TR/BR-5/1999-2000

APPLICATION OF ARTIFICIAL NEURAL NETWORK IN FLOOD STUDIES OF AJAY RIVER BASIN

आपो हि ष्ठा मयोभुवः

NATIONAL INSTITUTE OF HYDROLOGY
JALVIGYAN BHAWAN
ROORKEE - 247 667, UTTARANCHAL
INDIA
1999-2000

PREFACE

Hydrology may be broadly defined as the study of the life cycle of water. The most important process in this cycle is the section where rainfall occurs and results in runoff. The runoff is important to many activities such as designing flood protection works for urban areas and agricultural land, better water management etc.. The quantity of runoff resulting from a given rainfall event depends on a number of factors such as initial moisture content, land use and slope of catchment as well as intensity, distribution and duration of the rainfall. Mathematical models have been developed based either on physical considerations or on statistical analyses to estimate runoff resulting from a given rainfall event. Both approaches are difficult and the types of forecast they provide are not completely satisfactory. Unlike mathematical models which require precise knowledge of all the contributing variables, an Artificial Neural Network (ANN) model can model process behaviour even with incomplete information. It is a proven fact that once the ANN have been properly trained, they are able to provide accurate results even for cases they have never seen before.

In hydrological context, as in many other fields, ANN are increasingly used as black-box, simplified models. For hydrological applications, ANN models can take advantage of their capability to reproduce the unknown nonlinear relationship existing between a set of input variables descriptive of the system, for example, rainfall, and a set of output variables, for example, river flow rate.

Based on the ANN technique an ANN model has been developed at National Institute of Hydrology for Ajay river basin to forecast 6 hr ahead forecast at Sarath gauging site of Jharkhand (South Bihar). The performance of the developed model is evaluated based on root mean square error (r.m.s.e.) on training and test data. The study has been carried out by Dr Sanjay Kumar, Shri Rakesh Kumar and Dr. Chandranath Chatterjee, scientists of the institute; as per the work program of Ganga Plains North Regional Centre, Patna. Technical assistance has been provided by Shri A. K. Sivdas, technician. It has been illustrated that ANN are capable of modeling nonlinear flood events and may be used effectively to provide reasonable forecast with sufficient lead time to mitigate the undesirable effects of floods.

(K. S. Ramasastri)
DIRECTOR

CONTENTS

LIST OF TABLES

LIST OF FIGURE

ABSTRACT

In recent years there has been a growing interest in artificial neural networks (ANN) which operate in a manner analogous to that of biological neurons system. The ANN technique offers several advantages over conventional computing methods. Most important among these is ANN's ability to generalise a solution to a problem from a set of example. The ability of ANN to model nonlinear events advocates their use in hydrology to model various hydrological events which are dominantly nonlinear in nature. In the study a comprehensive review of ANN technique is presented. The study shows how and why such a system works. A case study has been performed for Ajay river basin to demonstrate the applicability of ANN techniques in model development.

In the case study, an Artificial Neural Networks (ANN) model is developed for Ajay river basin up to Sarath gauging site of Jharkhand (South Bihar) for forecasting floods which are generally sudden, flashy and of short duration and require solutions in quick succession. Artificial Neural Network (ANN) is important, where the time required to generate solutions is critical and is also capable of modeling nonlinear relationship between rainfall-runoff process as compared to other methods which assume the linear relation. The ANN model is developed utilising the available limited rainfall and runoff data of the Ajay river basin up to Sarath. The model is used to forecast 6-hr ahead runoff at Sarath. In the process of model development the 1-hourly rainfall–runoff data are divided into two groups. One part is used to find the network connection weights to represent the relationship between rainfall and runoff. The backpropagation algorithm is used to optimise the network connection weights. The developed model is then validated on the second set of data. The performance of the model on training and test data has been evaluated based on root mean square error (r.m.s.e.) index. The result shows that ANN based rainfall-runoff model can develop a nonlinear relationship for rainfall-runoff process even when the training data are limited and contain noise components. The trained ANN model is then applied to problems other than those used for training for validation. The validated model can be used to issue 6-hr ahead forecast at Sarath. However the performance of the model can be significantly improved if more rainfall-runoff events, with low noise components are made available for the study.

APPLICATION OF ARTIFICIAL NEURAL NETWORK IN FLOOD STUDIES OF AJAY RIVER BASIN

1.0 INTRODUCTION

Flood forecasting is an important non-structural solution for reducing flood damages and is used to provide warning to people residing in flood plains. Flood forecasts also help water management personnel to take optimal decision to operate flood control structures and reservoirs. Forecasting of floods involve the estimation of flood magnitude for the future using the past experiences. The process of flood forecasting involves gathering real time precipitation and stream flow data and using these data into rainfall-runoff and stream flow routing programs to forecast flood flow rates and water levels for periods varying from few hours to few days ahead, depending on the size of basin.

Floods are natural phenomena and are inherently complex to model. The quantity of runoff resulting from a given rainfall event depends on a number of factors, thus rainfall-runoff process are not amenable to mathematical modelling. Statistical solutions are possible. Several stochastic models have been proposed for modelling hydrological time series and generating synthetic stream flows. These include auto regressive moving average (ARMA) models (Box and Jenkins, 1970), deaggregation models (Valencia and Schaake, 1973), models based on the concept of pattern recognition (Panu and Unny, 1980). Stedinger and Taylor (1982) studied the performance of five different models for stream flow simulation. Statistical correlation techniques have been employed (Mutreja et al., 1987) for direct prediction of water levels. The routing techniques are more useful when the travel time is longer and downstream flow is low or controlled (Singh, 1988). It has been observed that the dynamic stochastic time series models are most suitable for on line forecasting of floods (Kalman, 1960; Sage and Husa, 1969; Eykhoff, 1974; Kashyap and Rao, 1975; Kumar, 1980; O'Connell, 1980; Chander et al., 1980, 81,84). Time series modelling and its application to online forecasting of floods were evolved in 1970's and have been successfully applied in real time forecasting.

Recently another class of black box models in the form of Artificial Neural Network (ANN) has been introduced in modelling real time problems. The ANN model has wide applicability in Civil Engineering applications and many research papers have been published on its application. The use of ANN in real time flood forecasting is of very recent origin and is still in the evolution stage. The present study has concentrated on the most popular class of neural networking system (The layered feed forward, back propagation network) and looked at how and why such a system works. The objective of this preliminary research is to examine an alternative empirical method with the hope of achieving two things: (a) Greater accuracy and (b) increased convenience for the user to develop an appropriate model for the flow history. In the study an ANN model has been designed and developed for the flow history of Ajay river basin to generate runoff at Sarath gauging site in Jharkhand (South Bihar).

2.0 LITERATURE RIVIEW

Neural networks offer several advantages over the more conventional approaches in computing. The most frequently cited of these are their ability to develop a generalised solution to a problem from a set of examples. The attribute of generalisation permits them to be applied to problems other than those used for training and to produce valid solutions even when there are errors in the training data (Hecht-Nelson, 1991; Haykin, 1994).

Another important area that is benefited from ANN approach is, where the time required to generate solutions is critical, such as real-time applications that require many solutions in quick succession. In particular, the ability of ANN to produce solutions in a fraction of a second, irrespective of the complexity of the problem, makes them valuable even when alternative techniques are available which can produce more optional or more accurate solutions. These factors make the ANN a powerful tool for modelling problems in which functional relationship between dependent (runoff) and independent (rainfall) variables are poorly understood and subject to uncertainty.

The potential of ANN techniques to solve civil engineering problems has been discussed in recent studies (Flood and Kartam, 1994a,b). However, their applications to simulation and forecasting problems in water resources are a few and relatively recent.

French et al. (1992) developed an ANN model to forecast rainfall in space and time. Kang (1993) used multi-layer perception model to forecast stream flows at Pyunchang river basin in Korea. They showed that the rainfall-and runoff simulation problem can be considered as a pattern recognition problem. Buch et al. (1993) have used ANN in runoff simulation of a Himalayan glacier. It was found that ANN performance was superior compared to the energy balance and multiple regression models. In addition it was observed that the ANN was faster in learning and exhibited excellent system generalisation characteristics.

Crespo and Mora (1993) have used ANN to derive stream flow from precipitation data and further extended their work to drought estimation. Karunanithi et al. (1994) developed a ANN for the simulation and prediction of daily flow of Huron river at Dexter sampling station. Smith (1995) used a back-propagation ANN model to predict only the peak discharge and the time to peak resulting from a single rainfall pattern. The data used were simulated from synthetic watershed that generated runoff from a stochastically generated rainfall. Raman and Sunil Kumar (1995) compared the performance of neural network with the statistical method for synthesising monthly inflow records for two reservoir sites.

