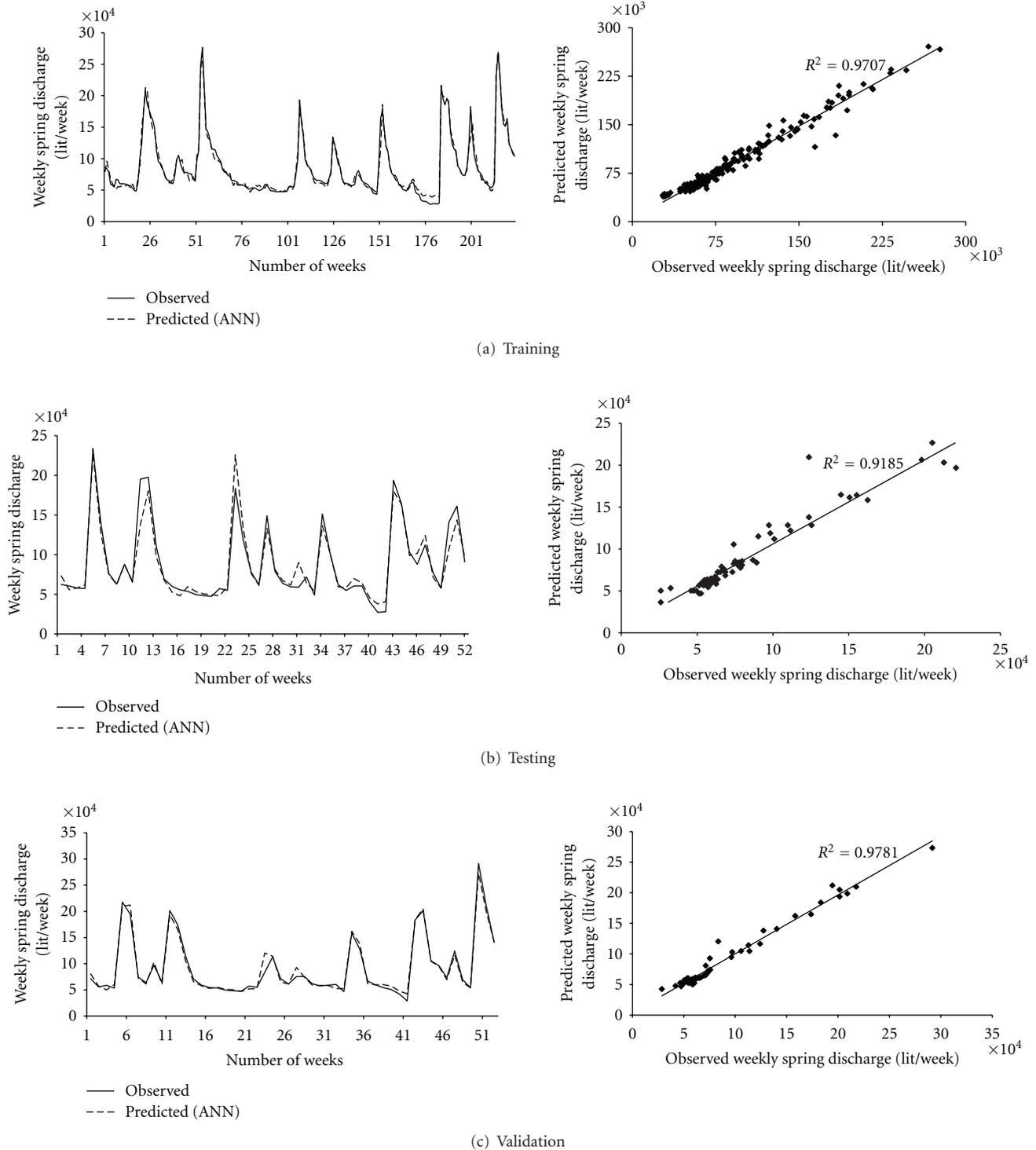


FIGURE 4: Observed and predicted (ANN5) weekly spring discharge—batch backpropagation algorithm.

It is observed from the presented figures that there is a very little mismatch between the observed and estimated spring discharges during all the three phases. Scatter plots also show a good correlation and regression values and no much shift from the ideal line.

