Available online at www.CivileJournal.org

Civil Engineering Journal

Vol. 3, No. 2, February, 2017

# Modeling of Rainfall-Runoff Correlations Using Artificial Neural Network-A Case Study of Dharoi Watershed of a Sabarmati River Basin, India

Ajay B. Patel [a*], Geeta S. Joshi [b]

[a] *Faculty of Technology and Engineering, Maharaja Sayajirao University of Baroda, Gujarat, India.*

[b] *Associate Professor, Faculty of Technology and Engineering, Maharaja Sayajirao University of Baroda, Gujarat, India.*

Received 31 December 2016; Accepted 10 February 2017

**Abstract**

The use of an Artificial Neural Network (ANN) is becoming common due to its ability to analyse complex nonlinear events. An ANN has a flexible, convenient and easy mathematical structure to identify the nonlinear relationships between input and output data sets. This capability could efficiently be employed for the different hydrological models such as rainfall-runoff models, which are inherently nonlinear in nature. Artificial Neural Networks (ANN) can be used in cases where the available data is limited. The present work involves the development of an ANN model using Feed-Forward Back Propagation algorithm for establishing monthly and annual rainfall runoff correlations. The hydrologic variables used were monthly and annual rainfall and runoff for monthly and annual time period of monsoon season. The ANN model developed in this study is applied to Dharoi reservoir watersheds of Sabarmati river basin of India. The hydrologic data were available for twenty-nine years at Dharoi station at Dharoi dam project. The model results yielding into the least error is recommended for simulating the rainfall-runoff characteristics of the watersheds. The obtained results can help the water resource managers to operate the reservoir properly in the case of extreme events such as flooding and drought.

*Keywords:* Artificial Neural Networks (ANN); Feed-Forward Back Propagation Algorithm; Rainfall-Runoff Modeling.

## 1. Introduction

The rainfall-runoff relationship is one the most complex hydrological phenomenon due to the tremendous spatial and temporal variability of watershed characteristics and rainfall patterns as well as a number of variables involved in the physical processes [1]. Also, this process is non-linear in nature and thus difficult to arrive at solutions. The runoff needs to be estimated for efficient utilization of water resources. The rainfall-runoff models play a significant role in water resource management, planning and hydraulic design [2]. The study on rainfall-runoff relationship also helps in planning and developing distribution policies from the available water resources [3]. Evaluating this process with accuracy is what allows rational management of the different water uses, such as: supply, irrigation, electric power generation, to forecast extreme flood events and dry periods, to generate scenarios of streamflow from precipitation scenarios resulting from climate change and others [4]. Generally mathematical models known as rainfall-runoff models perform the evaluation of this process. Rainfall-runoff models are divided into two major groups: conceptual and empirical models. The conceptual models describe mathematically the processes of the hydrologic cycle based on physical laws governing each of these processes [5]. However, despite generally good results are achieved, some aspects of the conceptual models are challenging. Calibration is not easy and in many cases, depends on field surveys of data often not available. Also, the use of basin averages for relevant parameters together with the nonlinear character of those processes leads to additional difficulties [6]. These characteristics often render the implementation

* Corresponding author: aajay1982@gmail.com

of conceptual model difficult and financially burdensome. Empirical models are an alternative to the conceptual models. The main characteristic of this type of model consists of establishing a stable relationship between input and output variables without accounting to the physical laws that govern the natural processes when rainfall is transformed into runoff. These models are easy to apply and supposedly cheaper. Examples of these models are multivariable equations with parameters estimated by Artificial Neural Networks ANNs [7].

The Artificial Neural Network (ANN) approach is extensively used in the water resources [8]. In this study, the Feed Forward Back Propagation method (FFBP) was employed to train the neural networks. As well known the FFBP algorithm has some drawbacks. It is very sensitive to the selected initial weight values and may provide performances differing from each other significantly. Another problem faced during the application of FFBP is the local minima issue. The distinct advantage of an ANN is that it learns the previously unknown relationship existing between the input and the output data through a process of training, without a prior knowledge of the catchment characteristics [9]. The ANN is also described as a mathematical structure, which is capable of representing the arbitrary complex nonlinear process relating the input and the output of any system [10]. The ANN model developed in this study is applied to Dharoi reservoir watershed of Sabarmati river basin of India. The hydrologic data were available for twenty-nine years at Dharoi station at Dharoi dam project. The model results yielding into the least error is recommended for simulating the rainfall-runoff characteristics of the watersheds [11]. The nonlinear nature of the relationships, availability of long historical records, and the complexity of the physical based models in this regard are some of the factors that have attracted researchers to consider alternative models in which, ANNs have been a one of the viable alternative choice [12].

## 2. Materials and Methods

### 2.1. Study Area and Data Collection

Sabarmati river is one of the major west flowing rivers of India. The Sabarmati basin extends over the states of Rajasthan and Gujarat having an area of 21,674 Sq. km with maximum length and width of 300 km and 150 km respectively. It lies between $70°58'$ to $73°51'$ east and $22°15'$ to $24°47'$ north. The basin is bounded by Aravalli hills in the north and north-east, Rann of Kutch in the west and Gulf of Khambhat in the south. The Sabarmati basin extends over parts of Udaipur, Sirohi, Pali and Dungarpur districts of Rajasthan, Sabarkantha, Kheda, Ahmedabad, Mahesana, Gandhinagar and Banaskantha districts of Gujarat. In Gujarat, the basin occupies an area of 17,550 $Sq.km$ accounting to 81% of the total basin area. In Rajasthan, it covers an area of 4,124 $Sq.km$ which accounts for 19% of the total basin area.