Patrick et al. (1996), designed and developed a Virtual Runoff Hydrograph System (VROHS) based on ANN to generate runoff hydrograph. Data from 45 lab experiment sets were used to develop the VROHS. Twenty-nine of the 45 lab experiment sets were randomly selected to train the network, while 16 experiment sets were selected to test the VROHS. It was found that the VROHS could predict the runoff hydrograph system very accurately for sets of input data that it had never seen before.

Konda et al. (1998) highlights their use in real time forecasting of water levels at a given site. The ANN is trained by using three algorithms, namely, back propagation, cascade correlation, and conjugate gradients. The results are compared with trained and untrained data.

In the study an ANN model is developed for Ajay river basin for forecasting floods which are generally sudden, flashy and of short duration and require solutions in quick succession. A brief introduction of the Ajay river system, its topograhic and hydrometeorological characteristics with flood problems is given in the following sections.

3.0 DESCRIPTION OF THE STUDY AREA

The catchment of the Ajay river spreads between Latitude 23°25'N to 24°35'N and Longitude 86°15'E to 88° 15'E. The Ajay river system originates in the low hills near Deoghar in the Santhal Pargana district of Jharkhand (South Bihar) and flows in a South-Easterly direction passing through Monghyr district of Jharkhand (South Bihar) and Birbhum and Burdwan district of West Bengal. Ajay river ultimately falls into the river Bhagirathi at Katwa about 216 Km above Catcutta. The Ajay river system lies between the Mayurakshi on the North and the Damodar and the Banka/Khari river system on the South.The portion of Ajay river basin lying in Jharkhand (South Bihar) along with other river basins of Jharkhand (South Bihar) is shown in Figure 1. The habitation in the catchment, on the whole, is dense and requires a reliable flood warning system which can provide timely warning to the people with sufficient lead time to take precautionary measure to save lives and their valuables. Major centres of habitation in the river system are Chakai in Monghyr, Jasidih, Deoghar, Sarwan,Karon, Jamtara etc. in the district Santhal Pargana, Khyarasota, Jayadeb, Kenduli, Ilambazar, Bolpur, Nanur etc. in district Birbhum, Rupnarayanpur, Churulia, Ukhra, Kakas, Gushhara, Katwa etc. in district Burdwan annd Kagram in district Murshidabad.

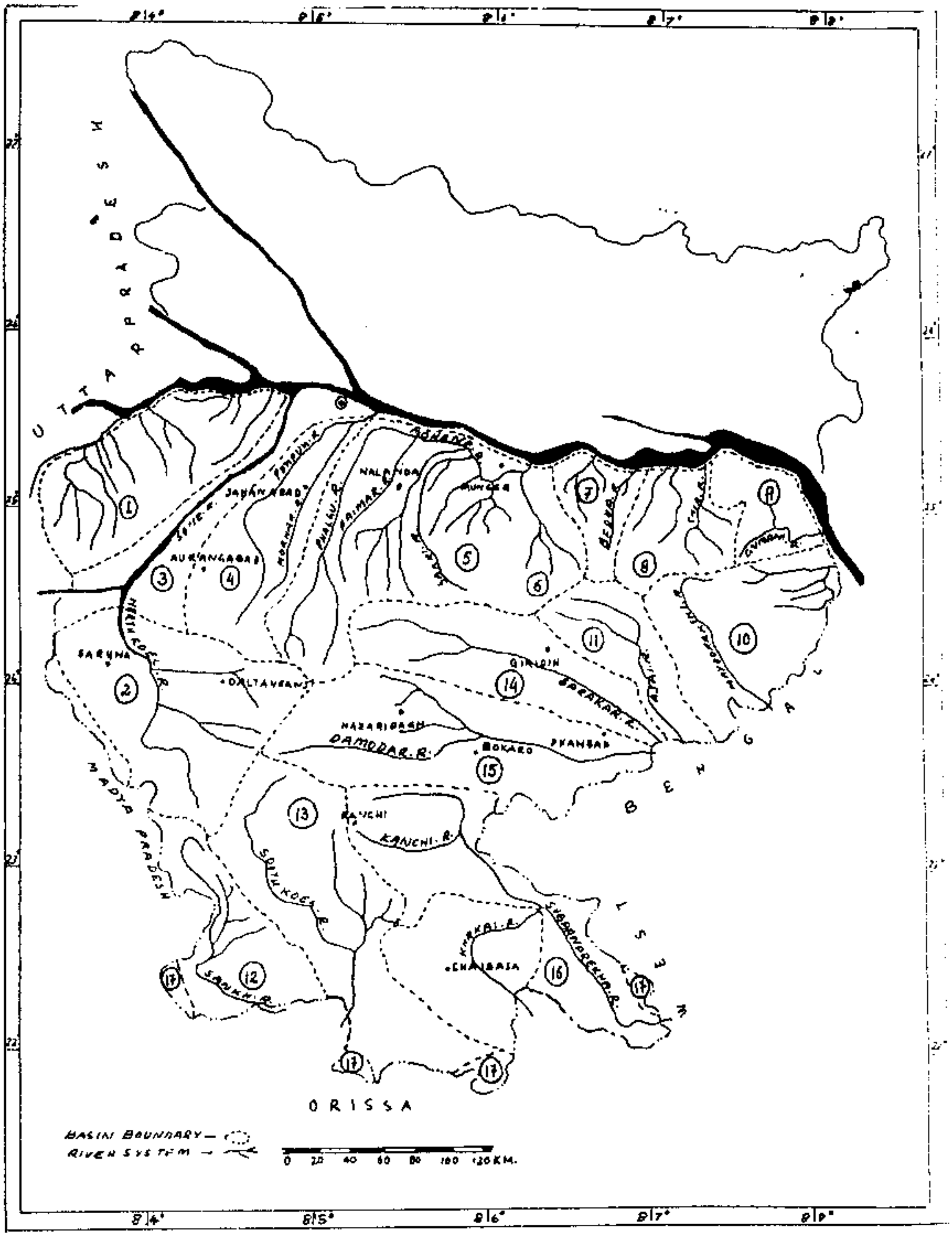
3.1 River System

Ajay river traverses a total length of 299 Kms, 24 Kms being in Monghyr, 102 Kms in district Santhal Parganas, 22 Kms along the boundary of Santhal Parganas and Burdwan, 115 Kms along the boundary between Singhbhum and Burdwan and rest of the total length falls in Burgwan district of West Bengal before it meets the Bhagirathi near Katwa. The various tributaries of Ajay river have been shown in Figure 2. Distances of confluence of various tributaries from origin are shown in Table-1. The river has a total catchment area of 6050.00 Sq.Km. The district wise distributions of its catchment area are shown in Table-2.

3.2 Topography

The catchment area of the river is long and narrow. The river course has remained more or less firm and there is no evidence of marked changes in the course. The bed slope of the river in different reaches vary considerably. The slope of the main river is flatter than that of the tributaries and slope in the plains of West Bengal is much flatter than that in its upper reaches.

Fig. 1.2 Map Showing Different River Basins of South Bihar (Jharkhand)



1 Karmnasa Basin
2 North Koel Basin
3 Sone-Kanhar and Kao-Gangi Composite Basin
4 Punpun Basin
5 Harohar Basin
6 Kiul Basin
7 Badua-Belharna Basin
8 Bilasi-Chandan-Chir Basin
9 Gumani and Koa-Bhena Basin

10 Basins of Mayurakshi and Other Adjoining Streams
11 Ajay Basin
12 Sankh Basin
13 South Koel Basin
14 Barakar Basin
15 Damodar Basin
16 Subarnarekha-Kharkai Basin
17 Basins of Small Streams Flowing Independently Outside the State

