STUDY OF PERFORMANCE OF DAILY RAINFALL- RUNOFF MODEL USING NEURAL NETWORKS

[1]Dr.C.S.V. Subrahmanya Kumar [2]Dr.G.K.Viswanadh
[1]Professor, [2] Professor & OSD to VC
[1]Department of Civil Engineering [2]Department of Civil Engineering
[1]ACE Engineering College, Hyderabad, India, [2]J.N.T.U College of Engineering, Hyderabad, India

ABSTRACT: Rainfall-runoff models are used to describe the hydrological behavior of a catchment. The relationship between rainfall and runoff is known to be highly non- linear, time varying and spatially distributed. This study presents the application of Artificial Neural Networks to model daily rainfall- runoff for Osmansagar catchment, Hyderabad, India using Levenberg- Marquardt back propagation algorithm. To study the performance of the model developed, various statistical performance indices namely correlation coefficient, normalised root mean square error, coefficient of efficiency, average absolute relative error and threshold statistic are computed during training and testing phases. The results indicate that ANN can effectively be used to model daily rainfall- runoff process.

Key words: Artificial Neural Networks, Catchment, Rainfall, Runoff, Modeling

I.INTRODUCTION

Runoff prediction is one of the most important aspects in hydrology, useful in water resources development, planning and management. A wide variety of rainfall- runoff models have been developed and used for flood forecasting. The transformation from rainfall to basin runoff involves many hydrological components like initial soil moisture, evaporation, evapotranspiration, infiltration and so on. All these hydrological components are believed to be highly non-linear, time varying, spatially distributed and cannot be properly described by simple models. The various models available are statistical, conceptual and black box to model rainfall-runoff process. . Recently, soft computing techniques such as fuzzy logic, Artificial Neural Networks, Adaptive Neuro-Fuzzy Inference System have emerged to model the various hydrological processes. Artificial neural networks (ANNs) have been proposed as black box models which are efficient tools for modeling and prediction in hydrology. ANNs are supposed to possess the capability to reproduce the unknown relationship existing between a set of input variables of the system and one or more output variables. Notable contributions in the field of hydrologic applications are French et al. (1992); Roger and Dowla (1994); Hsu et al. (1995); Tokar (1996).

1.1 ARTIFICIAL NEURAL NETWORKS MODELING APPROACH

Artificial Neural Network (ANN) is a massively parallel distributed information processing system that has certain performance characteristics resembling biological neural networks of the human brain (Haykin, 1994). ANN plays an important role in the field of hydrology, since the analysis of hydrologic systems deals with high degree of empiricism and approximation. As large number of publications have appeared in the recent past, to avoid duplication, the main concepts are highlighted in this section.

An ANN is composed of many non- linear and densely interconnected processing elements or neurons. In ANN architecture, neurons are arranged in groups called layers. Each neuron in a layer operates in logical parallelism. Information is transmitted from one layer to another in serial operations (Hecht- Nielsen, 1991). A network can have one or several layers. The basic structure of a network usually consists of three layers- the input layer, where the data are introduced to the network, the hidden layer(s), where the data are processed, and the output layer, where the results for the given input are produced. The neurons in the hidden layer(s) are connected to the neurons of a neighbouring layer by weighing factors that can be adjusted during the model training process. The networks are organized according to training methods for specific applications. Figure. 1 illustrates a three layer artificial neural network. The most distinctive characteristic of an ANN is its ability to learn from examples. Learning or training of an ANN model is a procedure by which ANN repeatedly processes data sets (input – output data pairs), changing the values of its weights. In the training or learning process, the target output at each output node is compared with the network output, and the difference or error is minimized by adjusting the weights and biases through some training algorithm. Back- propagation is the most commonly used supervised training algorithm in the multilayer feed forward networks. Although the back propagation algorithm based on gradient descent method is simple, it is very slow in convergence. Numerical optimization theory provides a rich and robust set of techniques which can be applied to neural networks to improve learning rates.

The gradient – descent method considers only the first order derivative of an error function. Using second order derivatives can improve the rate of convergence. Levenberg- Marquardt algorithm is one such method which uses second order derivatives of the error function and can be used, as a variant of back propagation to improve the learning rate. The weight update rule of this algorithm is given by (in vector notation)

$$\Delta W = (H + \mu \times I)^{-1} \times J^{T} \times e \qquad (1.1)$$

where $H = J^T \times J$ is the Hessian matrix of error vector and is equal to $\Delta^2 E$, is the Jacobian matrix of derivatives of each error to each weight, μ is a small value which controls the learning process, e is an error vector, ΔW is the change in weights, J^T is the transpose of the Jacobian matrix and I is the identity matrix. In practice, Levenberg- Marguardt algorithm with back propagation is faster and finds better optima for a variety of problems. In the present study, the training of ANNs was accomplished by Levenberg-Marquardt back propagation algorithm.

