

Research Article

An Assessment of a Proposed Hybrid Neural Network for Daily Flow Prediction in Arid Climate

Milad Jajarmizadeh, Sobri Harun, and Mohsen Salarpour

Department of Hydraulic and Hydrology, Faculty of Civil Engineering, Universiti Teknologi Malaysia, 81310 Johor, Johor Bahru, Malaysia

Correspondence should be addressed to Sobri Harun; sobri.b.harun@gmail.com

Received 28 May 2013; Revised 9 November 2013; Accepted 20 November 2013; Published 6 February 2014

Academic Editor: Dariusz J. Gawin

Copyright © 2014 Milad Jajarmizadeh 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.

Rainfall-runoff simulation in hydrology using artificial intelligence presents the nonlinear relationships using neural networks. In this study, a hybrid network presented as a feedforward modular neural network (FF-MNN) has been developed to predict the daily rainfall-runoff of the Roodan watershed at the southern part of Iran. This FF-MNN has three layers—input, hidden, and output. The hidden layer has two types of neural expert or module. Hydrometeorological data of the catchment were collected for 21 years. Heuristic method was used to develop the MNN for exploring daily flow generalization. Two training algorithms, namely, backpropagation with momentum and Levenberg-Marquardt, were used. Sigmoid and linear transfer functions were employed to explore the network's optimum behavior. Cross-validation and predictive uncertainty assessments were carried out to protect overfitting and overparameterization, respectively. Results showed that the FF-MNN could satisfactorily predict stream flow during testing period. The Nash-Sutcliffe coefficient, coefficient of determination, and root mean square error obtained using MNN during training and test periods were 0.85, 0.85, and 39.4 and 0.57, 0.58, and 32.2, respectively. The predictive uncertainties for both periods were 0.39 and 0.44, respectively. Generally, the study showed that the FF-MNN can give promising prediction for rainfall-runoff relations.

1. Introduction

A hydrologic model can be categorized as (i) mathematical, (ii) physical, and (iii) analog. Physical model is a small-scaled view of a real phenomenon. Analog model is used for observing one process to create another physically similar natural process. Mathematical model includes obvious series of numerical logical steps and equations that transfer numerical inputs to numerical outputs [1, 2]. Advances and improvements in the development of rainfall-runoff modeling appeared in the 1950s and 1960s [3]. One of the classifications of hydrological modeling is attributed to the concept of artificial intelligence (AI). The artificial neural networks in an AI can be roughly likened to the structure of a brain [4]. ANN researches have so far given rise to three periods of widespread activity. The first was in the 1940s, pioneered by McCulloch and Pitts. The second happened in the 1960s with Rosenblatt's perceptron convergence theory and Minsky and Piper's work showing the boundaries and limitation of

a simple perceptron. In the early 1980s, ANNs began receiving considerable renewed attention [5]. In recent years, neural networks have become tremendously accepted for forecasting and predicting in various disciplines such as that done for the rainfall-runoff relationships in hydrology [6–12]. Artificial networks used for solving problems may be of multilayer perceptron, radial basis function, and generalized feedforward networks. Indeed, every network has its own advantages. A full discussion on artificial neural networks in hydrology can be found in basic literature such as [5, 7, 8, 13].

In this research, feedforward modular neural network (FF-MNN) is proposed for the development of rainfall-runoff modeling as a new generation of neural networks in Roodan watershed. A short literature review for hybrid structures such as modular neural network revealed some interesting studies. For instance, Wu [14] predicted rainfall via a modular radial basis function neural network (M-RBF-NN) method to forecast in a near real time manner. Results indicated that

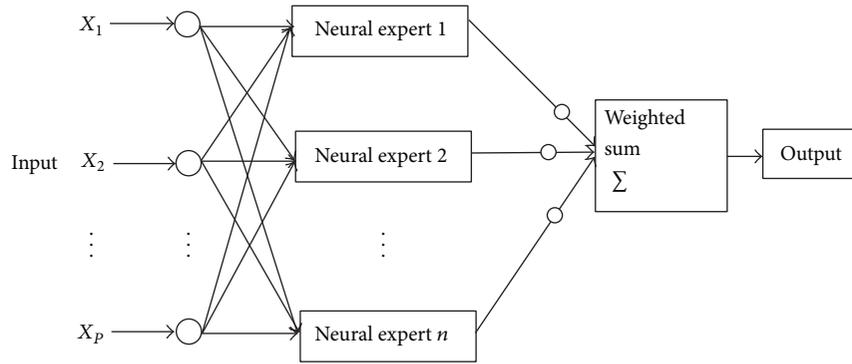


FIGURE 1: General construction of feedforward modular network.

the forecasting was more consistent and accurate. In another study, Wu and Chau [15] studied the optimal prediction of rainfall-runoff through a modular neural network (MNN) and an artificial neural network (ANN). Results showed that MNN provided better accuracy than a simple ANN. In one study in India, four types of feedforward modular neural networks were used to recognize the hourly rainfall-runoff pattern recognition; this study was done by Deshmukh and Ghatol [16]. Results revealed that feedforward modular neural networks are promising for water resource management in monsoon climate. Corzo and Solomatine [17] separated their base flow using modular neural network to predict the flow. The outcome showed that modular neural network is more accurate than traditional ANN models. Parasuraman et al. [18] developed spiking modular neural networks (SMMN) to predict stream flow and evaporation. Results indicated that SMNNs can give better generalization for prediction of high flows and evaporation than feedforward neural networks (FFNNs). Jin et al. [19] used a modular fuzzy neural network (MFNN) for climate prediction. They found that the MFNN model has an advantageous simple structure with no hidden layer, and their results showed that not only can the MFNN give better prediction but also its number of adjustable parameters is obviously less than that of common multilayer neural network. Almasri and Kaluarachchi [20] used modular fuzzy neural networks to simulate nitrate in an agriculture catchment where geographic information system (GIS) was used to prepare the input data. Results revealed that long-time simulations by MNN are effective for future water resource management. Zhang and Govindaraju [21] predicted monthly runoff using Bayesian concepts and MNNs. Their main objective was to develop modular topologies to overcome the complexity of rainfall-runoff modeling. They collected the average monthly rainfall and temperature as input and the output generated was the discharge. The results found satisfying prediction on runoff.