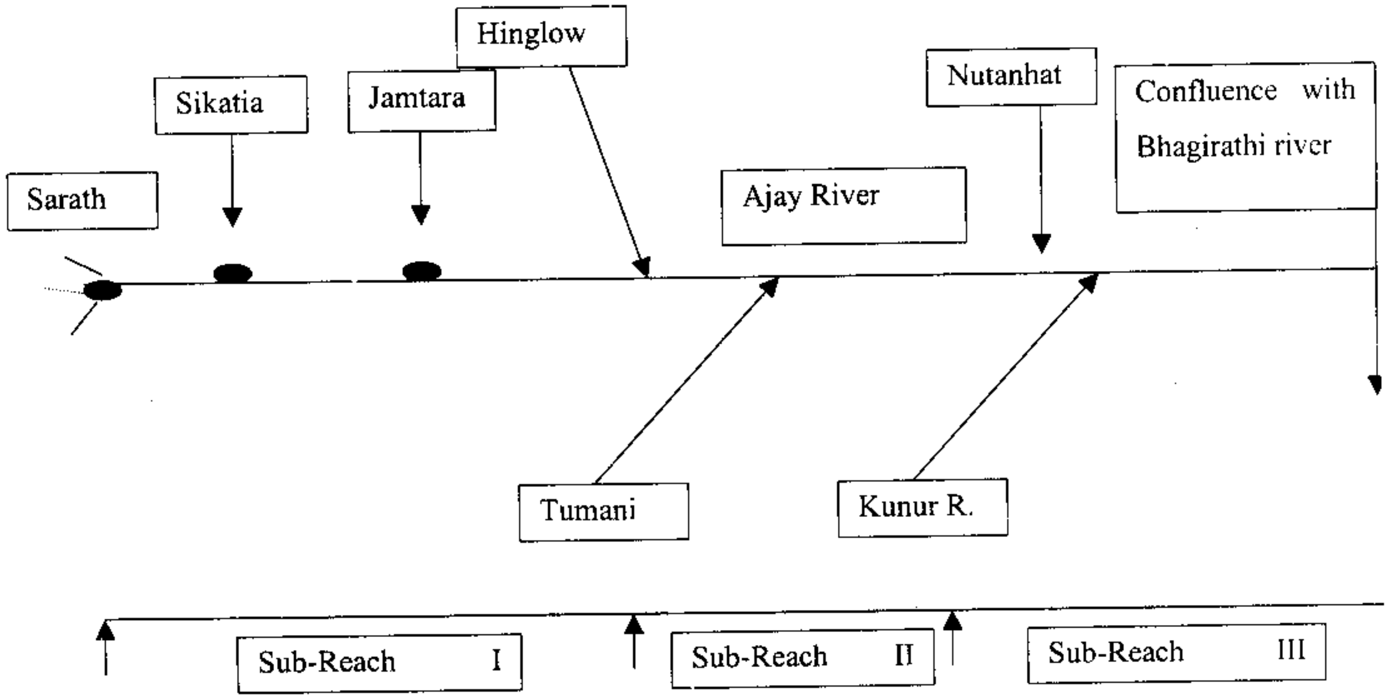


Figure 2: Schematic Line-Diagram of Ajay River

Table 1: Distance of confluence of various tributaries.

S.No.	Name of Confluence Nadi	Distance(Km)
1	Confluence with Darua Nadi	48
2	Confluence with Pathro Nadi	80
3	Confluence with Jainti Nadi	86.78
4	Confluence with Hinglo Nadi	180.29
5	Confluence with Tumuni Nadi	191.22
6	Confluence with Kane Nadi	240
7	Confluence with Kunur Nadi	252.8
8	Confluence with Kundur Nadi	293.89
9	Confluence with Bhagirathi	299

Table 2: State and district wise distribution of catchment area.

State	District	Area(Km^2)
Jharkhand (South Bihar)	Monghyr	375.88
Jharkhand (South Bihar)	Giridih	303.72
Jharkhand (South Bihar)	Santhal Paraganas	2778.44
West Bengal	Birbhum	760.86
West Bengal	Burdan	1784.40
West bengal	Murshidabad	46.70
Total		6050.00

3.3 Hydrometeorological Characteristics

The Ajay catchment lies in the path of tropical depression or cyclonic storms which forms in the Bay of Bengal in the monsoon period and moves generally in the North-West direction. Abnormally heavy rain spells are generally associated with depressions or cyclonic storms during monsoon. The average annual rainfall in the river catchment varies from 1280mm to 1380 mm. The rainfall is more in the hilly catchment than in plains. Nearly 75 to 80% of the rainfall is concentrated in the four monsoon months of June to September. There are 38 numbers of rainguage stations in the catchment, which are maintained either by IMD or by the state Government.

3.4 Flood Problem

The river system is divided into three reaches (Figure 2) from the point of view of floods. The upper reach is almost hilly and having comparatively steep slope. There is no flood problem in this reach. The middle reach is subjected to occasional flooding, caused due to breaches and overtopping of the embankment during high floods in the river. Whenever, the river is in high stage, flood locking of its tributaries takes place, which causes heavy spill on both sides of the rivers. Even during moderately high discharge the lower reach is the worst flood prone reach of the river and it suffers from very frequent inundation. Table 3 shows some of the severe floods and their effects over the years.

Table 3: Some of the severe floods and their effects.

Year	Nature of Flood	Breaches in Embankment	Extent of Flooding
1956	Extremely severe floods occurred in the last week of September.	Extensive breaches of embankments. Several bridges in Birbhum disdrict were washed away.	On both banks from Pandaveshwar to Katwa extensive areas were flooded.
1959	Extremely Severe flood occurred in the first week of October	Ex-Zamindari embankments were severely damaged	Extensive areas were flooded from the lower reach of schedule 'D' embankment to Katwa.
1961	High flood occurred in the month of October	Nil	On the right bank flooding occurred near Mangalkot, while on the left bank, the area between Khayrasol to Katwa was flooded. Total area affected was about 285 Km^2 .
1963	Towards the fag end of October there was heavy rainfall and the estimated peak at lambazar was 3197.9 cumec	Several breaches ocurred in the Ex-Zamindary Embankments at Najkhara.	About 129.50 Km^{2}were flooded in Birbhum and Burdwan District for a short duration but no crops were damaged.
1967	There was temporary flooding for 26 hours through the open mouth of the tributaries.	Nil	Approximately 518 Km^2 were flooded in Kalyanpur paschim Nabagram in P.S. Mangolkot for short duration.

1968	There was a high flood in June.	3 breaches occurred in the portion of right Exzamindary embankment from Jajkar to Taldanga	Extensive areas were flooded.
1969	There was a high flood in August.	One breach occurred at Mouza Paligram P.P. Mangolkot & 2 breaches occurred near Ch. 260 & 271.	Extensive area were flooded in Kaiyanpur Majghor, Pashchim, Nabagram, Paligram & Mangalkot.
1971	Flood occurred in the months of August – September when Bhagirathi ruled very high causing the most severe drainage congestion.	Nil	Area between Purucha and Katwa was flooded affecting mouzas Gopalpur, Kalyanpur, Taldanga, Ausgram, Kankore, Madhur, Pandhugarm.
1978	Extensive breaches in embankments on both banks occurred. Several road bridges and villages were washed out in the district of Burdwan and Birbhum.	Nil	An area of 1562 Km^2 in total on both banks from pandaveshar to Katwa was inundated.

The gauging site, establised near village Sarath and about 160 m upstream of Sarath – Madhupur road bridge, is maintained by Water resources Department, Govt of Jharkhand (South Bihar). The geographical location of the site is 24^0 13' 45" N latitute and 86^0 50' 43" E longitude which is approachable by road in all seasons. The catchment area upto the site is 1,191.40 sq km. The length of Ajay river up to Sarath gauging site is 82.18 sq km. For the present study, Ajay river basin upto Sarath forms the study area. Figure 3 shows the index map of the part of Ajay river basin lying in Jharkhand along

