CS/AR-2/2004

HYDROLOGICAL MODELING OF STREAMFLOWS USING ARTIFICIAL NEURAL NETWORKS FOR SINDH BASIN

आपो हिष्ठा मयोभुवः

NATIONAL INSTITUTE OF HYDROLOGY
JAL VIGYAN BHAWAN
ROORKEE-247 667 (UTTARAKHAND)

2004

DIRECTOR
Dr. K.D. Sharma

COORDINATOR
Dr. A.K. Bhar

STUDY GROUP
Sh. T. Thomas
Sh. R.K. Jaiswal
Dr. Surjeet Singh

PREFACE

The effective management of the water resources requires input from hydrological studies, mainly in the form of estimation or forecasting of the magnitude of the hydrological variables like rainfall and runoff. Deterministic and stochastic approaches are available to make these forecasts. In situations where information is needed only at specific sites in a river basin and where adequate meteorological and hydrological information is not available, the time and effort required in developing such models may not be justified. Site specific and simple neural network models seem attractive to apply under these circumstances. The current trend seems to be to model the data rather than the underlying physical process. An artificial neural network is a flexible mathematical structure, which is capable of identifying complex non-linear relationships, between input and output data sets, and has been found to be useful and efficient in problems for which the characteristics of the process are difficult to describe using physical equations. The success with which ANN has been used to model dynamic system in other fields of science and engineering, suggests that the ANN approach may prove to be an effective and efficient way to model the rainfall runoff process.

This report, titled *'Hydrological Modeling of Stream flows Using Artificial Neural Networks for Sindh Basin '*, presents a research study conducted to develop a rainfall-runoff model using ANN approach which has been trained and tested for the Sindh River Basin in Madhya Pradesh. The study demonstrates the applicability of ANN approach in developing effective non-linear models of the rainfall runoff process without the need to explicitly represent the internal hydrologic structure of the watershed. The study has been conducted by Sri. T. Thomas, *Scientist 'B'*, Sri. R. K. Jaiswal, *'PRA'* and Dr. Surjeet Singh, *Scientist 'C'* of Ganga Plains South Regional Centre, Sagar under the guidance of Dr. A. K. Bhar, *Scientist 'F'*, National Institute of Hydrology, Roorkee.

K. D. Sharma
Director

ABSTRACT

The problem of transformation of rainfall to runoff has been a very active area of research throughout the evolution of the subject of hydrology. The relationship of rainfall-runoff is known to be highly non-linear, complex, time varying and spatially distributed. It involves many highly complex components such as interception, depression storage, infiltration, overland flow, interflow, percolation, evaporation and transpiration. Transformation of rainfall to runoff is to be understood in order to forecast the stream flows for water supply, flood control, irrigation, drainage, water quality, power generation and wild life propagation. Every model is an attempt to capture the essence of the complex hydrologic system in a meaningful and manageable way, but it is important that the conceptualization involves considerable degree of simplification. Conceptual rainfall runoff models are designed to approximate within their structures the general internal sub-processes and physical mechanisms, which govern the hydrologic cycle. Conceptual models provide daily, monthly or seasonal estimates of the stream flow for short-term and long-term forecasting by mathematically formulating the entire physical process in the hydrologic cycle. A 6-parameter conceptual model of simple structure has been developed to represent the rainfall-runoff relationship. The efficiency of the model varies between 0.67 and 0.83 during calibration and between 0.76 and 0.82 during validation. The percentage difference in volume between the observed and computed annual flows vary between – 5.84 % and 25.65 %. The correlation coefficient between the observed and computed flow series varies between 0.90 and 0.96

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

In many parts of the world, rapid population growth, urbanization, and industrialization have resulted in the increased demand for water. These same pressures have resulted in altered watersheds and river systems, which have contributed to a greater loss of life and property damages due to flooding. It is becoming increasingly critical to plan, design, and manage water resources systems carefully and intelligently. Many years, hydrologists have attempted to understand the transformation of precipitation to runoff, in order to forecast stream flow for purposes such as water supply, flood control, irrigation, drainage, water quality, power generation, recreation, and fish and wildlife propagation. Therefore the efficient water resources development and management is necessary for any country for its economic growth. In India more than 80 percent of rainfall occurs in the four monsoon months from June to September. More than 3000 major and medium multipurpose reservoir projects have already been constructed in India to regulate the stream flow for various uses.

Modeling rainfall-runoff at the watershed scale is important for water resources management, safe yield computations and design of flood control structures. A model of the rainfall-runoff relationship is an essential component in the process of evaluation of water resource projects for which, most of the time a sufficient length of record of flow may not be available whereas the rainfall data may be available. This indeed is a major problem for developing countries like India where the use of a well-equipped measuring system is of recent origin. Even when such concurrent data are available, the length of record is generally very limited. Hence a rainfall-runoff model that can yield sufficiently accurate results with such short lengths of data is desirable and useful.

The response of a watershed to precipitation is complicated by various hydrologic components that are distributed within it in a heterogeneous manner. Watershed runoff depends on geomorphologic properties such as topology, vegetation and soil type of the watershed and the climatic factors such as precipitation, temperature etc. The influence of all these factors on the runoff is not understood clearly. The transformation of precipitation to watershed runoff involves many highly complex components, such as interception, depression storage, infiltration, overland flow, interflow, percolation, evaporation, and transpiration. Conceptual models provide daily, monthly, or seasonal estimates of stream flow on a continuous basis and are designed to approximate within their structures, the general internal sub-processes and physical mechanisms that govern the hydrologic cycle. The entire physical process in the hydrologic cycle is mathematically simulated in conceptual models, which usually incorporate simplified forms of physical laws and are generally non-linear, time invariant and deterministic with parameters that are representative of watershed characteristics. The accuracy of model predictions is very subjective and highly dependent on the user's ability, knowledge, and understanding of the model and of the watershed characteristics.

Artificial neural networks (ANN) have found increasing applications in various aspects of hydrology. ANN refers to computing systems whose central theme is borrowed from the analogy of biological neural networks. An ANN is described as an animation-processing system that is composed of many non-linear and densely interconnected processing elements or neurons. Artificial neural networks are also referred to as "neural nets", "artificial neural systems", "parallel distributed processing systems", and "connectionist systems." ANN is a non-linear mathematical structure, which is capable of representing arbitrarily complex non-linear processes that relate the inputs and outputs of any system. Previous studies have shown the potential of ANN for modeling rainfall-runoff relationships over the watershed. Neural networks are quite adept at modeling problems in pattern recognition and control applications. However, most implementations of the neural networks have been incapable of explaining their reasoning in a comprehensible manner because the knowledge of ANN is embedded in connection strengths and threshold values of the weights in an obscure fashion. Thus neural networks remain much like a black-box model. ANN approach is faster compared with its conventional compatriots, robust in noisy environments, flexible in the range of problems it can solve, and highly adaptive to the newer environments. Due to these established advantages, currently the ANN has numerous real world applications. The ANN approach may prove to be an effective and efficient way to model the rainfall runoff processes in situations where explicit knowledge of the internal hydrologic sub- processes is not required since it attempts to take care of the non-linearity involved in the transformation process of rainfall to runoff and due to its ability to generalize patterns in noisy and ambiguous input data and to synthesize a complex model without a priori knowledge or probability distributions. Because an ANN model is calibrated using automatic calibration techniques, it eliminates subjectivity and lengthy calibration cycles.

The main function of ANN paradigms is to map a set of inputs to a set of outputs. The following advantages of a neural network can be usefully exploited in constructing models of the water resource processes (Thirumalaiah et al., 1998): a) When the underlying problem is either poorly defined, complex or not clearly understood. b) When specific solutions do not exist to the problem posed. c) When prior knowledge of the underlying process is not known beforehand. d) Most suitable for dynamic forecasting problems because the weights involved can be updated when fresh observations are made available. d) Small errors in the input do not produce significant change in the output because of distributed processing.

In this study, ANN algorithms were used to model the daily rainfall-runoff relationship for the Sind river basin, Madhya Pradesh, India. The study demonstrates the applicability of ANN approach in developing effective non-linear models of rainfall runoff process without the need to explicitly represent the internal hydrologic structure of the watershed.

2.0 REVIEW OF LITERATURE

The concept of the artificial neurons was first introduced by McCulloch and Pitts *(Maier and Dandy, 2000)* in 1943 in biophysics. The application of neural networks to solve civil engineering problems began in late 1980's (*Flood and Kartam, 1994 a, b*). The last decade has witnessed many applications of neural networks in water resources. These include rainfall forecasting (French et al. 1992), multivariate modeling of water resources time series *(Raman and Sunil Kumar, 1995)*, modeling of rainfall-runoff process *(Hsu et al., 1995 and Crespo and Mora, 1993)*, and flow forecasting *(Karunanithi et al.,1994; Zealand et al., 1999)*. Neural networks learn from experience and then perform recognition without definition. The architecture of the ANN is designed by weights between neurons, a transfer function that controls the generation of output in a neuron, and learning laws that define the relative importance of weights for input to a neuron *(Caudill, 1987)*. When developing the ANN models, the statistical distribution of the data need not be known *(Burke, 1991)* and non-stationarity in the data such as trends and seasonal variations are implicitly accounted for by the internal structure of the ANN *(Maier & Dandy, 1996)*. The feed-forward neural networks with back-propagation learning algorithm are the most widely used neural networks *(Free-man and Skapura, 1991; Anderson, 1995; Hornik et al., 1989; Lippman, 1987)*. Details on the ANN and the back propagation-training algorithm can be found in *(Maren et al., 1990; Freeman and Skapura, 1991; Anderson, 1995; Dhar and Stein, 1997)*.

Even through the development of the feed-forward ANN in its present form was fore shadowed by the work of *Werbos, (1974)* and reinvented separately by *Rumelhart et al., (1986)*, its application in rainfall-runoff modeling is of recent origin. Two types of networks, namely a feed-forward multi-layer perceptron (MLP) network (Rumelhart et al., 1986) and Counter propagation network (CPN) (Hecht-Nielsen, 1987) are usually used for pattern mapping problems. *Chang and Tsang, (1992)*, compared the multiple linear regressions and ANN approaches to modeling snow water equivalent and reported that ANN yielded better results. *Karunanithi et al., (1994)*, used a cascade correlation algorithm for predicting the flow for the time dependence of the phenomenon. In a discussion of this work, *Zhu and Fujitha, (1994)*, compared the performance of fuzzy logic in rainfall-runoff modeling and applied a feed-forward ANN model in predicting a 3–hour lead runoff. *Lorrai and Sechi, (1995)*, verified the possibility of utilizing ANN to predict rainfall-runoff relation when only the information about the variation of the basic input variables, namely rainfall and temperature is available. *Raman and Sunilkumar, (1995)* used artificial neural network for the synthesis of inflows **to** two reservoirs Mangalani and Pothundy located in the Bharathapuzha, Kerala. *Smith and Eli, (1995)* used a feed-forward network to predict the runoff peak value and the time to peak for spatially distributed rainfall. However, the training (calibration) was attempted using simulated data. Even though this exercise demonstrated ANN's capability in learning the rainfall-runoff relationship through the training procedure, it is of limited practical use.

Hsu et al., (1995), showed that a non-linear ANN model provided a better representation of the rainfall-runoff relationship of the medium sized Leaf river basin near Collins, Mississippi than linear ARMAX (autoregressive moving average with exogenous inputs) time series approach or the conceptual SAC-SMA (Sacramento soil moisture) model. *Raman and Chandramouli, (1996)* used feed forward back propagation neural network for deriving better operating policy for the Aliyar Dam in Tamil Nadu and compared the operating policies using three models, dynamic programming (DP) model, stochastic dynamic programming model (SDP) and standard operating policy (SOP). General operating policies were derived using neural network model (DPN) from the DP model and it was concluded that the neural network procedure based on the dynamic programming algorithm provided better performance than the other models. *Kao, (1996)* used artificial neural networks to determine the drainage pattern from DEM data of to a sub-watershed located on Chin-Mei Creek, Taipei County, Taiwan using the back propagation algorithm was used for training the neural network model. *Carriere et al., (1996),* designed a virtual runoff hydrograph system based on ANN and obtained good correlation between the observed and predicted data.

Minns and Hall, (1996) used a feed-forward ANN to predict runoff from rainfall data, however, the purpose of the paper was to demonstrate the learning ability of the network, namely learning from example rather than applying it to a practical situation. The black-box type modeling of the rainfall-runoff relation can be classified under the category of pattern mapping. ANN have also been applied for prediction of water quality parameters and real-time forecasting of water quality *Dandy and Maier, (1996). Cheng and Noguchi, (1996),* obtained better results modeling the rainfall-runoff process with ANN using previous rainfall, soil moisture deficits and runoff values as model inputs.

Yang et. al., (1997) developed an artificial neural network (ANN) model to simulate fluctuations in mid span water table depths and drain outflows as influenced by daily rainfall and potential evapo-transpiration rates. *Dawson and Wilby, (1998),* used the multi-layered feed forward network structure with back propagation algorithm for rainfall-runoff modeling. They applied it for two flood prone catchments in UK and compared the performance of ANN with conventional flood forecasting systems. *Jain, Das and Srivastava, (1999)* used artificial neural network for reservoir inflow prediction and the operation for Upper Indravati Multipurpose project, Orissa and they found that ANN was suitable to predict high flows and auto-regressive integrated moving average time series model was suitable to predict low flows. *Thirumalaiah and Deo, (1998)* used artificial neural networks in real time forecasting of water levels at a given site continuously throughout the year based on the same levels at some upstream gauging station and/or using the stage time history recorded at the same site and concluded that the continuous forecasting of a river stage in real time sense was possible through the use of neural networks.

Maier and Dandy, (1999) used six methods to optimize the connection weights of feed forward ANN. These were the generalized delta (GD) rule, the normalized cumulative delta (NCD) rule, the delta bar delta (DBD). *Zealand, Burn and Simonovic, (1999)*, used the artificial neural networks to forecast the short-term stream flow for Winnipeg river system in north-west Ontario, Canada. *Saji Kumar and Thandeswara, (1999),* concluded that an ANN was the most efficient of the models tested for calibration periods as short as 6 years. *Tokar et al., (1999),* used the back propagation feed-forward neural network to forecast daily runoff as a function of daily precipitation, temperature and snowmelt for Little Patuxent river watershed in Maryland and compared the results with existing techniques like statistical regression and conceptual model and concluded that ANN proved better accuracy in the forecasts. *Maier and Dandy, (2000)* reviewed the works done in the application of artificial neural networks to predict and forecast water resources variables. A review of 43 papers in the prediction and forecasting of water resources variables were considered for laying down the procedure to model the ANN structure.

Elsorbagy et al., (2000), conducted a performance evaluation of artificial neural networks for spring runoff prediction in Red river valley, southern Manitoba, Canada by comparing the results with the linear and non-linear regression techniques. *Tokar and Markus, (2000),* compared the ANN models with the traditional conceptual models in predicting watershed runoff as a function of rainfall, snow water equivalent and temperature. Three basins with different climatic and physiographic characteristics namely, Fraser river in Colarado, Raccoon Creek in Iowa and Little Patuxent river in Maryland. For the Fraser river basin, the ANN technique was used to model the monthly stream flow and compared with the conceptual water balance model (WATBAL). The daily rainfall-runoff process was modeled in the Raccoon Creek watershed and compared to the Sacramento soil moisture accounting model (SAC-SMA). The daily rainfall-runoff process was also modeled in the Little Patuxent river basin and the training and testing results were compared to those of a simple conceptual rainfall runoff model (SCRR). Most of the studies carried out using the ANN technique indicate that ANN can be a powerful tool in modeling the precipitation runoff process for various time scales, topography and climate patterns.

3.0 OVERVIEW OF ARTIFICIAL NEURAL NETWORKS

The speed and efficiency with which the human brain processes information, has been fascinating science, for quite a long time. The quest to understand these processes and solve the associated types of problems, have led to the development of artificial intelligence (AI). Among the many fields of AI, the artificial neural network (ANN) is gaining a status prime importance even in other field of engineering, owing to its interesting properties such as "learning" from examples, the ability to represent non-linearity by means of a smaller number of parameters and the least requirement of information regarding the process to be modeled. Artificial neural networks are massively parallel-distributed adaptive networks of simple non-linear computing elements called neurons, which are intended to attract and model some of the functionality of the human nervous system in an attempt to partially capture some of its computational strengths. It resembles the brain in two aspects: i) knowledge is acquired by the network through a learning process. ii) Inter-neuron connection strengths known as synaptic weights are used to store the knowledge. The common task that can be accomplished both by the brain and ANN may be categorized into different groups such as pattern association, mapping and clustering. The pattern association is the process of linking an input or an output, depending on whether it is an auto-associative or a hetero-associative neural network, to the corresponding input, by means of the inter-neuron connection strengths. In contrast, in pattern mapping, the under-lying input–output relationship is captured by means of a suitable "learning" strategy. The pattern clustering deals with grouping of patterns based on the proximity of each pattern with others *(Thandaveswara and Sajikumar, 1998)*.

