Journal of Environmental Informatics 19(1) 38-50 (2012)

Journal of Environmental Informatics

www.iseis.org/jei

Investigating the Ability of Artificial Neural Network (ANN) Models to Estimate Missing Rain-gauge Data

V. Nourani[1,2], A. H. Baghanam[2,*], and M. Gebremichael[3]

[1]Department of Civil Engineering, University of Minnesota, Minneapolis, MN 55414, USA
[2]Department of Water Engineering, Faculty of Civil Engineering, University of Tabriz, Tabriz 5166614711, Iran
[3]Department of Civil and Environmental Engineering, University of Connecticut, Storrs, Connecticut 06268, USA

Received 5 November 2010; revised 22 August 2011; accepted 2 September 2011; published online 12 March 2012

ABSTRACT. The correct forecasting of unrecorded data could be enormously helpful in designing water projects and preventing related damages. The conventional methods available for rainfall estimation usually take a long time to estimate the missing data, and their estimations may have many errors in the long-term simulations. In this study, the capabilities of different Artificial Neural Networks (ANNs) were analyzed in estimating missing data from the Ardabel plain rain gauge stations located in northwestern Iran. Accordingly, six different structures of ANNs were used, and their efficiencies in terms of the mean squared error, training, and validation determination coefficients to select better-estimated missing data were examined. The results revealed that the best model is composed of the feed-forward networks, trained by the Levenberg-Marquardt algorithm and considering only one hidden layer. For each of the stations with a complete data set, an ANN was trained. Data gaps from other stations were obtained by the proposed ANN models. Furthermore, an integrated ANN was developed to investigate the hidden spatial relationships among the rainfall data of the stations as well as temporal auto-correlations. The results indicated the superiority of the proposed integrated model. After the estimation of the rain data gaps, the K-means clustering method was also employed as a data pre-processing method to improve the accuracy of the estimation, and the method led to better results.

Keywords: artificial neural networks, black box model, rainfall forecasting, clustering, Ardabel plain

1. Introduction

Water resource system planning, development, management, and optimal operation require different information in meteorology, hydrology, economics and social areas. Among atmospheric precipitations, rainfall is one of the most important processes of the hydrological cycle. Processes such as floods, erosion, sedimentation, and surface and groundwater pollution happen only when rainfall occurs. The rainfall process shows large spatiotemporal variations and quantifying rainfall in space and time is one of the meteorologists' issues for investigation.

Considerable variations of rainfall in space and time and the shortage of rain stations for recording the amount of rain indicate the need for reliable rainfall estimation models in developing countries. The deficiency of climatological and hydrological information in an area could be because of damage to measurement instruments or negligence in measurements. Because it is impossible to postpone the implementation of a project because of gaps in long-term hydrological data, the estimation of missing data in hydrological studies is inevitable, and the role of reliable data in hydraulic and hydrological designs cannot be ignored. Correct predictions of the missing data decrease uncertainties and could help to establish suitable plans for water projects.

Principally, conventional methods of data gap estimation, such as interpolation and extrapolation, differentials, proportional, mean, and sampling methods, are time-consuming, and they are usually associated with many errors (Karamoz and Araginejad, 2006). Nonlinearity and natural uncertainty of a stochastic process such as precipitation, the need for long-term historical information, and the complexity of physical-based methods are the reasons that researchers have attempted to develop black box models such as Artificial Neural Network (ANN). When data is insufficient and accurate prediction is more important than conceiving the physics of a problem, black box models could be a good option (Nourani and Mano, 2007). ANN models are black box models with particular properties for modeling nonlinear systems. As a black box tool, ANNs are used to relate the inputs to the outputs; they are less sensitive to input data errors and have the ability of parallel data processing (Nourani et al., 2009). ANNs are capable of recognizing systematic noise, such as a measurement instrument's noise pattern, in the input data. In addition, ANNs have the ability to recognize and isolate systematic noise from the inherent noise

* Corresponding author. Tel.: +98 411 3392409; fax: +98 411 3344287.
E-mail address: hosseinibaghanam@gmail.com (A. H. Baghanam).

ISSN: 1726-2135 print/1684-8799 online
 doi:10.3808/jei.201200207

of the data, and the ability to identify a relationship from given patterns makes it possible for ANNs to solve complex hydrologic problems (Nourani et al., 2008).

ANNs were first provided as an experimental model to formulate human brain capabilities. In the past decade, with due attention to calculation instruments' necessity in studies, many researchers have studied the ability of ANNs to process data from multivariable space to the other spaces. By providing a mathematical framework to the ANNs by Rummelhart and McLelland (1986), the calculation mechanism was developed.

In meteorological and hydrological studies, non-linear ANN models are extremely useful because they do not employ complicated differential equations. At the end of the 1990s, ANNs were used in hydrological and meteorological problems, including precipitation and river flow prediction, ground water modeling, water quality, water management, and river sediment (ASCE, 2000; Chau, 2006). The advantages of ANNs over traditional statistical and numerical weather prediction approaches have been discussed by McCann (1992), Kuligowski and Barros (1998), Silverman and Dracup (2000), and Chau and Cheng, (2002). Marzban (2002) used ANNs for post-processing the temperature forecasts of the Advanced Regional Prediction System. Marzban and Witt (2001) developed two ANNs for the estimation of severe-hail size, and they found that the ANNs designed to predict severe-hail size outperform the existing methods. Because of the importance of rainfall among other atmospheric events, ANNs and other classic methods have been used in various studies to model this hydrological process. Bodri and Cermak (2000) investigated an ANN model in which precipitation in a given month was forecasted using the rainfall values of the previous two months of the current year. Aksoy and Dahamsheh (2009) forecasted the monthly precipitation in arid regions by means of feed-forward back-propagation ANNs. ANN approach was used to forecast the time series of average summer-monsoon rainfall over India by Chattopadhyay and Chattopadhyay (2008). Hung et al. (2009) used meteorological parameters in developing an ANN rainfall forecast model. Furthermore, there are studies that report the suitability and superiority of the ANN models over conventional statistical rainfall prediction procedures (e.g., Greischar et al., 1995; Toth et al., 2000; Chattopadhyay, 2007). Mogaddam et al. (2009) used ANN technique for precipitation modeling of the Tabriz plain in Iran and presented an experimental formula for determining the number of hidden layer neurons. Nourani et al. (2009) developed a combined neural-wavelet model to predict the Ligvanchay Watershed precipitation in Tabriz, Iran. In their study, a link between wavelet analysis and the ANN was established for temporal pre-processing of rainfall data and precipitation prediction.

Wu et al. (2010) applied an ANN in rainfall time series modeling. They used different methods, but the modular artificial neural networks (MANNs) performed the best among the various models used; in the study, they indicated that advantages of MANNs over other models are quite noticeable, particularly for daily rainfall forecasting, whereas the predominance of MANN over ANN in monthly rainfall series forecasting was not proven. Ramesh et al. (2005) developed improved weighting methods and deterministic and stochastic data-driven models to estimate the precipitation records**.** A data-driven model that uses the ANN concept and a stochastic interpolation technique was applied in the study. Luk et al. (2001) predicted the spatial distribution of rainfall over the Aperparamata River basin in western Sydney, Australia. They used different structures of ANNs such as partial recurrent, time-delay, and feed-forward ANNs that provided suitable predictions of precipitation. Ramirez et al. (2005) employed the ANN technique for spatial modeling of precipitation in state of Sao Paulo, Brazil. For this purpose, the precipitation data of six sub-regions of the state along with the meteorological data such as pressure, moisture, temperature, and precipitation were used in the model as input data to predict the precipitation of the next time step. French et al. (1992) trained an ANN model for spatiotemporal modeling of rainfall, and the ability of the model over other conventional methods was concluded.

There are also studies regarding the application of conventional methods to fill in missing rainfall data. Singh and Chowdhury (1986) compared thirteen rainfall estimation methods and found that the isohyetal method yielded higher estimates of the mean daily and monthly rainfall than other methods in the area of their study. DeSilva et al. (2007) employed the Arithmetic Mean (Local Mean), Normal Ratio, Inverse Distance and Aerial Precipitation Ratio methods to estimate missing rainfall data in Sri Lanka. Kim and Pachepsky (2010) developed a new technique to reconstruct missing daily precipitation data in the central part of the Chesapeake Bay Watershed.