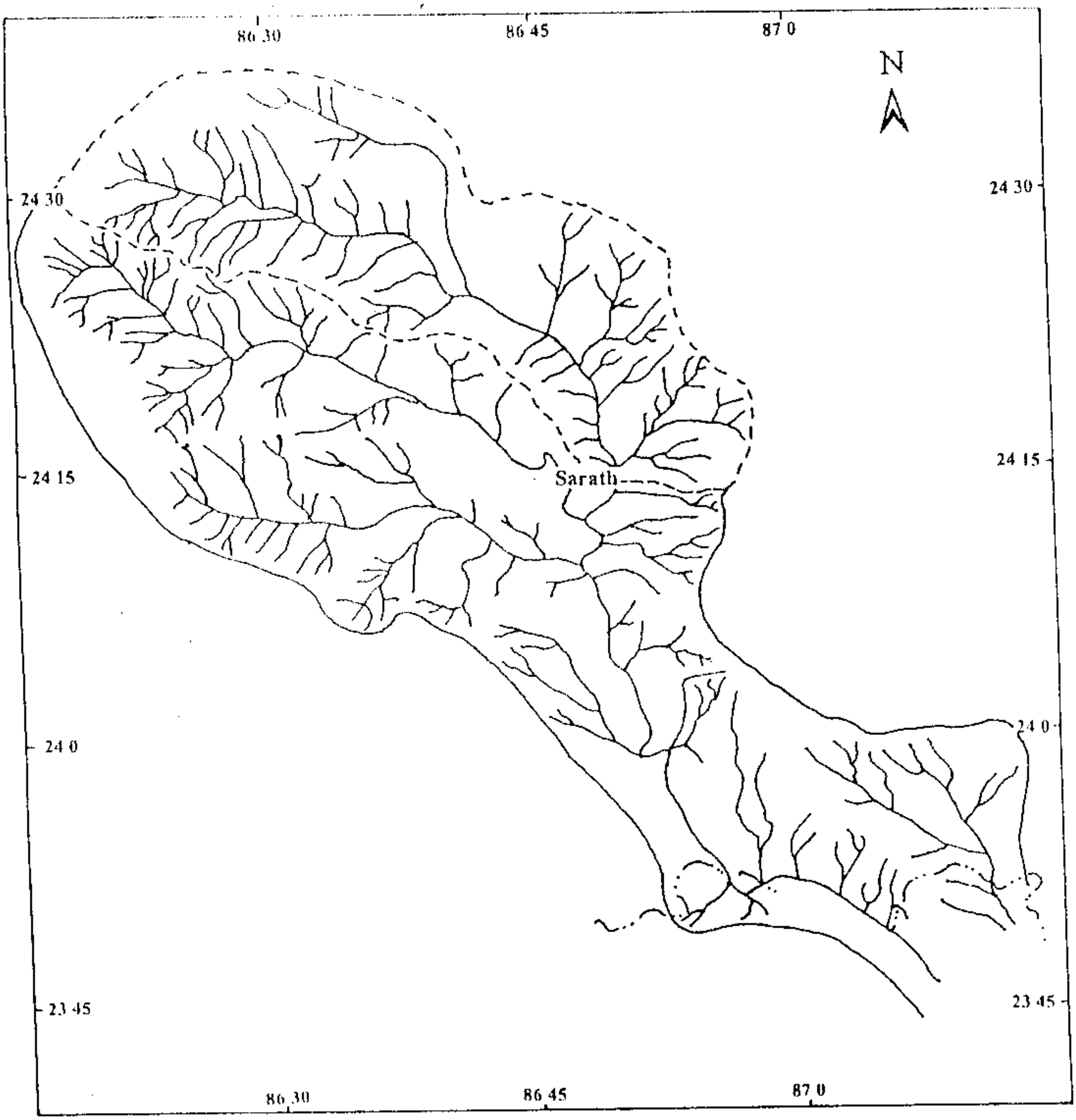


Figure 3: Index Map of Part of Ajay River Basin in Jharkhand (South Bihar) with Catchment Defined by Sarath Gauging Site

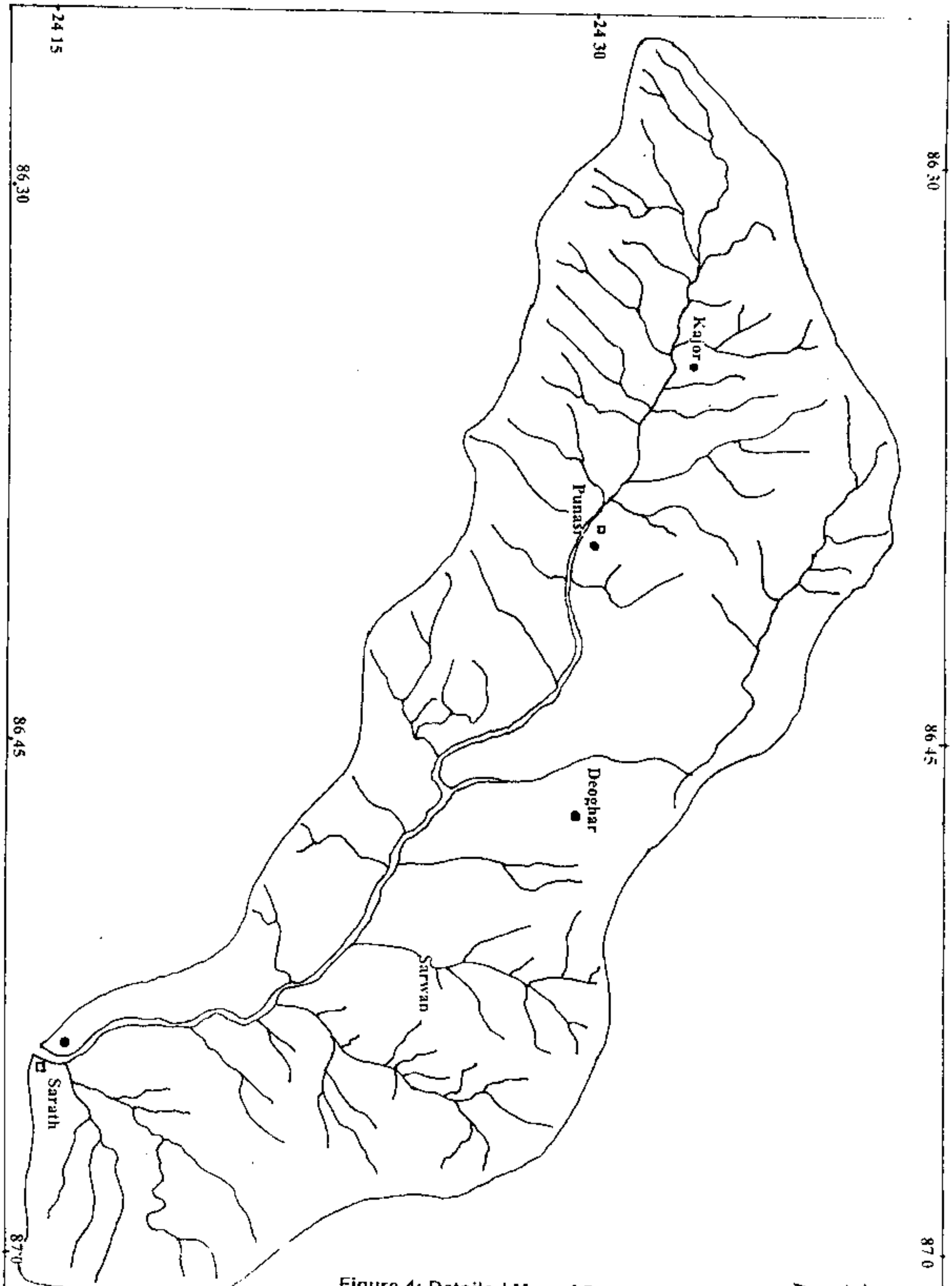


Figure 4: Detailed Map of Study Area

with the catchment defined by Sarath gauging site which forms the study area. Figure 4 shows the detailed map of study area showing drainage network of the study area and location of different raingauge stations.

On account of the peculiar nature of its catchment and rainfall characteristics, the floods in the river are sudden, flashy and of short duration and require solutions in quick succession. ANN model being capable of producing solutions in quick succession for nonlinear flood events may be best suited for flood forecasting for Ajay river basin.

4.0 DATA AVAILABILITY

Data from five raingauge stations located in the sub-basin (as shown in Figure 4) were collected. The daily rainfall data at these stations and three hourly runoff data at the gauging site at Sarath have been collected for the years 1977 to 1982 from Hydrology cell, Department of Water Resources. List of raingauge stations along with their location is given in Table 4. As there is no self-recording raingauge available in the basin, rainfall pattern for the neareby stations Sabour and Bhagalpur for the same period, are also collected from Indian Meteorolgical Department, Patna and the same were used based on their rainfall data for converting the daily rainfall data into hourly data.

Table 4: Location of rain gauge stations

S. No.	Station	Longitude (E)	Latitude (N)
1	Kiajori	86°28'22"	24°29'40"
2	Punasi	86°32'16"	24°28'57"
3	Deoghar	86°45'21"	24°29'18"
4	Sarwan	86°47'26"	24°22'50"
5	Sarath	86°50'34"	24°13'57"

Table 5 gives the details of periods of various storms whose rainfall-runoff data have been used in the study. Figure 5 shows the Theissen polygon of raingauge stations. The Theissen weights of the raingauge stations in the catchment whose data have been used in the study are given in Table 6.

Table 5: Periods of various rainfall-runoff events

S. No.	Period of the Events
1	13.08.1977 at 09 hrs. to 13.08.1977 at 20 hrs.
2	05.08.1978 at 21 hrs. to 06.08.1978 at 02 hrs.
3	16.08.1979 at 05 hrs. to 16.08.1979 at 16 hrs.
4	26.08.1980 at 15 hrs. to 26.08.1980 at 07 hrs.
5	22.08.1982 at 24 hrs. to 23.08.1982 at 06 hrs.
6	12.09.1987 at 13 hrs. to at 12.09.1987 24 hrs.