The basin is divided into 2 sub-basins as Sabarmati upper and Sabarmati lower sub-basin. They have been further clustered into 51 watersheds each of which represents a different tributary system. Sabarmati originates from Aravalli hills at an elevation of 762 $m$ near village Tepur in Udaipur district of Rajasthan. The total length of river from origin to outfall into the Arabian Sea is 371 $km$. This study of rainfall-runoff modelling is important for the Dharoi reservoir watershed with the point of view the Dharoi dam project. Figure 1. shows the index map for Sabarmati river basin and Dharoi reservoir watershed map.

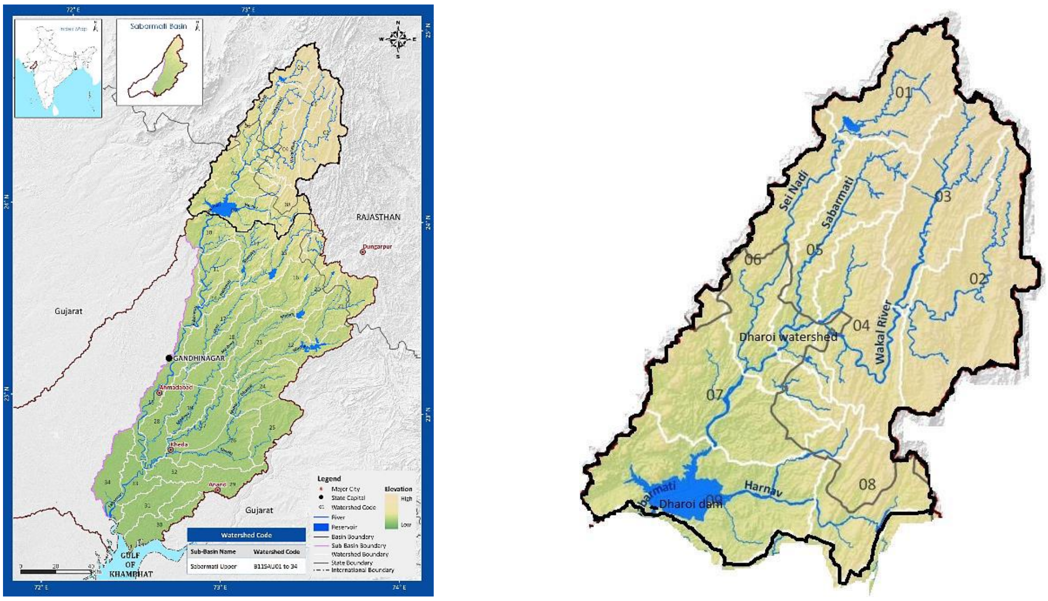


**Figure 1. Location of study area, Sabarmati river basin and Dharoi Watershed (**http://www.india-wris.nrsc.gov.in/wris.html**)**

In India, there is a wide timely variability of rainfall. The rainfall is occurring only during monsoon season that is from month June to October. The rainfall-runoff correlation also varying widely month to month and the rule level for reservoir operation is also decided monthly. So, there is a need to establish the monthly rainfall-runoff model. Dharoi dam is located on Sabarmati River near village Dharoi in Kheralu taluka of district Mehsana, 103 km from the source of the river.

The rainfall and runoff data have been collected from 1986 to 2014 at Dharoi dam station in Dharoi watershed.

## 2.2. Artificial Neural Networks Procedure

An ANN is a structure of elements formed by nodes or neurons, similar to the structure of the human brain, mathematically interconnected, representing a function. The coefficients and intercepts of the input variables of this function are called weights and biases. One major application of ANN in hydrology has been related to streamflow or rainfall forecasting [13, 14]. Artificial Neural Networks employ a mathematical simulation approach, which adopts a biological system in order to process the acquired information and derive the output after the network has been trained properly for pattern recognition. The main theme of ANN model is, it considers the brain as a parallel computational device for various computational tasks that were performed relatively poor by traditional serial computers [15, 16]. The neural network structure in the present study possessed adaptation of three-layer learning network consisting of an input layer, a hidden layer and an output layer consisting of output variable as shown in Figure 2. The input nodes pass on the input signal values to the nodes in the hidden layer unprocessed [17]. The values are distributed to all the nodes in the hidden layer depending on the connection weights Wij and Wjk. Where Wij and Wjk are the weights between the input node and the hidden nodes and the weights between the hidden nodes and the output nodes respectively. Connection weights are the interconnecting links between the neurons in successive layers [18, 19]. Each neuron in a certain layer is connected to every single neuron in the next layer by links having an appropriate and an adjustable connection weight.

In the present study, the Feed Forward Back Propagation (FFBP) algorithm was used for training using Levenberg–Marquardt optimization technique. This optimization technique is reported to be more powerful than the conventional gradient descent techniques [20, 21]. The study showed that the Marquardt algorithm is very efficient when training networks which has few hundred weights. Although the computational requirements are much higher in iterations of the Marquardt algorithm. Its efficiency is higher. This is especially true, when high precision is required. The Feed Forward Back Propagation (FFBP) distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units [22, 23]. The function of hidden neurons is to intervene between the external input and the network output in useful manner. Figure 2. shows the architecture of the Artificial Neural Network Model with the Feed Forward Back Propagation (FFBP) algorithm.