In rainfall models, accurate data acquisition without missing or wrong values is very important because most hydrological models require accurate observed data to estimate the current situation. However, missing or wrong data caused by accidental situations and/or malfunctions are often recorded, and the performance of the models might be poor if these missing and wrong data are used without modification. Therefore, prediction, calibration, and validation of such data are necessary. In developing countries, there are many problems concerning the lack of data recording. The estimation of unrecorded data in countries with deficiencies in hydrological information is an inevitable task. Therefore, our aim is to apply the ANN approach to predict the missing rainfall data that are important for the purposes of evaluation and prediction. Some researchers have already examined the ability of ANNs for estimation of other missing hydrological parameters such as water discharge and stage. For examples, Starrett et al. (2010) used ANNs to estimate missing data from the peak annual flow rate records of the Santa Clara River Watershed. Llunga and Stephenson (2005) studied the use of ANNs to estimate missing stream flow data in Africa where water resource data are commonly limited. Several methods that incorporate various ANNs were used to estimate missing stream flow data by Elshorbagy et al. (2002). Kalteh et al. (2009) employed and discussed imputations of missing values by means of self-organizing map, multilayer perceptron, multivariate nearest-neighbor, regularized expectation–maximization algorithm and multiple imputations in the context of a precipitation-runoff process database in northern Iran to construct a complete database. Current advances in estimation techniques to predict missing stream flow data continues to incorporate basic ANN concepts (Ng et al., 2009).

The aim of this study is to compare different structures of the ANNs that were trained to estimate missing monthly precipitation data from Ardabel plain rain gauge stations in Iran. To achieve this goal, an attempt was also made to improve the ANN model efficiency by presenting a new categorized integrated ANN model using pre-processed data via a spatial clustering method.

2. Artificial Neural Networks (ANNs)

An ANN may be described as a network of interconnected neurons (sometimes called nodes). The common structures of the ANNs consist of three layers. The first layer that connects with the input variables is named the input layer and the last layer relevant to the output variables is called the output layer. The layers in-between the input and output layers are the hidden layers, and there can be more than one hidden layer. Neurons that connect the layers have weights. The optimal set of weights is determined through learning process. The learning process consists of known input and output values, which train the network. While finding the known output using a known input, the best weights are calculated. The way that nodes in input and output layers are arranged and the direction that the data are processed in a network create networks with various characteristics. Feed-forward neural networks (FNNs) and recurrent neural networks (RNNs) are the classifications for these networks.

A FNN is an ANN where connections between the units do not form a directed cycle. In this network, the information moves in only one direction, forward, from the input nodes through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. On the other hand, RNN allows signals to travel in both directions by introducing loops in the network. RNNs are networks with one or more feedback connection. A feedback connection is used to pass the output of a neuron in a certain layer to the previous layer.

There are different mathematical learning algorithms available to train a network. In the current paper, three algorithms with two structures of ANNs have been applied. A comparison of these algorithms is available in part 5 of this paper. The three algorithms are as follows: the back-propagation (BP) algorithm, the conjugate gradient (CG) algorithm and the Levenberg-Marquardt (LM) algorithm. The BP algorithm is a common method of training that involves two phases (Werbos, 1974; Rumelhart, et al., 1986):

The Forward Phase; during this phase, the free parameters of the network are fixed, and the input signal is propagated through the network, layer by layer. The forward phase finishes with the computation of an error signal.

The Backward Phase; during the second phase, the error signal is propagated through the network in the backward direction, hence the name of the algorithm. It is during this phase that adjustments are applied to the free parameters of the network to minimize the error in a statistical sense.

There are two schemes for the BP training algorithm: gradient descent, and gradient descent with momentum (Haykin, 1994). These two schemes are often too slow for practical problems; therefore, several high-performance modified sub-schemes that can converge faster were presented. These faster algorithms fall into two categories. The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm. The second category of fast schemes uses standard numerical optimization techniques.

The CG is the most prominent iterative method for solving sparse systems of linear equations, and it is an effective method for symmetric positive definite systems. The method proceeds by generating vector sequences of iterates (i.e., successive approximations to the solution), residuals corresponding to iterates, and search directions used in updating iterates and residuals. Although the length of these sequences can become large, only a small number of vectors need to be kept in the memory. At each iteration of the method, two inner products are performed to compute update scalars that are defined to make the sequences satisfy certain orthogonality conditions. On a symmetric positive definite linear system, these conditions imply that the distance to the true solution is minimized in some norm.

The LM algorithm is the most widely used optimization algorithm. It is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real-valued functions (Levenberg, 1994). The mathematical notation for training FNNs with the LM algorithm is fully described by Hagan and Menhaj (1994). The LM algorithm has become a standard technique for nonlinear least-squares problems and could be thought of as a combination of the steepest descent and the Gauss-Newton method. The LM algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as (Hagan and Menhaj, 1994):

$$H = J^T J \tag{1}$$

and the gradient can be computed as:

$$g = J^T e \tag{2}$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases and e is a vector of the network errors. The Jacobian matrix can be computed through a standard BP technique, which is much less complex than computing the Hessian matrix. The LM algorithm uses this approximation to the Hessian matrix in the following Newton-like update (Hagan and Menhaj, 1994):

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \tag{3}$$

when the scalar μ is zero, this is simply Newton's method using the approximate Hessian matrix. When μ is large, this becomes the gradient descent method with a small step size. Newton's method is faster and more accurate near an error minimum, thus the aim is to shift toward Newton's method as quickly as possible. Therefore, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. Accordingly, the performance function is always reduced at each iteration of the algorithm.

In this study, six different structures of ANNs have been selected for comparison: FNN-BP, FNN-CG, FNN-LM, RNN-BP, RNN-CG, and RNN-LM. Among the applied neural networks, the FNN with the BP training algorithm are the most commonly used methods in solving various engineering problems (Nourani and Kalantari, 2010). Hornik et al. (1989) proved that the FNN-BP network with three-layers is satisfactory for forecasting and simulating as a general approximation. Thus, a three-layer FNN-BP network model trained using the LM optimization algorithm (Haykin, 1994) was chosen for this study. Three-layered FNNs are based on a linear combination of the input variables, which are transformed by a nonlinear activation function. Considering the authors' previous studies (e.g., Rajaee et al., 2009) and the pre-processing method, which scales the data between zero and one, Tangent Sigmoid (Tansig) was used as the activation function in this study. The application of an ANN model for predicting purposes consists of two steps. The first step is the model training, and the second is the model-testing step. In ANN modeling, two points are important and more attention must be paid to them; firstly, the ANN architecture and secondly, the training iteration number (epoch) in which the appropriate selection could progress the model efficiency in both the training and testing steps. In addition, they prevent the ANN model from being over trained. One problem that could occur when training ANNs is that the network may over-fit during the training step; the high epoch number and poor quality or quantity of data could cause this problem. Therefore, the ANN cannot adequately generalize new data outside of the training set. Using early stopping in the training algorithms could prevent this. The best overall performance for the given problem was achieved by the FNN trained with the LM algorithm, and the second best by the RNN trained with the same algorithm (Nourani et al., 2008). Another major point is to determine the number of neurons in the input and hidden layers, which provides the best training results. Because existing formulas are experimental and there is no specific mathematical algorithm to determine how many nodes are required in the hidden layers for simulating functions, the number of neurons in the hidden layer of the ANN model is usually established after many trails. However, there are some experimental formulas that can estimate the number of neurons in the hidden layers (Mogaddam et al., 2009). Fletcher et al. (1993) estimated the number of neurons in the hidden layers using the number of neurons in the input and output layers.

Considering previous studies (e.g., Rajaee et al., 2009) the optimum learning rate and the momentum coefficient were assumed to be 0.1 and 0.09, respectively. The learning rate provides the step size during the gradient descent and governs the rate at which the weights are allowed to change in any given arrangement. Higher learning rates speed the convergence process; however, higher learning rates could result in non-convergence. Slower learning rates produce more reliable results at the expense of increased training time. Generally, to assure rapid convergence, large step sizes that do not lead to oscillations are used. The momentum rate term is used to improve the convergence, which determines the effect of the previous weight changes on the present change in the weight space (Raman and Sunilkumar, 1995).

Input and output variables are usually pre-processed by scaling between zero and one to eliminate their dimensions and to ensure that all variables receive equal attention during the training of the models. The following simple linear mapping of the variables is the most common method for this purpose (Grimes et al., 2003):

$$r_i = (R_i - R_{min})/(R_{max} - R_{min}) \tag{4}$$

where, R_i is the actual value and is the respective normalized value. R_{min} and R_{max} are the minimum and maximum of all of the values used, respectively.