An ANN is an information processing system composed of many non-linear and densely interconnected processing elements or neurons. In most general form, the neural network can be viewed as comprising of eight components namely, (a) neurons, (b) activation state vector, (c) signal function, (d) pattern of connectivity, (e) activity aggregation rule, (f) activation rule, (g) learning rule and (h) environment. There can be three type of *neurons* namely, input neurons which are designated to receive external stimuli presented to the network, output neuron which gives the output signals generated from the network and hidden neurons which compute intermediate functions and their states are not available to external environment. The *activation state vector* is a vector of the activation level x_i of individual neurons in the neural network, $X = (x_1, \dots\dots, x_n)^T \; R^n$. The *signal function* generates the output signal of the neuron based on its activation. The common signal functions are binary threshold, linear threshold and sigmoid functions. Most networks are fixed-homogeneous, in the sense that all neurons within a field or layer have the same signal function. The *pattern of connectivity* essentially determines the interconnection architecture or the graph of the network. The memory of the network resides in the connections and it is the connections together with the neuron signal functions that determine the global behavioral properties of the network and thus the function it performs. The *activation rule* is a function that determines the new activation level of a neuron on the basis of its current activation and external inputs. The learning rule provides a means of modifying connection strengths based on both external stimuli and network performance. The *environment* within which the neural networks can operate could be noiseless (deterministic) or noisy (stochastic).

3.1 Architecture of the Network

A neural network can be viewed a weighted direct graph in which the artificial neurons are nodes and directed weighted edges represent connections between neurons. The architecture in which the local group of neurons can be connected may be either of (a) *Feed forward architecture* – in which the network has no loops, or (b) *Feedback architecture* – in which loops occur in the network because of feedback connections. Different network architectures yield different behavioral patterns of varying complexity.

The neurons in a network are arranged in groups called layers and each neuron in a layer operates in logical parallelism. Information is transmitted from one layer to others in serial operation (Hecht-Nielsen 1990). A network can be composed of one or many layers. The architecture of an ANN is designed by weights between neurons, a transfer function that controls the generation of output in a neuron, and learning laws that define the relative importance of weights input to a neuron (Caudill 1987).

3.2 Multilayered Feed Forward Networks

The multi-layered feed forward network is shown in Fig. 3.1.The input layer has n-linear neurons that receive real valued external inputs in the form of an n-dimensional vector in R^n. This layer also includes an additional bias neuron that receives no external input but generates a signal $+1$ that feeds all bias connections of the neurons of the hidden layer. The hidden layer has q-sigmoid neurons that receive signals from the input layer. A bias neuron has also been added in the hidden layer to generate a $+1$ signal for bias connections of the output layer neurons. The output layer comprises of p-sigmoid neurons. The neuron layers compute in a feed forward fashion, i.e., the signals from one layer of the neurons act as inputs to the next layer, and so on. Finally the network signals that emanate from the last layer of neurons comprise a p-dimensional vector of real numbers in vector R^p. The neural network thus maps a point in R^n (the input space) to a point in R^p (the output space). The most distinctive characteristic of an ANN is its ability to learn from examples. Learning (or training) is defined as self-adjustment of the network weights as a response to changes in the information environment. When a set of inputs is presented, a network adjusts its weights in order to approximate the target output (observed or measured output) based on certain algorithms.

3.3 Back Propagation Learning Algorithm

Back-propagation is the most commonly used neural network algorithm in the field of water resources applications. In the back-propagation algorithm, the network weights are modified by minimizing the error between target and computed outputs. In feed forward back-propagation networks,

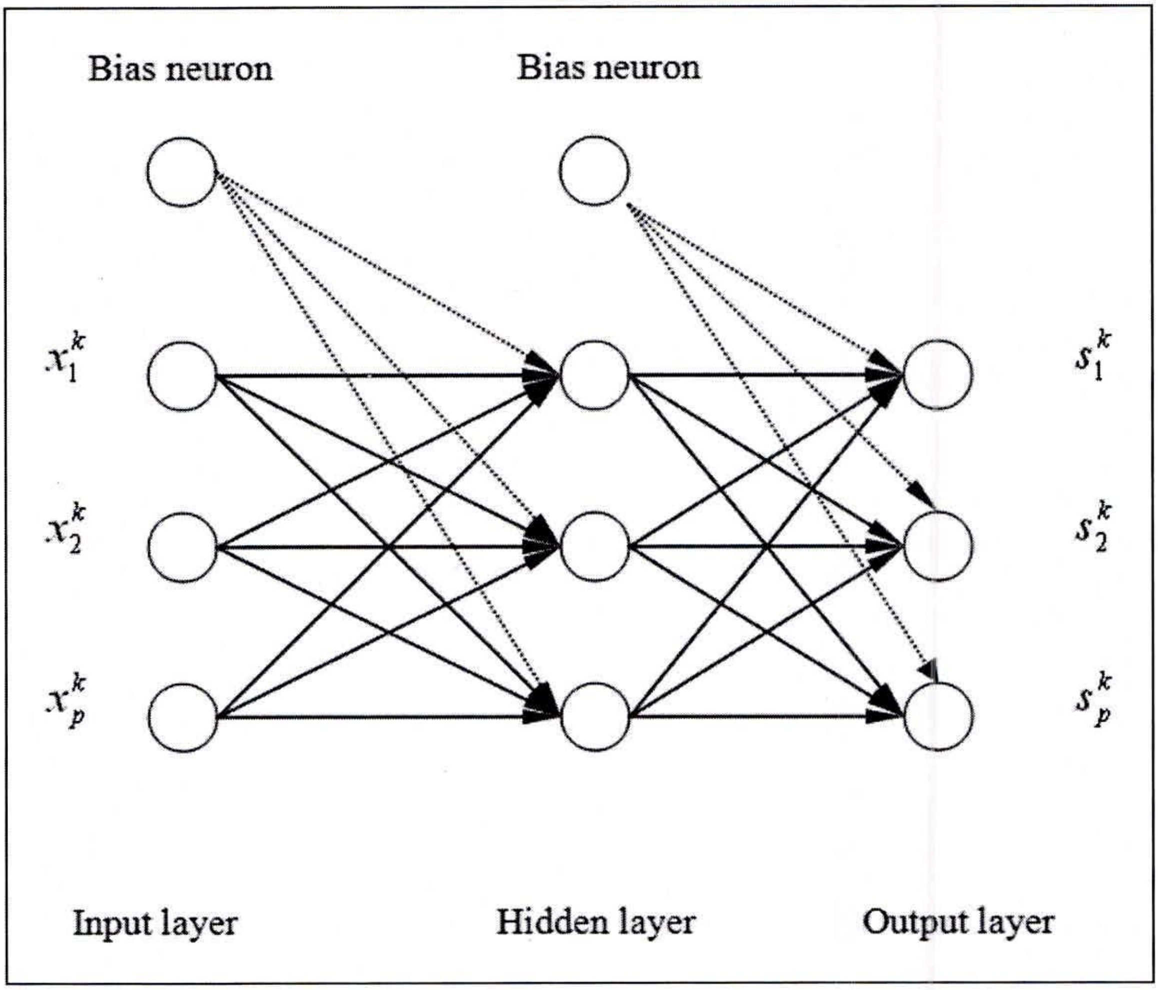

Fig.3.1: Schematic diagram of a feed-forward back propagation network

the information is processed in the forward direction from the input layer to the hidden layer(s), and then to the output layer. The objective of a back-propagation network is to find the weight that approximates the target values of the output with a selected accuracy. The least-mean square error method, along with the generalized-delta rule, is used to optimize the network weights in back-propagation networks. The gradient-descent method and the chain rule of derivative are employed to modify the network weights *(Rumelhart et al., 1986).*

Training is composed of two major phases namely, the forward propagation step followed by a back propagation step. In the forward propagation step, firstly the input data is presented to the input layer of the network, which is multiplied by the initial weights, and then the weighted inputs are added by simple summation to yield the *net input* to each neuron. The *net input* of a neuron is then passed through an activation function or transfer function to produce the output of a neuron. After the output the neuron is transmitted to the next layer as input, this procedure is repeated until the output layer is reached. So in each successive layer, every processing unit sums its inputs and then applies a transfer function to complete its output. The output layer of the network then produces the output of the network. The backward propagation step begins with the comparison of the network's output pattern to the target vector when the error between the output of the network and the target outputs are computed. If an error is higher than a selected value, the backward propagation step then calculates the error values for the hidden units and makes changes for their incoming weights, starting with the output layer and moving backward through the successive hidden layers. In this back propagation step the network corrects its weights in such a way as to decrease the observed error. The training stops when the error is within the prescribed tolerance levels.

In the back-propagation networks, the modification of the network weights is accomplished with the derivative of the activation function. Generally the continuous transfer functions namely sigmoid or hyperbolic-tangent functions are used as activation function. The modification of weights in the output layer is different from the modification of weights in the hidden layers. In the output layer, the target outputs are provided, whereas in the intermediate layers, target values do not exist. Therefore, back-propagation uses the derivatives of the objective function with respect to weights in the entire network to distribute the error to neurons in each layer in the entire network. The back propagation algorithm can be described as follows:

The network considered is assumed to be homogeneous i.e. all neurons in a layer use similar signal functions. For linear neurons in the input layer

$$\delta(x) = x \tag{3.1}$$

where,
x is the activation and *(x)* is the signal
For sigmoid neurons in the hidden and output layers,

$$\delta(x) = \frac{1}{1+e^{-\lambda x}} \tag{3.2}$$

where,
= sigmoid gain scale factor, generally taken as one.

The set of Q training pairs are represented by

$$T = \{(X_k, D_k)\}_{k=1}^{Q} \quad X_k \; R^n, \; D_k \; R^p \tag{3.3}$$

where,

D_k = vector response desired

X_k = vector input to the network

The vector pairs in T are assumed to be samples of some unknown function $f: R^n R^p$ for the neural network to approximate. The gradient of the pattern error is employed to reduce the global error over the entire training set. Such weight changes are effected for a sequence of training pairs (X_1, D_1), (X_2, D_2), ……………, (X_k, D_k) from the training set. Each weight change perturbs the existing neural network slightly in order to reduce the error on the pattern in question. The k^{th} training pair (X_k, D_k) then defines the instantaneous error :

$$E_k = D_k - \delta(y_k) \tag{3.4}$$

where,

$$E_k = \left(e_1^k, \ldots\ldots\ldots, e_p^k\right) = \left(d_1^k - \delta\left(y_1^k\right), \ldots\ldots\ldots, d_p^k - \delta\left(y_p^k\right)\right)^T \tag{3.5}$$

The instantaneous sum squared error $_k$ is the sum of squares of each individual output error e_j^k, scaled by one-half:

$$\varepsilon_K = \frac{1}{2}\sum_{J=1}^{P}\left(d_j^k - \delta\left(y_j^k\right)\right)^2 = \frac{1}{2} E_k^T E_k \tag{3.6}$$

The mean square error , is computed over the entire training set T,

$$\varepsilon = \frac{1}{Q}\sum_{k=1}^{Q}\varepsilon_k \tag{3.7}$$

The error calculated above is used to compute the change in the hidden to output layer weights, and the change in input to hidden layer weights (including all bias weights), such that the global error measure gets reduced. The weights of the network are then updated in accordance with these weight changes as given below:

For Hidden to Output layer weights

$$\begin{aligned} w_{hj}^{k+1} &= w_{hj}^k + \Delta w_{hj}^k \\ &= w_{hj}^k + \eta\left(-\frac{\partial \varepsilon_k}{\partial w_{hj}^k}\right) \\ &= w_{hj}^k + \eta \zeta_j^k \delta\left(z_h^k\right) \end{aligned} \tag{3.8}$$

For Input to Hidden layer weights

$$w_{ih}^{k+1} = w_{ih}^{k} + \Delta w_{ih}^{k}$$

$$= w_{ih}^{k} + \eta\left(-\frac{\partial \varepsilon_k}{\partial w_{ih}^{k}}\right) \qquad (3.9)$$

$$= w_{hj}^{k} + \eta \zeta_{h}^{k} x_i$$

The learning rate in the back propagation algorithm with pattern update has to be kept small in order to maintain a smooth trajectory in weight space. An elegant way of increasing the learning rate while maintaining stability is to introduce a momentum term in Equations 3.8 and 3.9:

$$\Delta w_{hj}^{k} = \eta \delta_{j}^{k} \delta\left(z_{h}^{k}\right) + \alpha \Delta w_{hj}^{k-1} \qquad (3.10)$$

$$\Delta w_{ih}^{k} = \eta \delta_{h}^{k} x_{i}^{k} + \alpha \Delta w_{ih}^{k-1} \qquad (3.11)$$

where >0 is the momentum. When the weights in the network are updated in accordance with the momentum, the algorithm is called generalized delta rule; but if =0 the above equation reduces to standard back propagation learning rule without momentum.

3.4 Salient Properties of ANN

The salient features of the neural network include robustness, the capacity for associative recall, the capacity for function approximation and generalization. Neural networks are robust in the sense that they can tolerate significant distortions in inputs for which they are programmed. This property is called fault tolerance due to which the network performance continues to be satisfactory when the inputs are distorted. The robustness of neural networks stems from the fact that it stores information in a distributed fashion. Due to the capacity of associative recall, one concept invokes related memories. Because of the dense inter-connectivity and reinforcing structure embedded in the ensembles of connected neurons, associative recall becomes a natural capability of the neural network. Neural networks because of their learning capability are often referred to as adaptive function estimators. Neural networks have a model free estimation capability in the sense that they create internal representations through examples, without being supplied a mathematical model of how outputs depend on inputs. Also the neural networks are able to generalizations based on previously learnt pattern class information.

4.0 THE STUDY AREA

The Sindh river basin up to Madhikhera dam site has been considered for applying the artificial neural networks for modeling and forecasting the river inflows. The details about the study are given in the following sub-sections.

4.1 Sindh River Project

Harnessing of the Sindh river to utilize its available yield for irrigation and power purpose has engaged the attention of the erstwhile Gwalior state as early as in the year 1900. Adjoining the Sindh basin to its North is Parvati basin. River Parvati has been harnessed at two places by constructing reservoirs at Kaketo and Harsi. In the first phase of the project a diversion weir was constructed on river Sindh near village Mohini in Shivpuri district and a feeder canal 6.4 km long to divert the water to the existing Harsi reservoir. This was necessitated since the command area of the Harsi system, which was completed in 1935, developed fast and exceeded the designed irrigation area. The construction of the Mohini pickup weir was taken up in 1972 and project including the appurtenant works and feeder canal was commissioned in 1977. The second phase of the project envisages the construction of the Mohini Sagar (Marhikheda) dam across river Sindh at Marhikheda village 16 km. u/s of the existing Mohini pickup weir in Shivpuri district. The location of the dam is at 25°33'20" N latitude and 77°51'10" E longitude. The reservoir formed is designated as Mohini Sagar dam at Marhikheda.

4.2 Catchment Area and its Location

The Sindh basin is situated in the northern part of Madhya Pradesh and has its origin at village Gopi talai in Lateri tehsil of Vidisha district in Madhya Pradesh at an elevation of 335 m. above m.s.l. Sindh river which is a tributary of Jamuna, flows mostly through M.P. through the districts of Vidisha, Guna, Shivpuri, Gwalior, Datia and Bhind. It then joins river Jamuna near village Jagwanpur in Etawa district of Uttar Pradesh. The total length of the river from its origin to its confluence with Jamuna is 500 km. The river flows through narrow valleys and is joined by numerous tributaries. The important tributaries are Nawari, Chonch Tndernala, Khair, Aer, Barasi, Amar, Mahwar, Parwati, Noon and Yasuli. The river flows through Malwa plateau in Vidisha and Guna districts. In Shivpuri, it flows through a thick forest and hilly stretch and enters the plains after the town Narwar in Shivpuri district. The total drainage area up to the confluence with Jamuna River is 18,389 sq. km. and up to the proposed dam site is 5,540 sq. km. The average bed slope of the river is 0.754 m per km. in the first 129 km. and 1.9 m. per km. in next 80.265 km. The catchment area is elongated and narrow with a length of 152 km. and average width of 37 km. The catchment area lies between elevation 274 m. and 533 m. above mean level. The important tributaries joining river Sindh up to Mohini Sagar dam are Barahi nallah and Khair nallah on the left bank and Aer nallah, Inder nallah and Chonch nallah on the right bank. The location map of the study area is given in Fig. 4.1.