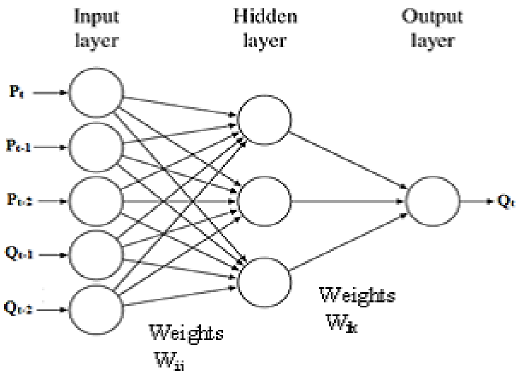


**Figure 2. Architecture of the Neural Network Model [35]**

The neurons go through an activation function to generate the result. The system, therefore, needs continuous transfer functions in order to determine the output of neurons based on its input. This transfer function is a continuous, differential and monotonically increasing function, which is typically employed in back propagation network [24, 25]. Later, the signal transmits from the second to third layer and the error is transmitted from the output layer back to the earlier layers. This process is called back propagation because the output error goes back to the input nodes in order to revise the weights.

## 2.3. Model development

The input layer comprised one layers (rainfall) and the runoff constituted the output layer. The whole computation was performed by using MATLAB capability of develop ANN by using nntoolbox. In the present study, the Feed Forward Back Propagation (FFBP) algorithm was used for training using Levenberg–Marquardt optimization technique. In this paper, trial and error method has been applied and from the total number of input data set, 70% have been used as training data set, while 15% have been used as testing data set and 15% have been used as validation dataset. Initially network is created with 1 hidden layer, as it has not given the desired output, so network is trained with 2-hidden layers.

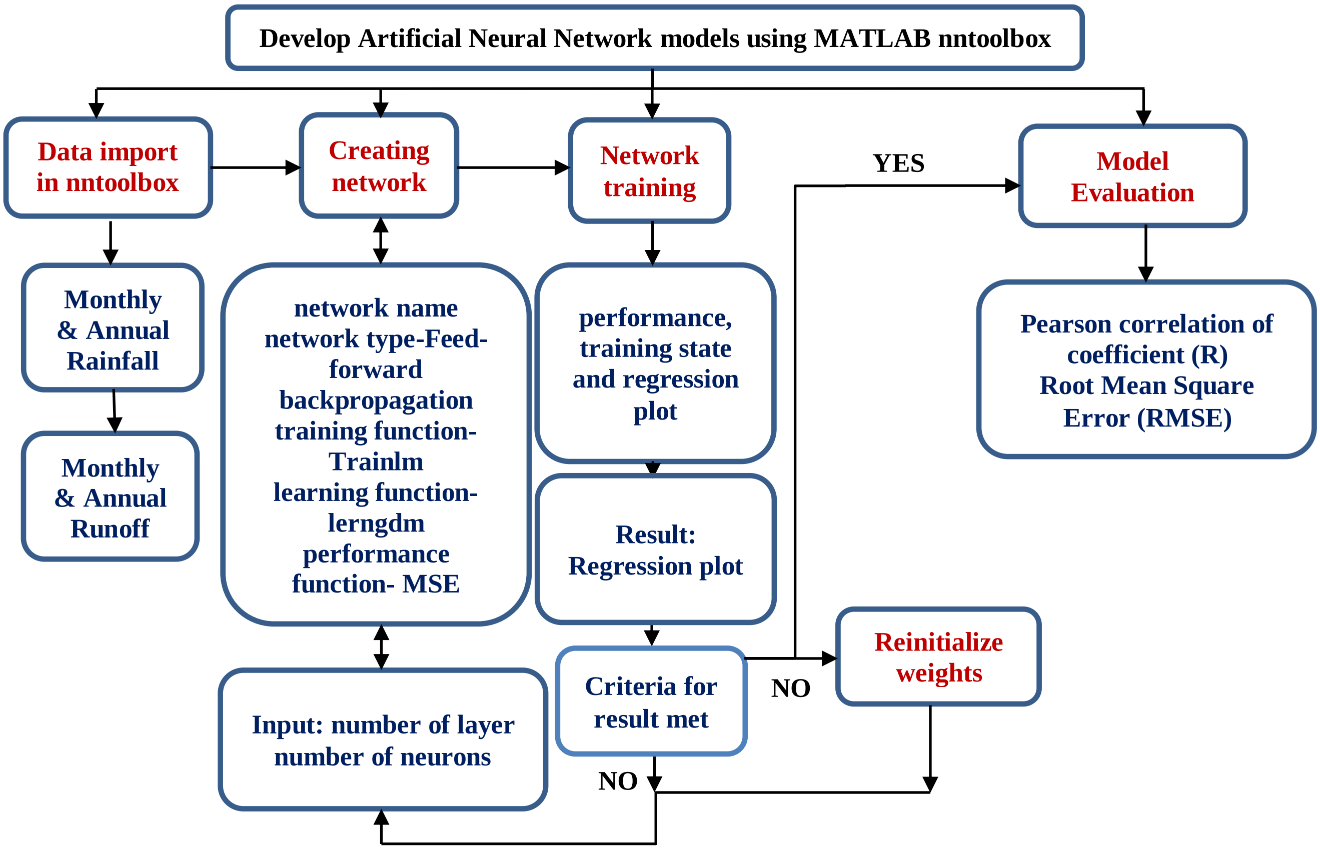

Develop Artificial Neural Network models using MATLAB nntoolbox
Data import in nntoolbox
Creating network
Network training
YES
Model Evaluation
Monthly & Annual Rainfall
network name network type-Feed-forward backpropagation training function-Trainlm learning function-lerngdm performance function- MSE
performance, training state and regression plot
Pearson correlation of coefficient (R) Root Mean Square Error (RMSE)
Monthly & Annual Runoff
Result: Regression plot
Reinitialize weights
Criteria for result met
NO
Input: number of layer number of neurons
NO