3. Model Precision Evaluation

In this study, three different criteria are used to measure the networks efficiency: the Root Mean Square Error (RMSE), the Error Mean (EM), and the R^2 (determination coefficient).

The Root Mean Square Error (RMSE) (Nourani, 2010) and the Error Mean (EM) (De Silva et al., 2007) are defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(I_i - \hat{I}_i)^2}{n}} \tag{5}$$

$$EM = \frac{abs(\sum_{i=1}^{n}(I_i - \bar{I}_i)^2)}{\sqrt{n}} \tag{6}$$

and R^2 (determination coefficient) is described in Equation (7), which represents the discrepancy between the computed and observed values (Nourani, 2010):

$$R^2 = \frac{\sum_{i=1}^{n}(I_i - \hat{I}_i)^2}{\sum_{i=1}^{n}(I_i - \bar{I}_i)^2} \tag{7}$$

In Equations 5, 6 and 7, n is the data number, I_i and $\bar{I}_i$ are the observed and the calculated rainfall values, respectively, and I_i s the averaged value of the observed rainfall. In analyzing the different structures of the ANN, these criteria have been used to compare the results. In the best answer for the model, R^2 and *RSME* go to one and *zero*, respectively.

Table 1. Statistics of 11 Rain Gauge Stations of Ardabel Plain

Date	Gauge Station	Rain (mm)			
		Max	Min	Mean	S.D.
From 1997 to 2005	Namin	90.5	0	22.31	20.20
	Nir	105.5	0	28.66	24.10
	Koloor	159	0	35.52	31.80
	Foladloo	101.5	0	25.76	23.65
	Bilvareg	122.1	0	35.93	27.70
	Vilkidge	101	0	28.96	24.38
	Khoshkeh	118	0	34.32	24.96
From 2001 to 2005	Kareg	76	0	19.93	22.12
	Konsol	196.6	0	48.38	36.43
	Hir	102	0	23.18	16.19
	Loron	78	0	17.34	18.76

4. Study Region and Data

In this study, the Ardabel plain was selected as the study region to estimate missing rainfall data of the relevant rain gauge stations. This plain, with an area of approximately 5,113 km^2 is one of northwest plains of Iran, and it is considered to be one of the most fertile areas with the highest precipitation. The topographic map of the study area shows that lowest elevation in the Ardabel plain is 1300 meters, and the maximum elevation in the north Vilkidge rural district is 1,845 meters above sea level. Most of the study area (85 percent) is flat plain, and the other areas are hills. This study area is located at high latitudes with unique topographic conditions, it is surrounded by mountains, and it is close to the Caspian Sea, which is source of moisture. Thus, the area has distinctive climate properties. The annual average temperature of the area is 9.5 °C, and the monthly average temperature varies from -6.9 °C in January to 25.4 °C in August. In the plain, eleven rain-gauge stations are operated to measure precipitation rainfall. The data sampling has been reported daily at all of the stations. Figure 1 shows the position of the stations. Among the eleven stations in the Ardabel plain, seven of the stations have recorded nine years of monthly rainfall data from 1997 to 2005, and the remaining four stations have recorded five years of data from 2001 to 2005, which have been used in this study. Brief statistics of the rainfall data are presented in Table 1. For instance, Figure 2 shows the monthly rainfall time series of the Koloor and Foladloo stations. The time series of the rainfall data mostly have high fluctuations, which confirm that it is difficult to employ classic estimation methods for modeling. Furthermore, this paper attempts to estimate the four years of monthly rainfall data missing (from 1997 to 2000) in the four rain stations (Hir, Loroon, Kareg, and Konsol) using the developed models.

5. Proposed Models and Results

5.1. Separate Modeling of Rain-gauge Stations

Following previous research, an ANN model was developed for each rain gauge station. Accordingly, normalized data at each station were divided into training and verification sets. Typically, about 75 percent of the data are used for the model training, and the remaining 25 percent are used for validation purposes. For seven stations, which have nine years of observed monthly rainfall data, the data from the first seven years were used to train the models, and the remaining two years of data were used for verification of the models. For the other four stations, which have only five years of data, the data from the first four years were employed for training, and one year of rainfall data was used for validation purposes. By considering different input neurons with different time delays (t, t - 1, t - 2 …) in the input layer, the ANN structure for each station was obtained (Table 2). The ANN model of each station was trained by different structures. In the current study, two structures of ANNs (i.e., FFN and RNN) were utilized. Three training algorithms were applied to these structures: the BP, CG and LM algorithms. The NNTOOL within the MATLAB software was used for network training purposes (Math Works, 2010). According to the results, the best structure for all of the networks is a three-layer feed-forward network with different input and hidden neurons. In this study, to show and compare the various architectures of the ANNs, the inputs are counted as layers per convention. The best convergence between the model's output and target was obtained using LM training algorithm.

The results show that in the ANN model for the Khoshkeh station, the monthly rainfall is auto-correlated for the four previous months. However, for the Bilevareg, Nir, Loron, and Kareg stations, this auto-regression degree is two months; for the other stations, it is only one month. These results are related with the high spatiotemporal variability in the rainfall. Spatial autocorrelation is a frequent phenomenon in ecological data because the observations from nearby locations are often more similar than what would be expected on a random basis (Legendre, 1993). This is especially true for species distribution data because they are inherently spatially structured (see, Dark, 2004; Guisan et al., 2006; Kissling et al., 2007). In the obtained structures, the maximum number of hidden layer neurons belongs to the Nir and Bilevareg stations with nine and five neurons, respectively. The output layer neurons in all networks are equal to the next month rainfall (I_{t+1}). One of the most important criteria in any ANN modeling is the training epoch number. Determining the correct number of this repetition in the network training is a crucial task. A low epoch number could cause a defect in the training, and a higher epoch number may lead to network over training. Therefore, an optimal value for the training epoch numbers has been determined so that the model performance in both the training and verification steps is suitable. In this study, different epoch numbers (30, 50, 80, and 100) have been examined in the network training as shown in Table 2. In the next step, an integrated ANN will be introduced to improve the modeling efficiency.

5.2. Integrated ANN model

In spite of temporal auto-correlation of the rainfall data for each station, which has been detected in the developed ANN models (section 5.1), it is expected that there is also a spatial correlation among the rainfall data for the stations in a desired time step. To detect this spatial relationship, an integrated ANN

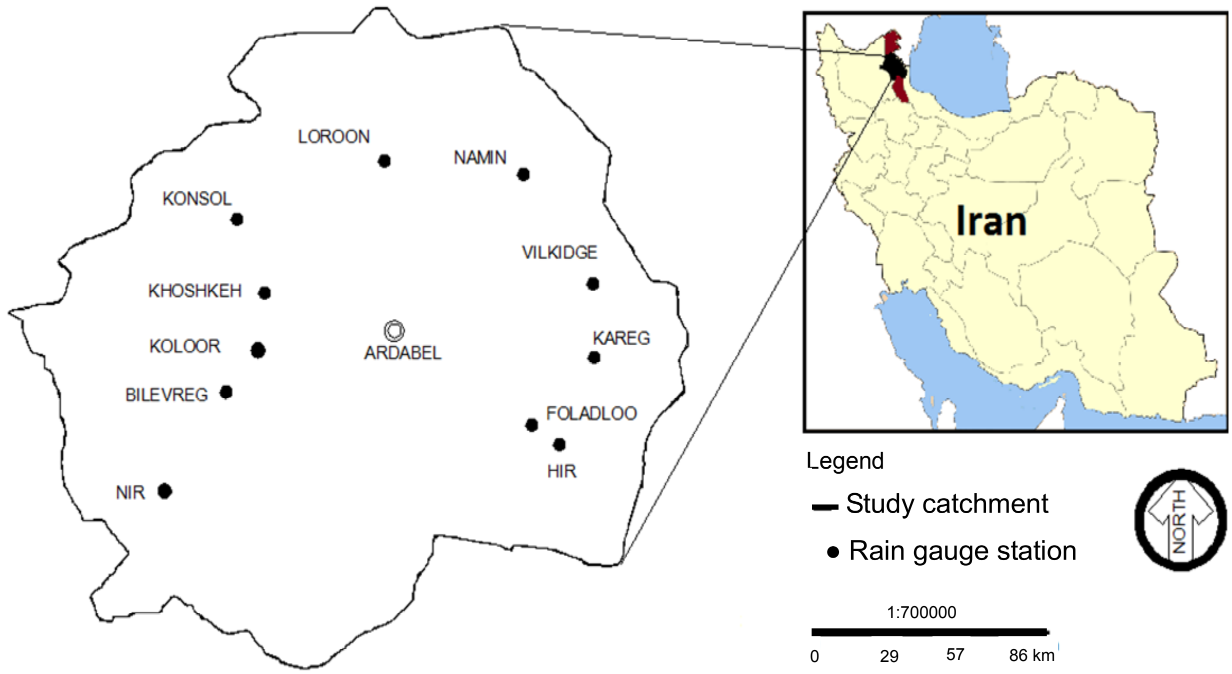

Figure 1. Map of Ardabel plain (Iran) and positions of rain gauges.