This study, on the other hand, was done to perform pattern recognition via feedforward modular neural network (FF-MNN) for the Roodan catchment situated at the southern part of Iran. As far as the authors are concerned, no similar studies using MNN have ever been done for the aforesaid catchment. The MNN in this study was developed to simulate daily flow via training, cross-validation, and testing.

2. Introduction of Feedforward Modular Neural Network

Feedforward modular neural network (FF-MNN) is a special class of multilayer perceptron (MLP) and is often defined as an extension to multilayer perceptron. This also means that its hypothesis and training rules are the same as MLP [5]. Generally, FF-MNN considers its input using two parallel MLPs networks and then recombines the results to generate the output. Wang et al. [22] stated that MNN is structured from neural expert or modules where each module (neural expert) is designed for individual input-output pairs. The weights of the module are adjusted by applying the attributed algorithm simultaneously during training phase. This procedure improves function specialization in each module to generate more options for the development of topology (architecture).

Generally, MNN can learn pattern recognition and speed up training times. The general representation of FF-MNN is shown in Figure 1. FF-MNN has several modules in hidden layers attributed to the number of neurons and transfer functions, but all learning rules are the same. The modules learn patterns for different input-output pairs using transfer function and the number of neurons. In this study, the FF-MNN had two modules, and this was adequate to configure the nonlinearity. A motivation to apply MNN can be found in Zhang and Govindaraju [21].

3. Methodology

3.1. Case Study. The study area is located in the southern part of Iran between the Hormozgan and Kerman provinces, which is the Roodan watershed. The area of catchment is 10570 km² and lies between northern geographical latitude of 26 degrees and 57 minutes to 28 degrees and 31 minutes and the eastern longitude of 56 degrees and 47 minutes to 57 degrees and 54 minutes (Figure 2). For the period of 1978 to 2008, the average annual precipitation was 215 mm. Generally, the climate of Roodan is arid to semiarid with short and high intensity rainfall. The most important and dominant land uses of Roodan watershed are as shrub land (range brush) mix grassland, and Minab dam is located at the outlet of catchment and is important in collecting surface water for downstream development. Precipitation and discharge were

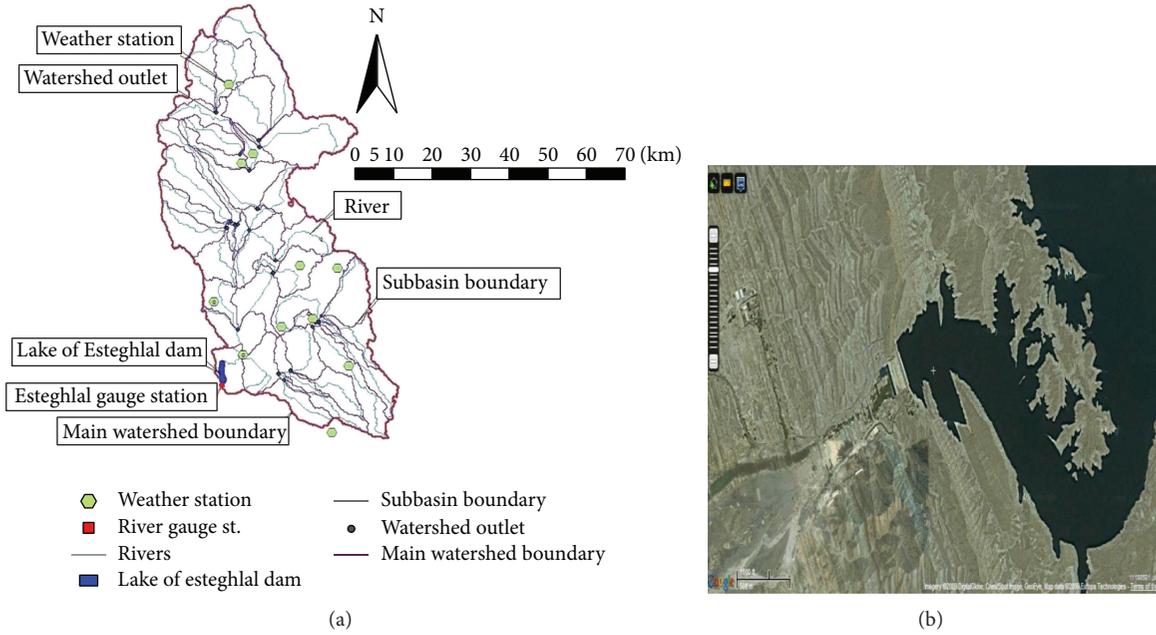


FIGURE 2: Roodan watershed in south of Iran (a); satellite image for the reservoir of Esteghlal (Minab) dam (b).

collected for the Roodan watershed in daily time step from 1988 to 2008.