**Figure 3. Flowchart of Methodology**

Another factor, which is one of the most significant characteristics of ANN, is the number of neurons in the hidden layers. If the numbers of neurons are insufficient, the network cannot configure the complex data set and the obtained results will be a poorly fit. Conversely, if the number of neurons is too high, the time required for network training will be long and the network might over-fit the data. In the present research, the number of neurons was determined by number of trials in nntoolbox. Initially in nntoolbox numbers of neurons are taken as 10 and the weight are also considered by default according to input data. But by trial and error, the numbers of neurons are obtained according to the desirable accuracy. The best result was obtained for 18 to 68 neurons in the present study for various months modelling. The present study used different number of neurons for different data as shown in Table 2 and 3. In the present study tansig function is used as transfer function, TRAINLM function is used as training function, LEARNGDM function is used as adaption learning function and MSE is used as performance function for the input and target values that are better suiting to output values for network training.

### 2.3.1. ANN Learning Process

The learning process initializes the network weights and biases and also prepares the network to do some task. There is a bias connected to each layer, the input is connected to layer 1, and the output comes from layer 2. Also, layer 1 is connected to layer 2. The train command will automatically configure the network and initialize the weights in nntoolbox but that may be required to reinitialize. When the network weights and biases are initialized, the network is ready for training [26].

### 2.3.2. ANN Training Procedure

This topic describes two different styles of training. In incremental training the weights and biases of the network are updated each time an input is presented to the network. In batch training the weights and biases are only updated after all the inputs are presented [27]. The batch training methods are generally more efficient in the MATLAB environment, and they are emphasized in the Neural Network Toolbox software. In the present study, the batch training method is used in nntoolbox. In nntoolbox the 'Show Window parameter' allows to specify whether a training window is visible when it is trained. The training window appears by default. Two other parameters, ‘Show Command Line and Show’ determine, whether command-line output is generated and the number of epochs between command-line feedbacks during training. Networks can use the tan-sigmoid transfer function (tansig). The feedforward neural network is the workhorse of the Neural Network Toolbox software. It can be used for both function fitting and pattern recognition problems [28, 29].

When training the multilayer networks, the general practice is to first divide the data into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. The network weights and biases are saved at the minimum of the validation set error. The default ratios for training, testing and validation are 0.7, 0.15 and 0.15, respectively. In present study for all network used default ratio. For instance, the first argument is an array containing the number of neurons in each hidden layer. The default setting is 10. The second argument contains the name of the training function to be used. If no arguments are supplied, the default number of layers is 2, the default number of neurons in the hidden layer is 10, and the default training function is trainlm. As number of hidden layer and neurons are set according to output accuracy required. The error is minimized by trial and error method [30]. The default transfer function for hidden layers is tansig. The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance.

The fastest training function is generally trainlm, and it is the default training function for feed forward network [31]. Also, trainlm performs better on function fitting (nonlinear regression) problems. The training window will appear during training. This window shows that the data has been divided using the dividerand function, and the Levenberg-Marquardt (trainlm) training method has been used with the mean square error performance function. Recall that these are the default settings for Feed Forward Back Propagation network.

During training, the progress has been constantly updated in the training window. The magnitude of the gradient and the number of validation checks are used to terminate the training. The gradient became very small as the training reaches a minimum of the performance. If the magnitude of the gradient is less than 1e-5, the training will stop by default. In present study, this limit was adjusted by setting the gradient parameter for various monthly models as 1e-50 to minimize the error. The number of validation checks represents the number of successive iterations that the validation performance fails. By default, this number reaches 6 (the default value), the training is stopped. In this study, the number of validation checks has been input varying for various monthly models and annual model varying from 600 to 3000. There are other criteria that have been used to stop network training, and on meeting any of the below mentioned criteria, the training will stop. They are listed as,

| **Parameter** | **Stopping Criteria** |
|---|---|
| min_grad | Minimum Gradient Magnitude |
| max_fail | Maximum Number of Validation Increases |
| time | Maximum Training Time |
| goal | Minimum Performance Value |
| epochs | Maximum Number of Training Epochs (Iterations) |

From the training window, three plots were accessed: performance, training state and regression. The performance plot shows the value of the performance function versus the iteration number. It plots training, validation and test performances. The training state plot shows the progress of other training variables, such as the gradient magnitude, the number of validation checks, etc. The regression plot shows a regression between network outputs and network targets. Regression plots were used to validate network performance [32].

### 2.3.3. Analysis of Results

The sensitivity analysis has been carried out for different number of neurons and hidden layers and the results for parameter correlation coefficient R between observed value of runoff and predicted value of runoff is shown in Table 1.

**Table 1. Model sensitivity analysis for August month ANN model**

| Network Architecture | No. of neurons | Correlation Factor (R) |
|---|---|---|
| 1−1−1 | 10 | 0.840 |
| 1−1−1 | 20 | 0.827 |
| 1−2−1 | 10 | 0.945 |
| 1−2−1 | 30 | 0.950 |
| 1−2−1 | 45 | 0.948 |
| 1−2−1 | 68 | 0.999 |
| 1−2−1 | 70 | 0.951 |

### 2.4. Model Evaluation

The performance of the developed model was evaluated by statistical evaluation measurements, such as Pearson correlation of coefficient (R) of observed and simulated runoff and Root Mean Square Error (RMSE) [33, 34]. RMSE is statistics evaluate the efficiency of the model in terms of its ability to predict data from a calibrated model. The other statistics R quantifies the effect of the ANN model in capturing the dynamic, complex and nonlinear rainfall-runoff processing as the correlation coefficient (R-value) between the outputs and targets [35].