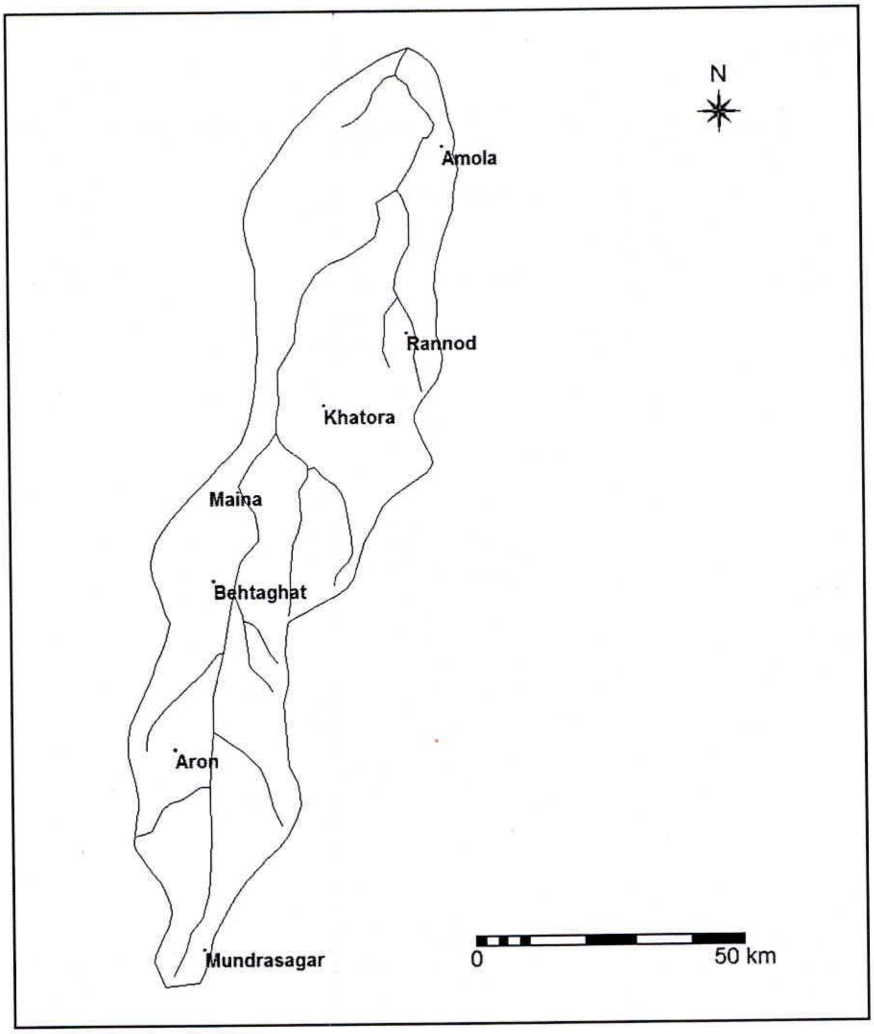

Fig. 4.1: Map showing Sindh river basin up to Madhikheda dam site

4.3 Climate

The northern region of Madhya Pradesh around Gwalior, Datia and Shivpuri districts is semi-arid and the climate in the area in hot and dry. There are three distinct types of climate. During summer months of March to June, which is hot and dry, and the temperature goes as high as 45.6° C. The rainy season from July to October which is humid and winter season from November to February, which is dry and cold and the temperature goes down to freezing point. The direction of the prevailing wind is generally southwest during the period of April to July. It is northwest during September to November. During the rest of the period, i.e. winter, it is generally calm. The wind speed is maximum during the summer months.

4.4 Rainfall

The monsoon generally arrives in the last week of June and continues till the end of September, receding thereafter by middle of October. Over 90% of the rain falls during the monsoon season and the rest 10% is distributed during the winter and summer months. There is spatial variation in the distribution of rainfall in the catchment and it is heavier near the head reaches of Vidisha and Guna districts reducing gradually towards Shivpuri district. The mean annual rainfall in the basin is 923.29 mm

4.5 Geology

The Sindh river flows through narrow valleys and at the reservoir site the hills rise rapidly from the river bed. The hills are covered with Vindhyan sand stone formation on the top. The river has cut its way through the sand stone formation at top into the granite gneiss on the sides and in the riverbed. On the left flank the hills are continuous without any break, but on the right flank the hills have gaps.

4.6 Soils and Land use

The catchment area in Vidisha and Guna districts lies in the Malwa Plateau and is covered with a top layer of black cotton soil and agriculture is being carried out. The catchment in Shivpuri district is hilly and is covered with thick forest.

4.7 Floods

The Sindh river flows in deep gorge and the banks are quite high which accounts for confining the floods within its banks. No case of serious damage due to floods in the river has been reported so far and there are no big towns on its banks. However the Sindh river being a tributary of river Jamuna, contributes to the floods of the Jamuna river in Uttar Pradesh.

4.8 Reservoir

The topography of the reservoir plan of Sindh River is such that the river flows through narrow valleys and practically the storage is confined in valleys and there are no big pockets. Sheet rock of granite formation outcrops in the riverbed. The flanks rise rapidly from them riverbed and are plain at the top the hills. The intake structure for the powerhouse will be in the left flanks of the reservoir. The dead storage of 113.7 M.cum (4,018 Mcft.) is fixed on the basis of site storage studies by C.W. & P.C. at Nandan site. The total love storage available both at Mohini Sagar and Mohini pick-up weir is 976.34 M. Cum. The water-spread area at F.R.L. of 346.285 m. above m.s.l. is 5679 hectares.

5.0 DATA AVAILABILITY

The present study intends to develop a rainfall-runoff relationship for the Sindh basin from the available historical data records, so as to develop effective water management policies to meet the demand from all sectors.

5.1 Rainfall

The Sind basin has seven rain gauge stations inside the catchment area namely, Mundrasagar, Aron, Behtaghat, Maina, Khatora, Rannod and Amola. These are concentrated mostly in upper and middle portions **of** the basin. There are breaks in continuity of data of some of the stations. After checking the data of all the stations for period of availability, all seven rain gauge stations were considered for the study. The daily rainfall data from 1992 to 2001 for the monsoon season have been considered for training and testing the developed network.

5.2 Stream flow

The Water Resources Department, Govt. of Madhya Pradesh maintains a gauge-discharge (G-D) site at the dam site intercepting a catchment area of 5540 sq. km.. Daily values stream flow data at this G-D site available from 1992 to 2001 for the monsoon season have been considered for training and testing the developed network except for 1994 for which data is not available. The study has been restricted to-monsoon season (June to October) alone, since the interest was to model the inflows in the basin using the developed model so as to assess the availability of the water in the dam.

5.3 Computation of Mean Aerial Rainfall

The mean aerial rainfall over the catchment was computed by the Thiessen Polygon method. The seven rain gauge stations within the study area were considered for computing the mean aerial rainfall in the catchment during the monsoon season from June to October. The catchment area was digitized and the Thiessen weight evaluated using ILWIS 3.0 (Integrated Land & Water Information System). The map showing the Thiessen Polygons is given as Fig. 5.1 and the Thiessen weights for the representing rain gauge stations in the basin are given in Table 5.1.

5.4 Processing and Analysis of Data

The rainfall data at seven rain gauge stations in and around the basin have been used in the study. The data was checked for consistency by comparing the records of the rainfall for the station under consideration with the rainfall at the surrounding stations. Gaps in the records were filled up. Similarly, the discharge data at the dam site were processed to check for errors and inconsistencies. The seasonal

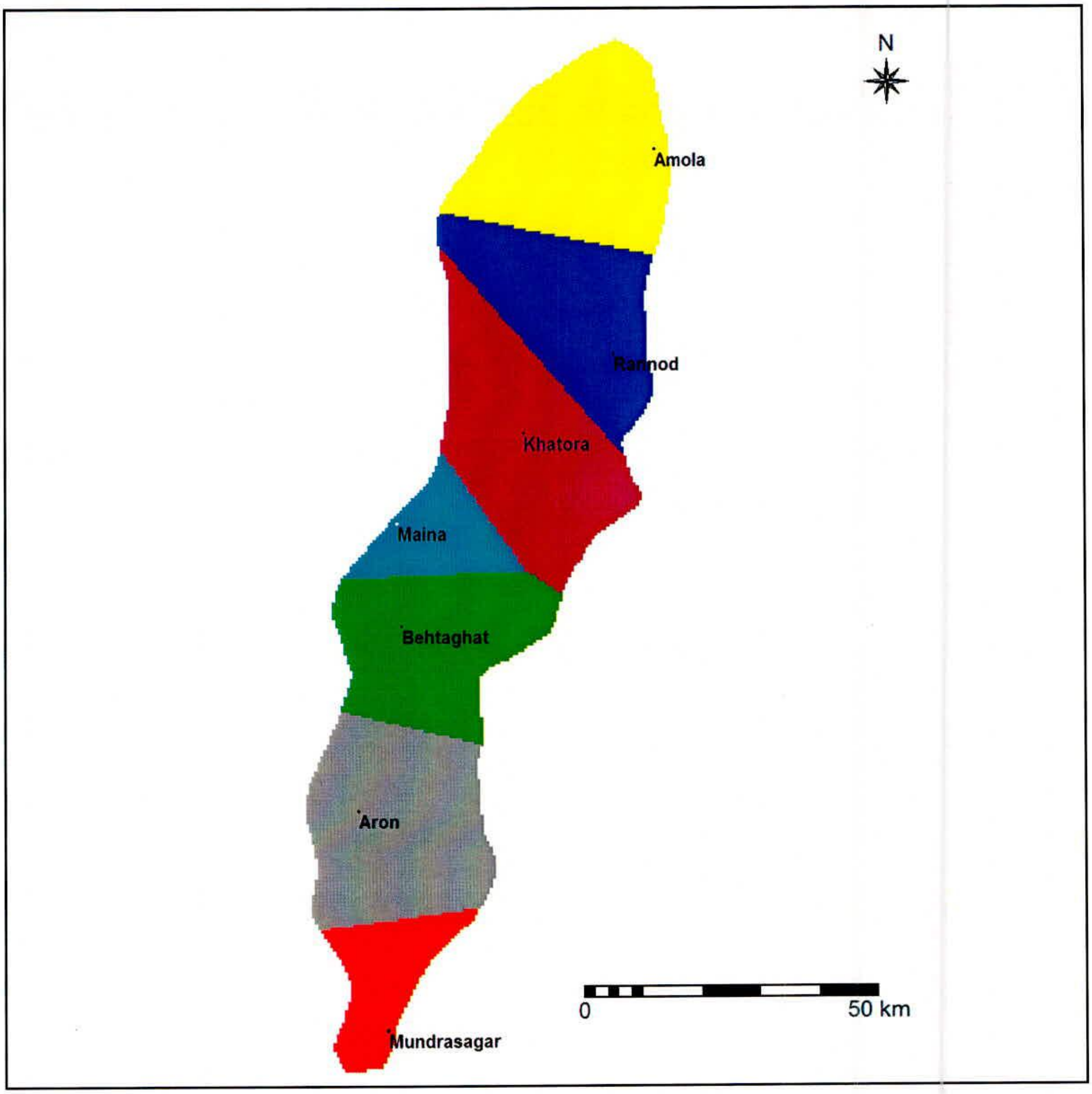

Fig. 5.1 : Thiessen polygon map of Sindh river basin upto dam site

rainfall runoff relation was developed for all the years under consideration. The runoff coefficient for the basin varied from 0.501 to 0.682. The monsoon season rainfall and runoff along with the runoff coefficients are given in Table-5.2. The variation of runoff coefficient with the seasonal rainfall is given in Fig. 5.2.

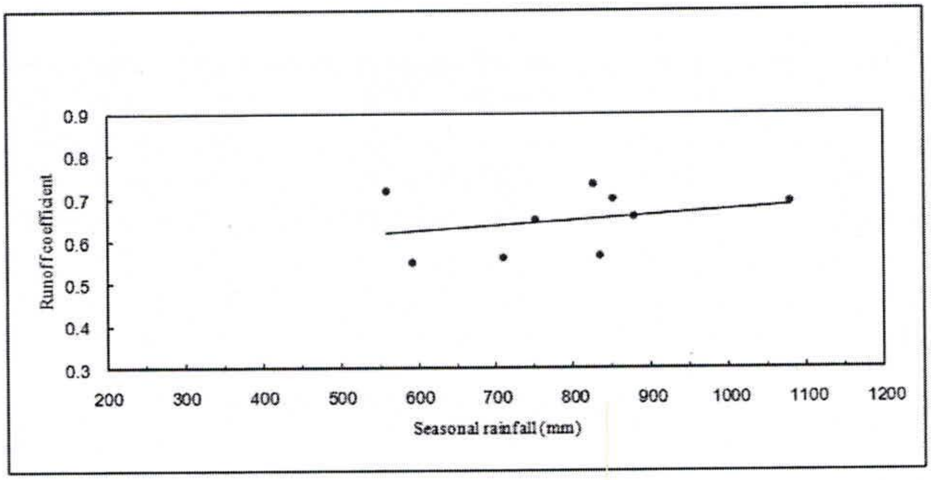

Fig. 5.2 : Variation of runoff coefficient with rainfall

6.0 MODEL DEVELOPMENT

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 or layers, where data are processed; and the output-layer, where the results of given input are produced. *When applying neural networks to modeling, a number of decisions must be made. It is imperative to choose an appropriate neural network structure in terms of input vector and output vector, apart from the hidden neurons. Determination of appropriate network architecture is one of the most important and also one of the most difficult tasks in the model building process. Unless carefully designed an ANN model can lead to over parameterization resulting in unnecessarily large network. Secondly, one must choose an appropriate training algorithm and select suitable training and validation periods or data sets. Also one must decide how to pre-process and post-process the input and output data. While some of these operations may be automated using appropriate modifications to training algorithms, many decisions must still be made through a process of trial and error.* There are multitudes of different types of ANN. The present study has employed the feed-forward back-propagation neural network.
The steps involved in the identification of a dynamic model of a system are:

(a) selection of the input and output data suitable for calibration and validation,
(b) selection of a model structure and estimates of its parameters, and
(c) validation of the identified model.

This study involves the development of a suitable ANN model based on the back propagation feed forward artificial neural network and assessing the accuracy of the model in simulating the flows with independent test data. In the study only monsoon season (June to October) data were used for calibration as well as validation, the data for the years 1992 to 1998 were used for calibration of ANN models and the models were validated for the years 1999 to 2001. The historical flow series from 1992 to 2001 is shown in Fig. 6.1.

6.1 ANN Model Identification

The ANN model structure is ideally suited for modeling highly non-linear input relationship such as those encountered in the transformation of rainfall to runoff. The main objective of the study was to use an ANN to simulate the stream flow from the available distributed rainfall and discharge data. Most of the previous work considered rainfall data averaged over the basin scale; this has the advantage of reducing the number of input variables to the network. As reported by Minns and Hall (1996), rainfall information alone is not sufficient to compute flow rate, since the state of the basin plays an important role in determining flow rate behavior. For this reason, flow data at certain time intervals before the time of predictions have been used as additional input information to the network. The selection of the number or previous flow data as input to the network was done by statistical analysis as briefed below.