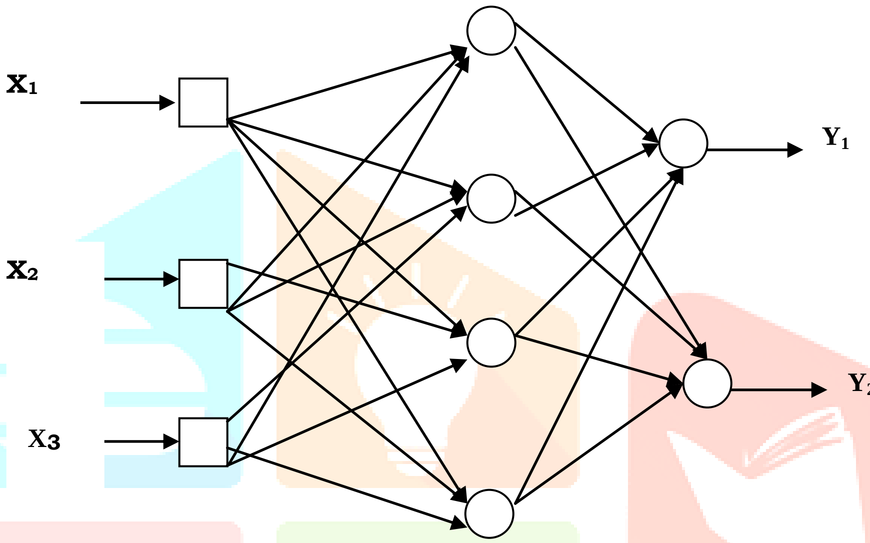


Fig: 1 A Typical Three Layer Feed Forward ANN configuration

II. STUDY AREA

Osmansagar catchment in Hyderabad, A.P, India, was selected to demonstrate the methodology for modeling daily rainfall- runoff relationship using ANN technique. The catchment has a drainage area of 750 sq km. The area is situated between 17.2^o – 17.5^o N latitude and 78.25^o to 78.35^{o}E longitude. The climate in the study area is semi arid and the major contribution of the rainfall is the South West monsoon with average annual rainfall varying from 525mm to 965mm. A view of Osmansagar reservoir with Osmansagar reservoir is shown in Fig.2. In the present study monsoon period is considered, which starts on 1st of June and continues upto 31st of October.

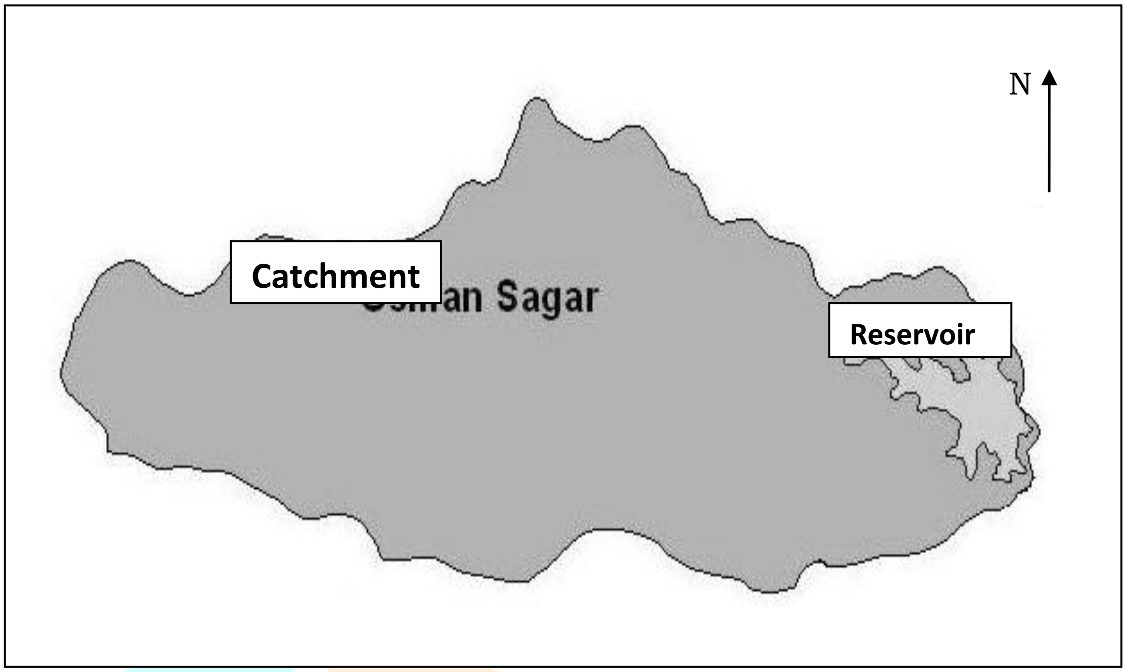


Fig: 2 A View of Osmansagar Catchment with Osmansagar Reservoir

III. EVALUATION METHODOLOGY

The steps involved in the present study in the formulation of various hydrological models are as follows:

(1) Selection of data sets for calibration and validation of the model.
(2) Normalization of the selected data.
(3) Formulation of the model by the identification of the input and output vectors.
(4) Determination of the structure of the Artificial Neural Network i.e., number of neurons in the input layer, hidden layer and the output layer.
(5) Training the Artificial Neural Network model using Levenberg- Marquardt Back Propagation algorithm.
(6) Validation of the model by presenting the test data to the developed ANN model.
(7) Computation of the statistical performance indices for both training and validation phases.

IV. DAILY RAINFALL- RUNOFF MODEL FORMULATION

The daily rainfall- runoff data considered in this study consist of 3200 data pairs collected from the records of the monsoon period of the years 1970- 2000. The ANN model was trained using 2000 data pairs and then tested or validated with 1200 data pairs.

In the present study, the input and output data vectors are normalized in the range 0 to 1 using equation (4.1).

$$(x_i)nor = \frac{(x_i)act - (x_i)min}{(x_i)\max - (x_i)min} \tag{4.1}$$

where, $(x_i)nor$ is the normalized value of the variable under consideration, $(x_i)act$ is the actual value of the variable, and $(x_i)_{max}$ and $(x_i)_{min}$ are the maximum and minimum values in the data series of a variable under consideration.

In rainfall- runoff modeling, the input vector presented to the network was decided by the procedure outlined by Sudheer et al. (2002). The input vector presented to the network was decided by carrying out statistical analysis such as computing cross correlation function (CCF) between rainfall and runoff series, auto correlation function (ACF) and partial auto correlation function (PACF) for rainfall and runoff series. In cross correlation analysis, CCF was found between the rainfall in the past i.e., for various lag periods and flow at time't'. A qualitative examination of the CCF between rainfall and runoff series would reveal which antecedent rainfall heavily influences the runoff at a certain time. Similarly, ACF and PACF was also determined to know upto what lag period the previous rainfall/ runoff is influencing the runoff at time 't'. The ACF and PACF would thus suggest the influencing antecedent runoff patterns in the flow at a given time.

By carrying out the above analysis, the variables that may not have a significant effect on the performance of the model can be omitted from the input vector, resulting in more compact network.

In daily rainfall-runoff modeling, CCF between the rainfall and runoff series, and ACF for rainfall series as depicted in Fig. (3) & Fig. (4) showed a significant correlation upto 3 days lag in rainfall data on the flow at any time. The ACF, PACF of runoff time series, shown in Fig. (5) and Fig. (6) indicates the runoff values upto lag 1 is significant. The gradual decaying pattern of autocorrelation exhibits the presence of a dominant autoregressive process.

The above analysis suggests that rainfall values up to lag 3 and runoff value upto lag1 in the input vector are required to model the rainfall- runoff process.

Thus the input vector to the network can be defined with rainfall at time‘t’, three antecedent rainfall intervals and one antecedent runoff, thus making the input a five element vector. Thus the functional relationship between rainfall and runoff is represented as,

$$R_t = f\{ P_t, P_{t-1}, P_{t-2}, P_{t-3}, R_{t-1} \} \quad (4.2)$$

In the above equations, R_t is runoff at time t; P_t is daily rainfall at time t; P_{t-1} is daily rainfall at time (t-1) and so on. R_{t-1} is daily runoff at time (t-1), i.e., one time lag runoff at basin outlet.