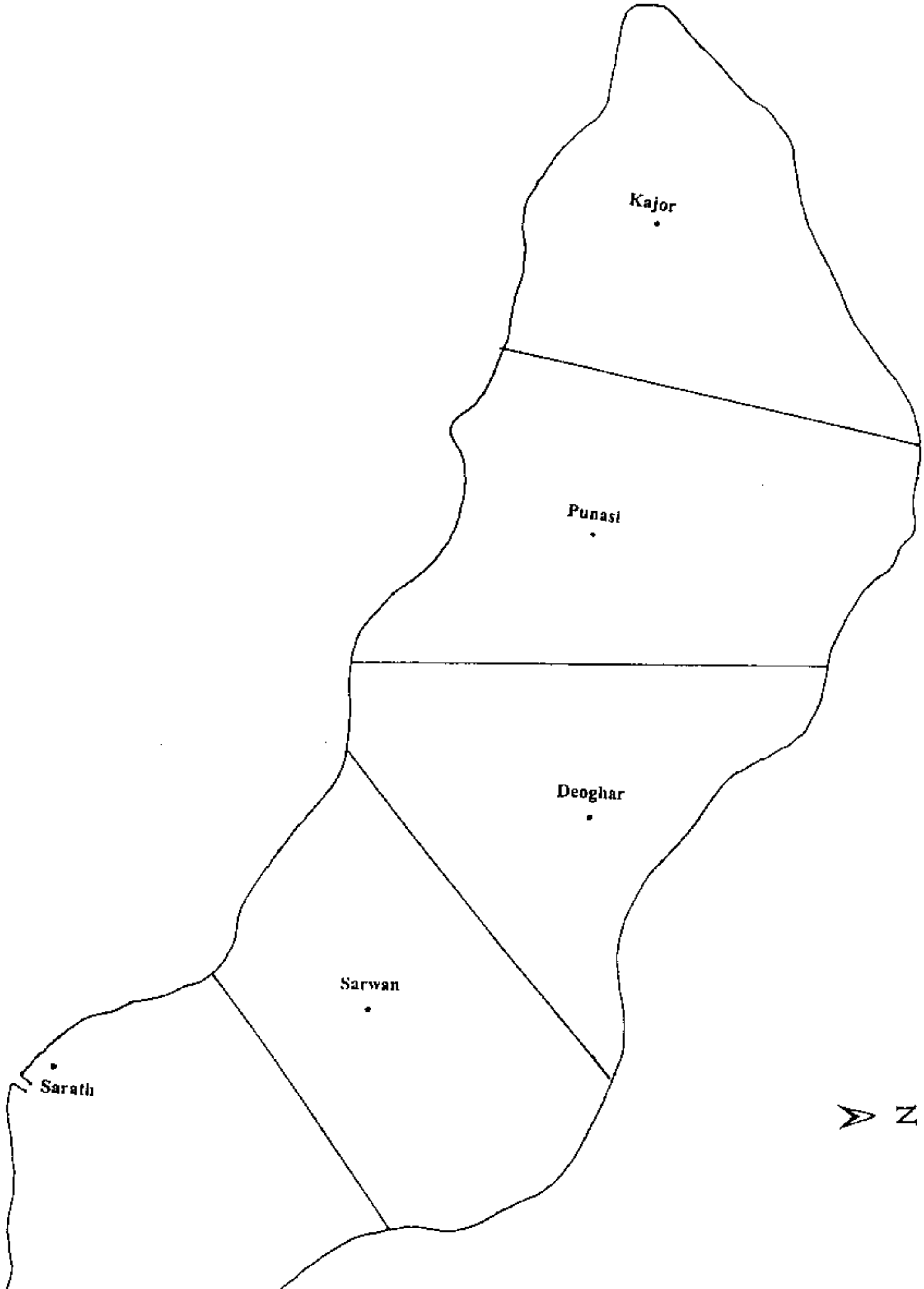


Figure 5: Theissen Polygon of Raingauge Stations

Table 6: Theissen weights of the raingauge stations

S.No.	Name of Raingauge station	Theissen weight
1	Kijor	0.15
2	Punasi	0.26
3	Deoghar	0.20
4	Sarwan	0.20
5	Sarath	0.19

5.0 ARTIFICIAL NEURAL NETWORK (ANN) METHODOLOGY

5.1 Preliminaries

The neural-network approach, also referred to as connectionism or paralleled distributed processing, adopts a "Brain metaphor" of information processing. Information processing in a neural network occurs through interactions involving large number of simulated neurons, such as the one depicted in Figure 6.

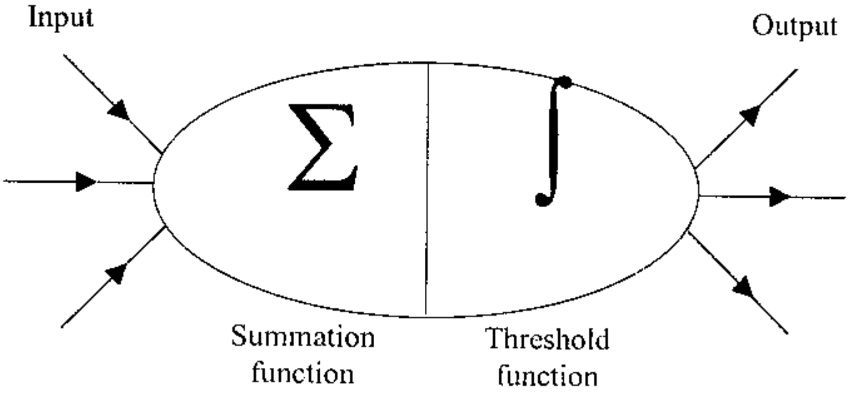


Figure 6: Simulated Neuron.

This simulated neuron, or unit, has four important components.

(i) Input connections (synapses), through which the unit receives activation from other units.

(ii) Summation function that combines the various input activations into a single activation.

(iii) Threshold function that converts this summation of input activation into output activation.

(iv) Output connections (axonal paths) by which a unit's output activation arrives as input activation at other units in the system.

Artificial neural-networks (ANNs) are massively parallel systems composed of many processing elements connected by links of variables weights. The back propagation network is the most popular (Lippman, 1987) of many ANN paradigms. The network consists of layers of neurons, with each layer being fully connected to the proceeding layer by inter connection strengths or weights W . Figure 7 illustrates a three-layer neural network consisting of input layer (L_i), hidden layer (L_H) and the output layer (L_o) with the inter-connection weights W_{ih} and W_{ho} between layers of neurons.as shown in Figure7. Initial estimated weight values (i.e W_{ih} and W_{ho}) are progressively corrected during a training process that compares predicted outputs to known outputs and back propagates any errors to determine the appropriate weight adjustments necessary to minimise the errors. The methodology used here for adjusting the weights is called "momentum back propagation" and is based on the "generalised delta rule" (Rumelhart et al., 1986). The mathematical description of generalised delta rule is given in the next section

In modelling rainfall – runoff process, the input neurons in the input layer (L_I) and output neuron in output layer L_o consists of runoff (y_k) and rainfall (w_k) (k being any instant of time). However in the subsequent section for generalisation the output of hidden neurons in the hidden layer is denoted by H_{oj} for j^{th} hidden neuron and output of the output neurons in the output layer L_o is denoted by O_j for the j^{th} output neuron.

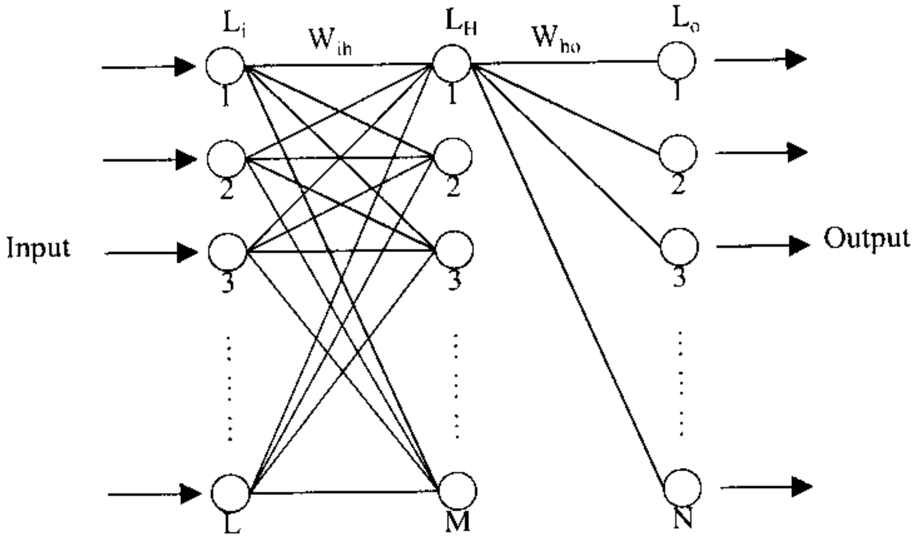


Figure 7 : Configuration of Three-Layer Neural Network.

