Daily Runoff Forecasting using Artificial Neural Network

G. Sreenivasa Rao[1] and M.V.S.S Giridhar[2]
[1]Assoc. Prof., Department of Civil Engineering, M.V.S.R College of Engineering, Hyderabad, India. Email: gumpallisrinivas@gmail.com
[2]Assis. Prof., & Head, Centre for Water Resources, Institute of Science and Technology, JNTUH, Hyderabad, India. Email: mvssgiridhar@gmail.com

ABSTRACT: Rainfall-Runoff is the most important hydrological variables used in most of the water resources applications. Watershed based planning and management requires thorough understanding hydrological process and accurate estimation of runoff. An Artificial Neural Network (ANN) methodology was employed to forecast daily runoff for the Kadam watershed of G-5 sub-basin of Godavari river basin. On the contrary ANN can be deployed in cases where the available data is limited. The hydrologic variables used were eleven years' rainfall and runoff data out of which 70% of the data is used for training, 15% for testing and the remaining 15% for validation, Runoff being the desired output for the years 2001 to 2011. Effect of number of layers in the network is also studied. The performance of ANN is evaluated based on the efficiency and the error. The results obtained in the present study have been able to demonstrate that the ANN models are able to provide a good representation of an event-based rainfall-runoff process.

KEYWORDS: Rainfall-Runoff, Hydrological variables, Validation, Kadam Watershed, ANN.

INTRODUCTION:

Rainfall-runoff models play an important role in water resource management planning and therefore, different types of models with various degrees of complexity have been developed for this purpose. These models, regardless to their structural diversity generally fall into three broad categories; namely, black box or system theoretical models, conceptual models and physically-based models (Dooge 1977). The rainfall-runoff relationship is one of the most complex hydrological phenomena to comprehend, owing to the tremendous spatial and temporal variability of watershed characteristics and precipitation patterns, and to the number of variables involved in the modelling of the physical process (Kumar et al., 2005).

The artificial neural network (ANN) approach differ from the traditional approaches in stochastic hydrology in the sense that it belongs to a class of data-driven approaches as opposed to traditional model driven approaches. There has been a tremendous growth in the interest of application the ANNs in rainfall-runoff modelling in the 1990s (Hsu et.al., 1995). Hydrologists are often confronted with problems of prediction and estimation of runoff, precipitation, contaminant concentrations, water stages, and so on.

ANN models have been used successfully to model complex non-linear input output relationships in an extremely interdisciplinary field. The natural behaviour of hydrological processes is appropriate for the application of ANN method. In terms of hydrologic applications, this modelling tool is still in its nascent stages (Govindaraju 2000). ANNs were usually assumed to be powerful tools for functional relationship establishment or nonlinear mapping in various applications.

(Shamseldin et.al.,1997), examined the effectiveness of rainfall-runoff modelling with ANNs by comparing their results with the Simple Linear Model (SLM), the seasonally based Linear Perturbation Model (LPM) and the Nearest Neighbour Linear Perturbation Model (NNLPM) and concluded that ANNs could provide more accurate discharge forecasts than some of the traditional models.

Several studies indicate that ANN have proven to be potentially useful tools in hydrological modelling such as for modelling of rainfall-runoff processes, flow prediction, water quality predictions, operation of reservoir system and groundwater reclamation problems (Buch A. M., 1993). Among the daily, monthly and annual scales, the monthly rainfall–runoff relationship is probably the most difficult since it has to take into account both short-term and long-term hydrological processes (Anmala et al., 2000).

In this paper, a neural network computer program was developed to carry out Rainfall-Runoff modelling of Kadam watershed of Godavari basin in Telangana. The present objective of the study is to experiment for the generation of fully distributed rainfall runoff. For the validation this observed data, a model is established for estimating observed runoff data using ANN technique.

STUDY AREA: The area selected for present study is Kadam watershed of G-5 sub basin i.e., 'Middle Godavari' sub basin of Godavari River Basin. The Godavari basin is situated in the Deccan plateau covering large areas in the states of Maharashtra, Madhya Pradesh, Chhattisgarh, Orissa, Karnataka and Telangana. The Godavari basin extends over an area of 13,812 sq.km which is nearly 10% of the total geographical area of the country. Godavari catchment is divided in to eight sub basins. The Kadam watershed of River Godavari which lies between latitudes 19 05' and 19 35' North and longitudes 78 10' and 78 55' East in the state of Telangana. The areal extent of the study area is 2651 km^2, which constitutes 7.4% of the sub basin area. The climate in the study area is semi-arid with an average rainfall of 765mm. The main geological formations of the sub basin are Deccan traps and gneiss. The formations have considerable effect on the runoff of the sub basin.