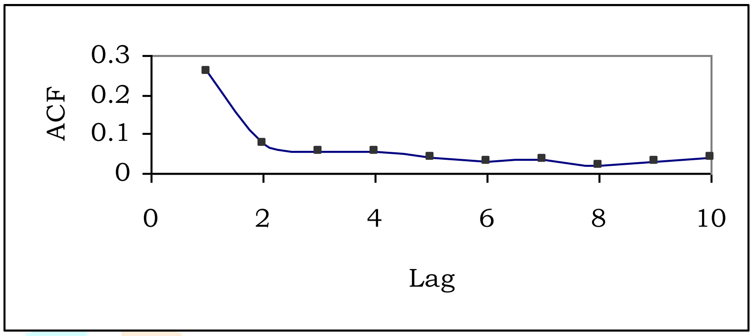


Fig: 3 The Cross correlation function between the rainfall-runoff series

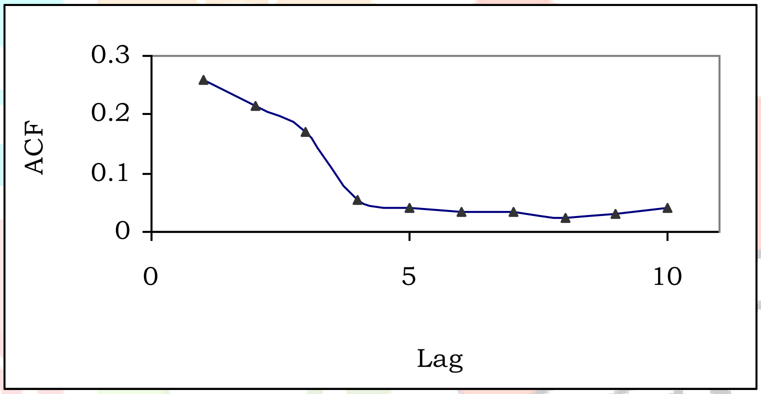


Fig: 4 Auto correlation function for the rainfall series

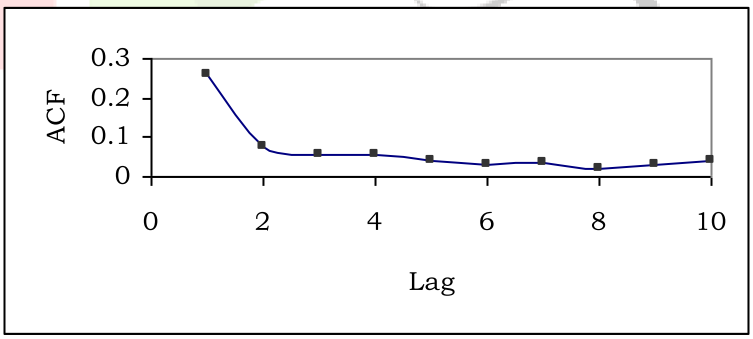


Fig: 5 Autocorrelation function for the runoff series

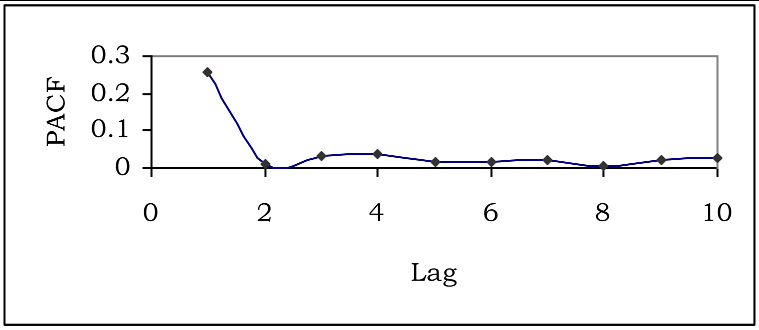


Fig: 6 Partial Autocorrelation function plot for the runoff series

A three layer ANN model was employed to study the rainfall – runoff relationship. The number of neurons in the hidden layer is finalised by trial and error. In the trial process during training, the number of neurons in the hidden layer was varied between 1 and 20. The configuration that gives the minimum MSE and maximum correlation coefficient was selected for each of the options. Sigmoid function is used as the activation function in the network training process. The final ANN architecture arrived consists of four hidden neurons. For the selected models, the computed values of runoff are denormalised and the performance criteria such as Correlation coefficient, Average absolute relative error (AARE), Nash coefficient of efficiency, Threshold Statistics (TS_x), Normalised Root Mean Square Error (NRMSE), Normalised Mean Bias Error (NMBE), are evaluated during training and testing and are presented in Table 2.

4.1 STATISTICAL PERFORMANCE INDICES

(1) Correlation Coefficient (R): The correlation coefficient is given as,

$$R = \frac{[y_o(t) - y'_o\ (t)] * [y_p(t) - y'_p\ (t)]}{\sqrt{\Sigma[\mathrm{yo(t) - y'o(t)}]2} * \sqrt{[y_p(t) - y'_p\ (t)]2}} \qquad (4.3)$$

Where $y_o(t)$ and $y_p(t)$ are the observed and computed values of a variable and $y'_o(t)$, $y'_p(t)$ are the mean of the observed and computed values.

(2) Average Absolute Relative Error (AARE):

Average Absolute Relative Error gives average error prediction. It is the average of the absolute values of the relative errors in forecasting. Mathematically *AARE* is calculated using the following equations.

$$RE(t) = \frac{y_p\ (t) - y_o\ (t)}{y_o(t)} * 100 \qquad (4.4)$$

$$AARE = \frac{1}{n}\sum |RE(t)| \qquad (4.5)$$

where $y_o(t)$ and $y_p(t)$ are the observed and computed values of a variable at time t, *RE(t)* is the relative error in predicting the variable at time *t* and *n*, the number of observations. Smaller the value of *AARE* better is the performance of the model.