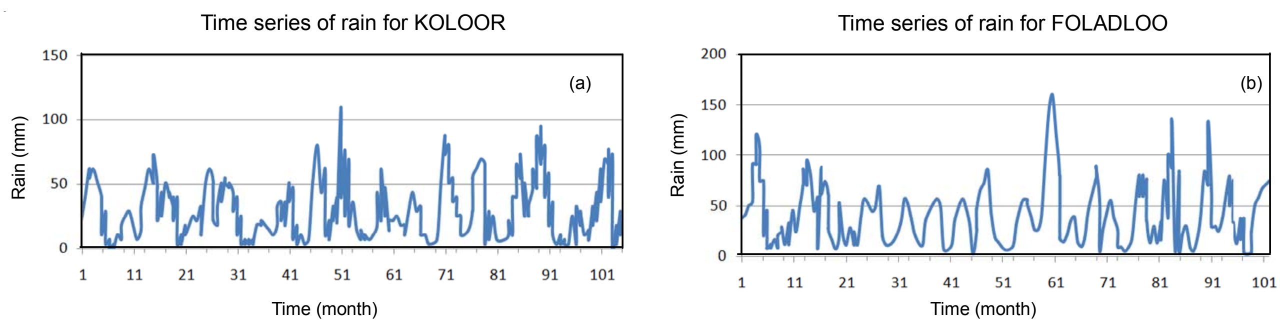

Figure 2. Time series of monthly rainfall of (a) Koloor rain gauge station, (b) Foladloo rain gauge station.

model was developed so that the data for all seven stations that have complete nine year datasets were entered into a single ANN model to predict the rainfall one month ahead at these seven stations (Foladloo, Khoshkeh, Bilevareg, Namin, Vilkidge, Nir, and Koloor). The first seven years of data were used for the model training, and the last two years of data were used for verification purposes. Figure 3 shows a schematic diagram of the proposed integrated ANN model. For each stage in Figure 3, N is equal to seven, which denotes the stations with nine years of monthly rainfall data, H is the number of neurons in the hidden layer, and *k* denotes the lag time of the stations data imposed in the input layer. The results of the integrated ANN for different structures and training epochs are presented in Table 3. According to Table 3, considerable increases in the training and verification determination coefficients demonstrates the neural network's ability when one month is lagged, and the current monthly rainfall values of all seven stations were entered to the model as input neurons. According to the results, by increasing the number of hidden neurons, the training and verification determination coefficients have not been improved considerably. The best structure obtained is (7-8-7) (which denotes a network with seven, eight and seven neurons in the input, hidden and output layers, respectively) and is trained with 50 epochs. The rainfall data of all seven stations at the current month when there are seven neurons in the input layer and the data of seven stations at the current and previous months when there are 14 input neurons are imposed in the model as dependent variables to predict simultaneously the next month rainfall data at all seven stations as independent variables in the output layer. Comparing the results in Tables 2 and 3, the superiority of the proposed integrated model over separate ANNs in monthly rainfall forecasting of the stations is demonstrated.

The main reason for this superiority may be related to the existence of spatial correlation between the different stations' data, which could be detected when all of stations' data are imposed in a single integrated ANN model. Therefore, the monthly

Table 2. Best Structure of Artificial Neural Network for Each Rain Gauge Station

Gauge Station	Training epoch	Input neurons	ANN architecture	Training				Verification			
				R^2	R^2(LR)	RMSE	EM	R^2	R^2(LR)	RMSE	EM
Bilevareg	80	I(t), I(t-1)	2-5-1	0.755	0.072	0.031	0.022	0.712	0.057	0.034	0.024
Nir	80	I(t),I(t-1)	2-9-1	0.790	0.086	0.024	0.014	0.515	0.148	0.037	0.028
Koloor	100	I(t)	1-2-1	0.599	0.112	0.036	0.029	0.593	0.002	0.037	0.028
Vilkidge	100	I(t)	1-3-1	0.652	0.088	0.042	0.036	0.636	0.001	0.043	0.035
Khoshkeh	80	I(t),I(t-1),I(t-2),I(t-3)	4-3-1	0.765	0.151	0.029	0.020	0.663	0.061	0.035	0.025
Namin	80	I(t)	1-1-1	0.609	0.000	0.038	0.031	0.534	0.016	0.042	0.034
Foladloo	100	I(t)	1-4-1	0.603	0.026	0.043	0.037	0.590	0.169	0.044	0.036
Konsol	100	I(t)	1-4-1	0.653	0.069	0.049	0.044	0.328	0.034	0.068	0.065
Hir	100	I(t)	1-1-1	0.577	0.156	0.041	0.034	0.497	0.278	0.045	0.037
Loron	100	I(t),I(t-1)	2-3-1	0.535	0.035	0.041	0.035	0.461	0.169	0.045	0.037
Kareg	80	I(t),I(t-1)	2-2-1	0.530	0.061	0.039	0.032	0.472	0.046	0.041	0.032
Average	/	/	/	0.642	0.078	0.036	0.030	0.546	0.089	0.040	0.035

Note: LR stands for Linear Regression.

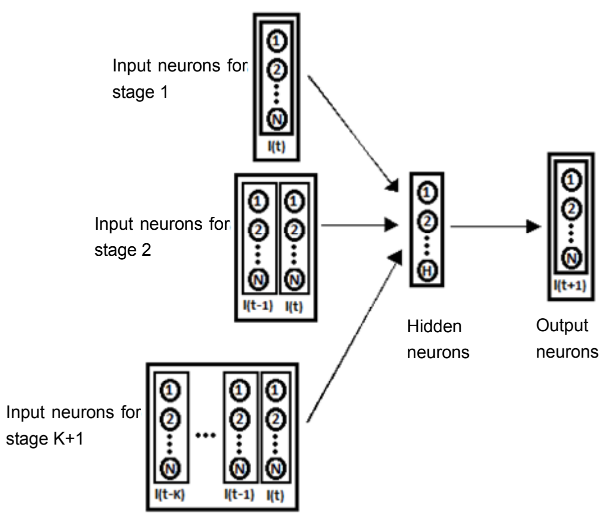

Figure 3. Schematic presentation of integrated ANN.Input neuron for stage k+1: the numbers 1 to N shows the number of stations that have been used in the input layer. I (t): represent the rainfall data of stations without any delay. (rainfall data of the current month), I (t-1): represent the rainfall data of stations with one step delay, (rainfall data of one month before the present month), I (t-k): represent the rainfall data of stations with k step delay, (rainfall data of k month before the current month).Hidden neuron: the numbers 1 to H show the number of hidden neurons that are usually established after many trails.Output neuron: the numbers 1 to N shows the number of stations that are relevant to the output values, obtained from ANN. I (t+1): represent the rainfall data of stations one month ahead, (rainfall data of the next month).

rainfall at all seven stations has been forecasted using only one integrated ANN model instead of using seven separate ANNs (one ANN for each station). Figure 4 shows the scatter diagram for the stations with observed and calculated data for both calibration and verification steps; the diagram was obtained from the proposed integrated ANN model. Similarly, because of the attention paid to the results of previous stage, another integrated ANN model was developed to estimate the four years (1997 ~ 2000) of missing monthly data for four stations (Hir, Loron, Kareg, and Konsol). Accordingly, the ANN was trained using the five years of available data from all of the stations (data from 2001 to 2005), the data from seven stations (Foladloo, Khoshkeh, Bilevareg, Namin, Vilkidge, Nir, and Kolor) as the input and the data from the remaining four stations, which have only five years of data (Hir, Loroon, Kareg, and Konsol), as the target model. Subsequently, the last four years of data (1997 ~ 2000) from the seven stations were imposed to the trained model to simulate the monthly time series for the remaining four stations, which have no observed monthly rainfall data for the years from 1997 to 2000. Therefore, the four years of missing data in the observed monthly rainfall data of the four stations have been filled by the proposed single integrated model. The modeling stages have been shown graphically in Figure 5.