3.2. Building the FF-MNN. Generally, an artificial neural network such as FF-MNN functions by learning variables relationships in training and then extending them to test conditions [23]. The quantity and quality of data, that is, the information content, is paramount for the modeling [24]. Generally, the collection of training data represents the hydrometeorological patterns' features in a basin and is very significant in neural networks as opined by Yapo et al. [25]. In this study, the hydrometeorological data, namely, precipitation and stream flow, were obtained from IRIMO from 1988 to 2008 in daily time scale for the Roodan watershed. Average daily precipitation (mm) and average daily discharge flow (m^3/s) were used as the available predictors.

The modular feedforward network selected was first introduced by Deshmukh and Ghatol [16]. FF-MNN includes two modules for its hidden layer, which means that there are two parallel calculations for input vectors. Every module has specific number of neurons and attributed transfer functions, and every layer has a training algorithm. The outputs of every module are summed up in the output layer computation. It is important that a desirable number of neurons are selected for every module in the hidden layer to advantage the transfer functions and optimize the training algorithm; the number of layers in this case can be selected heuristically through trial and error [26]. This entire process demands that the modeler remains patient before a neural network is developed. The transfer function is a required component for every process element (neuron) because the generating of output vectors in a neuron is related to the transfer function types [27]. In rainfall-runoff modeling, the sigmoid and linear transfer

TABLE 1: Applied transfer functions.

Transfer function	Description
Sigmoid	Sigmoid (0/1)
Linear sigmoid	Piecewise linear (0/1)
Linear	Adds a bias and scales

TABLE 2: Applied Training algorithms.

Name	Description
Backpropagation with momentum	Gradient and weight change
Levenberg-Marquardt	Improved second order method for gradient

functions are the most popular functions, as mentioned in [28]. Table 1 indicates the applied transfer functions for this research.

Generally, learning rule determines the relative significance of input weights to a process element. The most popular training algorithm is backpropagation (Momentum), which has been derived from gradient descent rule [22]. The Levenberg-Marquardt algorithm, which is associated with the optimization numerical technique as the learning rule, is undergoing more evaluation for rainfall-runoff relationships [29]. In our study, two training algorithms were chosen, and these are as shown in Table 2.

In neural networks, the data has to be standardized according to the training algorithm and then the data sets need to be divided for training and testing. The data should be normalized (standardized) because of equal consideration during the training stage. In this study, since sigmoid and

linear functions were used, the data were scaled separately for sigmoid to be between 0 and 1; this method was suggested by Zadeh et al. [30].

Three data sets were involved in the development procedure—the training set, the cross-validation set, and the test set (validation). The training dataset was first applied to train the FF-MNN model configuration. The cross-validation dataset was then applied to decide the training's stopping time to prevent overfitting [27]. Finally, the test set was applied to assess the selected model alongside independent data. Generally, about 70% of the input-output pairs were used for learning and remaining 30% of input-output pairs were used for validation (test). 10% of the training data was used as the cross-validation set for Roodan watershed.

Several architectures (topologies) were developed in this study to find the optimum generalization for the training algorithms, transfer functions, and number of neurons. In the first step, a series of configurations were tested by combining training algorithms and transfer functions in the hidden and output layers. Table 3 shows the designed architectures for Roodan watershed. There is no definite algorithm that can show the optimal number of neurons needed in a hidden layer, but this can be determined through trial and error [26]. In this regard, the number of neurons in the hidden layer for the two modules was increased gradually. Two evolution schemes were involved here for Module 1: (1) increase of the neurons for Module 1 and Module 2 while fixing two neurons in Module 2 and (2) increase of the neurons for Module 2 while fixing the optimum neurons for Module 1 (optimum neuron means that more neurons do not improve the generalization further). The same procedure was repeated for Module 2 where (1) the number of neurons for Module 2 was increased while two neurons in Module 1 were fixed and (2) the number of neurons in Module 1 was increased while the optimum neurons for Module 2 were fixed. The general aim here is to obtain a consistent generalization.

3.3. General Methodology of FF-MNN for Roodan Watershed.

The development of FF-MNN was initiated by first choosing the data set for model learning and validation. This was followed by determining the input and output variables and scaling the data. After that, the network topology and specification for the number of cells for hidden layer were set. Then, the crude FF-MNN was trained and tested to find the optimum results. This FF-MNN had three layers, which were the input, hidden, and output layers. The hidden layer had two modules (neural expert).

It was obvious at this point that the number of cells in the input layer was correlated with the input data vector and the same relationship was found between the output layer and its vector. Normally, one hidden layer is enough for rainfall-runoff modeling by ANNs. However, to find the optimal network architecture is a task that is highly dependent on the number of hidden layers. In this study, the layers were evaluated layer by layer, and this was done using the heuristic approach, deemed as a usual method, as suggested by Bowden et al. [7]. This is a stepwise approach where the inputs (forward approach) are increased stepwise by decreasing

the inputs (backward approach). In this regard, the modeler should consider the complexity of the neural network with the number of inputs variables (i.e., attributed lags and various variables). In this study, the input pairs were determined in a forward approach to find the optimum generalization. The FF-MNN model was trained using the resulting daily data of runoff and rainfall. The input vector was represented by rainfall (PCP) and runoff (Q) values for the previous five days. The reason that a five-day lag was chosen was due to the existence of rainfall events in five previous days (i.e., $t-1$, $t-2$, $t-3$, $t-4$, and $t-5$) [28]. The FF-MNN model can also be showed in the following compact format:

$$Q_t = \text{FF-MNN} [\text{PCP}_t, \text{PCP}_{t-1}, \text{PCP}_{t-2}, \text{PCP}_{t-3}, \text{PCP}_{t-4}, \text{PCP}_{t-5}, Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}]. \quad (1)$$

By using the forward approach to combine the input variables, the FF-MNN architectures were trained to capture the dynamic, complex, and nonlinear rainfall-runoff mechanism in the Roodan watershed with harmony of transfer function and normalized data. At this stage, the FF-MNN was developed by combining input data, transfer functions, and training algorithms in the hidden and output layers (Table 3). As mentioned before, the optimal number of neurons in the hidden layer was determined using a trial-and-error procedure via the two evolution schemes suggested by [7]. Special consideration was also taken to the fact that smaller amount cells could be inadequate to capture difficult relations between predictors and calculated output [31]. Finally, the results were compared to calculate the various performance evaluation indices (i.e., MSE, R^2 , and NS) for both training and testing data sets before the optimum developed topology was decided.