It is a measure of how well the variation in the output is explained by the targets. If this number is equal to 1, then there is perfect correlation between targets and outputs. These statistical criteria are calculated according to the following equations by nntoolbox of MATLAB environment,

Correlation Coefficient (R):

$$R = \frac{\sum_{i=1}^{n}(X_i - \overline{X_\iota}).(Y_i - \overline{Y_\iota})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X_\iota})^2 .(Y_i - \overline{Y_\iota})^2}} \tag{1}$$

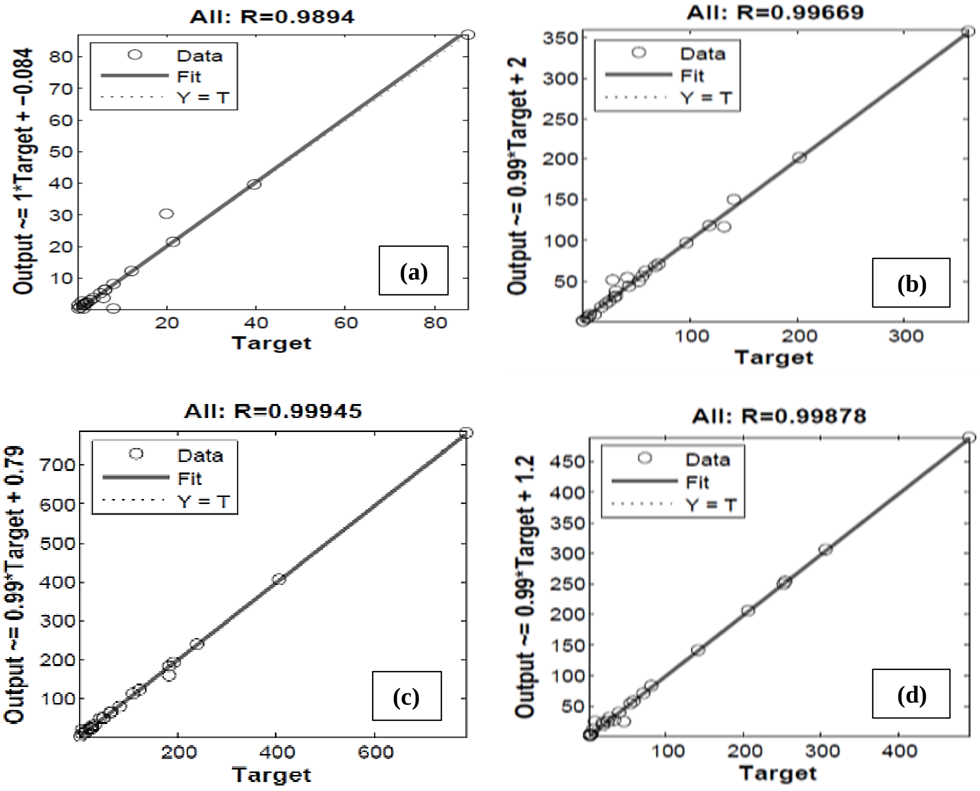
Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}} \tag{2}$$

In these equations, $X_{model,i}$ and $X_{obs,i}$ are simulated runoff and observed runoff, respectively and n is the number of samples. Furthermore, $X_i$ and $Y_i$ are the rainfall and runoff values of monthly as well as annually of the ANN model respectively and $\overline{X_\iota}$ and $\overline{Y_\iota}$ are the mean values of rainfall and runoff data, respectively.

## 3. Results and Discussion

The regression plot for target runoff and output runoff, for each monsoon month, obtained as a result of ANN monthly models have been shown below in Figure 4.



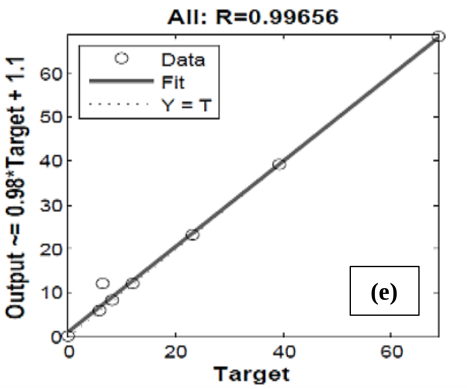


**Figure 4. Regression plot for target runoff and output runoff of the Dharoi watershed for, (a) June month; (b) July month; (c) August month; (d) September month; (e) October month**

The Regression plot for training, validation and testing for target annual runoff and output annual runoff, obtained as a result of ANN model have been shown below in Figure 5.

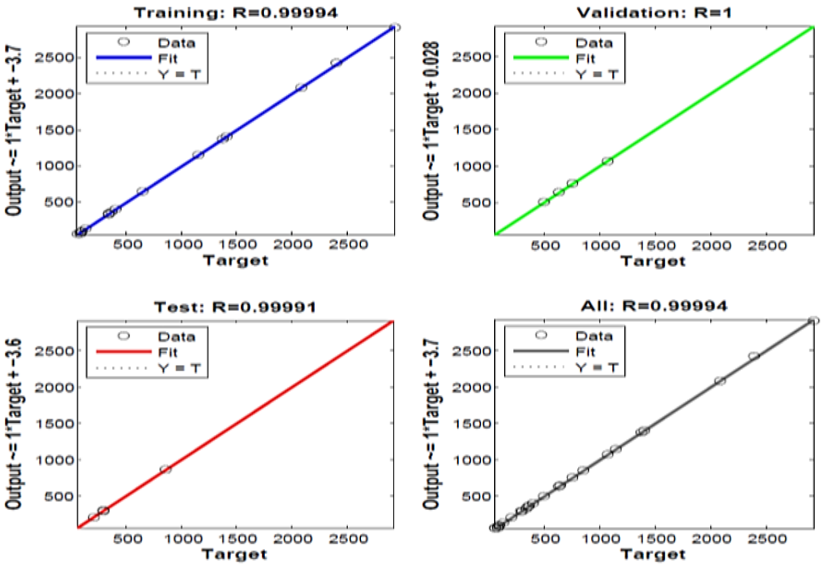


**Figure 5. Regression plot for target annual runoff and output annual runoff of the Dharoi watershed**

The ANN model results for monthly rainfall-runoff are shown in Table 2 and 3. shows results of ANN model for annual rainfall-runoff.