7. Comparison of the Best ANN Models with LMR Models

Table 4 gives comparison of performance of the best five ANN models with Linear Multiple Regression (LMR) mod-

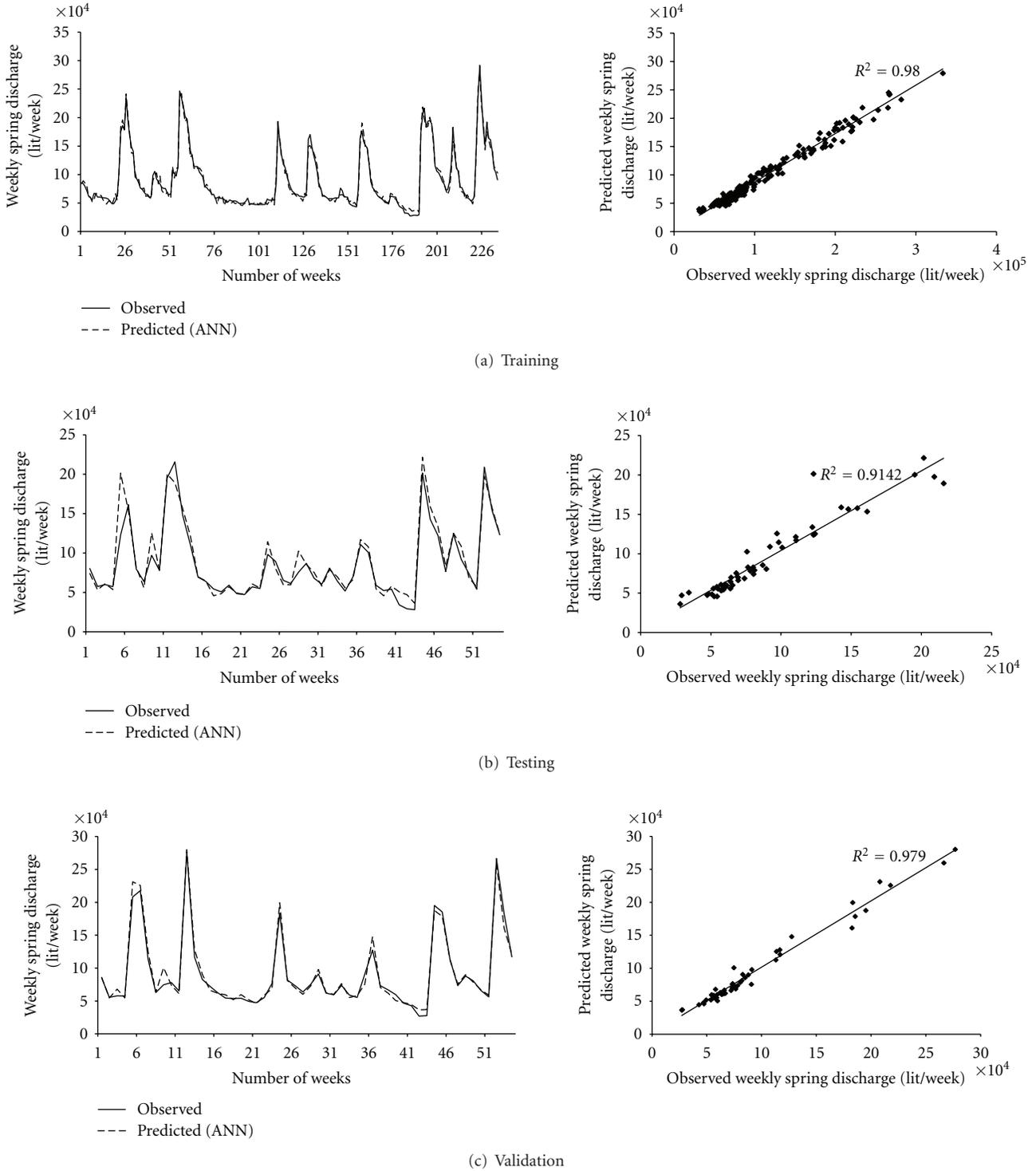


FIGURE 5: Observed and predicted (ANN3) weekly spring discharge—Levenberg Marquard’s algorithm.

TABLE 4: Comparison of the best ANN and LMR models.

Model inputs	No. of inputs	ANN quick propagation		LMR model	
		R	DC	R	DC
$Q_{t-1}, R_t, R_{t-1}, E_t, E_{t-1}, T_t, T_{t-1}$	7	0.983	0.964	0.970	0.941
$Q_{t-1}, Q_{t-2}, R_t, R_{t-1}, R_{t-2}, E_t, E_{t-1}, E_{t-2}, T_t, T_{t-1}, T_{t-2}$	11	0.990	0.960	0.976	0.948