METHODOLOGY:

COMPUTATIONS IN ANN:

The computational process associated with an ANN is as follows: An ANN is a computing system made up of a highly interconnected set of simple information processing elements, analogous to a neuron, called units. The neuron collects inputs from both a single and multiple sources and produces output in accordance with a predetermined non-linear function. An ANN model is created by interconnection of many of the neurons in a known configuration.

Generally, there are four distinct steps in developing an ANN- based solution. The first step is the data transformation or scaling. The second step is the network architecture definition, where the number of hidden layers, the number of neurons in each layer, and the connectivity between the neurons are set. In the third step, a learning algorithm is used to train the network to respond correctly to a given set of inputs. Lastly, comes the validation step in which the performance of the trained ANN model is tested through some selected statistical criteria.

NEURAL NETWORK APPLICATION:

A rainfall-runoff model using ANNs for the Kadam watershed in Telangana state. The

ISSN 2229-5518

available hydrologic parameters were Rainfall and Runoff values for 11 years (2001-11). There are three basic layers or levels of data processing units viz., the input layer, the hidden layer and the output layer.

Each of these layers consists of processing units called nodes of the neural network. The number of input nodes, output nodes and the nodes in the hidden layer depend upon the problem being studied. If the number of nodes in the hidden layer is small, the network may not have sufficient degrees of freedom to learn the process correctly. If the number is too high, the training will take a long time and the network may sometimes over-fit the data. The process of determining ANN weights is called training, which forms the interconnection between neurons. The ANNs are trained with a training set of input and known output data. At the beginning of training, the initial value of weights can be assigned randomly or based on experience

The learning algorithm systematically changes the weights such that for a given input, the difference between the ANN output and the actual output is small. Many learning examples are repeatedly presented to the network, and the process is terminated when this difference is less than a specified value. At this stage, the ANN is considered trained. An ANN is better trained as more input data are used. Several learning algorithms have been reported in the literature. In the present study, the most widely used three-layer feed forward error back propagation algorithm has been used for training. After training is over, the ANN performance is validated. Depending on the outcome, either the ANN has to be retrained or it can be implemented for its intended use. A large number of statistical criteria are available to compare the goodness of any given model. The performance evaluation statistics used for ANN training in the present work are root mean square error (RMSE), coefficient of correlation (R) and coefficient of determination (DC). These parameters have been determined using the following equations

$$\mathrm{DC} = \frac{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)^2 - \sum_{i=1}^{n}\left(Q_i - \bar{q}\right)^2}{\sqrt{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)^2}}$$

where $\bar{Q} = \frac{1}{n}\sum_{i=1}^{n} Q_i$, $\bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i$,

Q = observed discharge (cumec), q = calculated discharge (cumec).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(Q_i - q_i\right)}{n}^{2}}$$

$$\mathrm{R} = \frac{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)\left(q_i - \bar{q}\right)}{\sqrt{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)^2 \left(q_i - \bar{q}\right)^2}}$$

Large variation in the input data can slow down or even prevent the training of the network. To overcome this potential problem, the data are usually scaled using linear, logarithmic, or normal transformations. It is also important that the absolute input values are scaled to avoid asymptotic issues. In the present study, the input data for a variable x were standardised through the ANN software.

NEURAL NETWORK TOOL (NNT) IN MATLAB:

We have four different types of neural networks in Matlab. In the present case we use the “fitting app” as the remaining are not appropriate for the process. In fitting

ISSN 2229-5518

problems, you want a neural network to map between a data set of numeric inputs and a set of numeric targets. The Neural Fitting app will help you select data, create and train a network, and evaluate its performance using mean square error and regression analysis.

Since black box models such as ANNs derive all their 'knowledge' from the data that is presented to them it is clear to see that the question, which input and output data to present to an ANN, is of the utmost importance. ANNs try to approximate a function of the form.

$$Y^m = f\,(X^n) \text{ .... (A)}$$

Where

- X^n is an n -dimensional input vector consisting of variables x1, xi, …, xn and
- Y^m is an m-dimensional output vector consisting of output variables y1, yi…ym.

The user can select input and/or output variables of a feed forward ANN after loading a single Matlab data file that contains variables. The number of inputs and outputs can be chosen freely. ANN architecture since the hydrological problems for which the tool was designed are not very complex, the number of hidden layers had been limited to two. The user can choose between one, two or no hidden layers and can freely choose the number of neurons of which possible hidden layers consists. Four types of transfer functions for each layer can be chosen: two sigmoid functions, a purely linear function and a saturating linear function.