(3) Threshold Statistics (TS_x):

This performance index gives the distribution of the errors. The threshold statistic for a level of x% is a measure of the consistency in forecasting errors from a particular model (Nayak et al. 2005). It is designated by TS_x and is expressed in percentage. This criterion can be expressed for different levels of absolute relative error from the model. It is computed for x% level as,

$$TS_x = \left[{}^{Y_x}/_{N}\right] * 100 \qquad (4.6)$$

where 'Y_x' number of data points forecasted whose absolute relative error is less than x% and N the total number of data points predicted. Threshold Statistics were computed for absolute relative error levels of 1%, 5%, 10%, 25%, 50%, and 100% in this study. Higher the value of threshold statistic better is the model performance.

(4) Nash- Sutcliffe coefficient of efficiency (η):

The Nash coefficient of efficiency (Nash- Sutcliffe, 1970) compares the computed and the observed values of the variable and evaluates how far the model is able to explain the total variance in the data set.

The Nash coefficient of efficiency is calculated as,

$$\eta = \frac{[y_o(t) - y'_o(t)]^2 - y_p(t) - y_o(t)]^2}{\Sigma[y_o(t) - y'_o(t)]^2} * 100 \qquad (4.7)$$

where $y'_o(t)$ is the mean of observed values and all other variables are same as explained earlier. Higher the value of efficiency better is the model performance.

(5) Normalised Mean Bias Error (NMBE):

The Normalised Mean Bias Error (Nayak et al. 2005) indicates whether the modeled values of the output are under or over predicted. It is computed as,

$$\mathrm{NMBE} = \frac{{}^1/_n\, \Sigma[y_p(t) - y_o(t)]}{{}^1/_n\, \Sigma y_o(t)} *100 \qquad (4.8)$$

Positive NMBE would indicate overall over prediction while negative value would mean overall under prediction from the model.

(6)Normalised Root Mean Square Error (NRMSE):

The Normalised Root Mean Square Error is computed using the following equation.

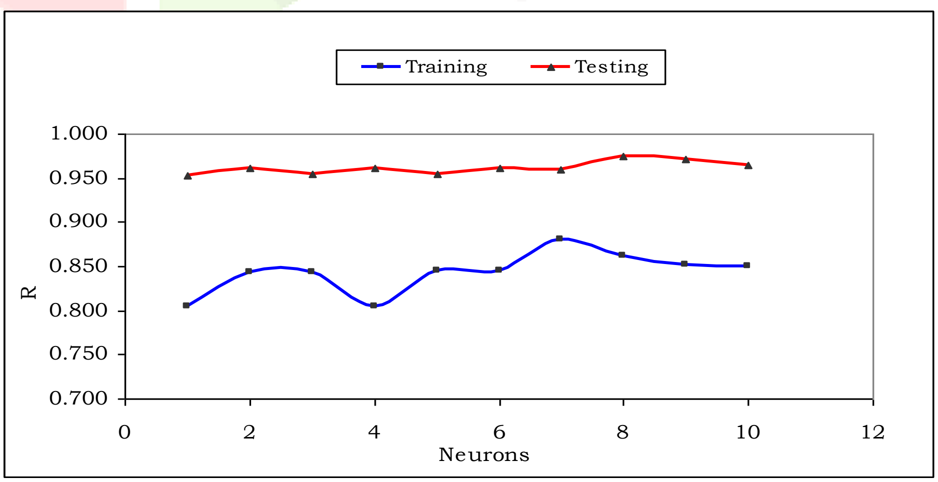
$$\mathrm{NRMSE} = \frac{{}^1/_n\, \sqrt{\Sigma[y_p(t) - y_o(t)]^2}}{{}^1/_n\, \Sigma y_o(t)} \qquad (4.9)$$

Better model performance is indicated by lower value of NRMSE.

V. RESULTS AND DISCUSSION

In daily rainfall- runoff model formulation, using Levenberg- Marquardt Back Propagation algorithm, it is found that the ANN model with architecture, 5 input neurons, 7 hidden neurons, 1 output neuron is the most suitable model. The graphical representation of variation of R and MSE with number of neurons is shown in Fig. (7) and Fig. (8). The variation of R and MSE with number of epochs for the selected ANN architecture 5-7-1 is shown in Fig. (9) and Fig. (10).

From the Fig. (10), it is observed that after 300 epochs there is no significant change in MSE during training and testing and thus the training is stopped at 400 epochs. The correlation coefficient is found to be 0.880 and 0.960 and the Mean square error is 9.69E-05 and 9.57582E-05 during training and testing phases at 100 epochs stage. With increase in from 100 to 400, the correlation coefficient has improved from 0.880 to 0.885 during training and from 0.960 to 0.979 during testing phase. The MSE has further decreased from 0.000101 to 9.70E-05 during training and 0.000116 to 9.60E-05 during testing.