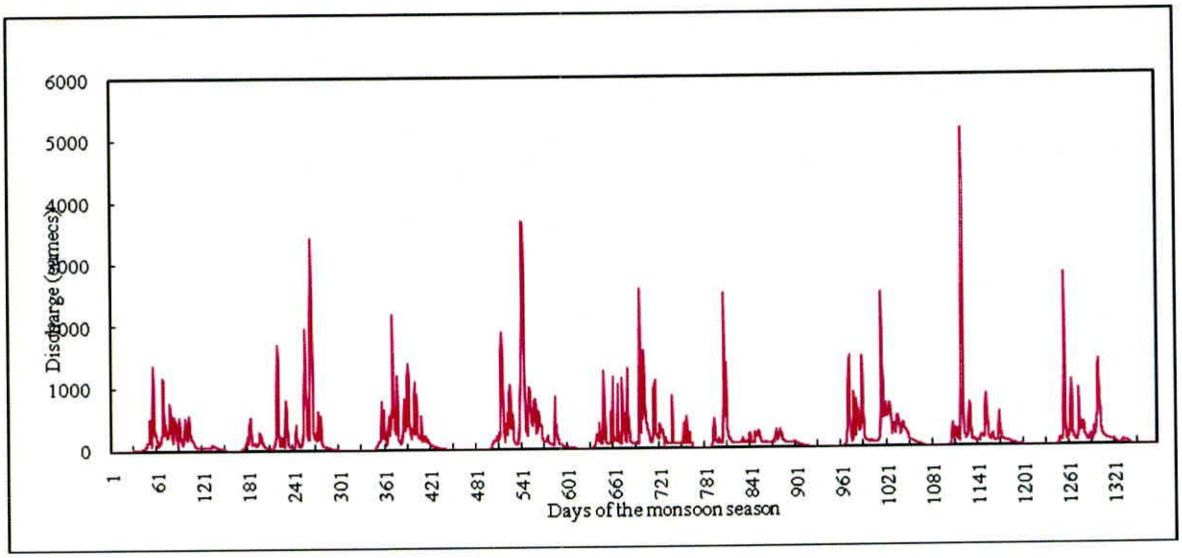

Fig. 6.1 : Historical flow series for the years 1992-2001

6.1.1 Standardization of time series

For the historical flow series, the daily means and standard deviations have been calculated for the monsoon season and are shown in Fig. 6.2 and Fig. 6.3 respectively. Daily mean flow was lower at the beginning and end of the season as compared to that of the mid of the season. The daily standard deviation values was also lower at the beginning and end of the season as compared to that of the mid of the season.

A time series may often contain periodic components that tend to repeat over a period of time intervals, due to astronomic cycles. The behavior of time series is known as a periodicity, which means that the statistical characteristics change periodically within the year. Within the year periodicity is due to the annual revolution of the earth around the sun, by the moon and daily rotation of the earth. These seasonal effects are repeated in the same time in each year and are thus deterministic. Seasonality can be removed by pre-whitening i.e. by standardizing and removing the periodicity. The periodic component can be removed from the time series as given by,

$$z_{t,\tau} = \frac{x_{t,\tau} - \bar{x}_{\tau}}{\sigma_{\tau}} \qquad 6.1$$

where,

$z_{v,}$ = standardized time series

$\bar{x}_{\tau}$ *and* σ_{τ} are the mean and standard deviation of the τ_{th} day ;

τ time interval within the year

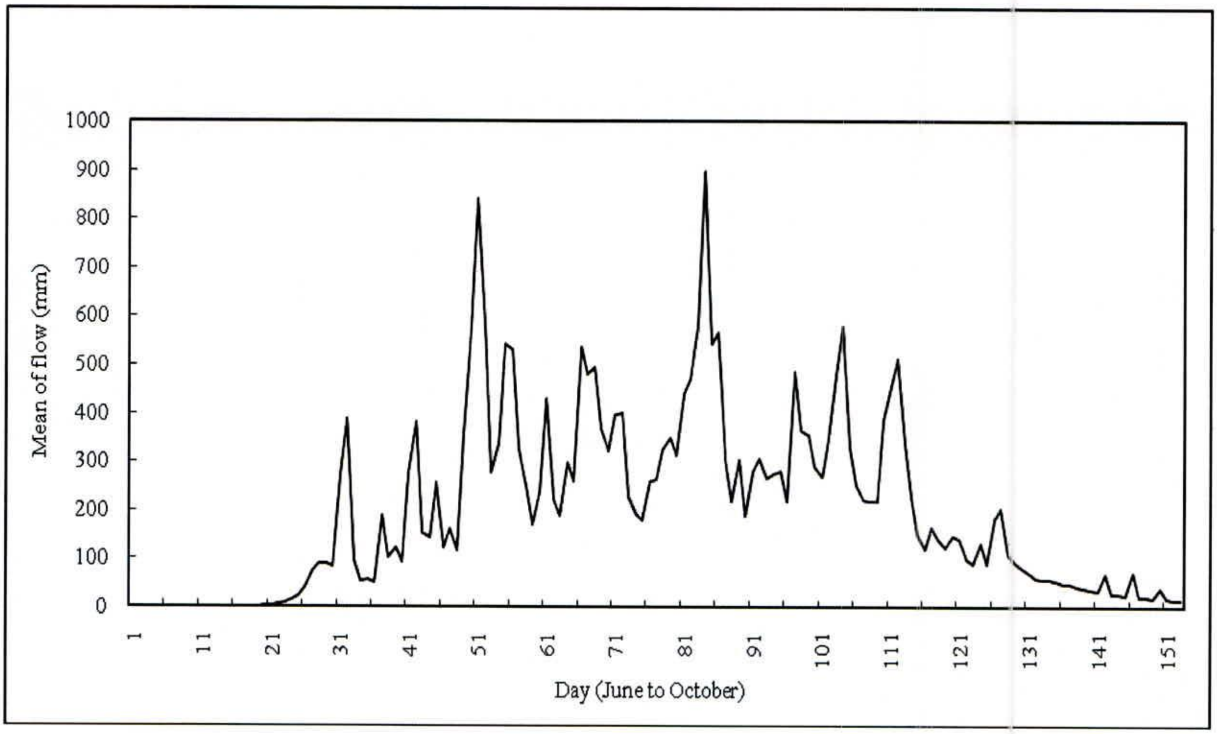

Fig. 6.2 Plot of sample mean of flow derived from data of 1992-2001

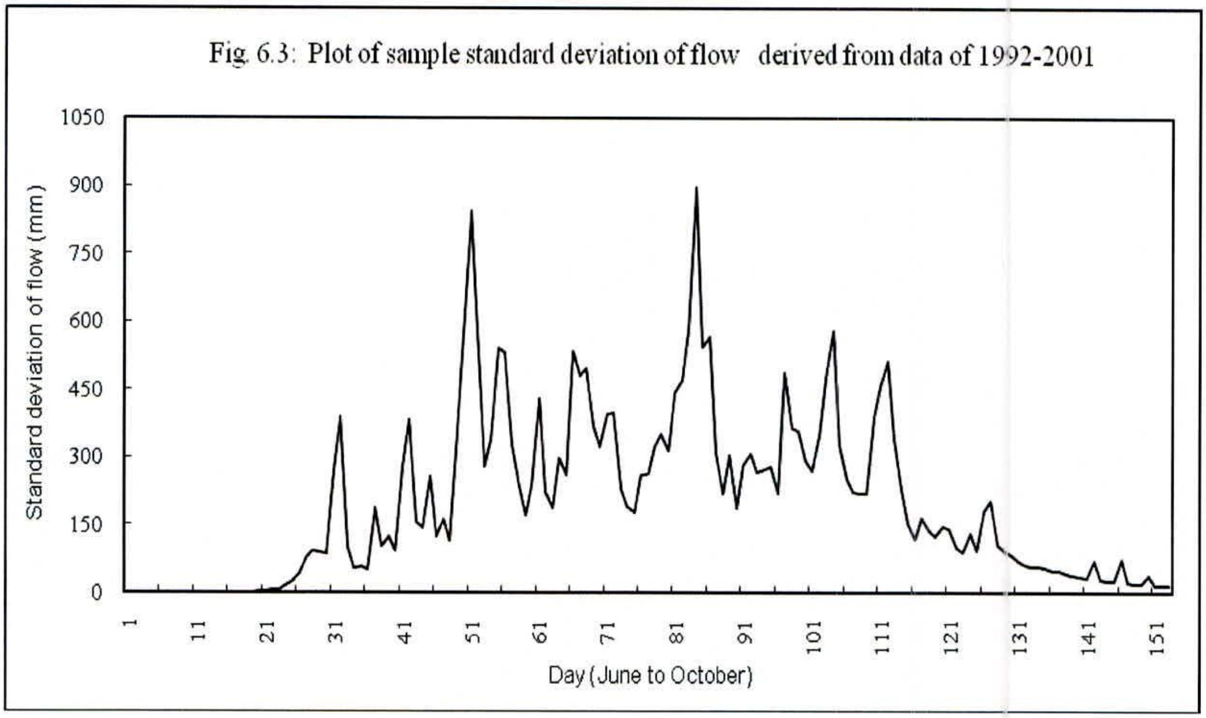

Fig. 6.3 : Plot of sample standard deviation of flow derived from data of 1992-2001

The standardization procedure preserves the first two moments i.e. mean and standard deviation of the historical time series. The sample periodic means and standard deviations can be estimated from the observed time series for each day and can be substituted in equation to obtain the standardized flow series. Salas et. al., (1988) mentioned that the sample estimates of means and standard deviations are subjected to larger errors, since they are usually estimated from a relatively small number of year's data, as compared to the population estimates. Also, the use of too many-estimated parameter violates the principle of statistical parsimony in the number of parameters. To reduce the number of estimated parameter and to obtain better estimates of these parameters, the values of mean and standard deviation are smoothened by harmonic analysis. The estimates of periodic parameters were obtained in this study by using the Fourier series analysis as given below,

Let V_τ represents any periodical parameter of the flow series, such as the daily mean or standard deviation then its representation by harmonics can be given as,

$$v_\tau = v_x + \sum_{j=1}^{m}\left[A_j \cos(2\pi j\tau/\omega) + B_j \sin(2\pi j\tau/\omega)\right] \tag{6.2}$$

$$A_j = \frac{2}{\omega}\sum_{\tau=1}^{\omega} V_\tau \cos(2\pi j/\omega) \quad \text{for j=1,......,h} \tag{6.3}$$

$$B_j = \frac{2}{\omega}\sum_{\tau=1}^{\omega} V_\tau \sin(2\pi j/\omega) \quad \text{for j=1,......,h} \tag{6.4}$$

$A_j and B_j$ =Fourier series coefficients
ω = number of seasons in a year, for daily flows is 365
j=harmonic
V_τ = un-smoothened parameter for the season, i.e. mean or standard deviation
m = number of harmonics fitted to smoothen the parameter
v_x = overall mean of the parameter

The Fourier series fit procedure requires the selection of the number of the significant harmonics. Salas et. al (1988) provides a procedure for selecting the number of significant harmonics by plotting the periodogram. However, this procedure added too many harmonics to the function (Aboitiz et. al., 1986). The significance of the harmonics has been tested by using the P_{max} and P_{min} test. Though the maximum number of harmonics that can be fitted to any seasonal parameter is $\omega/2$, yet out of $\omega/2$ only few harmonics may be significant. The steps in the test are:

For each harmonic, the variance explained in it is given by,

$$Var(h_j) = \frac{c_j^2}{2} = \frac{A_j^2 + B_j^2}{2} \tag{6.5}$$

The ratio of the variance explained by the j^{th} harmonic and the original variance is given by

$$\Delta P_j = \frac{Var(h_j)}{Var(v_\tau)} \tag{6.6}$$

The sums of ΔP_j for j=1,2,......., $\frac{\omega}{2}$ are given as

$$P_1 = \Delta P_1$$

$$P_{\omega/2} = \Delta P_1 + \Delta P_2 + ...+ \Delta P_{\omega/2} \tag{6.7}$$

The P_{max} and P_{min} is computed as given by

$$P_{\min} = a\left(\frac{\omega}{CN}\right)^{0.5} \tag{6.8}$$

$$P_{\min} = 1 - P_{\min} \tag{6.9}$$

where,

a=0.33 and *C=1* for mean and *2* for standard deviation

The harmonics explaining variance up to P_{max} are considered for smoothening the parameter. The Fourier series model with six harmonics were selected and they fitted well to both the mean and standard deviation and the fitted models resulted in smooth functions, which can be expected with a large sample size. Estimates of the periodic mean and standard deviation obtained from the fitted Fourier series models were utilized to obtain the standardized flow series using equation 6.10. The removal of the periodic component by the parametric approach is given by,

$$Z_{t,\tau} = \frac{X_{t,\tau} - \overline{X}_{\tau,s}}{\sigma_{\tau,s}} \tag{6.10}$$

where, $X_{\tau,s}$ = smoothened mean and

$\sigma_{\tau,s}$ = smoothened standard deviation

The standardized $Z_{t,\tau}$ series generally may not have overall mean of zero and standard deviation of one. The mean and standard deviation of the resulting standardized series were found to be 0.14 and 2.64 cumecs, respectively, which are not close to the theoretical values. The completely standardized series having zero mean and unit variance is obtained by

$$Z'_{t,\tau} = \frac{Z_{t,\tau} - \overline{Z}_{t,\tau}}{\hat{Z}_{t,\tau}} \tag{6.11}$$

where,

$Z'_{t,\tau}$ = completely standardized flow series

$Z_{t,\tau}$ = standardized series in the previous step

$\overline{Z}_{t,\tau}$ = mean of the standardized series in the previous step

$\hat{Z}_{t,\tau}$ = standard deviation of the standardized series in the previous step

6.1.2 Identification of the input vector

The identification of number of flow series in the input vector is done based on the sample auto correlation and partial auto correlation functions. These functions reveal the correlation structure of the time series and thus, are helpful in determining the underlying stochastic process. The theory is based on the assumption of second order stationarity. The assumption can be explained by letting (z_t, z_{t+h}) be a pair of flow measurements at t and $t+h$ in time, separated by a vector h i.e. lag. Each z_t, is a realization of the random variable (Z_t, t within the time domain of interest) is called a random function and is said to be second order stationarity if:

(i) the expected value $E(Z_t)$ exists and is the same within the time domain:

$$E(Z_t) = m \tag{6.12}$$

(ii) the covariance for each pair of random variables (z_t, z_{t+h}) exists, is the same in time, and depends on h,

$$Cov(h) = E(z_t, z_{t+h}) - m^2 \tag{6.13}$$

Stationarity of the covariance implies stationary of the variance.

6.1.3 Autocorrelation function

The autocorrelation function expresses the degree of dependency among neighboring observations. It is a process of self-comparison expressing the linear correlation between an equally spaced series and the same series at a specified lag. Let $z_0, z_1, z_2, \ldots\ldots z_{N-1}$ be a realization of a stationary stochastic process, then the population autocorrelation function can be defined as the quotient of the population auto covariance, cov(z_t, z_{t+h}) and variance, var(z_t):

$$\rho(h) = \frac{Cov(z_t, z_{t+h})}{Var(z_t)} \tag{6.14}$$

where,

(h) = *auto correlation function*

z_t = *the value of the variable at t^{th} time*

h = *time lag*

Since the series analyzed is just one particular realization out of an infinite number of realizations of a stochastic process produced by the underlying probabilistic *mechanism, the population autocorrelation function can be estimated using the simple autocorrelation function* r(h)*:*

$$r(h)=\frac{\sum_{t=1}^{N-h}(z_{t+h}-\bar{z})(z_t-\bar{z})}{\sum_{t=1}^{N-h}(z_t-\bar{z})^2}-1\leq r(h)\leq 1 \qquad 6.15$$

Where, $\bar{z}$ $= sample\ mean$

The 95% confidence band for the sample autocorrelation function given by Anderson and Jenkins, 1970:

$$r(h)=0\pm\frac{1.96}{\sqrt{n}}\geq\left[1+2\sum_{j=1}^{q}r_j^{\,2}\right]^{\frac{1}{2}} \quad \text{h>q} \qquad 6.16$$

where,

$q = order\ of\ the\ process$

$n = number\ of\ observation\ in\ the\ series$

The autocorrelation function is a diagnostic of the moving average process. Therefore, the value of a variable at a given time can be estimated from a purely random series using the weighted sum of the values at previous time steps.

6.1.4 Partial autocorrelation function

The partial autocorrelation function is another way of representing the time dependence structure of a series or of a given model. It is useful for diagnosing the order of the autoregressive processes. The autoregressive process has a relationship with the previous time steps. Therefore, the idea of autocorrelation, which measures the correlation of variable separated by assigned lags, can be extended to that of correlation, where dependence on the intermediate terms can be removed. Mathematically, it can be defined as,

$$\phi_k(k)=corr(z_t, z_{t-k}/z_{t-1},\ldots\ldots\ldots,z_{t-k+1}) \qquad 6.17$$

and is the correlation between z_t and z_{t-k}, excluding the effects of $z_{t-1}, z_{t-2},\ldots,z_{t-k+1}$. In this equation, k is the distance or time lag measured between the measured quantities. In general, for an autoregressive process of order k, the partial autocorrelation coefficient $_k$ (k), is a measure of the linear association between $_j$, and $_{j-k}$ (autocorrelation function at lag j and lag j-k, for j k. It is the kth autoregressive coefficient and $_k$(k), for k=1,2,…., is the partial autocorrelation function. Lag j autocorrelation for an autoregressive [AR(k)] process can be written as:

$$\rho_j = \phi_1(k)\rho_{j-1} + \phi_2(k)\rho_{j-2} + ... + \phi_k(k)\rho_{j-k} \qquad 6.18$$

where,

$_k(k) = j^{th}$ *autoregressive coefficient of the AR(k) model.*

The equation 6.18 constitutes a set of linear equations, which can be written in terms of sample partial autocorrelation functions $_k(k)$. Thus the sample partial autocorrelation function can be obtained by solving equation 6.18. Bartlett (1946), gave the 95% confidence band for the sample autocorrelation function as,

$$\phi_k(k) = 0 \pm \frac{1.96}{\sqrt{n}} \qquad 6.19$$

where,

= number of observations in the series.