4. Model Performance Assessment

The hydrological model's accuracy can be evaluated using many approaches, for example, the methods proposed by the World Meteorological Organization (WMO) which can be generally divided into graphical evaluation and numerical assessment [32]. WMO [33] has suggested two indicators as graphical evaluation for observed and simulated data, which are

- (i) linear scale plot of the predicted and measured data,
- (ii) double mass plots of the estimated and real data.

The numerical assessment can be carried out in two forms as well, that is, absolute goodness-of-fit and relative goodness-of-fit [34]. Relative goodness-of-fit is a nondimensional criterion that offers a relative comparison between the observed and simulated data [35], represented as the coefficient of determination (R^2) and Nash-Sutcliffe coefficient of efficiency (NS). The absolute goodness-of-fit statistics has dimension [35] and is represented as the root mean square error (RMSE).

TABLE 3: Developed configurations of MNNs for Roodan watershed.

Architecture	Hidden layer 1			Output layer	
	Module	Transfer function	Training algorithm	Transfer function	Training algorithm
1	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Sigmoid			
2	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Linear sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Sigmoid			
3	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Linear	*Levenberg-Marquardt or momentum
	Module 2	Sigmoid			
4	Module 1	Linear sigmoid	*Levenberg-Marquardt or momentum	Sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear sigmoid			
5	Module 1	Linear sigmoid	*Levenberg-Marquardt or momentum	Linear sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear sigmoid			
6	Module 1	Linear sigmoid	*Levenberg-Marquardt or momentum	Linear	*Levenberg-Marquardt or momentum
	Module 2	Linear sigmoid			
7	Module 1	Linear	*Levenberg-Marquardt or momentum	Sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear			
8	Module 1	Linear	*Levenberg-Marquardt or momentum	Linear sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear			
9	Module 1	Linear	*Levenberg-Marquardt or momentum	Linear	*Levenberg-Marquardt or momentum
	Module 2	Linear			
10	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear sigmoid			
11	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Linear sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear sigmoid			
12	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Linear	*Levenberg-Marquardt or momentum
	Module 2	Linear sigmoid			
13	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear			
14	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Linear sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear			
15	Module 1	Sigmoid	*Levenberg-Marquardt or momentum	Linear	*Levenberg-Marquardt or momentum
	Module 2	Linear			
16	Module 1	Linear sigmoid	*Levenberg-Marquardt or momentum	Sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear			
17	Module 1	Linear sigmoid	*Levenberg-Marquardt or momentum	Linear sigmoid	*Levenberg-Marquardt or momentum
	Module 2	Linear			
18	Module 1	Linear sigmoid	*Levenberg-Marquardt or momentum	Linear	*Levenberg-Marquardt or momentum
	Module 2	Linear			

*For every configuration, both training algorithms, Levenberg-Marquardt and momentum, were examined separately for hidden and output layers.

(a) Coefficient of determination is presented as

$$R^2 = \left\{ \frac{\sum_{i=1}^n (Q_{obs} - Q_{obsavg})(Q_{sim} - Q_{simavg})}{\left[\sum_{i=1}^n (Q_{obs} - Q_{obsavg})^2 \sum_{i=1}^n (Q_{sim} - Q_{simavg})^2 \right]^{0.5}} \right\}^2, \quad (2)$$

where Q_{obs} is the observed value at time i ; Q_{sim} is the predicted value at time i ; n is the sum number of observations;

and Q_{obsavg} and Q_{simavg} are the average of observed and predicted values, respectively. R^2 ranges between 0 and 1, and a higher value indicates higher degree of harmony or agreement.

(b) Nash-Sutcliffe (NS) is presented as

$$NS = 1 - \left[\frac{\sum_i^n (Q_{simi} - Q_{obsi})^2}{\sum_i^n (Q_{obsi} - Q_{avg})^2} \right], \quad (3)$$

where n is the number of time steps; Q_{simi} and Q_{obsi} are the simulated and observed stream flow at time step i ; and Q_{avg} is

TABLE 4: Optimum architecture of FF-MNN in Roodan watershed.

Input variable*	Hidden layer 1		Output layer	
	Module	Transfer function	Learning rule	Transfer function
PCP _(t) , PCP _(t-1) , Q _(t-1) = 3 cells	Module (1): 38 cells	Sigmoid	Levenberg-Marquardt	Linear: 1 cell
	Module (2): 26 cells	Linear sigmoid		
Learning rule: Momentum (momentum = 0.8) (step size = 0.1)				

*PCP_(t) is the present precipitation meanwhile PCP_(t-1) is the precipitation for one day before present time. Q_(t-1) is the discharge for one day before present time.

the average observed stream flow over the simulation period. When NS is equal to 100%, it means that a perfect dependency between the observed and the predicted values has been achieved. Generally, the Nash-Sutcliffe coefficient is developed over correlation-based measures due to its responsiveness to the measured and predicted averages and variances [35].

(c) Root mean square error (RMSE) is presented as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (O_i - P_i)^2}{n}}, \quad (4)$$

where O_i is the measured value at time i ; P_i is the estimated value at time i ; and n is the total number of measured data. The RMSE is a dimensional measurement that shows the agreement between the observed and simulated data. When RMSE is close to zero, it means that the model's performance is good.