F ig: 7 Variation of R with Neurons

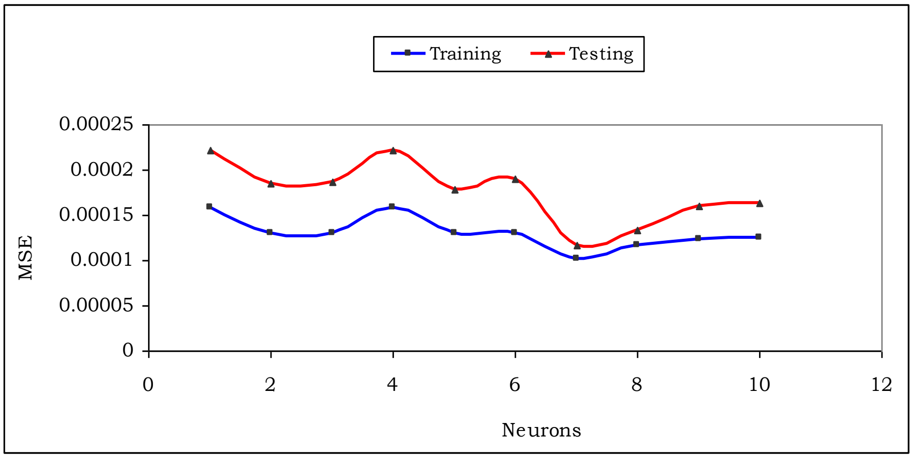


Fig: 8 Variation of MSE with Neurons

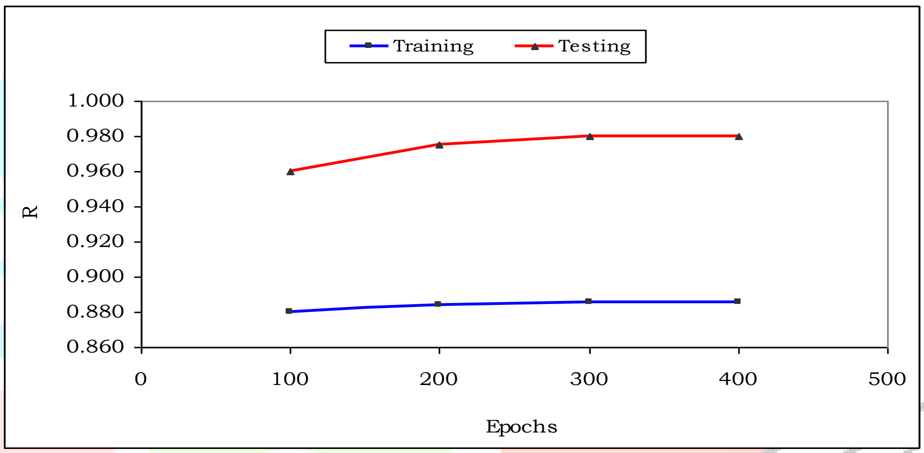


Fig: 9 Variation of R with epochs for 5-7-1 Architecture

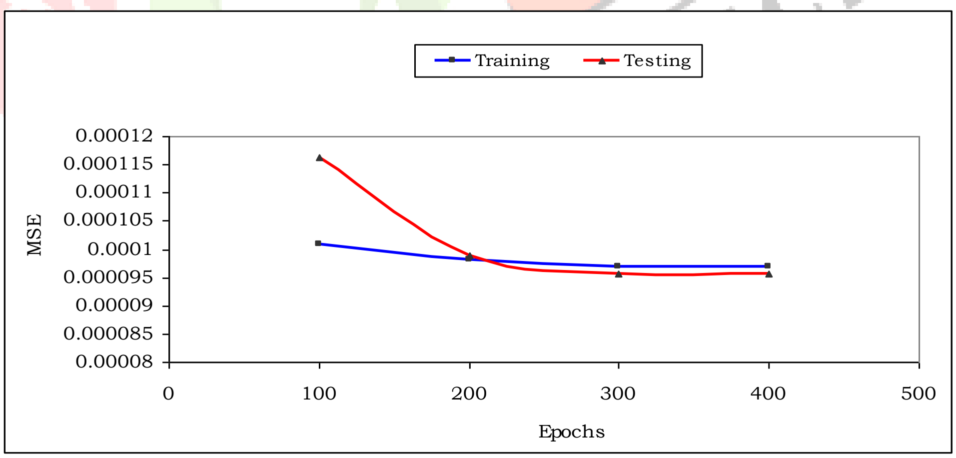


Fig: 10 Variation of MSE with epochs for 5-7-1 Architecture

To assess the potential of all the models developed, the first three statistical criteria namely, mean, standard deviation and skewness are determined for observed and modeled values during training and testing phases and are presented in Table 1. The analysis reveals that the statistical properties namely mean and standard deviation of the historical flow series are preserved by the model developed, both during training and testing. However, it is observed that skewness is preserved during testing.