6.1.5 Cross-correlation function

The number of previous day's rainfall which influences the flow rate to be predicted was determined based on the cross-correlation between the rainfall and discharge. The procedure that was used to identify the number of rainfall patterns as input to the network is summarized below. The lag-k correlation coefficient between the random variables X_j and X_h is estimated as given by,

$$r_{j,h}(k) = \left[\frac{\sum_{i=1}^{n-k} (X_{j\,i} - \bar{X}_j)(X_{h,i+k} - X_h)}{(n-k)\sigma_{x,j}\sigma_{x,h}} \right] \qquad 6.20$$

where,

$n =$ total number of observations on X_j and X_h

$X_{j.i} =$ ith observation on X_j

$\bar{X}_j =$ mean of the observation on X_j

$\sigma^2_{x,j} =$ variance of X_j

The *lag-k* cross-correlation coefficient is thus the correlation coefficient between the values of X_j and the values of X_h that are *k* units apart. In our case, X_j and X_h represents the standardized values of average rainfall and discharge respectively.

6.2 Model Performance Indicators

A mean squared error (MSE) is one of the most commonly used performance measure in hydrological modeling. Many researchers used MSE or its root (RMSE) as an accuracy measure *(Carpenter and Brathelemy 1994; Bastarache et al. 1997; Shamseldin 1997)*. The MSE and RMSE are given by,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Q_o(i) - Q_c(i)\right)^2 \qquad 6.23$$

$$RMSE = (MSE)^{0.5} \quad 6.23 \qquad 6.24$$

Others such as *Karunanithi et al., (1994),* used MSE and mean relative error (MRE) to try to fill some of the gaps left by considering only MSE and stated that the squared error and the relative error provide different types of information about the model's predicative capabilities. For each model, the fit to the training and testing data was done using the popular residual statistics. Some of the indicators considered for the model performance include bias (B), variance (V), relative bias (RB), mean absolute error (MAE), relative mean absolute error (RMAE).

$$B = \left[\overline{Q}_c(i) - \overline{Q}_o(i)\right] \qquad 6.25$$

$$V = MSE - (B)^2 \qquad 6.26$$

$$= \frac{B}{\overline{Q}_o} \qquad 6.27$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Q_c(i) - Q_o(i)\right| \qquad 6.28$$

$$RMAE = \frac{MAE}{\overline{Q}_o} \qquad 6.29$$

The percentage difference in peak between the observed and computed discharge is given by,

$$PK = \left[1 - \frac{\max(Q_c)}{\max(Q_o)}\right] * 100 \qquad 6.30$$

The Nash-Sutcliffe goodness of fit, which is a measure of the model efficiency is given by,

$$\xi = 1 - \frac{\sum_{i=1}^{n}\left[Q_o(i) - Q_c(i)\right]^2}{\sum_{i=1}^{n}\left[Q_o(i) - \overline{Q}_o(i)\right]^2} \qquad 6.31$$

where,

$Q_o(i) =$ observed discharge

$Q_c(i) =$ computed discharge

7.0 RESULTS AND DISCUSSION

The main objective of the study was to develop a rainfall runoff model for the Sindh river basin, Madhya Pradesh, which would be able to forecast the stream flow using historic time series data of rainfall and runoff. ANN algorithms, which are able to capture the non-linearity inherent in the rainfall runoff process, are capable of modelling the rainfall-runoff relationship due to its ability to generalize the patterns in noisy environment without a prior knowledge of probability distributions. The results pertaining to the study as the performance of the ANN models are presented here.

7.1 Identification of the Input Vector to the Network

The model building process is rather a very complicated process, which involves the proper identification of several features, which are very important in order to obtain a good model that is able to forecast with a reasonable degree of accuracy. The first and foremost step is the division of the available data into training set and testing set. Poor forecasts can be expected when the validation data contains values outside the range of that used for training. The division of the sub-sets should be done in such a manner that the training data as well as the testing data should be representative of the same population. Also the generalization ability of the model depends on the ratio of the training samples to the number of connection weights. If this ratio is small, over fitting of training data may occur that ultimately leads to poor prediction accuracy during testing of the model. The available data set was divided into two sub-sets namely for training the data from 1992 to 1998 was used where as the remaining data from 1999 to 2001 was used for testing the model.

The data pre-processing can also have a significant effect on the model performance. The data is divided into their respective sub-sets as mentioned above before any data pre-processing is carried out. The variables are standardized as explained in the previous section. The variables are generally measured in different units. By standardizing the variables and recasting them in dimensionless units, the arbitrary effect of similarity between objects are removed. Also by adopting the harmonic analysis for arriving at the smoothened means and standard deviation and then using the same for standardization helps to remove the seasonal effects from the variables which are essentially of deterministic nature. As the mean and standard deviation of this standardized series is 0.14 and 2.64 which is significantly different from 0.0 and 1.0 respectively, so the series has been completely standardized and the mean and standard deviation of the completely standardized series is 0.0 and 1.0 respectively. The plot showing the completely standardized series of the historical flow series from 1992 to 2001 is given in Fig. 7.1. Also it is imperative that the variables need to be scaled down in such a way as to commensurate with the limits of the activation function used in the output layer. As the outputs of the logistics function are between zero and one they are generally scaled in this range.

It is important to note that if the variables are scaled down to the extreme ranges of the transfer function, the size of the weight update becomes very small and it is possible that flat spots may occur in

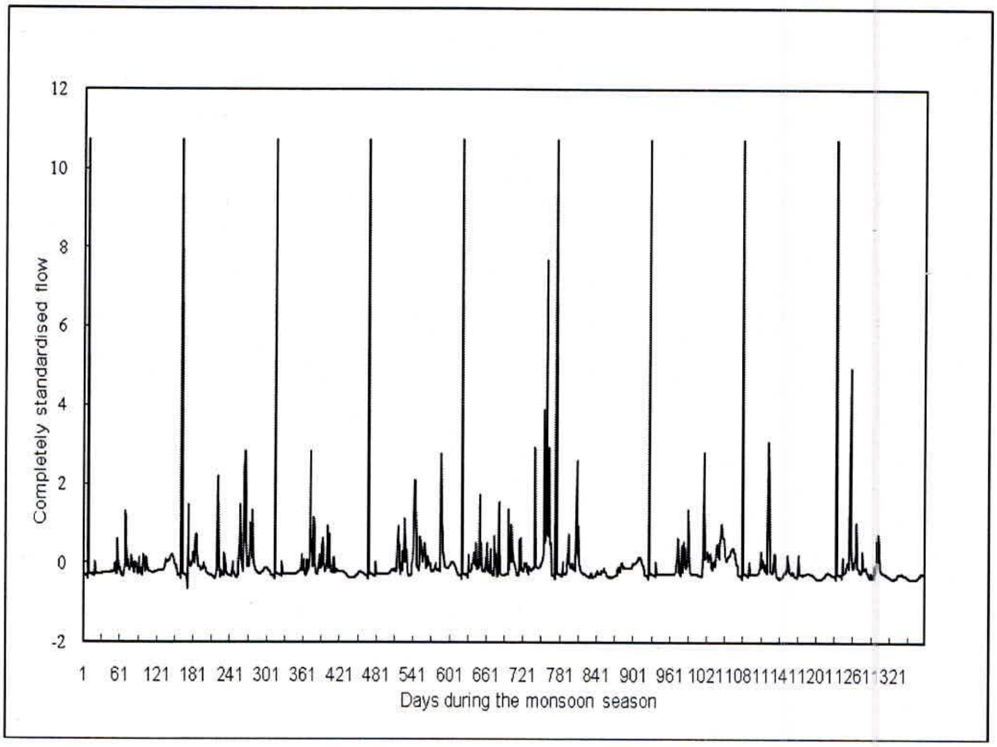

Fig. 7.1 : Completely standardized flow series for the years 1992-2001

training. When the transfer function in the output layer is unbounded, scaling may not be strictly required. However, scaling to uniform ranges is still advised. The data has been scaled by using the following function (Romesburg, 1984) given by equation 7.1,

Another important limitation to be kept in mind is that ANN cannot extrapolate beyond the range

$$Y_{i,j} = \frac{\left(X_{i,j} - X_{\min j}\right)}{\left(X_{\max j} - X_{\min j}\right)}$$

where,

$X_{\max j}$ = maximum of the j^{th} variable in all observations

$X_{\min j}$ = minimum of the j^{th} variable in all observations

of the training data. So ANN cannot account for trends and hetero-scedasticity in the data. One way to deal with the problem is by removing the deterministic components using methods commonly used in the time series analysis. The procedure of standardizing, differencing or classical decomposition is helpful for the same.

To create any rainfall-runoff model by system theoretic approach, such as ANN, it is required to determine from the available historical sequences of rainfall and runoff data, the choice of how many and which delayed runoff patterns and rainfall patterns affect the next output. This is one of the complexities, which make the forecast more difficult than the simple straight regression analysis. Conducting autocorrelation and partial correlation analysis of the river flow series and determining the flow lags that have significant effect on the next day's flow can contain this complication. The autocorrelation function (ACF) and the corresponding 95% confidence bands from lag 0 to lag 20 were estimated for the standardized flow. The autocorrelation plot of the standardized series is shown in Fig. 7.2. Lag zero auto correlation is always one as it is the correlation of the variable with itself. However, as the lag increases, the correlation between the variable and the same variable at the specified lag decreases, i.e. covariance decreases. The auto correlation function showed significant correlation, at 95% confidence level, up to lag-14, and thereafter fell near to or below the confidence band. The gradual decaying pattern of the auto correlation exhibits the presence of dominant autoregressive process.

Similarly the partial autocorrelation function (PACF) and corresponding 95% confidence were estimated for lag 0 to lag 20. The partial correlation plot of the standardized series is shown in Fig. 7.3. The PACF showed significant correlation at lag-2 and thereafter fell below the confidence band. The rapid decaying pattern of the PACF confirms the dominance of auto regressive process, relative to the moving average process. The above analysis of auto and partial correlation coefficients suggested incorporating flow values with *lag-2* in the input vector to the network.

In the present study, this analysis was carried out and it was found that the flow on a particular day is dependent on two previous days flows. Similarly, the number of previous rainfall patterns that are having significant effect on the next day flow has been identified after conducting the cross correlation analysis between the rainfall and runoff variables and then going for a trial and error procedure to check whether any additional rainfall pattern needs to be considered as input. The cross-correlation function (CCF) and corresponding 95% confidence were estimated and is given in Fig 7.4. The CCF showed significant correlation above 0.50 for the *lag-2*. Hence the initial number of rainfall patterns in the input vector to the network has been considered for a lag of 2 days.

The study is based on the feed-forward back-propagation neural network, which is the most widely used algorithm in the field of water resources. The trial started with the presentation of the input vector to the network, which consisted of two previous days rainfall along with two previous days runoff and estimating the goodness of fit statistics pertaining to the present input vector. The trial was continued by adding one more previous day's rainfall, i.e. lag of three days in the input vector. The performance of the new input vectors was examined based on statistical indices. The effect of rainfall lags in the input vector on the performance of the network is presented in Table 7.1. The model is selected on the basis of the performance as examined by the goodness of fit statistics.

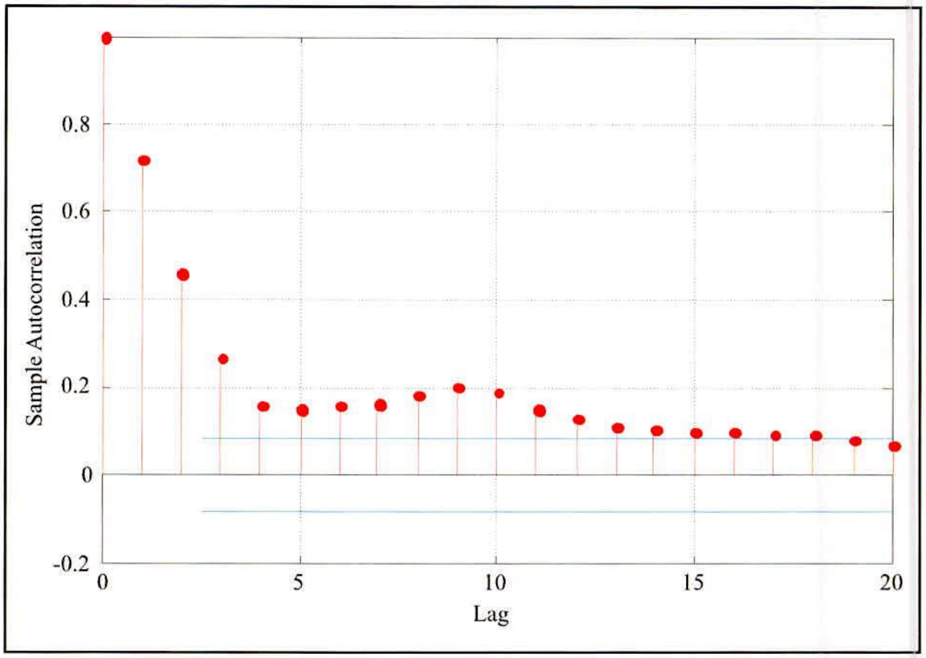

Fig. 7.2 : Autocorrelation plot of the standardized flow series

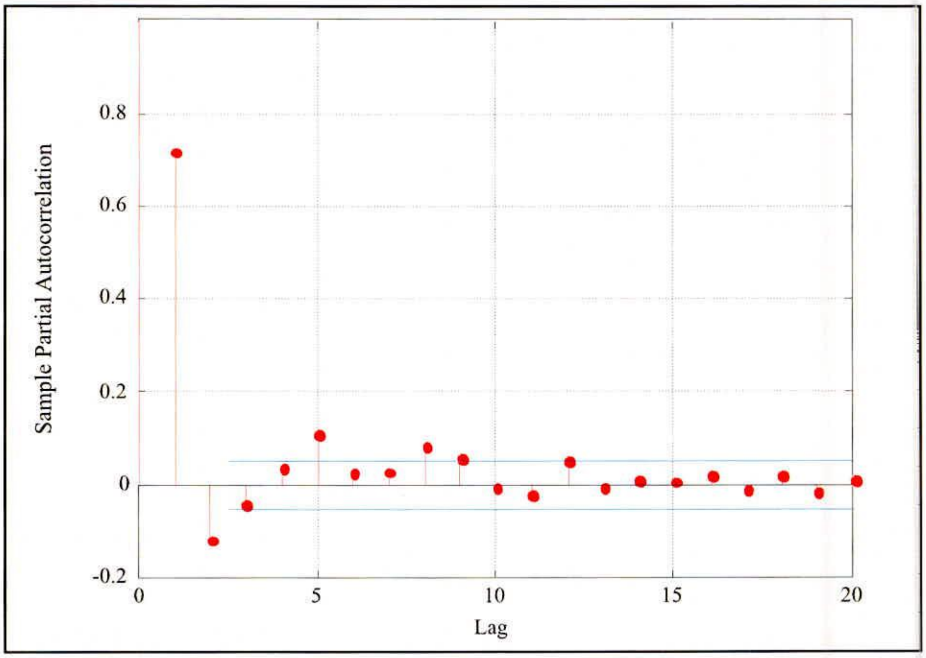

Fig. 7.3 : Partial autocorrelation plot of the standardized flow series

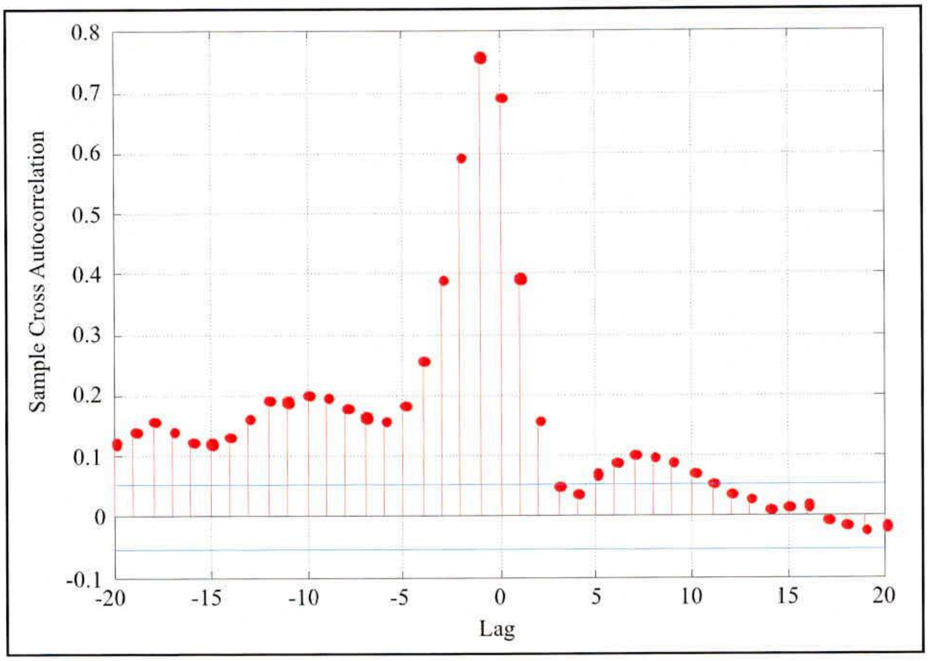

Fig. 7.4 : Cross correlation plot

The RMSE error has not improved much when the number of rainfall patterns in the input vector to the network increased from two, as can be seen from Table 7.1. These results lead to the conclusion that the number of previous rainfall data has a significant effect in the model performance. The experiment resulted in the conclusion that an input vector with 2-days rainfall lag along with 2-days runoff lag can produce river flow patterns in a satisfactory manner. Therefore, a sample model was selected by representing stream flow at the present time, *t* as a function of precipitation at *t-1* and *t-2* and stream flow at *t-1* and *t-2* as desired from the analysis of the autocorrelation and partial autocorrelation structure of the flow series and the cross-correlation structure between the rainfall and runoff. The model can be represented as