Generally, the predictive uncertainty (PU) of the ANN is assessed using the noise-to-signal ratio index [6]. In this case, it was calculated as the unbiased standard error (SEE). SSE is an unexplained variance which is compared with the standard deviation of observed values for the dependent variable (STD). Therefore, the ratio of SEE to STD (SEE/STD) is named as the noise-to-signal ratio or the predictive uncertainty index. It indicates the degree to which noise hides the information. The model can offer correct predictions if SEE is smaller than the STD. On the other hand, the model estimations will not be correct if the ratio is larger than 1. SEE is presented as

$$\text{SEE} = \left[\frac{1}{\nu} \sum_{i=1}^n (Q_p - Q_o)^2 \right]^{0.5}, \quad (5)$$

where ν is the degree of freedom and is equal to the number of observations in the training set minus the number of parameters. Q_o and Q_p are the observed and predicted values of flow, respectively. The predicted uncertainty (PU) is thus calculated as

$$\text{PU} = \frac{\text{SEE}}{\text{STD}}. \quad (6)$$

5. Results and Discussions

The optimum developed FF-MNN was found via challenging heuristic method that considered (i) different topologies

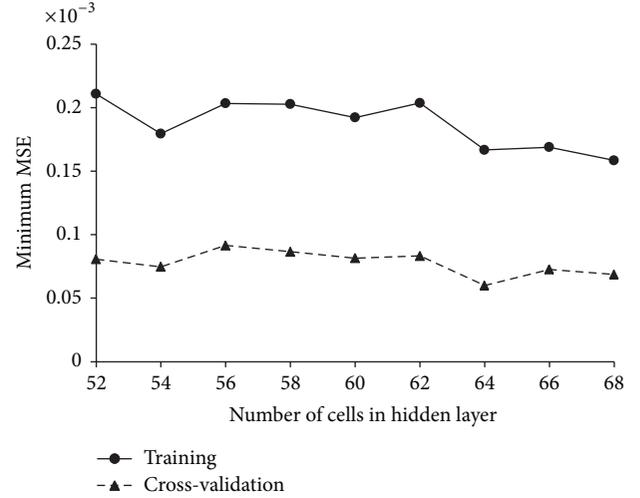


FIGURE 3: Training and cross-validation curves attributed with MSE for FF-MNN.

training; (ii) combination of input variables; (iii) increasing and decreasing number of neurons in hidden layer for both modules; and (iv) exploration on the learning rate and momentum term to bring better generalization through trial and error. Figure 3 shows the behavior of the minimum mean square error (MSE) for the training and cross-validating of the dataset on a normalized data for optimum architecture. The optimum numbers of cells which had increased and decreased throughout the heuristic procedure were 38 and 26 for Module 1 and 2, respectively, in the hidden layer. Generally, the increase in neurons that has resulted in better generalization can be attributed to the sigmoid transfer function for Module 1. However, the transfer function failed to incur any significant changes in Module 2. The results were not satisfactory since the neurons number was shorter than 52.

Table 4 presents the final developed FF-MNN model with related components in the hidden and output layers for Roodan watershed. To sum up, the linear transfer function in the output layer and sigmoid type transfer function in the hidden layer (module) have led to better generalization. Additionally, Levenberg-Marquardt algorithm in hidden layer and backpropagation in output layer were found suitable in improving generalization.

In this study, the suitable momentum and step size (learning rate) values were found through trial-and-error.

TABLE 5: Examining indices for accuracy of FF-MNN model.

Index	Calibration	Validation
Nash and Sutcliffe coefficient % (NS)	85	57
Coefficient of determination % (R^2)	85	58
Root mean square error	39.4	32.2

The momentum speeds up the training in very flat regions of the error surface. A learning rate is applied to increase the possibility of preventing trapping in local minima as an alternative of global minima [5]. The momentum value should be less than 1.0 (normally between 0.1 and 0.9) for making convergence in the network. Adjustment of step size is usually related with updating the weights space. Xu et al. [36] found that a momentum value of 0.85 and learning rate of 0.23 was suitable. Furthermore, Nourani and Kalantari [26] proposed values of 0.9 and 0.1 for momentum and learning rate, respectively. In this research, the momentum and step size were set at 0.8 and 0.1, respectively, after taking into consideration training time and stability of the optimum results and literature suggestions.

The NS-coefficients for calibration and validation period were 85% and 57%, respectively, for the optimal FF-MNN. The calibration period gave good performance while that of validation period was moderate, as defined according to Tombul and Oğul [37]. The R^2 coefficients obtained were 85% and 58% for calibration and validation periods, respectively. In a recent study in Pakistan where a feedforward neural network model was developed to predict the monthly runoff of an arid large watershed (9391 km²) with an annual precipitation of 191 mm [38], the NS values were 0.88 and 0.63 for calibration and validation periods, respectively; these are in fairly good agreement with this study. Clearly, the rainfall-runoff processes were extremely nonlinear. The differences between training and test (validation) could be derived from the complexity of the rainfall-runoff relationships that had become more significant in large-scale watersheds with aridity climate. Therefore, the model failed to capture this nonlinearity relationship in a perfect manner. Deshmukh and Ghatol [16] reported a satisfactory range between 0.5 and 0.8 for the correlation coefficient (r) of a 4000 km² catchment in India; this was obtained through a feedforward modular neural network for rainfall-runoff developed in hourly time step. Table 5 shows the accuracy criteria for the developed model in this study.