In present study we have used ten number of neurons or hidden layers and have calculated the regression factor, the one showing the extent of linear relation between the input and output. After the training is completed we have validated the output with the original output.

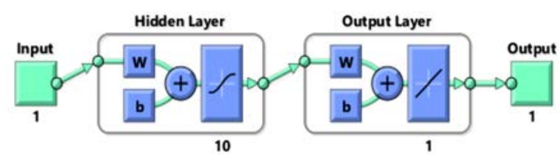


Fig 1: Frame work of Neural Network

Eight possible training algorithms are provided to train the ANN of choice: four conjugate gradient algorithm variations, the Levenberg-Marquardt algorithm, one quasi-Newton algorithm and two advanced back propagation variants: resilient back propagation and back propagation with regularization. A data set can be split into two or three parts: one for training, one for cross-training (optional) and one for testing. This split-sampling of the data can be done either continuously or distributed (i.e. divide the data into three continuous parts or take three random selections from the data). The cross training data is used when the user chooses to use the early-stopping technique of training with cross training. If you are not satisfied with the results you can retrain the network by importing larger data set or adjust network size or by changing the percentages used for testing and training.

RESULTS

After the training has completed we have obtained an equation for the relation between the rainfall and runoff.

$$y = B2 + LW*\text{tansig}\,(B1+IW*x) \text{ … (1)}$$

Where

ISSN 2229-5518

- y is the runoff
- x is the rainfall
- B1 & B2 are the bias and
- LW & IW are the weights

The average equation for the present case is

y = -1.051231857 +0.192358105*tansig (0.343305405+6.078387998x) …… (2)

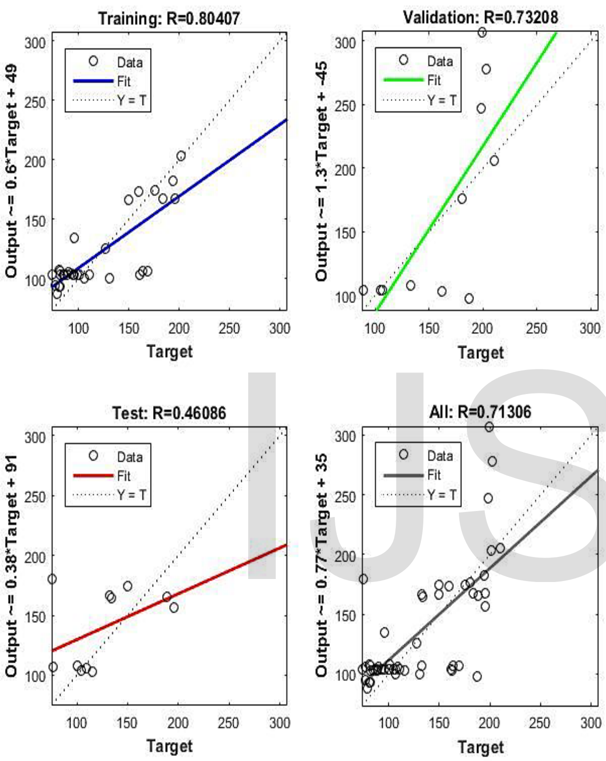


Fig 2: The regression plot of the trained network in Matlab

Table 1: Total Rainfall and Predicted Runoff data

Year	Total rainfall (mm)	Actual runoff (mm^3/mm^2)	Predicted runoff (mm^3/mm^2)
2001	1064.7	82.215789	89.94571
2002	1070.6	95.298864	81.39708
2003	1044.5	118.43582	96.10972
2004	1332.7	79.829628	66.43052
2005	705.7	88.692602	66.13920
2006	1272.1	129.59226	80.53714
2007	1145.3	70.853139	58.05534
2008	829.35	74.185834	42.64238
2009	974.66	85.950922	54.86834
2010	683.23	95.279281	65.48438
2011	1442.3	130.66504	106.1455

The following are the graphs drawn between actual runoff and predicted runoff for the years 2001 to 2011:

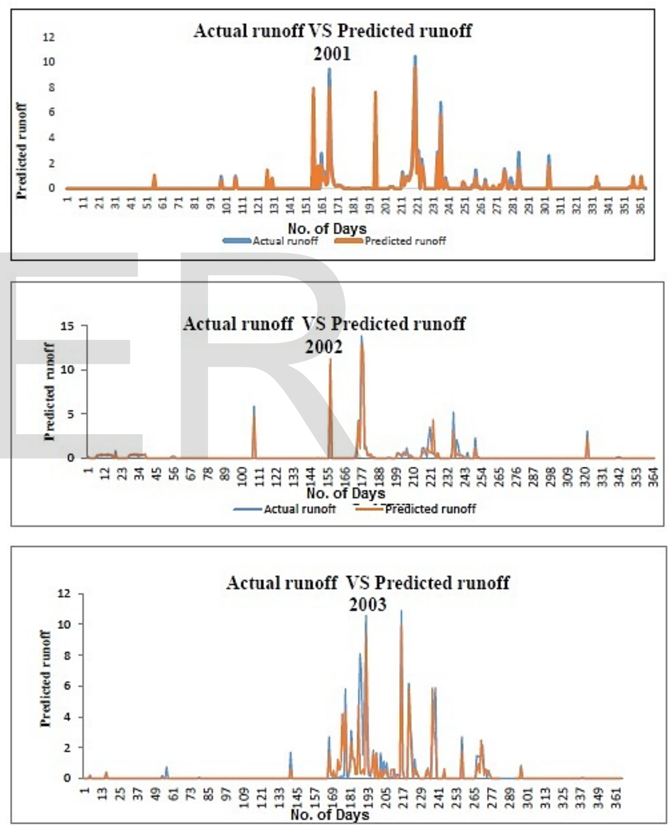


Fig 3: Graphs for the years 2001, 2002, 2003

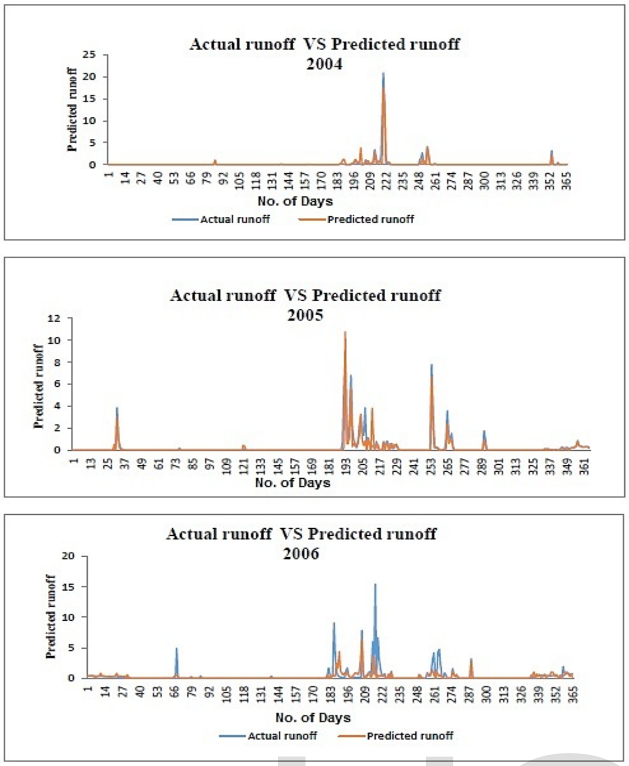
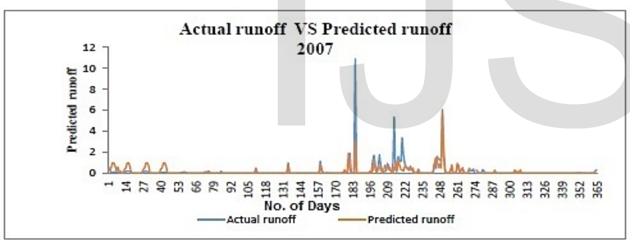


Fig 4: Graphs for the years 2004, 2005, 2006



The large different points can be ignored as it may be differ sometimes due to the error at peak points.