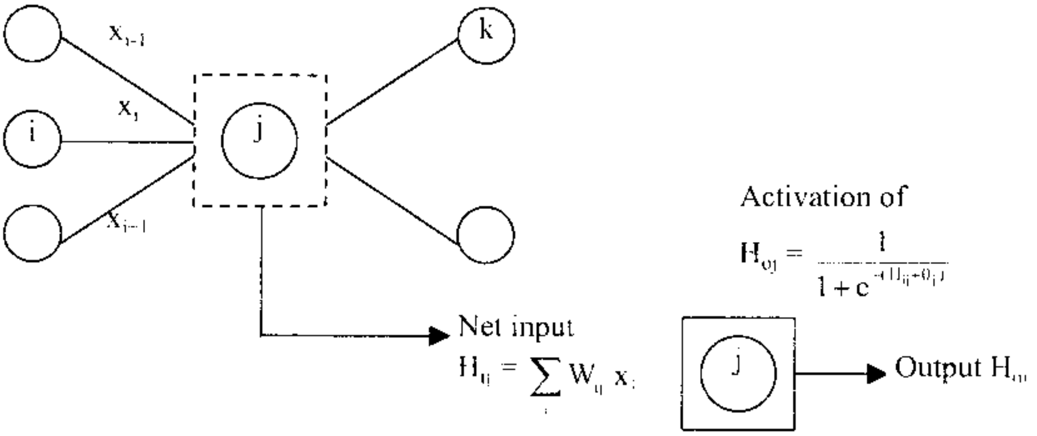
5.2 Generalised Delta Rule (Back Propagation Algorithm)

The generalised delta rule, which determines the appropriate weight adjustments necessary to minimise the errors can be explained through Figure 8. The Figure 8 shows a neuron (j) and its functions.

The total input H_{ij} to hidden units j is a linear function of outputs x_i of the units that are connected to j and of the weights w_{ij} on these connections i.e.

$$H_{ij} = \sum x_i w_{ij} \quad (1)$$

Units can be given biases (θ_j) by introducing on extra input to each unit which always has a value of 1.



x_{i-1}
x_i
x_{i+1}
i
j
k
Activation of
$H_{oj} = \frac{1}{1+e^{-(H_{ij}+\theta_j)}}$
Net input
$H_{ij} = \sum_i W_{ij} x_i$
j
Output H_{oj}

Figure 8 : A Hidden Neuron and Its Function.

A hidden unit has a real-value output H_{oj}, as shown in Figure 8, which is a non-linear function of its total input.

$$H_{oj} = \frac{1}{1+e^{-(H_{ij}+\theta_j)}} \quad (2)$$

The use of a linear function for combing the inputs to a unit before applying the non-linearity greatly simplifies the learning procedure.

The aim is to find a set of weights that ensure that for each input vector, the output vector produced by the network is the same as (or sufficiently close to) the desired output vector. If there is a fixed, finite set of input-output cases, the total error in the performance of the network with a particular set of weights can be computed by comparing the actual and desired output vectors for every case. The total error E, is defined as:

$$E = \frac{1}{2}\sum_{C}\sum_{i}(O_{j,c} - T_{i,c})^2 \qquad (3)$$

where 'c' is the an index over cases (input-output pairs), j is an index over output units, ‘O’ is the actual state of an output neuron in the output layer and T is its targeted state. To minimise E by gradient descent, it is necessary to compute the partial derivative of E with respect to each weight in the network i.e. $\partial E/\partial W_{ji}$. This can computed successively as follows:

Firstly differentiate Eqn (3) for a particular case, c,

$$\frac{\partial E}{\partial O_j} = (O_j - T_i) \qquad (4)$$

Next $\partial E/\partial x_j$ is computed using chain rule i.e.

$$\frac{\partial E}{\partial xj} = \frac{\partial E}{\partial Oj} \; \frac{\partial Oj}{\partial xj} \qquad (5)$$

Differentiating Eqn (2) to get the value of $\partial Oj/\partial xj$ and substituting in (5)

$$\frac{\partial Ej}{\partial xj} = \frac{\partial E}{\partial Oj} \; O_i\,(1 - O_j) \qquad (6)$$

Eqn (5) calculates how the change in the total input ‘x’ to an output unit, will affect the error E. The total input is just a linear function, of the states of the lower level units and it is also a linear function of the weights on the connections, it is, therefore, easy to compute how the error will be affected by changing these states and weights. For a weight w_{ij}, from i to j the derivative is

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial xj}\frac{\partial x_j}{\partial w_{ji}} = \frac{\partial E}{\partial xj} \, . \; O_i \qquad (7)$$

and for the output of the ith unit the contribution to $\partial E/\partial O_i$ resulting from the effect of i on j is simply.

$$\frac{\partial E}{\partial xj}\,\frac{\partial xj}{\partial Oj} = \frac{\partial E}{\partial xj}.\ w_{ji} \qquad (8)$$

So taking into account all the connections emanating from unit i we have

$$\frac{\partial E}{\partial Oi} = \sum_{j} \frac{\partial E}{\partial xj}.\ w_{ij} \qquad (9)$$

Given $\partial E/\partial O$ for all units j, in the previous layer, the $\partial E/\partial O_i$ in the penultimate layer can be computed using Eqn (9). This procedure can, therefore, be repeated for successive layers.

The simplest version of gradient descent is to change each weight by an amount proportional to the accumulated $\partial E/\partial w$. This proportional constant is called learning rate.

$$\Delta w = -\in \partial E/\partial w \qquad (10)$$

The convergence of Eqn (10) can be significantly improved, by an acceleration method wherein the incremental weights at t can related to the previous incremental weights given in Eqn (11).

$$\Delta w(t) = \in \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \qquad (11)$$

where α is an exponential decay factor between '0' and '1' that determines the relative contribution of the current gradient and earlier gradients to the weight change.

5.3 Network Training and Identification of ANN Model

A major concern in the development of a neural network is determining an appropriate set of weights that make it perform the desired function. There are many ways that this can be done; the most popular class of these algorithms are based on supervised training. Typically, supervised training starts with a network comprising an arbitrary number of hidden neurons, a fixed topology of connections, and randomly selected values for weights. The network is then presented with a set of training patterns, each comprising an example of the problem to be solved (the inputs) and its corresponding solution (the targeted output). Each problem is input into the network in turn, and the resultant output is compared to the targeted solution providing a measure of total error in the network for the set of training patterns. The weights are then adjusted by small amounts as prescribed by generalised delta rule (or any other rule) so that on the next occasion the example problem presented to the network, the error is reduced, and the output is closer to the required. The process is repeated many times until the network is able to reproduce, to within a specified tolerance, the corresponding solutions to each of the example problems. In case of the generalised delta rule, the first derivative of the total error with respect to a weight determines the extent to which that weight is adjusted. Thus, the more a weight appears, in this sense, to affect the total error, the more that weight is changed. The direction of change is that which decreases the total error.

Determining an appropriate configuration of hidden neurons for a given problem is usually troublesome. Hecht-Nielsen (1989) provides a proof that one hidden layer of neurons (operating sigmoidal activation functions) is sufficient to model any soluiton surface of practical interest. Despite Hecht-Nielsen (1989) proof, there are many solution surfaces that are extremely difficult to model using a sigmoidal network comprising one hidden layer. Alternative configuration, with several hidden layers, should also be considered, since they often lead to more effective solutions. Generally, there is no direct and precise way of determining the most appropriate number of neurons to include in each hidden layer. This problem becomes more complicated as the number of hidden layers in the network increases. In an attempt to resolve this problem, a range of different configuration of hidden neurons is normally considered, and that with the best performance is accepted. Another factor that can significantly influence a network's ability to learn and generalise is the number of patterns in the training set. Increasing the number of training patterns provides more information about the shape of the solution surfaces, and thus increases the potential level of accuracy that can be achieved by the network.

In the study the data is divided into two parts for supervised training. One part is used for training and the other part is used for testing the performance of the ANN. The training of the neural network is accomplished by adjusting the inter-connecting weights till such time that the root mean square error (r.m.s.e.) between the observed and the predicted set of values is minimised. The adjusting of inter-connecting weights is accomplished by using the back propagation algorithm as given in Section 5.2.

In the process of model development several network architectures with different number of input neuron in input layer, and different number of hidden layer with varying number of hidden neurons are considered to select the optimal architecture of the network. A trial and error procedure based on the minimum r.m.s.e. criterion is used to select the best network architecture. A case study has been done to demonstrate the applicability of ANN for Ajay river basin to forecast 6-hr ahead runoff at Sarath gauging site of Jharkhand (South Bihar).