Table.1 Summary Statistics during Training & Testing

Statistical Parameter	Training		Testing	
	Observed	**Modeled**	**Observed**	**Modeled**
Mean (cm)	3.481	3.449	3.928	3.953
Standard Deviation (cm)	1.510	1.347	3.487	3.429
Skewness Coefficient (cm)	6.094	7.842	13.437	14.066

Table. 2 Statistical Performance Indices during Training & Testing for 5-7-1 Architecture

Phase	Threshold Statistics						R	η (%)	NRMSE	AARE (%)	NMBE (%)
	TS1	**TS5**	**TS10**	**TS25**	**TS50**	**TS100**					
Training	15.55	55.90	88.70	93.70	97.80	99.70	0.885	78.32	0.202	7.37	-0.90
Testing	18.31	61.04	80.35	88.88	96.57	99.41	0.979	95.98	0.072	9.34	0.64

The Various statistical performance indices during training and testing are presented in Table.2. The correlation coefficient was found to be 0.885 and 0.979 during training and testing indicating good strength between computed and observed runoff. The AARE is 7.37% and 9.34% during training and testing phases. From Threshold statistic criteria it is observed that 88.70% and 80.35% of the computed values of runoff are within 10% absolute relative error. Based on NMBE it is inferred that, the runoff values are slightly underestimated during training and overestimated during testing. The Nash coefficient of efficiency is found to be 78.32% during training and 95.98% during testing. Thus based on the performance evaluation criteria it is concluded that the AAN based rainfall-runoff model is able to capture the dynamics of the catchment.

In addition to the numerical indicators, graphical indicators are also developed for model evaluation.The linear scale plot of the observed and modeled runoff v/s time during training and testing phases for the formulated model is shown in Fig. (11) and Fig. (12). The graphs show a good match between modeled and observed runoff values in the model formulated. The scatter plots of the modeled flow versus observed flows for the training and testing phases are shown in Fig. (13) and Fig. (14). The observed and modeled values appear to be closer to the 45^{o} line indicating good match between observed and modeled runoff values.

The results confirm that NN is capable of learning the rainfall- runoff process in the area under consideration and the model formulated are able to predict the runoff from the catchment with reasonable accuracy.