**Table 2. Results of ANN model for monthly rainfall-runoff**

| Hydrological region | Month | Network Architecture | No. of neurons | Correlation Factor (R) | RMSE |
|---|---|---|---|---|---|
| Dharoi | June | 1−2−1 | 62 | 0.989 | 2.792 |
| Dharoi | July | 1−2−1 | 66 | 0.997 | 6.371 |
| Dharoi | August | 1−2−1 | 68 | 0.999 | 5.462 |
| Dharoi | September | 1−2−1 | 61 | 0.999 | 6.030 |
| Dharoi | October | 1−2−1 | 18 | 0.997 | 1.935 |

**Table 3. Results of ANN model for annual rainfall-runoff.**

| Hydrological region | Network Architecture | No. of neurons | Correlation Factor (R) | RMSE |
|---|---|---|---|---|
| Dharoi | 1−2−1 | 76 | 0.9999 | 8.7334 |

In the present study, good correlation of coefficient (R) (0.99 overall) and Root Mean Square Error (RMSE) (between 1 to 7) for monthly rainfall-runoff ANN model have been obtained. Also, for annual rainfall-runoff ANN model, the good coefficient correlation (R=0.99) and Root Mean Square Error (RMSE) (8.73) have been obtained, which indicates a good fit for both monthly models and annual model.

Figure 6. Shows the plot have been prepared for each monthly observed rainfall (mm), observed runoff (Mm$^3$) and Predicted runoff (Mm$^3$) using ANN model.

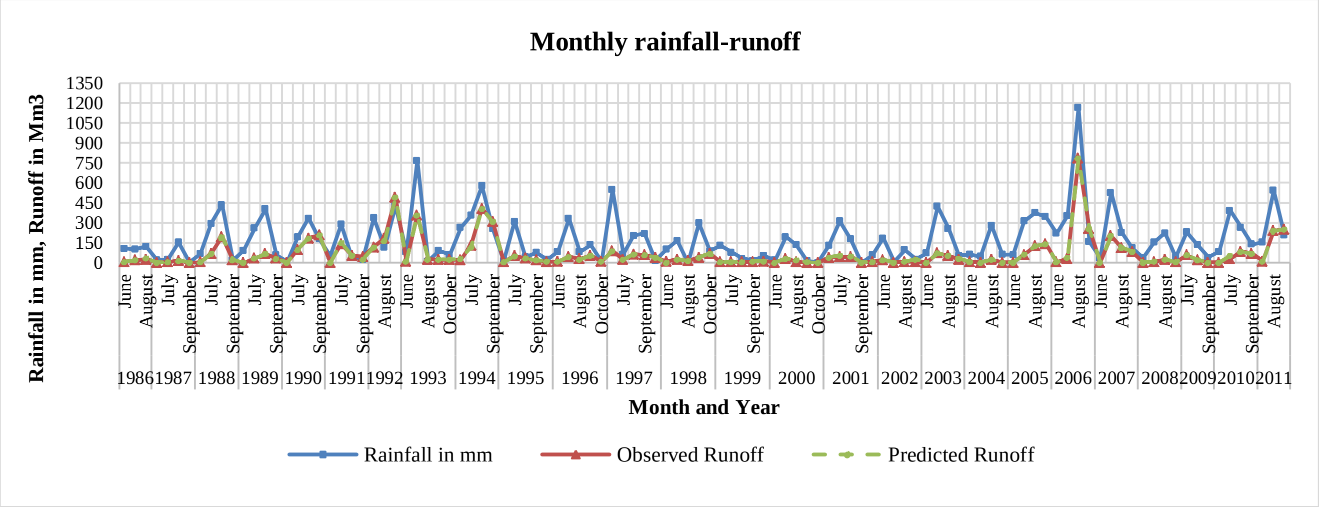


**Figure 6. Monthly observed rainfall (mm), observed runoff(Mm$^3$) and predicted runoff(Mm$^3$) using ANN model, for Dharoi watershed**

Figure 7. Shows the plot have been prepared for Yearly observed rainfall (mm), observed runoff (Mm$^3$) and Predicted runoff (Mm$^3$) using ANN model.

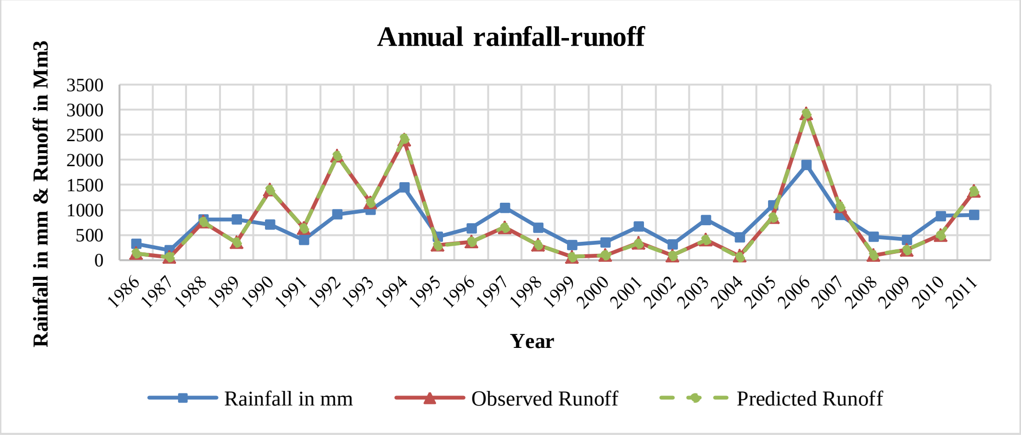


**Figure 7. Yearly observed rainfall (mm), observed runoff(Mm$^3$) and predicted runoff(Mm$^3$) using ANN model, for Dharoi watershed**