6.0 DEVELOPMENT OF ANN MODEL FOR AJAY RIVER BASIN

Six isolated storm events are chosen for the study. The 1-hourly rainfall runoff data are available for flood seasons. To identifying the structure of an ANN model three isolated storms were chosen. The ANN structure basically has three components: (i) the number of neurons in the input layers L_I (ii) the number of neuron in the hidden layers L_H the number of hidden layers and their inter-connection with the input layers and finally (iii) the number of output neurons in output layer L_o .

In modelling the rainfall – runoff process of Ajay river basin for Sarath gauging site, the input layer L_I consist of input neurons, which represents runoff y_{k-I} and rainfall w_{k-I} at various lag periods. Similarly as the the model is to forecast 6 hr ahead runoff at Sarath gauging site, the output neuron is taken as only one and its output is the forecasted value of the runoff, which is represented by y_k. The number of input neurons (which may consist of runoffs (y_{k-I}) and rainfalls (w_{k-I})) were assessed based on the lead time to forecast and the time of concentration of the catchment. Since the objective was to give 6 hours ahead forecast, the earliest input values to forecast y_k can be runoff y_{k-6} and rainfall w_{k-6}. The concentration time was observed to lie between 21 to 22 hours. Thus the rainfall earlier to w_{k-22} had practically no effect in estimating the runoff y_k. Thus the input neurons could be (y_{k-6}, y_{k-7}, w_{k-6}, w_{k-7}, w_{k-8},.......... w_{k-21}, w_{k-22}) Various combinations of y_k' and w_k's were tried without any hidden layer and that combination which gave the least root mean square error (r.m.s.e.) was identified. It was thus observed that the combination (y_{k-6}, y_{k-7}, w_{k-6}, w_{k-7},........ w_{k-21}, w_{k-22}) gave the best result. The input layer thus has eighteen neurons and the output layer has one neuron. Now, different combinations of hidden layers were introduced and various trials were attempted in order to reduce the r.m.s.e. It was observed that one hidden layer with nine neurons gives the least r.m.s.e. Thus the ANN model, identified is as follows:

Number of input neurons in input layer	=	18
Number of hidden layers	=	1
Number of hidden neurons	=	9
Number of neurons in output layer	=	1.

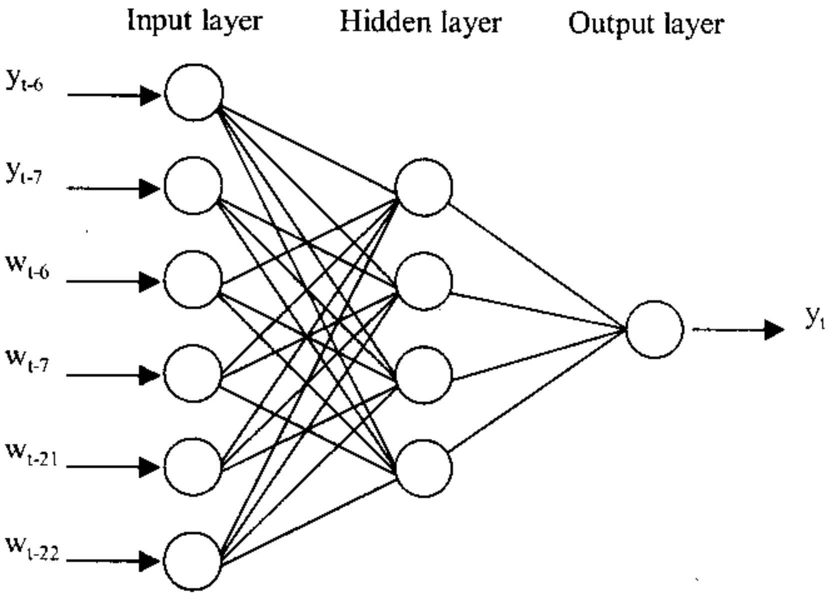
Graphically the ANN model can be represented as shown in Figure 9.

The three flood hydrographs FH1, FH2 and FH3 given in Figures 10, 11, 12 were used for training the ANN. The architecture was kept the same for all the three events and the weights were optimised using back propagation technique. Total number of iterative cycles were kept at 1000 with learning at 0.02. The observed and forecast values of these hydrographs are also shown in Figures 10, 11, 12. The Table 7 shows the weights between the input layer and hidden layer. The magnitude of weights were kept between –2.5 and 2.5. The biases for the hidden neurons in hidden layer are shown in Table 8. Table 9 gives the weights between hidden layer and output layer. The bias value for output neuron is 0.2204. The performance of the identified ANN with optimised weights for forecasting the 6 hours ahead forecast for the new rainfall – runoff events are given in Figures 13, 14 and 15. The observed or forecast values using the ANN are shown in these figures.

The performance of ANN model developed is evaluated based on r.m.s.e. which is defined as

$$\text{r.m.s.e} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(\text{obs}_j - \text{fitted } y_j)^2}$$

The root mean square error (r.m.s.e) of developed ANN model for three flood events (i.e FH1, FH2, FH3) in trainning (fitting) is 38.72. The r.m.s.e in forecasting the other three flood events is 194.9. Low r.m.s.e in the training session signifies the fact that ANN are capable of fitting nonlinear flood events efficiently. As three events for training are not sufficient for proper representation of rainfall – runoff process in the catchment, more events were included in the training process to determine better weight optimization between different layers of ANN model for more appropriate representation of rainfall – runoff process in the catchment. A weight optimization strategy is adopted for this



(18 input neurons) (9 hidden neurons)

Figure 9: ANN Model.

Table 7: Weight matrix between input layer (LI) and hidden layer (Lh)

-.7880	-0.9391	-0.0229	-0.3933	-0.2490	-0.3813	-0.4866	-0.4743	-0.4440	-0.8031	-0.0665	0.3141	0.2394	-0.0134	0.0515	0.2936	0.3388	0.4302
-1.0233	-0.6553	-0.1230	-0.1628	-0.3882	-0.6827	-0.6200	-0.3713	-0.5112	-0.7404	-0.0507	0.0676	0.0380	0.4511	0.4878	0.1836	0.0068	0.4231
-1.2291	-1.1251	-0.1062	-0.3644	-0.5345	-0.4274	0.0204	0.1569	0.1357	0.2803	-0.0422	-0.3479	-0.0683	0.0240	-0.2248	-0.4050	-0.1930	-0.2348
-2.4754	-1.4312	0.3573	0.0995	0.0902	0.2327	-0.0972	1.2850	0.9396	0.1555	-1.6163	-1.1478	-0.7125	-0.9140	-0.6931	-0.0698	-0.2411	-0.6300
-1.1760	-0.6617	-0.6801	-0.3174	-0.5527	-0.8358	-1.5763	-1.0190	-0.5236	-0.7854	0.0360	0.1349	0.3629	0.5337	0.6408	0.6308	0.3060	0.4866
-0.9867	-0.8906	-0.3084	-0.3261	-0.3181	-0.2419	-0.6462	-0.6837	-0.3671	-0.4755	0.2201	0.2413	0.4145	0.2186	0.1420	0.2566	0.1117	0.3767
-0.9254	-1.1044	-0.3788	-0.1587	-0.1056	0.0143	-0.2654	-0.1465	-0.3772	-0.0767	0.1214	0.0229	0.1673	-0.0247	0.2389	0.3077	0.0562	-0.0549
0.3039	-1.1176	0.1762	0.6969	1.3368	0.9708	1.0512	0.6242	0.9905	1.4422	1.2616	1.5575	0.9180	0.7273	0.8287	0.3477	0.3604	0.2415
-0.8770	-0.4301	-0.2907	-0.0877	-0.5068	-0.1141	-0.3920	-0.4569	-0.4739	-0.3875	-0.0131	-0.2327	-0.0975	0.2151	0.2544	0.0217	-0.2218	-0.1287

Table 8: Biases between input layer (LI) and hidden layer (Lh)

-0.1441	-0.3368	-0.4136	-0.6159	0.2833	-0.0069	-0.3544	-0.7266	-0.5413

Table 9: Weight matrix between hidden layer (LI) and output layer (Lh)

-1.0011	-1.0969	0.5480	2.3264	-1.7526	-1.1764	-0.3830	1.4705	-0.6898

Bias of the output unit:
0.2204

purpose. Five rainfall – runoff events were selected in turn to findout the best possible weights between the neurons of different layers. The weights thus found were used to forecast the remaining rainfall-runoff event. The weight matrix which corresponds to least r.m.s.e in forecast are recommended for use in the catchment. To further check the stability of the recommended weights, all six events were used to find out new weights and comparison was done to observe any significant change in the weights.

The possible combinations of rainfall-runoff events for better weight optimization are shown in Table 10.