In this study, as mentioned in the second section, the recurrent and the feed-forward networks with different training algorithms, i.e., CG, LM and BP (Hornik et al., 1989), were used to compare the network convergence and to provide results. Six different structures of the FNN and RNN (FNN–CG, FNN–LM, FNN–BP, RNN–CG, RNN–LM, and RNN–BP) were used; their efficiencies in the terms of the RMSE, the EM and the training and verification determination coefficients were tested, and the best structure was selected (first stage; Figure 5-a). In Table 4, the best models of the six different ANN structures have been compared. The best model is a feed-forward network trained by the LM algorithm. Subsequently, the selected structure was used to estimate the non-observed monthly rainfall data (se-

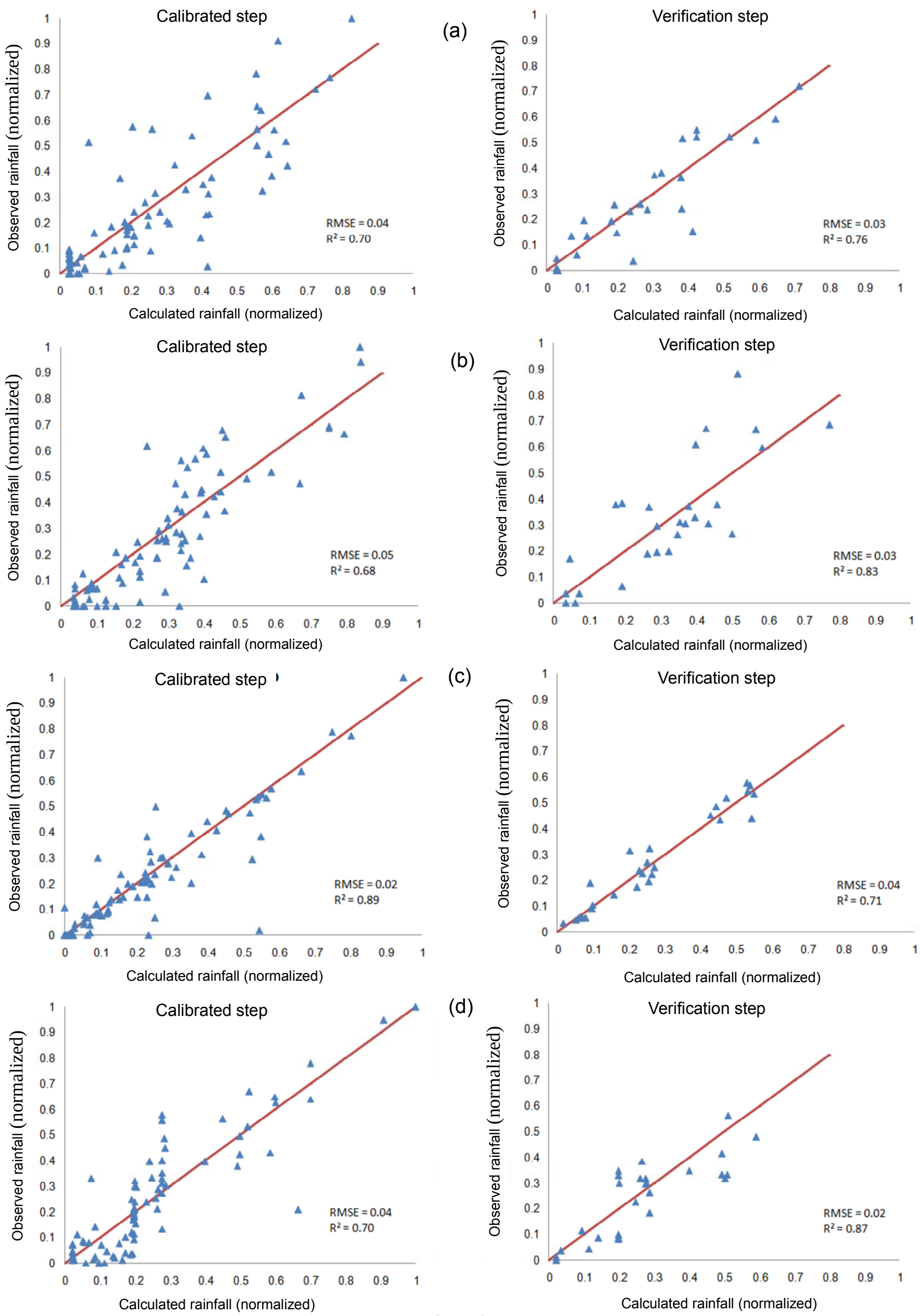

Figure 4. Scatter diagram of seven stations for training and verification rainfall data. (a) Namin; (b) Nir; (c) Koloor; (d)Foladloo.

cond stage; Figure 5-b). All structures were trained using different epoch numbers. The simulated time series of the normalized monthly rainfall for the non-observed monthly rainfall data at four rain stations (Hir, Loroon, Kareg, and Konsol) from 1997 to 2000 are presented in Figure 6.

5.3. Clustering-Based ANN Model

After estimating four years rainfall data (1997 ~ 2000) at four rain-gauge stations (Hir, Loron, Kareg, and Konsol) and acquiring the complete data set for nine years of rainfall at all eleven rain-gauge stations in the Ardabel plain, the data were

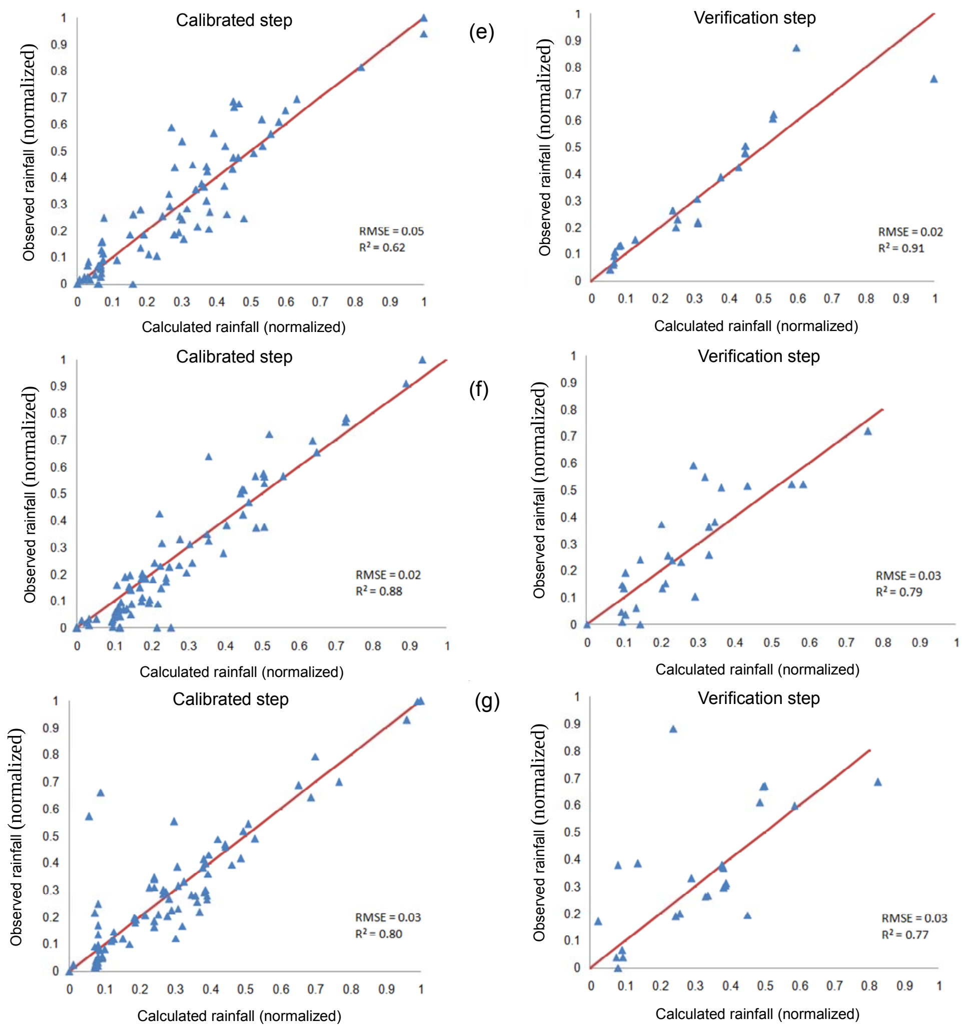

Figure 4. Scatter diagram of seven stations for training and verification rainfall data. (e) Bilvareg; (f) Vilkidge; (g) Khoshkeh.