The daily stream flow in m³/s (CMS) was evaluated through graphical visualizations and statistical analysis to give a good cognition for reviewing observed and simulated daily stream flows. Figures 4 and 5 show a comparison between measured and simulated stream flow in CMS for the training and test periods, respectively. Figure 4 depicted an acceptable simulation for peak flows by FF-MNN for training period. The 5 February 1993 event had the largest flow recorded in the Roodan watershed and had the most promising striking. On the contrary, the largest flow was recorded on

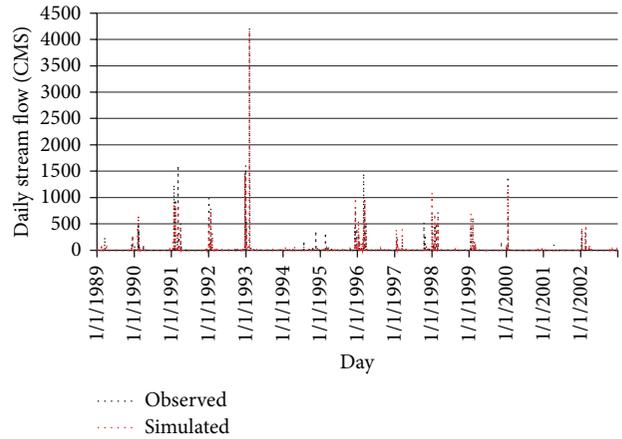


FIGURE 4: Measured and simulated stream flow (CMS) for training period.

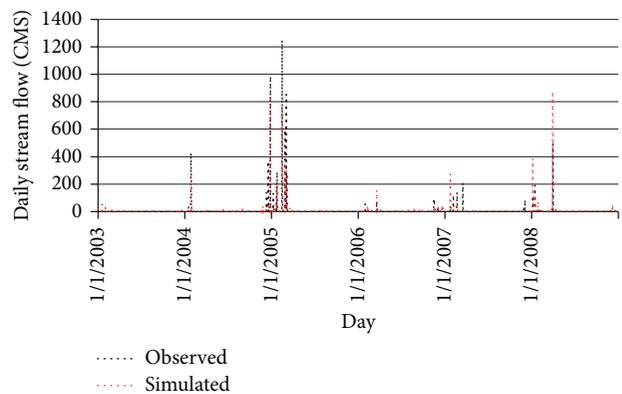


FIGURE 5: Measured and simulated stream flow (CMS) for test period.

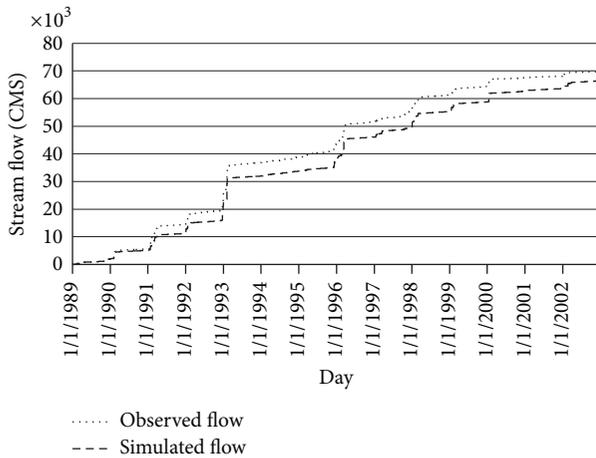
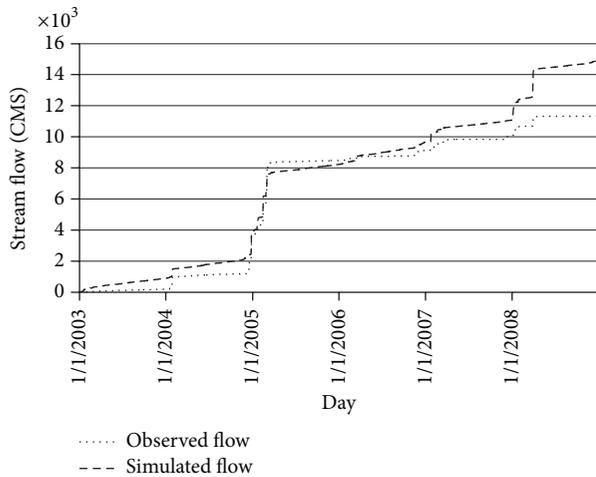
14 February 2005 for the test period, but the MNN had clearly underestimated this. In Figure 5, the daily stream flow on 29 March 2008 was obviously overestimated. Generally, the stream flow prediction was inspiring for both periods, and their predictions followed a similar trend to measure stream flows.

Figures 6 and 7 illustrate the cumulative daily stream flow for Roodan during calibration and validation periods. Figure 6 showed that the simulated daily stream flow has been underestimated over the period of 1991 to 2002. From 1989 to 1990, the simulated daily flows were satisfactory. For the test period, the FF-MNN predicted two similar trends as overestimation (Figure 7). To sum up, the simulated cumulative daily flow has been overestimated for the test period, but the similarity in early 2005 is acceptable.

Generally, the daily cumulative flow trend is similar between observed and simulated daily flow for the training period, though there is a slight sustainable underestimation. In addition, there is an overestimation for the test period except in early 2005. The dissimilarities between observed and predicted flows for both periods may be due to the capability of the network and the significance of nonlinearity in

TABLE 6: Percentile of absolute error value between observed and simulated flow (CMS).

	Percentile %						
	5	10	25	50	75	90	95
Validation	0.75	0.8	1	1.27	1.42	4.6	10
Calibration	0.05	0.09	0.34	0.87	2	6.2	19.7

FIGURE 6: Cumulative daily stream flow m^3/s (CMS) for calibration period.FIGURE 7: Cumulative daily stream flow m^3/s (CMS) for validation period.

rainfall-runoff relationships for large-scale watersheds with aridity climate.