## 4. Conclusions

Monthly and annual ANN model with Feed-Forward Back Propagation network is developed in the present study for Dharoi watershed of Sabarmati river basin, India. The regression plot between observed runoff and the simulated runoff for monthly models were available from ANN monthly models and ANN annual model as well for Dharoi watershed of Sabarmati river basin. The performance of the developed models was then evaluated by statistical evaluation measurements, such as Pearson correlation coefficient (R) and Root Mean Square Error (RMSE). For monthly ANN models, the evaluation shows Pearson correlation coefficient (R) has been obtained as 0.99. Also, for annual ANN model, the Pearson correlation coefficient (R) obtained as 0.99. The RMSE values for monthly models and annual model have been obtain within the limits 1 to 9. The results indicate that the ANN model had good ability to capture the relationship between input/output i.e., Rainfall/Runoff and nonlinearity of input/output (Rainfall/Runoff). The results indicated that ANN cam capture the nonlinearity of rainfall-runoff modelling very well with good predictive power for simulation in hydrological models. The Figures have been prepared for each month for monthly observed rainfall (mm), observed runoff (Mm$^3$) and predicted runoff (Mm$^3$) using ANN models. Also, the similar figures have been prepared for annual rainfall-runoff ANN model. It is seen from these figures that the observed runoff is matching well with simulated runoff. The models results provide valuable information, which can use to solve problems in water resources studies and management. For better prediction include in the future other environmental factors as deforestation, land use and agricultural activities.

## 5. References

[1] H. Maier, A. Jain, G. Dandy, K. P. Sudheer, "Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions." Environ. Model. Softw. 2010, 25, 891–909.

[2] Faridah Othman, Mahdi Naseri, "Reservoir inflow forecasting using artificial neural network." International Journal of the Physical Sciences, 2011, Vol. 6(3), pp. 434-440, 4 February, 2011.

[3] P. V. Singh, Kumar Akhilesh; J. S. Rawat, Kumar Devendra, "Artificial neural networks based daily rainfall-runoff model for an agricultural hilly watershed." I.J.E.M.S. 2013, Vol.4 (2) 2013: 108-112.

[4] A. Bahremand, F. De Smedt, "Predictive analysis and simulation uncertainty of a distributed hydrological model." Water resources management. 2010, vol.24(12), 2869-2880.

[5] Vahid Nourani, Mehdi Komasi, Mohammad Taghi Alami, "Hybrid Wavelet–Genetic Programming Approach to Optimize ANN Modeling of Rainfall–Runoff Process." Journal of hydrologic engineering ASCE, 2012 JUNE, 727.

[6] M. P. Rajkumar, U. C. Kothyari, U. C. Chaube, "Artificial neural networks for daily rainfall-runoff modelling." Hydrologkal Sciences-Jo umai-des Sciences Hydrologiques, 2002, 47(6), December 2002.

[7] Mehrdad Fereydooni, Mehrdad Rahnemaei, Hossein Babazadeh, Hossein Sedghi, Mohammad Reza Elhami, "Comparison of artificial neural networks and stochastic models in river discharge forecasting, (Case study: Ghara- Aghaj River, Fars Province, Iran)." African Journal of Agricultural Research 2012, Vol. 7(40), pp. 5446-5458, 23 October, 2012.

[8] Dae Jeong, Young-Oh Kim, "Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction." Hydrological Processes, 2005, 19, 3819–3835.

[9] Ankit Chakravarti, Nitin Joshi, Himanshu Panjiar, "Rainfall Runoff Analysis Using Artificial Neural Network." Indian Journal of Science and Technology. 2015, Vol 8(14), July 2015.

[10] Archana Sarkar, Rakesh Kumar, "Artificial Neural Networks for Event Based Rainfall-Runoff Modeling." Journal of Water Resource and Protection. 2012, vol.4, 891-897.

[11] C. Chandre Gowda, S. G. Mayya, "Determining the Sensitivity of Rainfall- Runoff Model using Artificial Neural Network Approach." International Journal of Advanced Structures and Geotechnical Engineering ISSN 2319-5347, 2015, Vol. 04, No. 01, January 2015.

[12] W. Christian Dawson, Robert Wilby, "An artificial neural network approach to rainfall runoff modelling." Hydrological Sciences—Journal—des Sciences Hydrologiques, 1998, 43(1) February 1998.

[13] C. Chandre Gowda, S. G. Mayya, "Rainfall Runoff Modelling Using Generalized Neural Network and Radial Basis Network." International Journal of Intelligent Systems and Applications in Engineering ISSN:2147-6799 IJISAE, 2014, 2(4), 76–79.

[14] Ivana Sušanj, Nevenka ODaniT, Ivan MaroviT, "Methodology for Developing Hydrological Models Based on an Artificial Neural Network to Establish an Early Warning System in Small Catchments." Hindawi Publishing Corporation Advances in Meteorology, 2016, Volume 14.