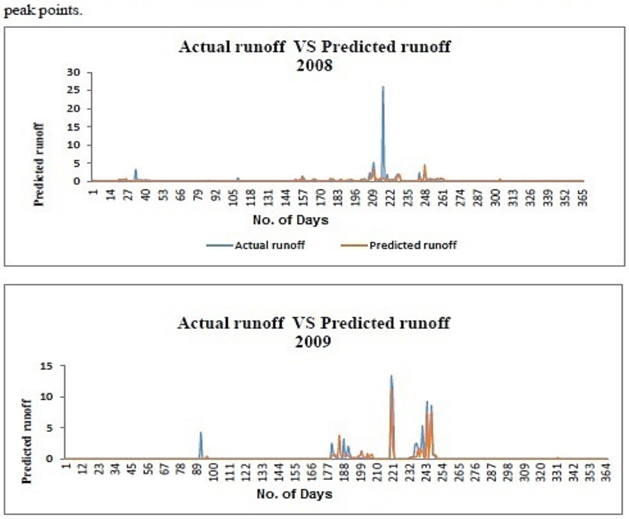


Fig 5: Graphs for the years 2007, 2008, 2009

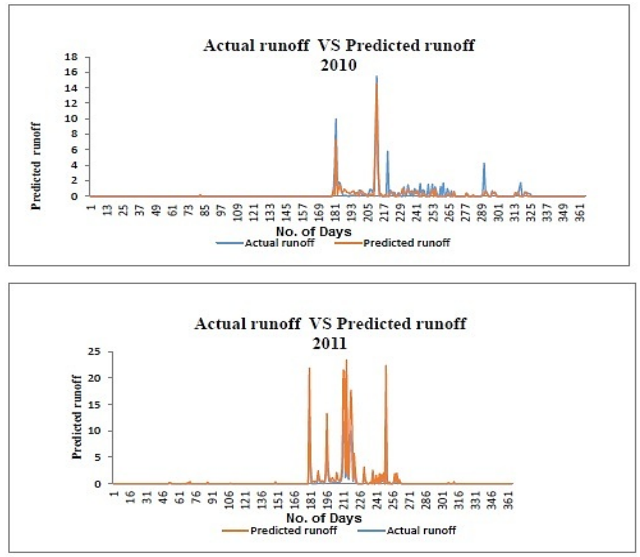


Fig 6: Graphs for the years 2010, 2011

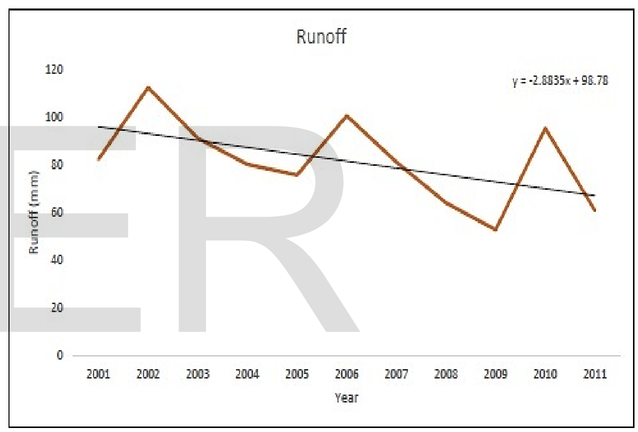


Fig 7: The above graph represents the total runoff over the study area of every year.

The maximum runoff occurred in the year 2002 and the minimum in 2009.

CONCLUSIONS

ANNs are very much capable of mapping the relationships between rainfall on a hydrological catchment and runoff from it. The performance of a data-driven approach such as ANN techniques, however, is obviously very dependent on the data that is used. The application of artificial neural network (ANN) methodology for modelling events of rainfall-runoff in a medium size catchment of the Godavari River in Telangana is presented.

Various combinations of the flood events have been considered by the data used for training. The results obtained in the present study have been able to demonstrate that the ANN models are able to provide a good representation of an event-based rainfall-runoff process. Trend line fitted for yearly observed runoff reflected a decreasing trend given by the equation y = -71.241x + 8690.4. The trend lines of both yearly rainfall and yearly observed runoff showed decreasing trend over the study area. Validation of the results of the model has shown good congruence with the observed data.

REFERENCES

1. Dooge, J.C.I., 1977. Problems and methods of rainfall-runoff modelling. In: Ciriani, T.A., U. Maione and J.R. Wallis (Eds.). Mathematical Models for Surface Water Hydrology: The Workshop Held at the IBM Scientific Centre, Pisa. Wiley, London, pp: 71-108.

2. Hsu, Kuo-lin, H.V. and Gupta, S. Sorooshian, 1995. Artificial neural network modelling of the rainfall-runoff process. Water Resource Res., 31 (10): 2517-2530.

3. Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling. J. Hydrol., 199: 272-294.

4. Kumar, A.R.S., Sudheer, K.P., Jain, S.K., and Agarwal, P.K. (2005), Rainfall-runoff 741 modelling using artificial neural networks: comparison of network types. Hydrological 742 Processes, 19 (6), 1277-1291.

5. Govindaraju R. S. (2000). “Artificial Neural Networks in Hydrology, Part I: Preliminary Concepts.” J. of Hydro Eng., 5(2): 115-123.

6. Buch A. M., Mazumdar H. S., and Pandey P. C. (1993). “A case study of runoff simulation of a Himalayan glacier basin.” Proc. Inter. Conf. on Neural Networks, 971-974.

7. Anmala, J., Zhang, B. & Govindaraju, R. S. (2000) Comparison of ANNs and empirical approaches for predicting watershed runoff. J. Water Resource Plan. Manage. 126(3), 156–166.