The percentile absolute errors between observed and simulated flows are shown in Table 6 for both training and testing periods. 95% of the data had a difference of $10 \text{ m}^3/\text{s}$ in testing period and $19.7 \text{ m}^3/\text{s}$ in training period. Generally, the percentiles of absolute error that were less than 50% were lesser for the testing period than the training period. They have been shown in bold version in Table 6.

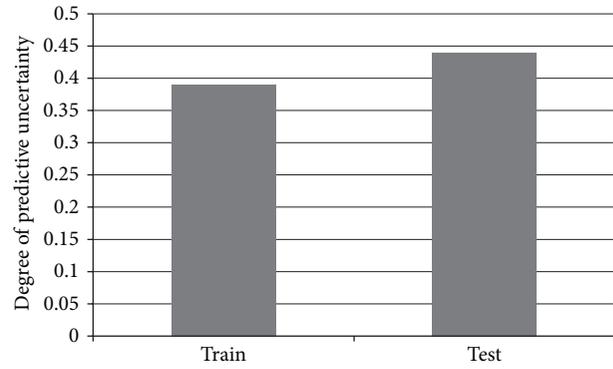


FIGURE 8: Uncertainty prediction of training and test period for FF-MNN.

The predictive uncertainty obtained for training and testing periods was 0.39 and 0.44, respectively (Figure 8). Generally, PU under unity is considered satisfying [6]. In the study of Tokar and Johnson [24], the predictive uncertainty evaluation for neural networks involving learning data included wet and dry or wet and average years. They opined that such can offer more acceptable predictive accuracy in comparison to networks trained using a combination of dry and/or average year data. Perhaps the reason of satisfying predictive uncertainty of MNN for Roodan watershed was due to the application of a large data period that involved wet, dry, and average years for training.

6. Conclusion

This study has proposed a feedforward modular neural network for a large catchment with aridity climate for rainfall-runoff prediction. The FF-MNN was developed through training, cross-validation, and testing. LevenbergMarquardt and backpropagation with momentum terms were used to develop the training algorithms. Sigmoid and linear transfer functions were applied to compute the neuron output. The developed FF-MNN gave good and fair predictions for training and test. The absolute errors according to 50 percentiles in the test period were less than those in training period. The uncertainty prediction obtained was satisfactory for both periods. To conclude, feedforward modular neural networks can be promising as new generation of neural networks for flow prediction.

Conflict of Interests

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

Acknowledgments

The authors appreciate the cooperation and help given by the Department of Hydraulic and Hydrology and Centre of Information and Communication Technology (CICT) of Universiti Teknologi Malaysia; consultant engineers of Ab Rah Saz Shargh Corporation in Iran; and the Regional Water Organization, Agricultural Organization, and Natural Resources Organization of the Hormozgan province, Iran.