$$Q(t) = f(P_{t-3}, P_{t-2}, P_{t-1}, Q_{t-2}, Q_{t-1}) \quad 6.21$$

7.2 Identification of the Network Architecture

The determination of the network architecture is the other important aspect. The network geometry determines the number of connection weights and how these are arranged. The network architecture is decided by number of layers in the network, number of neurons in these layers. The neurons in the input layer and the output layer are entirely dependent on the problem being considered, i.e. the number of variables in the input vector and the outputs desired from the output vector. The number of hidden layers can vary and so is the number of neurons in the hidden layer. This depends on the complexity of the underlying process to be modeled. Extensive research has pointed out that ANN with one hidden layer can approximate any function given that sufficient degrees of freedom are provided (*Hornik, 1989*). *Sridhar, (1996)* reported that increasing the number of hidden layer to two or more have no significant effect in the performance of the network. But identifying the number **of** neurons in the hidden layer is a very complicated task in finalizing the network architecture. The critical aspect is the choice of the number of neurons in the hidden layer and hence the connection weights. This is generally done by trial and error evaluation. The important consideration while deciding the number of neurons in the hidden layer is the relationship between the number of training samples and the number of connection weights. The number of weights should not exceed the number of training samples; the ratio between the number of training samples to the number of connection weights should be 2:1 and it is believed that over fitting does not occur if the number of training samples is at least 30 times the number of free parameters.

In the present study, a back propagation network with a single hidden layer and having five neurons in the input layer, two neurons in the hidden layer and one neuron in the output layer was selected with the logistic sigmoid transfer function for both hidden layer as well as the output layer. Initial random weights were assigned to the connections between the neurons from one layer to the next layer and the training was done in batch mode wherein the epoch size is equal to the size of the training set. The learning parameter and momentum parameter are other important consideration as the speed of the training as well as reaching to a global minima very much depend on these. The momentum factor is generally less than one for convergence. The mean square error is taken as the error function, which is minimized during the training. The mean square error function is the most widely used as it penalizes large errors and its partial derivative with respect to weights can be calculated easily. The goodness of fit statistics has been computed for the network with two neurons in the hidden layer. In the next trial, the number of neurons in the hidden layer is increased to three and goodness of fit statistics are evaluated. The effect of the number of neurons in the hidden layer is presented in Table 7.2.

From Table 7.2, it can be observed that the RMSE during training and testing gets reduced as the number neurons in the hidden layer increases from two to five. Thereafter with the increase in the number of neurons in the hidden layer, the RMSE gets reduced during training whereas it increases during testing. This indicates the model is able to reproduce the results well on the independent test data as long as the number of neurons in the hidden layer is limited to five. Similarly the efficiency of the model is maximum during training and testing with five neurons in the hidden layer namely, 85.14% and 84.42% respectively. Even though the efficiency of the model increases significantly during the training with the increase in the number of neurons in the hidden layer, but during testing it drops down to significantly lower values. The efficiency during training and testing with 15 neurons in the hidden layer is indicative of the fact.

7.3 Performance of the Model

In the present study, based on the performance in representing the rainfall runoff process, the feed-forward neural network with five neurons in the single hidden layer was finalized. The comparison of the observed and simulated runoff during the training period from 1992 to 1998 is given in Fig. 7.5 to Fig. 7.10. It is observed that the model is able to simulate the flows reasonably well except for one peak flow in 1992 and another peak flow during 1997. This corresponding rainfall for those particular periods was checked. Overall the model was able to simulate the flows with a fair degree of accuracy. The model was tested on the independent test data from 1999 to 2001. The comparison of the observed and simulated runoff during the testing period from 1999 to 2001 is given in Fig. 7.11 to Fig. 7.13. The model trained with the input vector is able to simulate the flows with the independent test data with reasonable accuracy.

The indicators of the model performance were also computed for the entire training set and testing set also separately for each year during the training and testing process. The performance of the three identified models for training as well as validation periods are critically examined using various statistical indices and are reported in Table 7.3.

The optimal value for the correlation coefficient is 1.00. The model tends to have correlation coefficient of 0.923 and 0.926 during training and testing respectively which indicates that the model is able to reproduce the flows quite well. The correlation coefficient during testing and training is very similar and the difference is insignificant. The optimal value for the RMSE statistic, which measures the residual variance, is 0.0.

The model tends to have smaller value of RMSE during training. The value of RMSE is found slightly deteriorating during validation. The optimal value for the percentage error in peak flow which measures the percent error in matching the maximum flow of the data record is 0.0 and positive values indicates overestimation whereas the negative values indicate underestimation. During training as well as testing the model matches the peak flow very well. The efficiency of the model as defined by Nash-Sutcliffe criteria is a measure of the performance of the model in predicting the output values. According to this statistic, the model predictions were fairly good during training. The model efficiency during training and validation is 85.14 % and 84.42 % respectively. The efficiency of the model, root mean squared error, relative mean absolute error, relative bias and correlation coefficient during the corresponding years of training and testing is given in Table 7.4. It can be seen that the efficiency of the model is rather very poor during the training period of year 1997. Similarly, the RMSE and RMAE is maximum whereas the correlation coefficient is minimum for the year 1997. However the results during training are satisfactory.

It is observed that the efficiency is minimum for the year 1997 whereas it is maximum for the year 1996. It was seen that during the monsoon season of 1997 a number of flood peaks have been

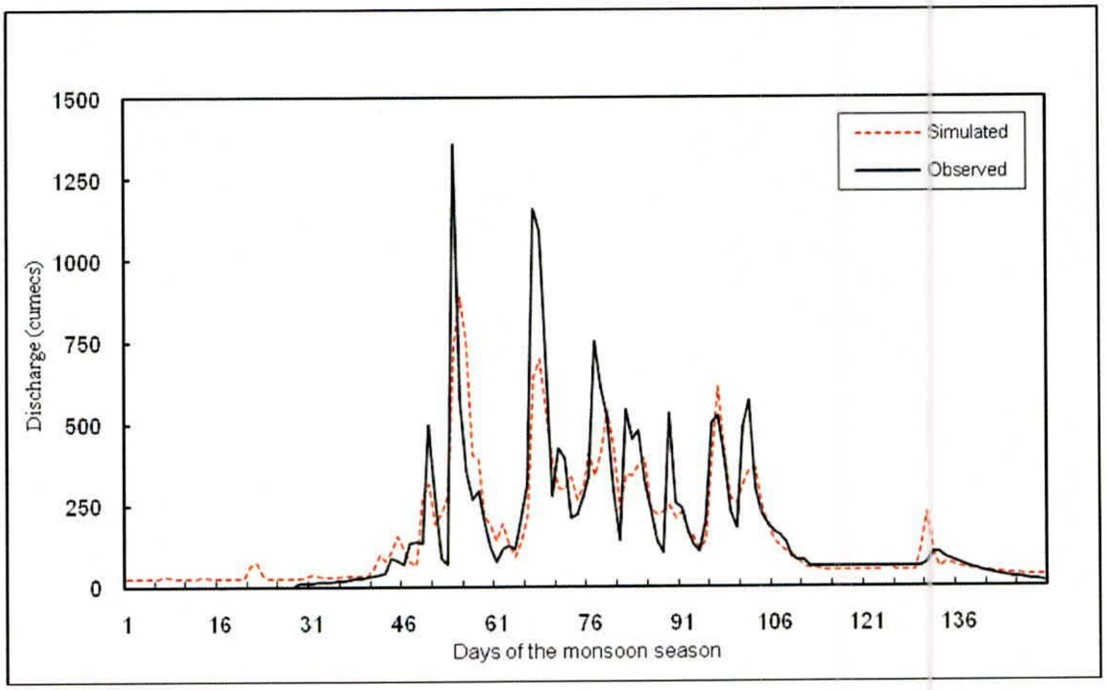

Fig. 7.5 Comparison of the observed and simulated discharge during training for 1992

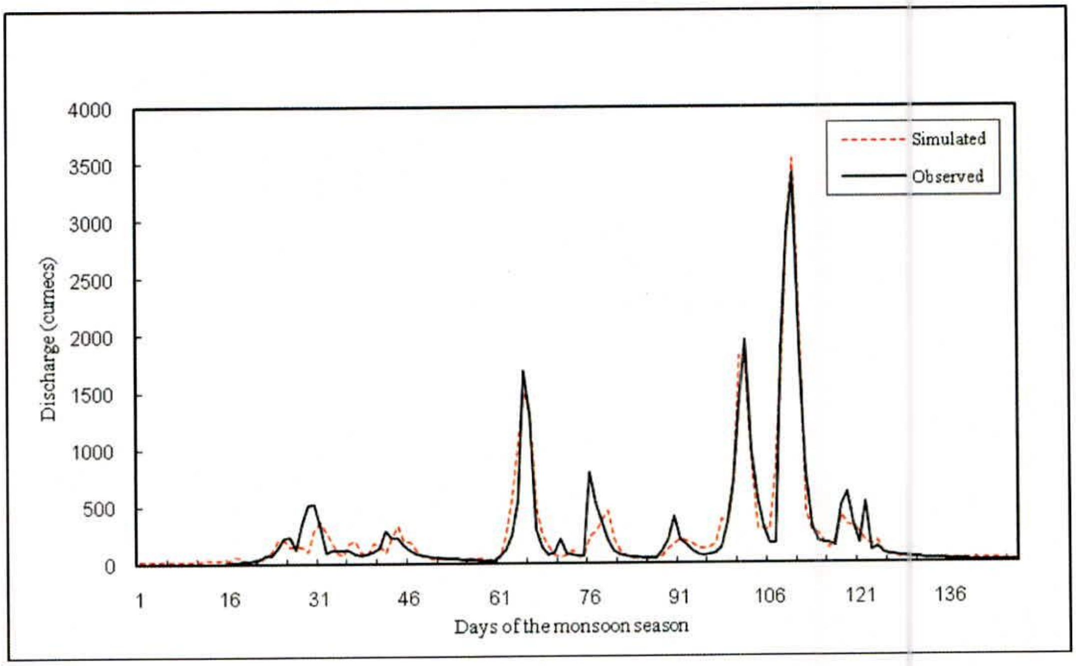

Fig. 7.6 : Comparsion of the observed and simulated discharge during training for 1993

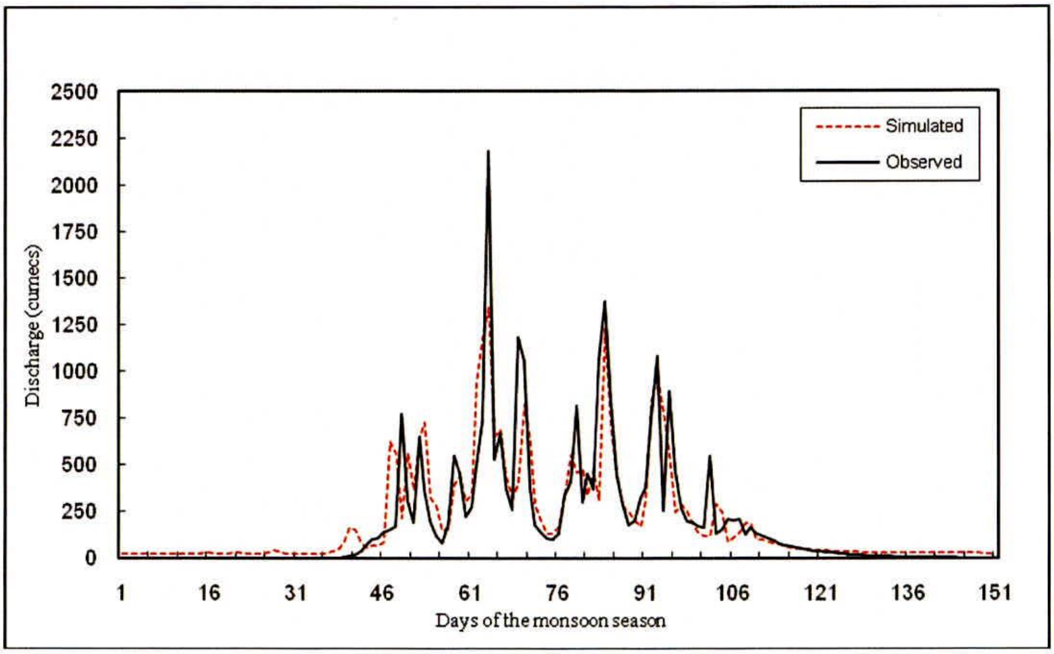

Fig. 7.7: Comparison of the observed and simulated discharge during training for 1995

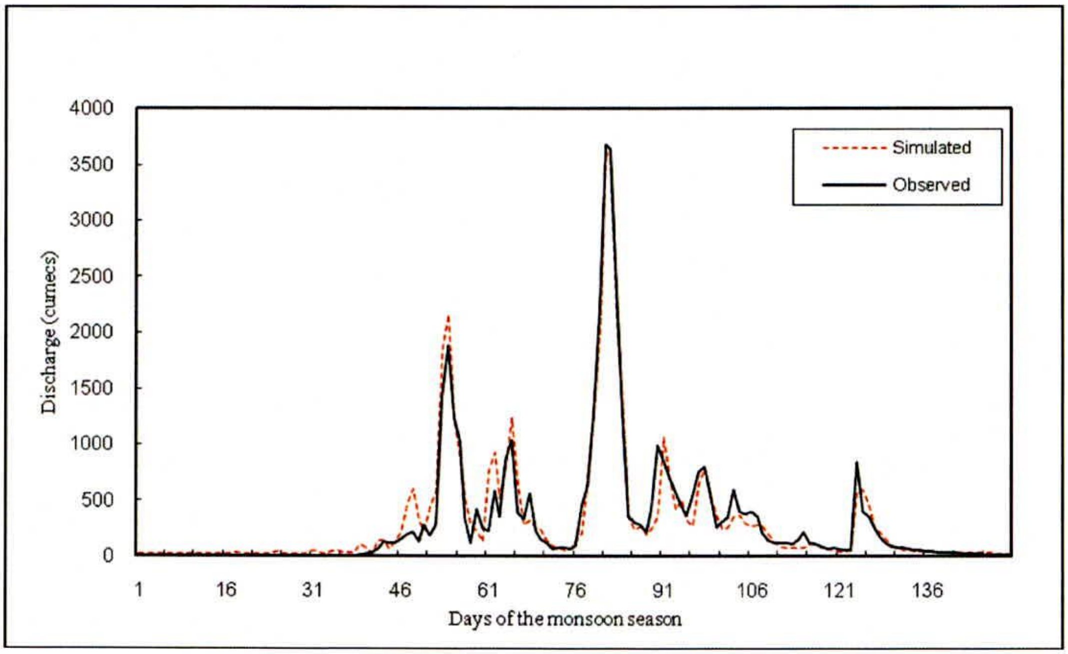

Fig. 7.8: Comparison of the observed and simulated discharge during training for 1996