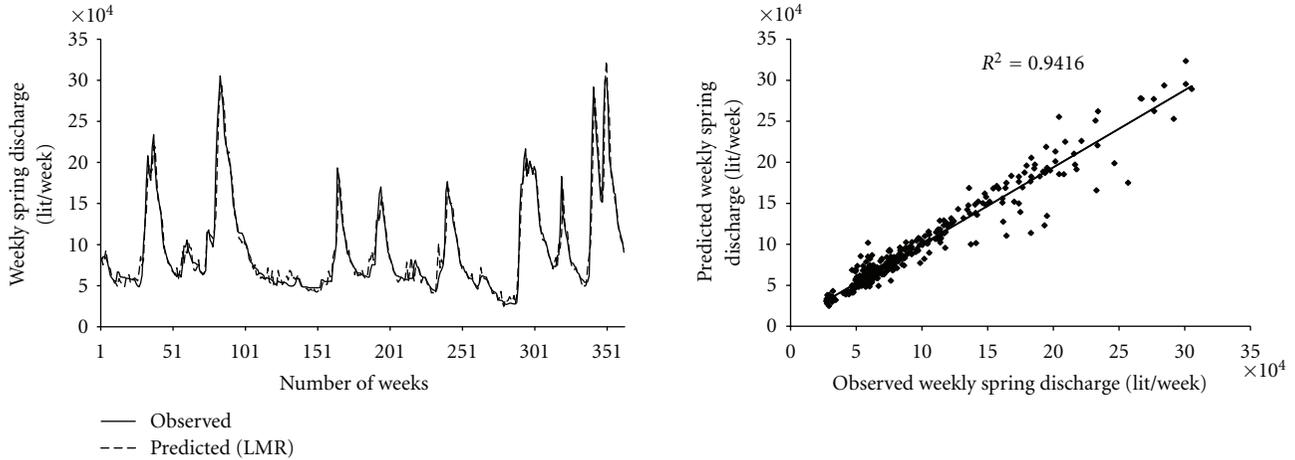


FIGURE 6: Observed and predicted (LMR-1) weekly spring discharge.

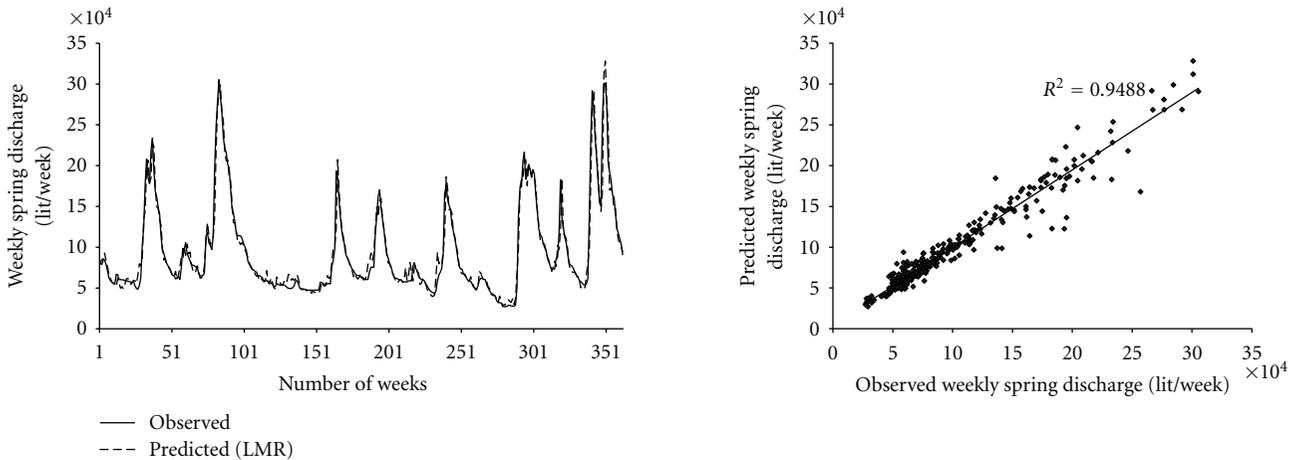


FIGURE 7: Observed and predicted (LMR-2) weekly spring discharge.

els. The performance indicators throughout the study are correlation coefficient (R), coefficient of efficiency (R^2). In the Tables 1, 2, and 3 the Correlation Coefficient of ANN models varies from 90.0% to as high as 99.8% for the mixed data (training, testing, and validation), whereas the Correlation Coefficient for LMR models varies from 97.0% to as high as 97.6%. Coefficient of Efficiency for ANN models varies from 90.1% to as high as 99.8%, while the same for the LMR models varies from 94.1% to as high as 94.8%. By the comparison made in Table 4, it is clear that the performance of ANN models are better than that of LMR models for spring discharge.