Table 4. Comparison of Networks for ANN Modeling

Structure	FNN			RNN		
	BP	CG	LM	LM	BP	CG
Training						
R^2	0.623	0.621	0.791	0.589	0.563	0.537
RMSE	0.044	0.044	0.033	0.046	0.047	0.049
EM	0.038	0.038	0.024	0.040	0.042	0.043
Verification						
R^2	0.588	0.533	0.75	0.551	0.513	0.511
RMSE	0.046	0.049	0.036	0.048	0.050	0.050
EM	0.038	0.042	0.026	0.041	0.043	0.043
ANN architecture						
	7-1-4	7-1-4	7-1-4	7-1-4	7-1-4	7-1-4
Training epoch						
	80	100	80	80	100	80

used for the spatial clustering of the stations as a data pre-processing technique to improve the ability of the time series forecasting of the rainfall. The K-means clustering method was used via SPSS software (SPSS, 2001). The K-means function partitions data into K mutually exclusive clusters, each cluster in the partition is defined by its member objects and by its centroid, or center. The centroid for each cluster is the point at which the sum of the distances from all objects in that cluster is minimized. The K-means computes the cluster centroids differently for each distance measure to minimize the sum with respect to the measure that is specified. It uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well separated as possible (Nourani and Kalantari, 2010). The stations were divided into two clusters using the K-means clustering method. The clustering of the stations had the advantage that the relatively homogeneous stations were clustered into the same cluster. Accordingly, seven of the eleven rain-gauge stations are included in cluster 1, and the others (four stations) are included in cluster 2. With regard to Figure 7, following the station clus-

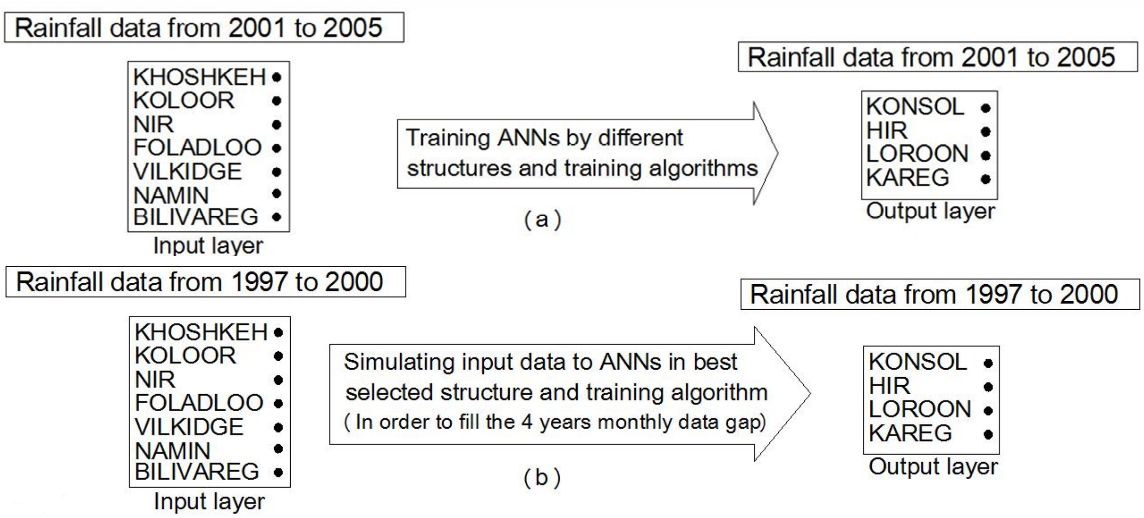

Figure 5. Schematic presentation of estimation of rainfall data gap for four rain stations from 1997 to 2000. (a) first stage: Training network by different structures and selecting best structure of artificial neural network (b) second stage: Imposing input data to ANN with best selected structure and training algorithm.

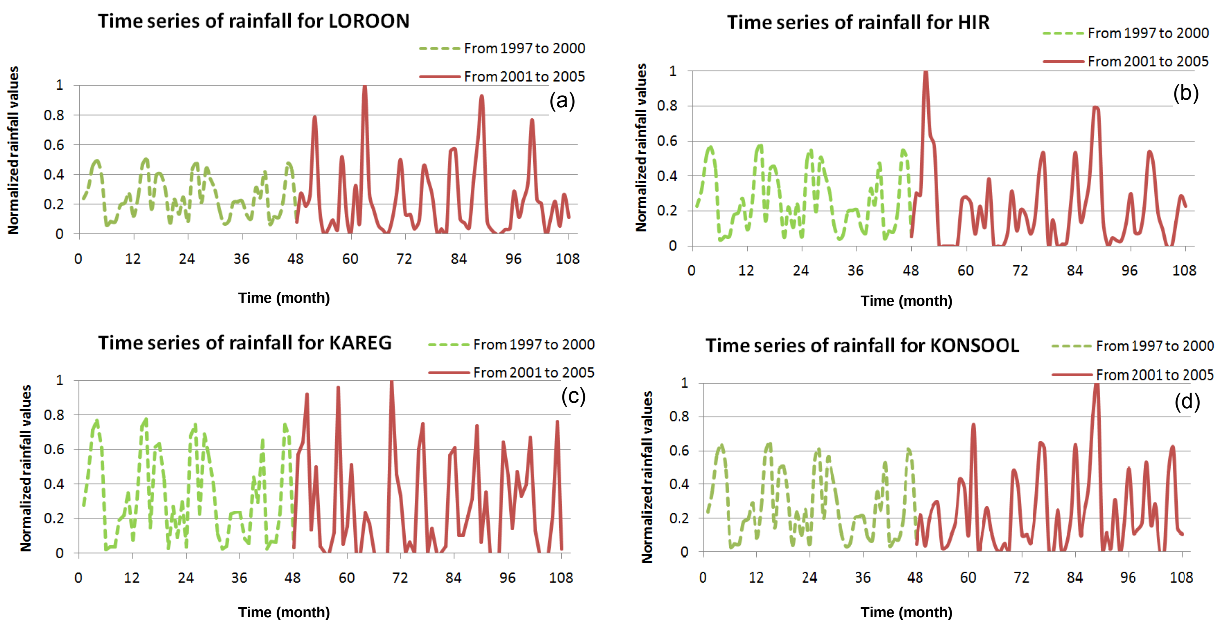

Figure 6. Time series of artificial neural networks estimation for 4 years rainfall data gap (1997-2000), also years (2001-2005) with available data, for 4 rain stations. (a) Loroon (b) Hir (c) Konsol, and (d) Kareg.

tering, which was carried out using the rainfall data from the stations, the Ardabel plain is divided into two regions, the northeast and the southwest. This division was carried out based on the station dispersion. The linear correlation coefficients of each station with the other stations were inserted in Table 5 to be compared with the clustering results. After the stations were divided into two clusters, following previously presented methodology for developing an integrated ANN model, two integrated networks were trained, one for the first cluster and another for the second cluster. The modeling results for both clusters are presented in Table 4. By comparing the results in Tables 3 and 6, which indicate the modeling before and after clustering, respectively, it is observed that the spatial pre-processing of the data is able to improve the efficiency of the neural network modeling of the stations' rainfall time series. The main reason for this progress may be related to the fact that after the station clustering, the compatible and relevant data for each cluster are input in to the related network, which could lead to results that are more reliable than when all data are input into a single network without any assortment. In the inte-