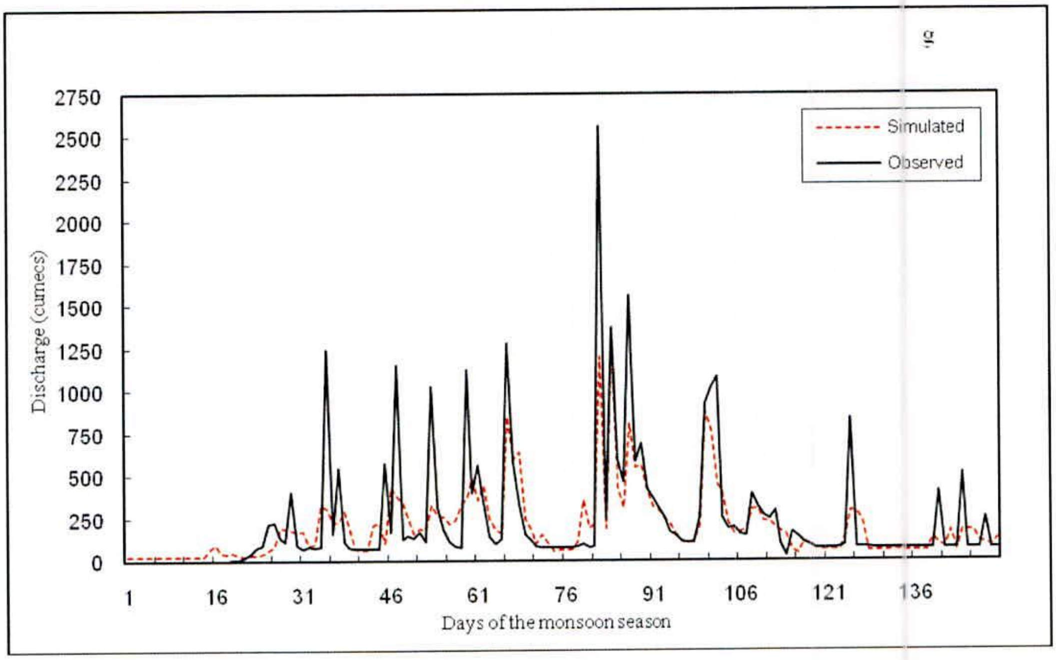

Fig. 7.9: Comparison of the observed and simulated discharge during training for 1997

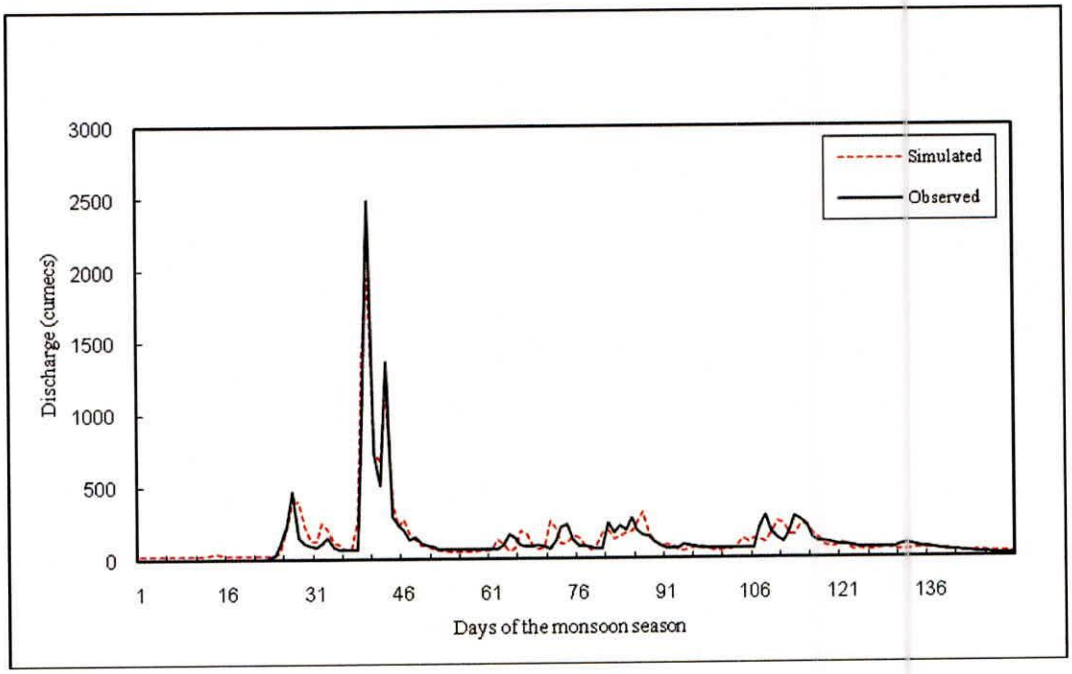

Fig. 7.10 : Comparison of the observed and simulated discharge during training for 1998

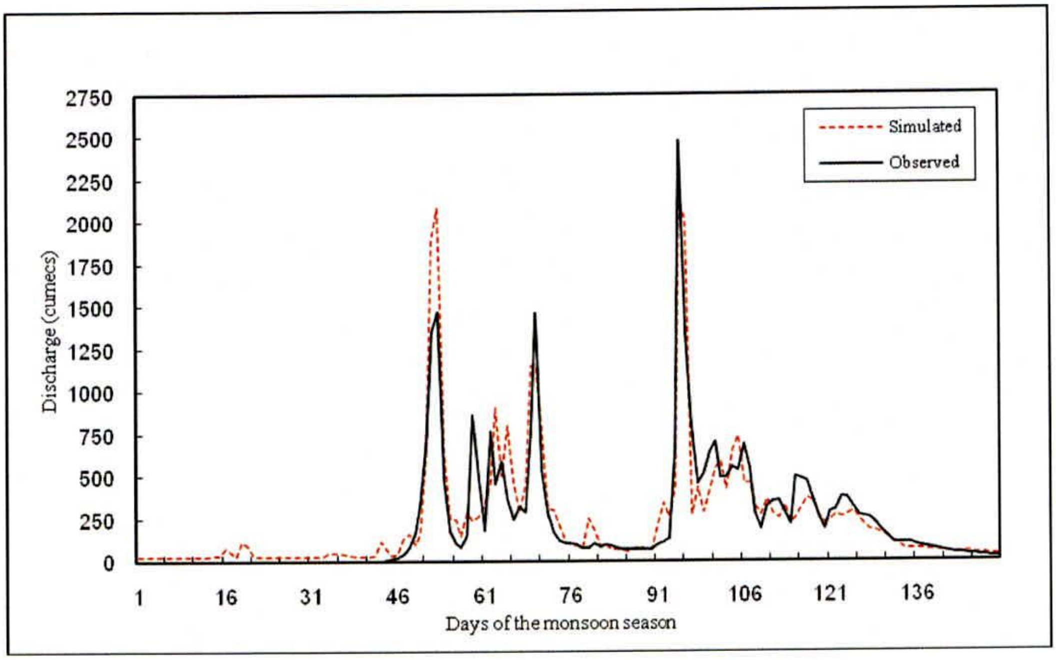

Fig. 7.11: Comparison of the observed and simulated discharge during training for 1999

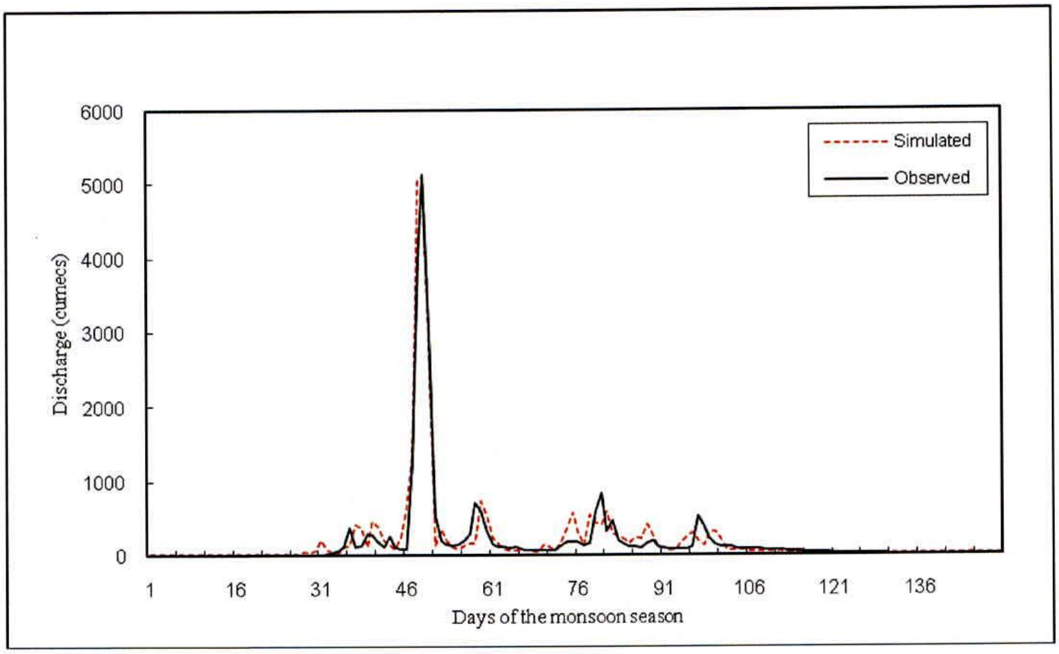

Fig. 7.12: Comparison of the observed and simulated discharge during training for 2000

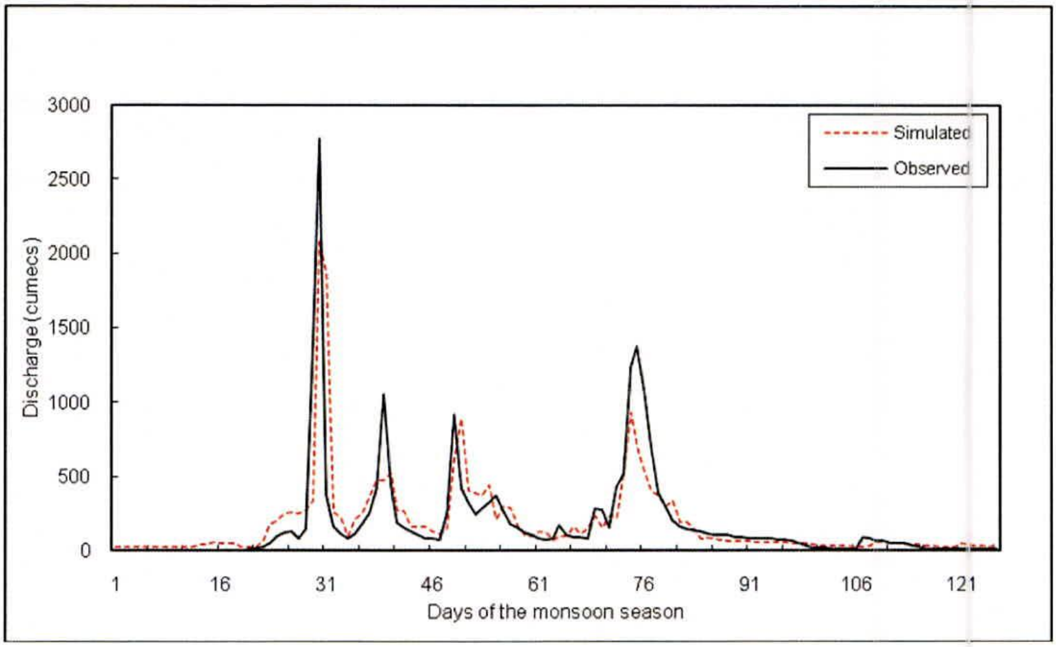

Fig. 7.13: Comparison of the observed and simulated discharge during training for 2001

observed and the model failed to predict the peaks correctly during the start of the monsoon season. Thereafter the model was able to simulate the peak flows with considerable accuracy. The correlation coefficient varies between 0.82 to 0.97 during training whereas it varies between 0.79 to 0.97 during training. Similarly the RMSE varied between 84.38 to 225.44 during training and varied between 157.73 to 199.35 during testing with the independent test data. These results do indicate that the model is capable enough to simulate the flows with a reasonable degree of accuracy. Therefore it can be concluded that the selected model i.e. the feed forward neural network with back propagation algorithm having a single hidden layer with five neurons in the hidden layer is able to model the rainfall-runoff process in the Sindh river basin in a satisfactory manner.

8.0 CONCLUSIONS

An attempt has been made to apply the artificial neural network techniques to develop a rainfall-runoff model for the Sindh river basin in Madhya Pradesh. The review of the literature indicates that the ANN methodology has been reported to provide reasonably good solutions for circumstances having complex systems that may be poorly defined or understood using mathematical equations, problems that deal with noise or involve pattern recognition and input data that are incomplete and ambiguous by nature. Earlier application of artificial neural networks in water resources revealed that the approach of neural computations was very effective in developing the required model, due to its various advantages. Therefore it was decided to apply the ANN to model the precipitation-runoff relationship in the Sindh river basin. Accordingly, the model based on feed- forward back-propagation ANN architecture was developed for the study area, to represent rainfall-runoff transformation. The architecture of the network was determined based on a trial and error procedure and after examining various goodness of fit statistics. An auto correlation and partial auto correlation analysis of the standardized daily flow series suggested that the flow at time *'t'* was highly correlated to previous two days flow. These parameters were included in the input vector of the network, apart from 2-days rainfall series prior to the day, at which the flow was to be predicted. The number of rainfall patterns in the input vector was finalized by cross-correlation analysis and then by trial and error.

Statistical analysis was done on the performance of each model in estimating the runoff. The study revealed that a feed-forward artificial neural network with back propagation algorithm having a single hidden layer with five neurons in the hidden layer was able to model the rainfall-runoff transformation quite accurately. The correlation coefficient during the training and testing varies between 0.793 to 0.973 respectively whereas the model efficiency varies between 70.36 % to 94.57 % with an overall efficiency of 85.14 % during calibration and between 63.02 % to 92.76 % with an overall efficiency of 84.42 % during validation. The study demonstrates that ANN can model accurately the non-linear relationship between rainfall and runoff and provides a systematic approach and shortened time spent on training of models compared to development and calibration of the conceptual models. Hence it is concluded that the feed- forward back propagation network model developed for rainfall- runoff process in the Sindh river basin might be employed for water resources planning and management in the basin. While such a model is not intended as a substitute for a physically based model, it can provide a viable alternative when the hydrologic application requires that an accurate forecast of stream flow be provided using only the available input and output time series data, and with relatively little conceptual understanding of the hydrologic dynamics of the particular basin under investigation. Although the artificial neural network technique has been applied to only one basin at present, the results presented here are encouraging and demonstrates a high potential for application of neural networks to various precipitation-runoff modelling scenarios.

REFERENCES

Aboitiz, M., Labadie J. W., and Heerman, D. F., (1986). *Stochastic soil moisture estimation and forecasting for irrigation fields.* Water resources research, 22(2):180 90.

Aithen, A.P., (1973). Assessing systematic error in rainfall- runoff models. J. of Hydrol. 20, 121-136.

Akaike, H. (1974). A new look at the statistical model identification. 1 EEE transactions of automation and control, AC-19:716-723.

Ambrus, S.Z., and Ward, R.J. (1990). A computerized real-time flood forecasting system on the Rideau River of Canada. Proc., 8th Int. Conf. on Computational Methods in Water Resour., Computational Mechanics, Southampton U.K., 243-250.

Amrrocho, J., and Orlob, G.T., (1961). Non-linear analysis of hydrologic systems, Water Resources Center, University of California (Berkeley) U.S.A. Contrib. 40, 147 pp. illus.

Amrrocho, J., and Brandstetter, A., (1971). Determination of non-linear function response functions in rainfall-runoff processes. *Water Resour. Res.* 7 (5), 1087-1101.

Anderson, J.A. (1995). An introduction to neural networks, MIT Press, Cambridge, Mass.

Bastarache, D., El-Jabi. N., Turkkan, N., and Clair, T.A. (1997). Predicting conductivity and acidity small streams using neural networks. Can. J. Civ. Engrg., Ottawa, 24, 1030-1039.

Bishop, C. M., (1994). Neural networks and their applications. Rev. Science Instruments., 65:1803-1832.