References

- [1] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, "Artificial neural networks in hydrology. I: pre-liminary concepts," *Journal of Hydrologic Engineering*, ASCE, vol. 5, no. 2, pp. 115–123, 2000.
- [2] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, "Artificial neural networks in hydrology. II: hydrologic applications," *Journal of Hydrologic Engineering*, ASCE, vol. 5, no. 2, pp. 124–137, 2000.
- [3] D. A. K. Fernando and A. W. Jayawardena, "Runoff forecasting using RBF networks with OLS algorithm," *Journal of Hydrologic Engineering*, ASCE, vol. 3, no. 3, pp. 203–209, 1998.
- [4] R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley, New York, NY, USA, 1991.
- [5] K. W. Hipel, A. Ian, U. S. Pann, and V. P. Singh, *Stochastic and Statistical Methods in Hydrology and Environmental Engineering*, The Series Analysis in Hydrology and Environmental Engineering, Kluwer Academic, Dodrecht, The Netherlands, 1994.
- [6] Kuo-Lin 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.
- [7] A. Sarkar, R. Kumar, R. D. Singh, G. Thakur, and S. K. Jain, "Sediment-discharge modeling in a river using artificial neural

- networks,” in *Proceedings of the International Conference ICON—HERP*, IIT Roorkee, India, October 2004.
- [8] M. N. French, W. F. Krajewski, and R. R. Cuykendall, “Rainfall forecasting in space and time using a neural network,” *Journal of Hydrology*, vol. 137, pp. 1–31, 1992.
- [9] V. R. Vemuri, *Artificial Neural Networks: Concepts and Control Application*, IEEE Computer Society Press, Los Alamitos, CA, USA, 1992.
- [10] N. Karunanithi, W. J. Grenney, D. Whitley, and K. Bovee, “Neural networks for river flow prediction,” *Journal of Computing in Civil Engineering*, ASCE, vol. 8, no. 2, pp. 201–220, 1994.
- [11] T. W. Kim and J. B. Valdes, “Nonlinear model for drought forecasting based on a conjunction of wavelet transforms and neural networks,” *Journal of Hydrologic Engineering*, ASCE, vol. 8, no. 6, pp. 319–328, 2003.
- [12] O. Kisi, “Development of streamflow-suspended sediment rating curve using a range dependent neural network,” *International Journal of Science & Technology*, vol. 2, no. 1, pp. 49–61, 2007.
- [13] N. S. Raghuvanshi, R. Singh, and L. S. Reddy, “Runoff and sediment yield modeling using artificial neural networks: Upper Siwane River, India,” *Journal of Hydrologic Engineering*, ASCE, vol. 11, no. 1, pp. 71–79, 2006.
- [14] R. K. Rai and B. S. Mathur, “Event-based sediment yield modeling using artificial neural network,” *Water Resources Management*, vol. 22, no. 4, pp. 423–441, 2008.
- [15] M. M. Raju, *Runoff-sediment modeling using artificial neural networks*, M. Tech. thesis, The G. B. Pant University of Agriculture & Technology, Uttarakhand, India.
- [16] M. P. Rajurkar, U. C. Kothiyari, and U. C. Chaube, “Artificial neural networks for daily rainfall-runoff modelling,” *Hydrological Sciences Journal*, vol. 47, no. 6, pp. 865–877, 2002.
- [17] M. P. Rajurkar, U. C. Kothiyari, and U. C. Chaube, “Modeling of the daily rainfall-runoff relationship with artificial neural network,” *Journal of Hydrology*, vol. 285, pp. 96–113, 2004.
- [18] S. Sajikumar and B. S. Thandaveswara, “A non-linear rainfall-runoff model using an artificial neural network,” *Journal of Hydrology*, vol. 216, no. 1-2, pp. 32–55, 1999.
- [19] A. Sarkar, “Real time flood forecasting using artificial neural networks in an Indian river,” in *Proceedings of the International Conference on Monitoring, Prediction and Mitigation of Water Related Disasters*, Kyoto, Japan, January 2005.
- [20] J. Smith and R. N. Eli, “Neural-network models of rainfall-runoff process,” *Journal of Water Resources Planning & Management*, ASCE, vol. 121, no. 6, pp. 499–508, 1995.
- [21] K. Thirumalaiah and M. C. Deo, “Hydrological forecasting using neural networks,” *Journal of Hydrologic Engineering*, ASCE, vol. 5, no. 2, pp. 180–189, 2000.
- [22] A. S. Tokar and P. A. Johnson, “Rainfall-runoff modeling using artificial neural networks,” *Journal of Hydrologic Engineering*, ASCE, vol. 4, no. 3, pp. 232–239, 1999.
- [23] A. S. Tokar and M. Markus, “Precipitation-runoff modeling using artificial neural networks and conceptual models,” *Journal of Hydrologic Engineering*, ASCE, vol. 5, no. 2, pp. 156–161, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