Table 5. Correlation Coefficient between the Stations

	Nir	Hir	Kareg	Vilkidge	Namin	Loron	Konsol	Koloor	Bilvareg	Foladloo	Khoshkeh
Nir		0.69	0.65	0.67	0.73	0.47	0.6	0.51	0.72	0.78	0.79
Hir	0.69		0.73	0.58	0.67	0.66	0.73	0.54	0.63	0.83	0.71
Kareg	0.65	0.73		0.56	0.70	0.56	0.71	0.47	0.69	0.83	0.66
Vilkidge	0.67	0.58	0.56		0.65	0.37	0.56	0.39	0.69	0.69	0.66
Namin	0.73	0.67	0.70	0.65		0.38	0.5	0.44	0.67	0.83	0.76
Loron	0.47	0.66	0.56	0.37	0.38		0.66	0.52	0.49	0.58	0.51
Konsol	0.60	0.73	0.71	0.56	0.5	0.66		0.46	0.73	0.70	0.70
Koloor	0.51	0.54	0.47	0.39	0.44	0.52	0.46		0.45	0.47	0.52
Bilvareg	0.72	0.63	0.69	0.69	0.67	0.49	0.73	0.45		0.72	0.81
Foladloo	0.78	0.83	0.83	0.69	0.83	0.58	0.70	0.47	0.72		0.79
Khoshkeh	0.79	0.71	0.66	0.66	0.76	0.51	0.70	0.52	0.81	0.79	

Table 6. The Artificial Neural Network Results for Clusters 1 and 2

Cluster	Training epoch	ANN Architecture	Training			Verification		
			R^2	RMSE	EM	R^2	RMSE	EM
1	100	7-7-7	0.814	0.022	0.012	0.710	0.028	0.016
2	100	4-6-4	0.792	0.026	0.016	0.716	0.029	0.019

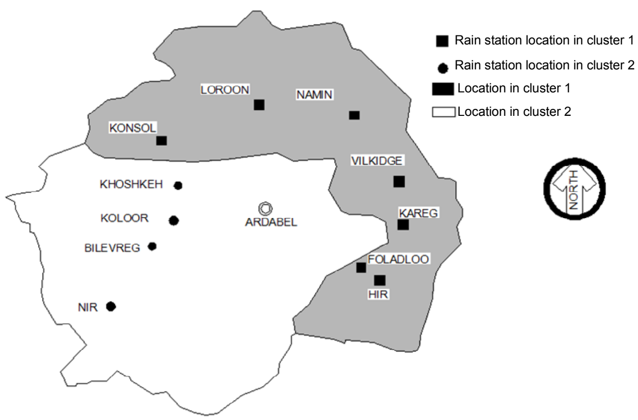

Figure 7. Clustering-based ardabel plain division.

grated ANN model, the data from the seven stations with complete observed data sets were used (Table 3). However, in Table 6, the modeling results of all 11 stations (seven in cluster one and four in cluster two) in which the data were synthetically produced and filled in the previous step are presented. Nevertheless, nearly comparable results in terms of the R^2 and RMSE indicate the quality and efficiency of the clustering process. According to the obtained result, the best-selected structure is (7-7-7) for cluster 1 and (4-6-4) for cluster 2. Moreover, the optimal training epoch number was 100 for both clusters.

6. Concluding Remarks

An ANN is utilized as an empirical model because it does not employ complicated differential formulas; it simply establishes a relation between the input and output data. Moreover, the model is less sensitive to error in the input data and processes data in parallel. In this study, the black box ANN approach was used to predict monthly rainfall time series for the Ardabel rain gauges. The modeling was performed over three stages. First, the conventional ANN model was employed so that one network was trained for each station, separately. At the second stage, an integrated ANN with all stations data as inputs was presented that was able to detect the spatial relationship among the stations rainfall data; thus, the result of this single network was more accurate than the eleven separate ANNs. Finally, at the last stage, a spatial clustering of the stations was applied as the data pre-processing method to cluster the stations into two groups according the stations rainfall data. It was concluded that this clustering could improve the ANN modeling efficiency. Furthermore, the ability of the proposed integrated network in estimating missing data was verified. In spite of the acceptable results of different structures, which have been discussed in this study, the most effective network is the FNN trained by the LM algorithm. The results of the forecasting reveal improvement while the numbers of hidden and input neurons in model are increased up to an optimal threshold.

The methodology presented herein not only is applicable in other regions with different climatic regimes, but also it could be utilized in other hydrological processes such as sediment, runoff etc. As a suggestion for future studies, and to improve the model results, in addition to the spatial pre-processing of the stations, a temporal data pre-processing (using wavelet transform, Nourani et al., 2009, 2011) may also be applied on the rainfall time series before any ANN training. Moreover, according to the uncertainty of the rainfall process and the ability of the Fuzzy concept in handling uncertainties, the conjunction of the ANN and FIS (Fuzzy Inference System) models, as an ANFIS (Adaptive Neural-Fuzzy Inference system) model could be a reliable choice for the model progress.

References

Aksoy, H., and Dahamsheh, A. (2009). Artificial neural network models for forecasting intermittent monthly precipitation in arid regions. *Meteorol. Appl.*, 16(3), 325-337. http://dx.doi.org/10.1002/met.127

ASCE Task Committee on the application of ANN in hydrology

(2000). Artificial neural networks in hydrology, II: hydrologic applications. *J. Hydrol. Eng.*, 5(2), 124-137.

Bodri, L., and Cermak, V. (2000). Prediction of extreme precipitation using a neural network: application to summer flood occurrence in Moravia. *Adv. Eng. Software*, 31(5), 311-321. http://dx.doi.org/10.1016/S0965-9978(99)00063-0

Chattopadhyay, G., and Chattopadhyay, S. (2008). Comparative study among different neural net learning algorithms applied to rainfall time series. *Meteorol. Appl.*, 15(2), 273-280. http://dx.doi.org/10.1002/met.71

Chattopadhyay, S. (2007). Feed-forward Artificial Neural Network model to predict the average summer-monsoon rainfall in India. *Acta Geophys.*, 55(3), 369-382. http://dx.doi.org/10.2478/s11600-007-0020-8

Chau, K.W. (2006). Particle swarm optimization training algorithm for ANN in stage prediction of Shing Mun River. *J. Hydrol.*, 329 (3-4), 363-367. http://dx.doi.org/10.1016/j.jhydrol.2006.02.025

Chau, K.W., and Cheng, C.T. (2002). Real-time prediction of water stage with artificial neural network approach. Advances in Artificial Intelligence, *Lect. Notes Contr. Inf.*, 2557, 715. http://dx.doi.org/10.1007/3-540-36187-1_64

Dark, S.J. (2004). The biogeography of invasive alien plants in California: an application of GIS and spatial regression analysis. *Divers. Distrib.*, 10(1), 1-9. http://dx.doi.org/10.1111/j.1472-4642.2004.00054.x

De Silva, R.P., Dayawansa N.D.K., and Ratnasiri, M.D. (2007). A Comparison of methods used in estimating missing rainfall data. *J. Agric.Sci.*, 3(2), 101-108.

Elshorbagy, A., Simonovic, and S.P., Panu, U.S. (2002).Estimation of missing stream flow data using principles of chaos theory. *J. Hydrol.*, 255(1), 123-133. http://dx.doi.org/10.1016/S0022-1694(01)00513-3

Fletcher, D., and Goss, E. (1993). Forecasting with neural networks: An application using bankruptcy data. *Inform. Manage.*, 24(3), 159- 167. http://dx.doi.org/10.1016/0378-7206(93)90064-Z

French, M., Krajewski, W., and Cuykendall, R.R. (1992). Rainfall forecasting in space and time using a neural network. *J. Hydrol.*, 137(1-4), 1-31. http://dx.doi.org/10.1016/0022-1694(92)90046-X

Greischar, L., Hastenrath, S., and Heerden, J.V. (1995). Prediction of the summer rainfall over South Africa. *J. Clim.*, 8(6), 1511-1518. http://dx.doi.org/10.1175/1520-0442(1995)008<1511:POTSRO>2.0.CO;2

Grimes, D.I.F., Coppola, E., Verdecchia, M., and Visconti, G. (2003). A neural network approach to real-time rainfall estimation for Africa using satellite data. *Journal of Hydrometeorology*. 4(6), 1119-1133. http://dx.doi.org/10.1175/1525-7541(2003)004<1119:ANNA TR >2.0.CO;2

Guisan, A., Lehmann, A., Ferrier, S., Austin, M., Overton, J.Mc.C., Aspinall, R., and Hastie, T. (2006). Making better biogeographical predictions of species' distributions. *J. Appl. Ecol.*, 43(3), 386-392. http://dx.doi.org/10.1111/j.1365-2664.2006.01190.x

Hagan, M.T., and Menhaj M.B. (1994). Training feed-forward networks with the Marquardt algorithm. *IEEE T. Neural Network.*, 5(6), 989-993. http://dx.doi.org/10.1109/72.329697

Haykin, S. (1994). *Neural network, a comprehensive foundation*. MacMillan College Publishing Co. New York.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*. 2(5), 359-366. http://dx.doi.org/10.1016/0893-6080(89)90020-8

Hung, N.Q., Babel, M.S., Weesakul, S., and Tripathi, N.K. (2009). An artificial neural network model for rainfall forecasting in Bangkok, Thailand. *Hydrol. Earth Syst. Sci.*, 13, 1413-1425. http://dx.doi.org/10.5194/hess-13-1413-2009

Kalteh, A.M., and Hjorth, P. (2009). Imputation of missing values in a precipitation-runoff process database. *Hydrol. Res.* 40(4), 420-432. http://dx.doi.org/10.2166/nh.2009.001

Karamoz, M., and Araginejad, S. (2006). *Advanced hydrology*. First edition, Amirkabir University of Technology (Tehran Polytechnic) aut.ac.ir publications, 313-351(in Persian).