Box, G.E.P. and Jenkins, G.M., (1970). Time Series Analysis, Forecasting and Control, Holden-Day Press, *San Francisco, California*, USA.

Brath, A., and Rosso, R., (1993). Adaptive calibration of a conceptual model for flash flood forecasting. *Water Resour. Res.* 29 (8), 2561-2572.

Burke, L. F., Ignizio, J.P., (1992). Neural networks and operations research: an overview. Computer operations research, 19(3/4), 179-189.

Burnash, R.J.C., Ferral, R.L., and McGuire, A., (1983). A generalized stream flow system: Conceptual modeling for digital computers. Report, Joint Federal State River Forecast Center, National Weather Service, Sacramento, Calif: Department of Water Resources.

Carpenter, W.C., and Barthelemy, J.F., (1994). Common misconceptions about neural networks as approximators. *J. Comp. in Civ. Engg.*, ASCE, 8(3), 345-358.

Carriere, P., Mohaghegh, S. and Gaskari, R., (1960). "Performance of a virtual runoff hydrograph system." *J. Water Resour. Plng. and Mgmt.*, ASCE, 122(6), 421-427.

Caudill, M., (1987). Neural network primer: Part I. *At Expert, (*December), 46-52.

Chakraborty, K., Mehrotra, K., Mohan, C. K. and Ranka, S., (1992). *Forecasting of the behavior of multivariate time series using neural networks,* Neural networks, 5:961-970.

Chander, S., and Prasad, T., (1981). For*ecasting* and Prediction of Floods, i*n: Indo-US* Workshop on Flood Mitigation (New Delhi), (1981).

Chander, S., Kapoor, P.N. and Natarajan, P., (1984). Newer Techniques in High Flow Range forecasting, Civil Engineering Department, Indian Institute of Technology, Delhi, (Indian National Committee the International Hydrological Programme).

Chen, C.H., (1991). *Neural networks in pattern recognition and their applications,* World Scientific Singapore.

Cheng, X., and Noguchi, M., (1996). Rainfall runoff modeling by neural network approach. Proc. Int. Conf. Water Res. Environ. Res. Vol. II, Oct. 29-31, Kyoto, Japan.

Crespo, L., and Mora, E., (1993). Drought estimation with n*eural networks*, Advances in Engineering Software, Vol. 993, pp. 167-170,.

Dawson,C.W., and Wilby,R., (1998). An artificial neural network approach to rainfall- runoff modeling. *Hydrological Sciences Journal,* 43(1), 47-66.

Dhar, V., and Stein R., (1997). Intelligent decision support methods, Prentice-Hall, Englewood Cliffs, N.J.

Flood, I and Kartam, N., (1994b). Neural Network in Civil Engineering-I, Principles and Understanding, Journal of Computing in Civil Engineering, ASCE, 8(2), 131-148.

Flood, I and Kartam, N., (1994 a) *Neural Network in Civil Engineering – II: Systems and Applications,* Journal of Computing in Civil Engineering, ASCE, 8(2), 149-162.

Freeman, J.A., and Skapura, D.M. (1991). Neural networks, algorithms, applications, and programming techniques. Addison -Wesley, Reading, Mass.

French, M.N., Krajewski, W.F. and Cuykendall, R.R., (1992). Rainfall forecasting in space and time using a neural network, Journal of Hydrology, Vol. 137, pp. 1-31.

Garrote, J.H. and Bras, R.L., (1995a). A distributed model real time flood forecasting using digital elevation models. *Journal of Hydrology,* 167(1), 279-306.

Garrote, J.H., and Bras, R.L. (1995b). An integrated software environment real-time used of a distributed hydrologic models. *Journal of Hydrology*., 167(1), 307-326.

Gupta, V. K. and Sorooshian, S., (1985). The relationship between data and the precision of parameter estimates of hydrologic model. *J. Hydrol.,* 81, 57-77.

Haykin, S., (1994). Neural Network, A Comprehensive Foundation, IEEE Press New York.

Hartman, E. J., Keeler, J. D., and Kowalski, J. M. (1990). *Layered neural networks with gaussian hidden units as universal approximations.* Neural computations, 2: 210-215.

Hecht-Nielsen, R., 1987. Counter-propagation networks. J. of Applied Optics 26 (3), 4979-4984.

Hecht-Nielsen, R., (1990). *Neurocomputing,* Addison-Wesley Publishing Company, Reading, Mass.

Hecht. Nielsen, R., (1989). Theory of the Backpropagation Neural Network. Proc. Int. Joint Conf. On

Neural Networks, IEEE, Washington D.C., Vol. 1, 593 - 605.

Hertz, J., Krogh, A. and Palmer, R.G., (1991). Introduction to the theory of *neural computations.* Addison-Wesley publishing company, New York, pp: 246-250.

Hornik, K., Stinchcombe, M., While, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359-366.

Hsu, K. L., Gupta, V.H., and Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. *Water Resour. Res.,* 31(10), 2517-2530.

Jain, S.K., Goel, M.K., Senthil Kumar, A.R. and Agarwal, P.K., (1996). Multiobjective optimization of operation of a dam. TR(BR)- 143, National Institute of Hydrology, Roorkee, India.

Jain, S.K., Chalisgaonkar, D. and Goel, M.K., (1997). Software for reservoir analysis (SRA). UM-/196-97, National Institute of Hydrology, Roorkee, India.

Jain, S.K., Das, A., and Srivastava, D.K., (1999). Application of ANN for Reservoir Inflow Prediction and Operation. *Journal of Water Resources Planning and Management, ASCE,* 125(5), 263-271.

Jain, S.K. and Chalisgaonkar, D., (2000). On Setting up stage-discharge relations using ANN, Journal of Hydrologic Engineering, ASCE, 5(4), 112-121.

Kao, J-J. (1996). Neural Net for Determining DEM-Based Model Drainage Pattern. *Journal of Irrigation and Drainage Engineering,* ASCE, 122 (2), 112-12 1.

Kang, K.W., Evaluation of hydrologic forecasting system based on n*eural network model, Proceedings of IAHS Floods and Droughts, Japan (1993).*

Karunanithi, N., Grenney, W.J., Whitley, D. and Bovee, K., (1994). Neural networks for river flow prediction. J. Comp. in Civ. Engrg., ASCE, 8(2), 201-220.

Kitanidis, P.K. and Bras, R.L., (1980). Real-time forecasting with a conceptual hydrological model. Applications and results." *Water Resour. Res., 16(6), 1034-1044.*

Konda, T. and Deo, M.C., *(1998).* River Stage forecasting Using Artificial Neural Networks, *Journal of Hydrologic Engineering, Vol. 3(1), 26 – 31.*

Leaf, C.F. and Brink, G.E., (1975). Land use simulation model of subalpine coniferous forest zone. USDA Forest Service Res. Paper RM- 135, U.S. Department of Agriculture, Washington, D.C.

Lippman, R.P., (1987). An Introduction to Computing with *Neural Networks,* IEEE ASSP Magazine, 4 – 22.

Lorrai, M., and Sechi, G.M., Neural nets for modeling rainfall runoff transformations. Water Res. Mgmnt., 9(4), 299-313.

Maier, H.R., and Dandy, G.C. (1999). Empirical comparison of various methods for training feed-forward neural networks for salinity forecasting. Water Resources Research, 35 (8), 2591-2596.

Maier, H.R., and Dandy, G.C. (2000). Neural Networks for the Prediction and Forecasting of Water Resources Variables: A Review of Modelling Issues and Applications. Environmental Modelling

& Software, 15, 10 1 - 124.

Markus, M. and Baker, D., (1994). The Fraser River: Stream flow forecasting and simulation computer package. Tech. Rep., Northern Colorado Water Conservancy District, Loveland, Colo.

Markus, M., Salas, J. D. and Shin, H., (1995). Predicting streamflows based on neural networks. 1st Int. Conf. On Water Resour. Engrg., ASCE, New York.

Marren, A., Harston, C.., and Pap, R., (1990). Handbook of neural computing applications. Academic Press, Sandiego, CA.

McCulloh, W.S., and Pitts, W., (1943). A logical calculus of ideas imminent in nervous activity. Bulletin of Mathematical Biophysics, 5, 115-133.

Mehrotra, K., Mohan, C.K., and Ranka, S. (1997). Elements of Artificial Neural Networks, Penram International Publishing (India), India.

Minai, A.A., and Williams, R.D. (1990). Acceleration of back-propagation through learning rate and momentum adaptation. Int. Joint Conf. Neural Networks. Lawrence Erlbaum Assoc., Hillsdale, N.J., Vol. 1, 676-679.

Minns, A. W., and M. J. Hall, (1996). Artificial neural networks as rainfall runoff models, Journal of hydrological sciences, 41(3): 399-417.

Murtaza, M.B. and Fisher, D.J., (1994). Neuromodex- Neural network system modular construction decision making. J. Comp. in Civ. Engg., ASCE, 8(2), 212-233.

Nash, J.E. and Sutcliffe J.V. (1970). River flow forecasting through conceptual models: 1.A discussion of principles. Journal of hydrology, 10: 282 - 290.

Nalbantis, I, (1995). Use of multiple-time-step animation in rainfall-runoff modeling. J. Hydro., 165(3), 135-159.

Raman, H. and Kumar, S., (1995). "Multivariate modeling of water resources time series using a*rtificial neural networks,"* Hydrological Sciences Journal, 40(2), 145-163.

Raman, H., and Chandramouli, V., (1996). Deriving a general operating policy reservoir using neural networks. Journal of Water Resources Planning and Management, ASCE, 122(5), 342-347.

Rissanen, J. (1978). Modeling by short data description, Automation, 14:465-47 1.

Romesburg, H. C. (1984). Cluster analysis for researchers. lifetime learning publications, Belmont., California.

Roger, L.L., and Dowla, F.U. (1994). Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resour. Res., 30(2), 457-481.

Rumelhart, D. E., McLelland, J. L., and the PDP Research Group. (1986). Parallel distributed processing, explorations in the micro structure of cognition, vol. 1: Foundations. MIT Press, Cambridge, Mass

Rumelhart, D. E., and MeCielland, J., eds. (1986). Parallel distributed processing, Vol. 1, M IT Press, Cambridge.

Rumelhart, D.E., Hinton, E., and Williams, J. (1986). Learning internal representation by error propagation in parallel distributed processing. MIT press, Cambridge, Vol 1: 318 - 362.

Sajikumar, N., (1998). Rainfall-runoff modeling in data scarce situation using artificial neural network, Ph.D. thesis, Department of Civil Engineering, Indian Institute of Technology Madras. Chennai, India.

Sajikumar, N., and Thandaveswara, B.S. (1999). A non-linear rainfall-runoff model using an artificial neural network. Journal of Hydrologv, 216, 32-55.

Salas, J. D., J. W. Delleur, V. Yevjevich, and W. 1. Lane, (1988). Applied modeling of hydrologic time series, Littleton, Colarado, Water Resources Publications.

Senthil Kumar, A.R., Jain, S.K., Goel, M.K., and Nema, R.K. (1997). Development of Operation Policy for Tawa Dam. CS(AR)-18/96-97, National Institute of Hydrology, Roorkee, India.

Shemseldin, A. Y., (1997). Application of a neural network technique to rainfall-runoff modeling. J. Hydro., 199, 272-294.

Smith, J., and Eli, R.N., (1995).Neural network models of rainfall-runoff process. J. Water Resour. Plng. and Mgmt., ASCE, 121(6), 499-508.

Sorooshian, S., Gupta, V.K., and Fulton, J.I. (1983). Evaluation of maximum likelihood parameter estimation techniques conceptual rainfall-runoff models: Influence of calibration data variability and length on model credibility. Water Resour. Res., 19(1), 252-259.

Sorooshian, S., Duan, Q., and Gupta, V.K.(1993). Calibration of rainfall-runoff models: Application of global optimization to the Sacramento soil moisture accounting model. Water Resour. Res., 29(4), 1185-1194.

Sudheer, K.P., Nayak, P.C., and Mohan R.D., (2000). Rainfall runoff modeling using ANN techniques. CS/AR-16/1999-2000. National Institiute of Hydrology, Roorkee, India.

Thandaveswara, B.S. and Sajikumar, N., (1999). Classification of river basins using artificial neural network, Journal of Hydrologic Engineering. ASCE (accepted for publication).

Thirumalaih, K., and Deo, M.C., (1998). River stage forecasting using ANN. Jour. Of Hydologic. Engg, ASCE, 3(1), 26-32.

Tokar, A.S., and Markins, M., (2000). Precipitation runoff modeling using ANN and conceptual models. Jour. Of Hydrolo. Engg, ASCE, Vol.5, No. 2, 156-161.

Werbos, P.J., (1990) Backpropagation through time: What it does and how to do it ?. Proc. Of IEEE, 79:1550-1560.

Wooishier, D. A. (1 996). Search for physically based rainfall runoff model - a hydrological El Dorado?, Journal of hydrologic engineering, 122: 122-129.

Yang, C-C., Prasher, S.O., Lacroix, R., Sreekanth, S., Patni, N.K., and Masse, L. (1997). Artificial neural network model for subsurface drained farmlands". Journal of Irrigation and Drainage Engineering, ASCE, 123 (4), 285-292.

Zealand, C.M., Bum, D.H., and Simonovic, S.P. (1999). Short Term Streamflow Forecasting using Artificial Neural Networks. Journal of hydrology, 214, 32-48.

Zhu, M.L., Fujitha, M., (1994). Comparison between fuzzy reasoning and Neural networks methods to forecast runoff discharge. J. Hydro. and Hydr. Engrg. 12 (2), 133-141.

Table 5.1: Thiessen weights for Sindh river basin up to Madhikhera dam site

Raingauge Station	Area under influence (sq. km)	Thiessen weight
Mundrasagar	387.80	0.07
Aron	1052.60	0.19
Behtaghat	886.40	0.16
Maina	332.40	0.06
Khatora	1052.60	0.19
Rannod	831.00	0.15
Amola	997.20	0.18

Table 5.2: Runoff coefficients for Sindh at Madhikhera dam site

Year	Monsoon season rainfall (mm)	Monsoon season runoff (mm)	Runoff coefficient
1992	559.8	373.65	0.667
1993	826.2	563.29	0.682
1995	751.1	451.30	0.601
1996	1079.0	692.02	0.641
1997	851.1	552.29	0.649
1998	592.5	296.89	0.501
1999	879.0	532.54	0.606
2000	834.7	430.59	0.516
2001	709.8	364.55	0.514

Table 7.1 : Goodness of fit statistics for the effect of rainfall lags in the input vector

Network	Rainfall pattern	RMSE		Efficiency (%)	
		Training	Testing	Training	Testing
BPN with 5 neurons	$P_{t-1}, \ldots\ldots, P_{t-2}$	150.90	173.08	85.14	84.42
	$P_{t-1}, \ldots\ldots, P_{t-3}$	140.09	245.99	87.19	69.05
	$P_{t-1}, \ldots\ldots, P_{t-4}$	136.91	256.98	87.76	66.78
	$P_{t-1}, \ldots\ldots, P_{t-5}$	136.98	311.16	87.75	52.02

Table 7.2: Goodness of fit statistics for the effect of number of neurons in hidden layer

No. of neurons in the hidden layer	RMSE		Efficiency (%)	
	Training	Testing	Training	Testing
2	193.72	269.61	75.50	62.20
4	158.01	203.97	83.70	78.37
5	150.90	173.08	85.14	84.42
8	117.45	228.62	90.99	72.82
10	113.42	306.21	91.60	51.24
15	100.10	342.57	93.46	38.98

Table 7.3: Performance indicators of the selected model during training and testing

S. No.	Performance Indicator	Training	Testing
1.	Correlation Coefficient	0.923	0.926
2.	RMSE	150.90	173.08
3.	% Difference in peak	1.335 %	0.319 %
4.	Efficiency	85.14 %	84.42 %

Table 7.4: Efficiency of the trained and tested model during 1992 to 2001

Year	Efficiency (%)	RMSE	RMAE	Rel. Bias	Correlation coefficient
Training					
1992	69.59	114.11	0.395	-0.003	0.87
1993	92.33	135.50	0.315	0.058	0.96
1995	70.19	173.49	0.469	0.053	0.84
1996	93.86	130.73	0.261	0.063	0.97
1997	56.22	225.44	0.494	-0.140	0.82
1998	86.63	84.38	0.361	0.089	0.94
Testing					
1999	82.89	157.73	0.400	0.081	0.91
2000	81.72	158.90	0.462	0.203	0.97
2001	71.04	199.35	0.552	0.066	0.79