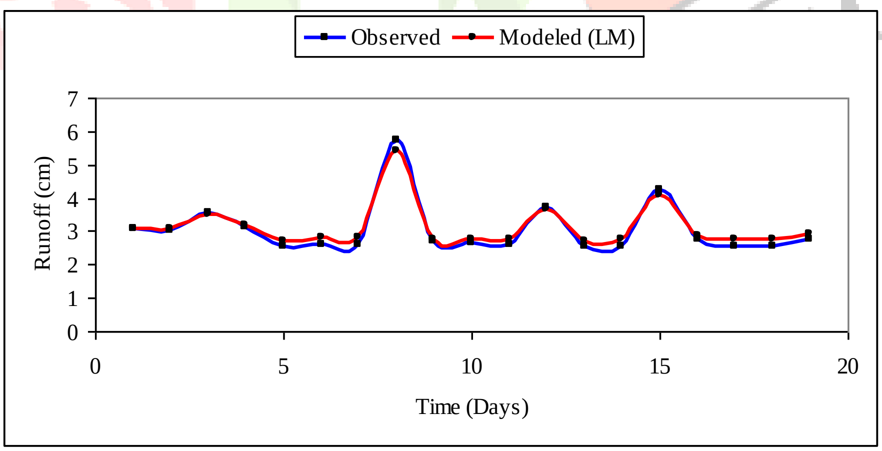


Fig: 11 Comparison of Observed and Modeled results for during Training

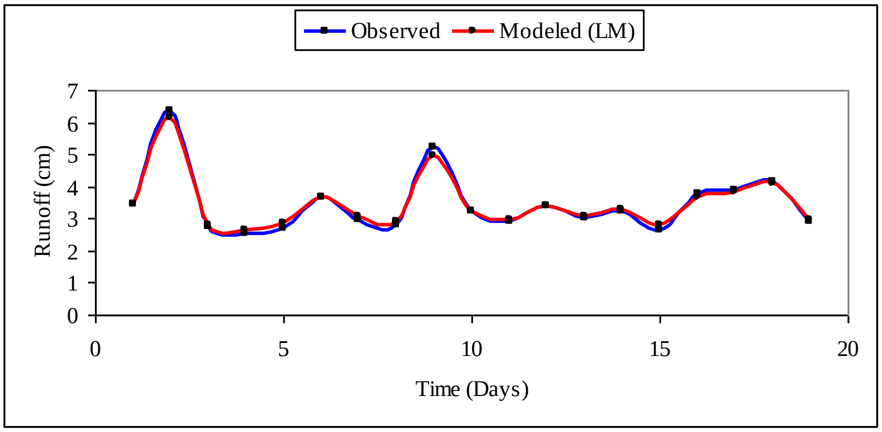


Fig: 12 Comparison of Observed and Modeled results for during Testing

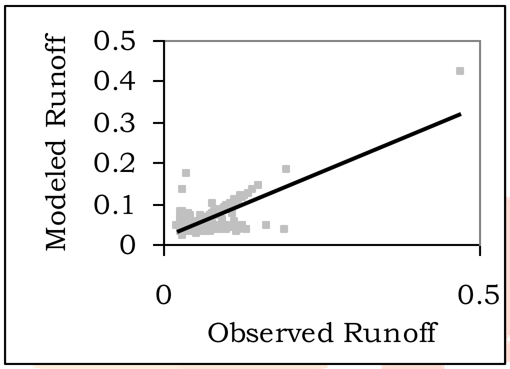


Fig: 13 Scatter Plots comparing modeled & observed flows during Training

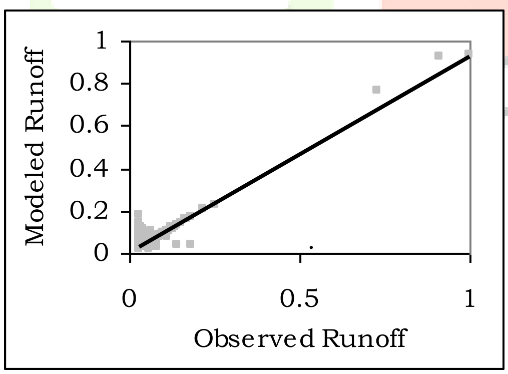


Fig: 14 Scatter Plots comparing modeled & observed flows during Testing

REFERENCES

[1] ASCE Task Committee (2000a). “Artificial neural networks in Hydrology I Journal of Hydrologic Engineering, 5(2), 115- 123.
[2] ASCE Task Committee (2000b). “Artificial neural networks in Hydrology II, Journal of Hydrologic Engineering, 5(2), 124- 132.
[3] French,M.N., Krajewski.W.F., and Cuykendall R.R. (1992). “Rainfall forecasting in space and time using a neural network.” Journal of Hydrology, Amsterdam, 137, 1-31.
[4] Haykin, S. (1994). “Neural networks: A comphrensive foundation.” Mac Millan, New York.
[5] Hect- Nielson, R.(1990). “Neuro Computing.” Addison-Wesley, New York.
[6] Hsu.K.,Gupta.V.H., and Sorooshian,S. (1995). “Artificial Neural Network Modeling of the Rainfall – Runoff Process.” Water Resource Research, 31(10), 2517-2530.

[7] Jain .A, Ormsbee L.E., (2002). "Evaluation of short term water demand forecast modeling techniques: conventional methods versus AI." Journal of the American Water Works Association, 94(7), 64 – 72.

[8] Jain.A, Varshney.A.K, Joshi.U.C., (2001). "Short term water demand forecast modeling at IIT Kanpur using artificial neural networks." Water Resources Management, 15(5), 299 – 321.

[9] Prakash,A.,(1986). "Current state of hydrologic modeling and its limitations." Proceedings International Symposium on Flood Frequency and Risk Analysis, Louisiana State University, Baton Rouge, Louisiana, USA.

[10] Raman.H., Sunil Kumar.N. (1995). " Multivariate Modelling of Water Resources Time Series using Artificial Neural Networks." Hydrological Sciences, 2(40), 145-163.

[(11] Roger,L.L., Dowla,F.U. (1994). "Optimization of Groundwater Remediation using ANN with Parallel Solute Transport Modeling." Water Resources Research, 30(2), 457-481.

[12] Subrahmanya Kumar. C.S.V., (2009), Ph.D dissertation on Performance Evaluation and Comparison of Different types of Hydrological Models using Artificial Neural Networks, JNTU, Hyderabad.

[13] Subrahmanya Kumar. C.S.V & Viswanadh G.K (2008) Artificial Neural Network Approach for Reservoir Stage Prediction, International Journal of Applied Mathematical Analysis and Applications, pp 13- 19, Vol 3, No.1, Jan - June 2008.

[14] Subrahmanya Kumar. C.S.V & Viswanadh G.K (2009), Performance Evaluation of Rainfall-Runoff Modeling using Artificial Neural Networks**,** International Journal of Scientific Computing, pp 245- 250, Vol 3, No.2, July – December, 2009.

[15] Tokar,A.S.(1996). "Rainfall -Runoff Modeling in an Uncertain Environment." Ph.D dissertation, University of Maryland, College Park.