References

- [1] S. L. Dingman, *Physical Hydrology*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
- [2] M. Jajarmizadeh and M. Sobri Harun Salarpour, "A review on theoretical consideration and types of models in hydrology," *Journal of Environmental Science and Technology*, vol. 5, no. 5, pp. 249–261, 2012.
- [3] E. M. Shaw, *Hydrology in Practice*, Chapman and Hall, London, UK, 1994.
- [4] D. Anderson and G. Mcneill, "Artificial Neural Networks Technology," Tech. Rep. RL/C3C, Rom laboratory, Griffiss AFB, Rome, NY, USA.
- [5] ASCE, "Artificial neural networks in hydrology. I: preliminary concepts," *Journal of Hydrologic Engineering*, vol. 5, no. 2, pp. 115–123, 2000.
- [6] A. R. Senthil Kumar, K. P. Sudheer, S. K. Jain, and P. K. Agarwal, "Rainfall-runoff modelling using artificial neural networks: comparison of network types," *Hydrological Processes*, vol. 19, no. 6, pp. 1277–1291, 2005.
- [7] G. J. Bowden, G. C. Dandy, and H. R. Maier, "Input determination for neural network models in water resources applications. Part 1—background and methodology," *Journal of Hydrology*, vol. 301, no. 1–4, pp. 75–92, 2005.
- [8] A. M. Kalteh, P. Hjorth, and R. Berndtsson, "Review of the self-organizing map (SOM) approach in water resources: analysis, modelling and application," *Environmental Modelling and Software*, vol. 23, no. 7, pp. 835–845, 2008.
- [9] M. Salarpour, Z. Yusop, M. Jajarmizadeh, and F. Yusof, "Annual peak flow prediction by multi layer perceptron network in tropical region: johor watershed," in *Proceeding of the 2nd International Conference on Water Resources (ICWR '12)*, Langkawi, Malaysia, 2012.
- [10] M. Salarpour, Z. Yusop, M. Jajarmizadeh, S. Shahid, and F. Yusof, "A generalized feed forward network for the prediction of runoff in Johor River Basin," in *Proceeding of the 4th International Graduate Conference on Engineering Science & Humanity (IGCESH '13)*, pp. 16–17, Johor, Malaysia, 2013.
- [11] E. K. Lafdani, A. Moghaddam Nia, C. A. Pahlavanravi A, M. Ahmadi, and Jajarmizadeh, "Daily rainfall-runoff prediction and simulation using ann, anfis and conceptual hydrological MIKE11/NAM models," *International Journal of Engineering & Technology Sciences*, vol. 1, no. 1, pp. 32–50, 2013.
- [12] E. Kakaei Lafdani, A. R. M. Nia, M. Jajarmizadeh, A. Ahmadi, and M. G. Gosheh, "Stream flow simulation using SVM, ANFIS and NAM models," *Caspian Journal of Applied Sciences Research*, vol. 2, no. 4, pp. 86–93.
- [13] 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.
- [14] J. Wu, "Prediction of Rainfall Time Series Using Modular RBF Neural Network Model Coupled With SSA and PLS," in *Intelligent Information and Database Systems*, vol. 7197, pp. 509–518, Springer, Berlin, Germany, 2012.
- [15] C. L. Wu and K. W. Chau, "Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis," *Journal of Hydrology*, vol. 399, no. 3–4, pp. 394–409, 2011.
- [16] R. P. Deshmukh and A. A. Ghatol, "Modular neural network approach for short term flood forecasting a comparative study," *International Journal of Advanced Computer Science and Applications*, vol. 1, no. 50, pp. 81–87, 2010.
- [17] G. Corzo and D. Solomatine, "Baseflow separation techniques for modular artificial neural network modelling in flow forecasting," *Hydrological Sciences Journal*, vol. 52, no. 3, pp. 491–507, 2007.
- [18] K. Parasuraman, A. Elshorbagy, and S. K. Carey, "Spiking modular neural networks: a neural network modeling approach for hydrological processes," *Water Resources Research*, vol. 42, no. 5, Article ID W05412, pp. 1–14, 2006.
- [19] L. Jin, J. Jin, and C. Yao, "A short-term climate prediction model based on a modular fuzzy neural network," *Advances in Atmospheric Sciences*, vol. 22, no. 3, pp. 428–435, 2005.
- [20] M. N. Almasri and J. J. Kaluarachchi, "Modular neural networks to predict the nitrate distribution in ground water using the on-ground nitrogen loading and recharge data," *Environmental Modelling and Software*, vol. 20, no. 7, pp. 851–871, 2005.
- [21] B. Zhang and R. S. Govindaraju, "Prediction of watershed runoff using Bayesian concepts and modular neural networks," *Water Resources Research*, vol. 36, no. 3, pp. 753–762, 2000.
- [22] W. Wang, P. H. A. J. M. van Gelder, J. K. Vrijling, and J. Ma, "Forecasting daily streamflow using hybrid ANN models," *Journal of Hydrology*, vol. 324, no. 1–4, pp. 383–399, 2006.
- [23] E. Yakhkeshi, *Application of artificial neural network models in flood warning system preparation [Ph.D. dissertation]*, Azad University, Tehran, Iran, 2007.
- [24] A. S. Tokar and P. A. Johnson, "Rainfall-runoff modeling using artificial neural networks," *Journal of Hydrologic Engineering*, vol. 4, no. 3, pp. 232–239, 1999.
- [25] P. O. Yapo, H. V. Gupta, and S. Sorooshian, "Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data," *Journal of Hydrology*, vol. 181, no. 1–4, pp. 23–48, 1996.
- [26] V. Nourani and O. Kalantari, "Integrated artificial neural network for spatiotemporal modeling of rainfall-runoff-sediment processes," *Environmental Engineering Science*, vol. 27, no. 5, pp. 411–422, 2010.
- [27] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox, Users Guide*, vol. 7, The Math Works, 2010.
- [28] S. Riad, J. Mania, L. Bouchaou, and Y. Najjar, "Predicting catchment flow in a semi-arid region via an artificial neural network technique," *Hydrological Processes*, vol. 18, no. 13, pp. 2387–2393, 2004.
- [29] A. Bhadra, A. Bandyopadhyay, R. Singh, and N. S. Raghuwanshi, "Rainfall-runoff modeling: comparison of two approaches with different data requirements," *Water Resources Management*, vol. 24, no. 1, pp. 37–62, 2010.
- [30] M. R. Zadeh, S. Amin, D. Khalili, and V. P. Singh, "Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions," *Water Resources Management*, vol. 24, no. 11, pp. 2673–2688, 2010.

- [31] B. Sivakumar, A. W. Jayawardena, and T. M. K. G. Fernando, "River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches," *Journal of Hydrology*, vol. 265, no. 1-4, pp. 225-245, 2002.
- [32] N. 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.
- [33] World Meteorological Organization, "Inter-comparison of conceptual models used in operational hydrological forecasting," Tech. Rep. 429, World Meteorological Organization, Geneva, Switzerland, 1975.
- [34] P. Srivastava, J. N. McNair, and T. E. Johnson, "Comparison of process-based and artificial neural network approaches for streamflow modeling in an agricultural watershed," *Journal of the American Water Resources Association*, vol. 42, no. 3, pp. 545-563, 2006.
- [35] P. Kneale, L. See, and A. Smith, "Towards defining evaluation measures for neural network forecasting models," in *Proceedings of The 6th International Conference on Geocomputation*, Geo Computation, University of Queensland Brisbane, Australia, 2001.
- [36] Z. X. Xu, J. P. Pang, C. M. Liu, and J. Y. Li, "Assessment of runoff and sediment yield in the miyun reservoir catchment by using SWAT model," *Hydrological Processes*, vol. 23, no. 25, pp. 3619-3630, 2009.
- [37] M. Tombul and E. Oğul, "Modeling of rainfall-runoff relationship at the semi-arid small catchments using artificial neural networks," in *Intelligent Control and Automation*, vol. 344 of *Lecture Notes in Control and Information Sciences*, pp. 309-318, 2006.
- [38] A. R. Ghumman, Y. M. Ghazaw, A. R. Sohail, and K. Watanabe, "Runoff forecasting by artificial neural network and conventional model," *Alexandria Engineering Journal*, vol. 50, no. 4, pp. 345-350, 2011.



Hindawi

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