Kim, J.W., and Pachepsky, Y.A. (2010). Reconstructing missing daily precipitation data using regression trees and artificial neural networks for SWAT stream flow simulation. *J. Hydrol.*, 394(3-4), 305-314. http://dx.doi.org/10.1016/j.jhydrol.2010.09.005

Kissling, W.D., Rahbek, C.,and Böhning-Gaese, K. (2007). Food plant diversity as broad-scale determinant of avian frugivore richness. *Proc. R. Soc. B.*, 274(1611), 799-808. http://dx.doi.org/10.1098/rs pb.2006.0311

Kuligowski, R.J., and Barros, A.P. (1998). Experiments in short-term precipitation forecasting using artificial neural networks. *Mon. Weather Rev.*, 126(2), 470-482. http://dx.doi.org/10.1175/1520-0493(1998)126 <0470:EISTPF>2.0.CO;2

Legendre, P. (1993). Spatial autocorrelation: trouble or new paradigm? *Ecology*, 74(6), 1659-1673. http://dx.doi.org/10.2307/1939924

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.*, 2(2), 164-168.

Llunga, M., and Stephenson, D. (2005). Infilling stream flow data using feed-forward back-propagation (BP) artificial neural networks: Application of standard BP and pseudo Mac Laurin power series BP techniques. *Water SA*, 31(2), 171-176.

Luk, K.C., Ball, J.E., and Sharama, A. (2001). An application of artificial neural networks for rainfall forecasting. *Math. Comput. Model.*, 33(6-7), 683-693. http://dx.doi.org/10.1016/S0895-7177(00)00272-7

Marzban, C. (2002). Neural network short course, Amer. *Meteor. Soc.*, Available at http://www.nhn.ou.edu/~marzban.

Marzban, C., and Witt, A. (2001). A Bayesian neural network for hail size prediction. *Weather Forecast.*, 16(5), 600-610. http://dx.doi.org/10.1175/1520- 0434(2001)016<0600:ABNNFS>2.0.CO;2

Math Works, Inc. (2010). *MATLAB: Neural Networks Toolbox (NNTOOL) User's Guide*, Version 7, The Math works, Inc., Natick.

McCann, D.W. (1992). A neural network short-term forecast of significant thunderstorms. *Weather Forecast.*, 7(3), 525-534. http://dx.doi.org/10.1175/1520-0434(1992)007<0525:ANNSTF>2.0.CO;2

Mogaddam, A.A., Nourani, V., and Nadiri, A. (2009). Precipitation modeling of Tabriz plain by artificial neural network. *Agriculture Sciences*. 18, 1-15(in Persian with English abstract).

Ng, W.W., Panu, U.S., and Lennox, W.C. (2009).Comparative studies in problems of missing extreme daily streamflowrecords. *J. Hydrol. Eng.*, 14(1), 91-100. http://dx.doi.org/10.1061/(ASCE)1084-0699(2009)14:1(91)

Nourani, V. (2010). Reply to comment on' Nourani V, Mogaddam A.A , Nadiri A. 2008' An ANN-based model for spatiotemporal groundwater level forecasting. *Hydrol. Process.*, 24(3), 370-371. http://dx.doi.org/10.1002/hyp.7469

Nourani, V., and Mano, A. (2007). Semi-distributed flood runoff model at the sub continental scale for southwestern Iran. *Hydrol. Process.*, 21, 3173-3180. http://dx.doi.org/10.1002/hyp.6549

Nourani, V., and Kalantari, O. (2010). An integrated artificial neural network for spatiotemporal modeling of rainfall-runoff-sediment process. *Environ. Eng. Sci.*, 27(5), 411-422. http://dx.doi.org/10.1089/ees.2009.0353

Nourani, V., Alami, M.T., and Aminfar, M.H. (2009). A combined neural-wavelet model for prediction of Ligvanchay watershed precipitation. *Eng. Appl. Artif. Intel.*, 22(3), 466-477. http://dx.doi.org/10.1016/j.engappai.2008.09.003

Nourani, V., Mogaddam, A.A., and Nadiri, A. (2008). An ANN-based model for spatiotemporal groundwater level forecasting. *Hydrol. Process.*, 22(26), 5054-5066. http://dx.doi.org/10.1002/hyp.7129

Nourani, V., Kisi, O., and Komasi, M. (2011). Two hybrid artificial intelligence approaches for modeling rainfall-runoff process. *J. Hydrol.*, 402(1-2), 41-59. http://dx.doi.org/10.1016/j.jhydrol.2011.03.002

Rajaee, T., Mirbagheri, S.A., Zounemat-Kermani, M., and Nourani, V. (2009). Daily suspended sediment concentration simulation using ANN and neuro-fuzzy models. *Sci. Total Environ.*, 407(17), 4916-4927. http://dx.doi.org/10.1016/j.scitotenv.2009.05.016

Raman, H., and Sunilkumar N. (1995). Multivariate modelling of water resources time series using artificial neural networks. *Hydrol. Sci. J.*, 40(2),145-163. http://dx.doi.org/10.1080/02626669509491401

Ramesh, S.V., Teegavarapu, R.S.V., and Chandramouli, V. (2005). Improved weighting methods, deterministic and stochastic data-driven models for estimation of missing precipitation records. *J. Hydrol.*, 312(1-4), 191-206. http://dx.doi.org/10.1016/j.jhydrol.2005.02.015

Ramirez, M.C.V., Velho, H.F.C., and Ferreira, N.J. (2005). Artificial neural network technique for rainfall forecasting applied to the Sao Paulo region. *J. Hydrol.*, 301(1-4), 146-162. http://dx.doi.org/10.1016/j.jhydrol.2004.06.028

Rummelhart, D.E., and McClelland, J.L. (1986). *Parallel distributed processing*. Exploration in the microstructure of cognition, Foundation, 1, MIT Press.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). *Learning Internal Representations by Error Propagation*, in D. E. Rumelhart and J. L. McCleland, eds. (Cambridge, MA: MIT Press), Vol. 1, Chapter 8.

Silverman, D., and Dracup, J.A. (2000). Artificial neural networks and long range precipitation prediction in California. *J. Appl. Meteorol.*, 39(1), 57-66. http://dx.doi.org/10.1175/1520-0450(2000)039<0057:ANNALR>2.0.CO;2

Singh, V.P., and Chowdhury, K. (1986). Comparing some methods of estimating mean areal rainfall. *Water Resour. Bull.*, 22(2), 275-282. http://dx.doi.org/10.1111/j.1752-1688.1986.tb01884.x

SPSS, Inc. (2001). *SPSS for Windows User's Guide*, Version 11, Englewood Cliffs, Prentice Hall.

Starrett, S.K., Starrett, S.K., Heier, T., Su, Y., Tuan, D., and Bandurraga, M. (2010). Filling in missing peak flow data using artificial neural networks. *J. Eng. Appl. Sci.*, 5(1), 49-55.

Toth, E., Brath, A., and Montanari, V. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *J. Hydrol.*, 239(1-4), 132-147. http://dx.doi.org/10.1016/S0022-1694(00)00344-9

Werbos, P.J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. Dissertation, Harvard University, Cambridge, MA., USA.

Wu, C.L., Chau, K.W., and Fan, C. (2010). Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. *J. Hydrol.*, 389(1-2), 146-167. http://dx.doi.org/10.1016/j.jhydrol.2010.05.040