[15] Jun Zhang, Chun-tian Cheng, Sheng-li Liao, Xin-yu Wu, Jian-jian Shen, "Daily reservoir inflow forecasting combining QPF into ANNs model." Hydrol. Earth Syst. Sci. Discuss., 2009, 6, 121–150, 2009.

[16] Manish Kumar Goyal; Chandra Shekhar Prasad Ojha. "Analysis of Mean Monthly Rainfall Runoff Data of Indian Catchments Using Dimensionless Variables by Neural Network." Journal of Environmental Protection, 2010, 1, 155-171.

[17] Pallavi Mittal, Swaptik Chowdhury, Sangeeta Roy, Nikhil Bhatia, Roshan Srivastav, "Dual Artificial Neural Network for Rainfall-Runoff Forecasting." Journal of Water Resource and Protection, 2012, 4, 1024-1028.

[18] Faramarz Keshvari, Seyed Amir Shamsnia, "Evaluation of Artificial intelligence models (Artificial Neural Networks) to forecasting the discharge of monthly river flow (Case study: Bahman section, Qareaghaj River)." Indian Journal of Fundamental and Applied Life Sciences ISSN: 2231– 6345 (Online), 2014 Vol.4 (S3), pp.8-13.

[19] Y. Shamseldin Asaad, "Artificial neural network model for river flow forecasting in a developing country." Journal of Hydroinformatics, 2010, 12.1.

[20] S. Raid, J. Mania, "Rainfall-Runoff Model Using an Artificial Neural Network Approach." Mathematical and Computer Modelling, 2004, 40, 839-846.

[21] Shraddha Srivastava, K. C. Tripathi, "Artificial Neural Network and Non-Linear Regression: A Comparative Study."

International Journal of Scientific and Research Publications, 2012, Volume 2, Issue 12, December 2012. ISSN 2250-3153.

[22] Kuok King Kuok, Nabil Bessaih, "Artificial neural networks (ANNS) for daily rainfall runoff modelling." Journal - The Institution of Engineers, Malaysia 2007 Vol. 68, No.3, September 2007.

[23] Takahiro Sayama, Yasuto Tachikawa, Kaoru Takara, Yutaka Ichikawa, "Distributed rainfall–runoff analysis in a flow regulated basin having multiple multi-purpose dams." IAHS Publ. 2006, 303.

[24] A. M. Kalteh, "Rainfall-runoff modelling using artificial neural networks (ANNs): modelling and understanding." Caspian J. Env. Sci. 2008, Vol. 6 No.1 pp. 53-58.

[25] Shahram Karimi-Googhari, Huang Yuk Feng, Abdul Halim B Ghazali, Lee Teang Shui, "Neural Networks for Forecasting Daily Reservoir Inflows." Pertanika J. Sci. and Technol, 2010, 18 (1): 33 – 41 (2010). ISSN: 0128-7680.

[26] Zulkarnain Hassana, Supiah Shamsudin, Sobri Harun, "Minimum Input Variances for Modelling Rainfall-runoff Using ANN." Jurnal Teknologi (Sciences and Engineering), 2014 Vol. 69(3), 2014, pages 113-118.

[27] J. M. Bravo, A. R. Paz, W. Collischonn, C. B. Uvo, O. C. Pedrollo, S. C. Chou, "Incorporating Forecasts of Rainfall in Two Hydrologic Models Used for Medium-Range Streamflow Forecasting." J. Hydrol. Eng. 2009, 14:435-445.

[28] R. Naik Arti, Prof. S. K. Pathan, "Weather Classification and Forecasting using Back Propagation Feed -forward Neural Network." International Journal of Scientific and Research Publications, 2012, Volume 2, Issue 12, December 2012, ISSN 2250-3153.

[29] Rosmina Bustami, Nabil Bessaih, Charles Bong, Suhaila Suhaili, "Artificial Neural Network for Precipitation and Water Level Predictions of Bedup River." IAENG International Journal of Computer Science, 2010, 34:2, IJCS_34_2_10.

[30] E. O. Oyebode, K. O. Adekalu, S. G. Fashoto, "Development of Rainfall-Runoff forecast model." International Journal of Engineering and Mathematical Intelligence, 2010, Vol. 1 Nos. 1 & 3, 2010.

[31] Anil Kumar Kar, Anil Kumar Lohani, Narendra Kumar Goel, Gopal Prasad Roy, "Development of Flood Forecasting System Using Statistical and ANN Techniques in the Downstream Catchment of Mahanadi Basin, India." J. Water Resource and Protection, 2010, 2, 880-887.

[32] Santosh Patil, Shriniwas Valunjkar, "Farecasting of daily runoff using artificial neural networks." International Journal of Civil Engineering and Technology (IJCIET), ISSN 0976 – 6308 (Print), ISSN 0976 – 6316(Online) 2014, Volume 5, Issue 1, January (2014), pp. 13-20.

[33] Mehmet C. Demirel, Anabela Venancio, Ercan Kahya, "Flow forecast by SWAT model and ANN in Pracana basin, Portugal." Advances in Engineering Software, 2009, 40, 467–473.

[34] Farzbod Amirhossien, Faridhossieni Alireza, Javan Kazem, Sharifi Mohammadbagher, "A Comparison of ANN and HSPF Models for Runoff Simulation in Balkhichai River Watershed, Iran." American Journal of Climate Change, 2015, 4, 203-216.

[35] P. Sundara Kumar, T. V. Praveen, Anjanaya Prasad, "Artificial Neural Network Model for Rainfall-Runoff -A Case Study." International Journal of Hybrid Information Technology. 2016, Vol.9, No.3, pp. 263-272.