Table 10: Different combinations of flood events to determine optimal weights

	FH1	FH2	FH3	FH4	FH5	FH6
Case 1	✓	✓	✓	✓	✓	✗
Case 2	✓	✓	✓	✓	✗	✓
Case 3	✓	✓	✓	✗	✓	✓
Case 4	✓	✓	✗	✓	✓	✓
Case 5	✓	✗	✓	✓	✓	✓
Case 6	✗	✓	✓	✓	✓	✓
Case 7	✓	✓	✓	✓	✓	✓

The ✓ sign indicates that the corresponding flood event is considered for training the ANN model whereas ✗ sign indicates that the corresponding flood event is not considered for training but used for validating the developed ANN model. Six possible cases were used to determine the six sets of weights between different layers for the given ANN model. The best set of weights was chosen based on minimum r.m.s.e for validating event. In case 7 all the flood events were used to determine the weighs between different layers of ANN model and compared with the best chosen weights from earlier six cases to determine any significant change in the weight matrix.

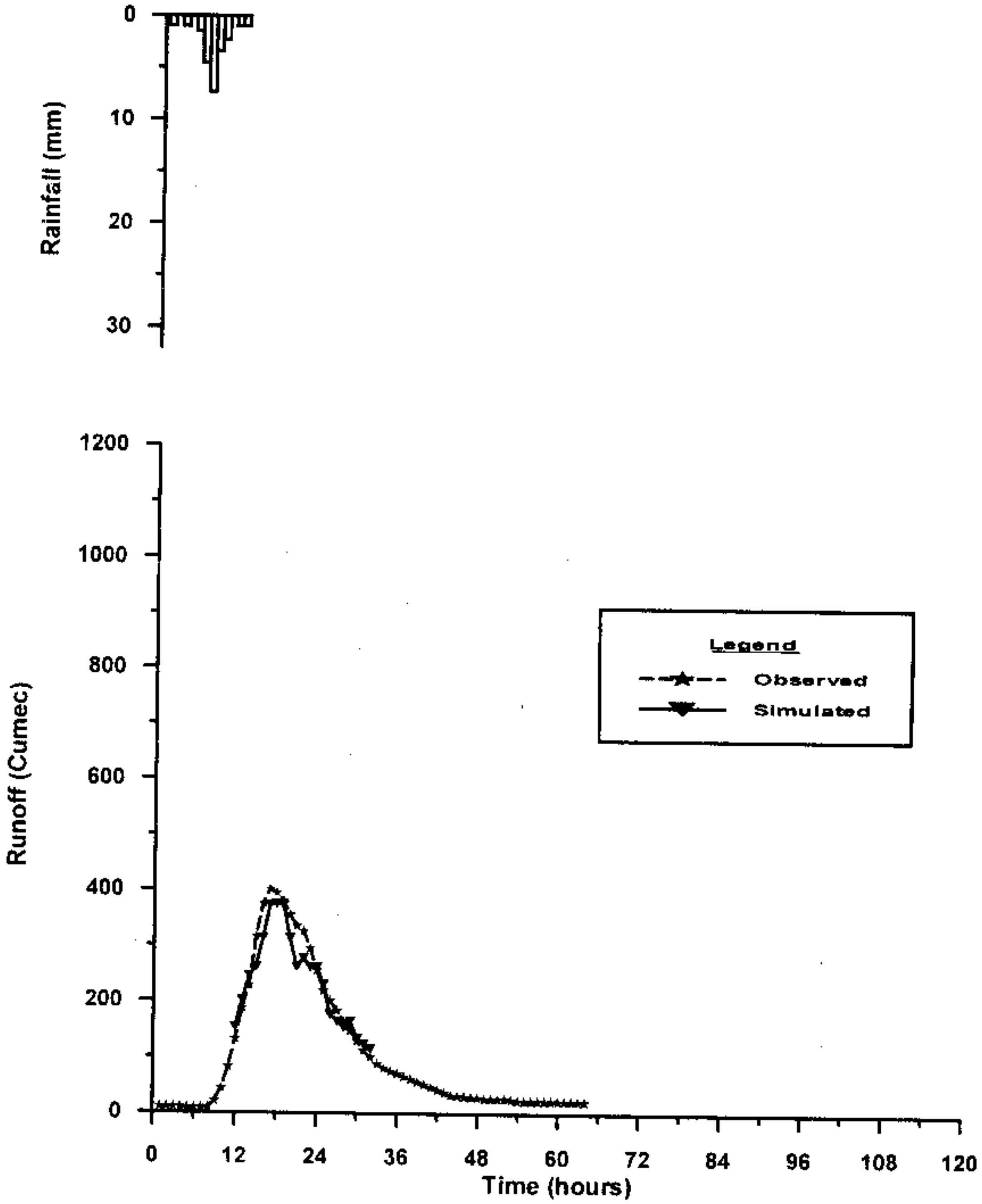


Figure 10: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -1 forTraining of the ANN model.

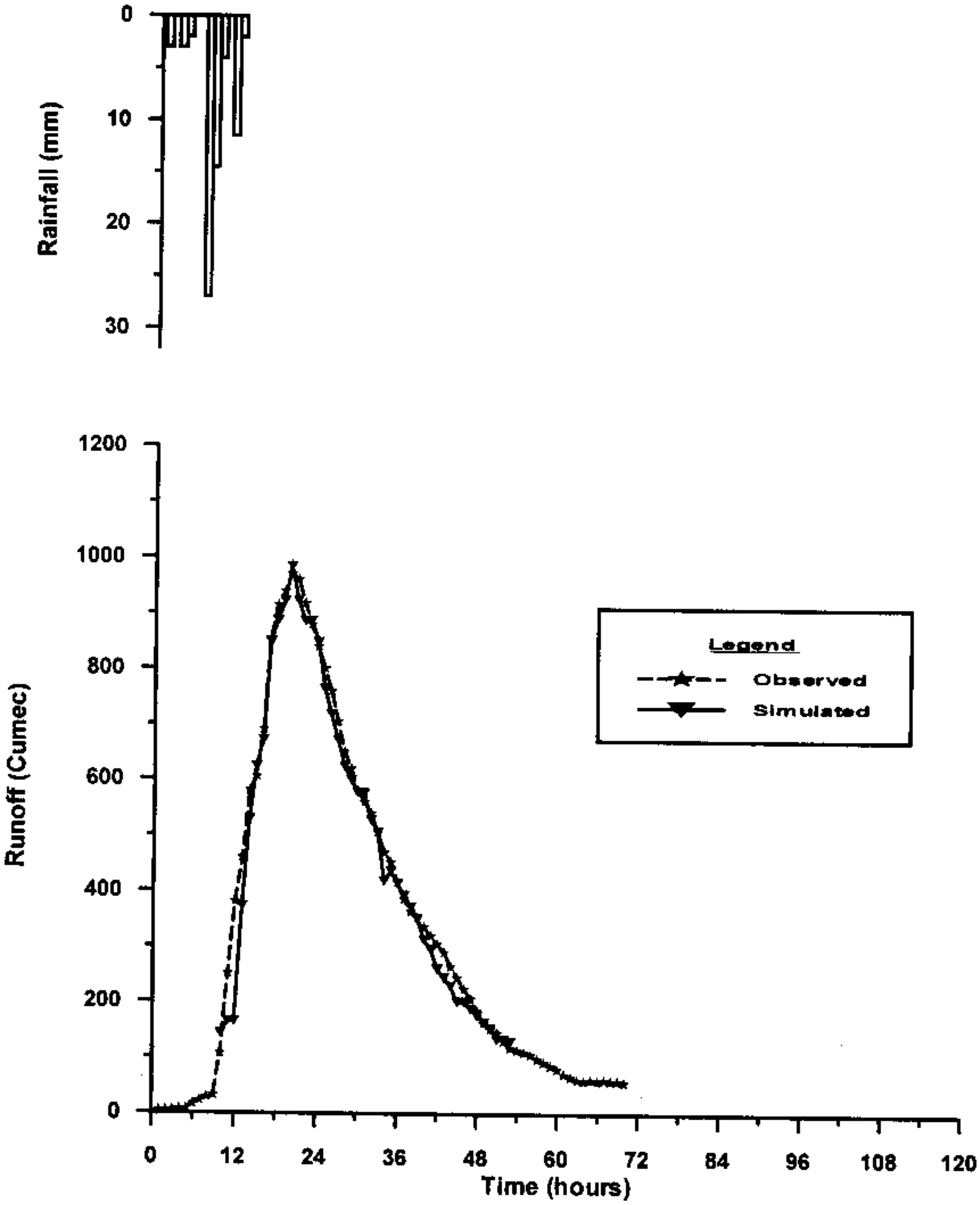


Figure 11: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -2 forTraining of the ANN model.

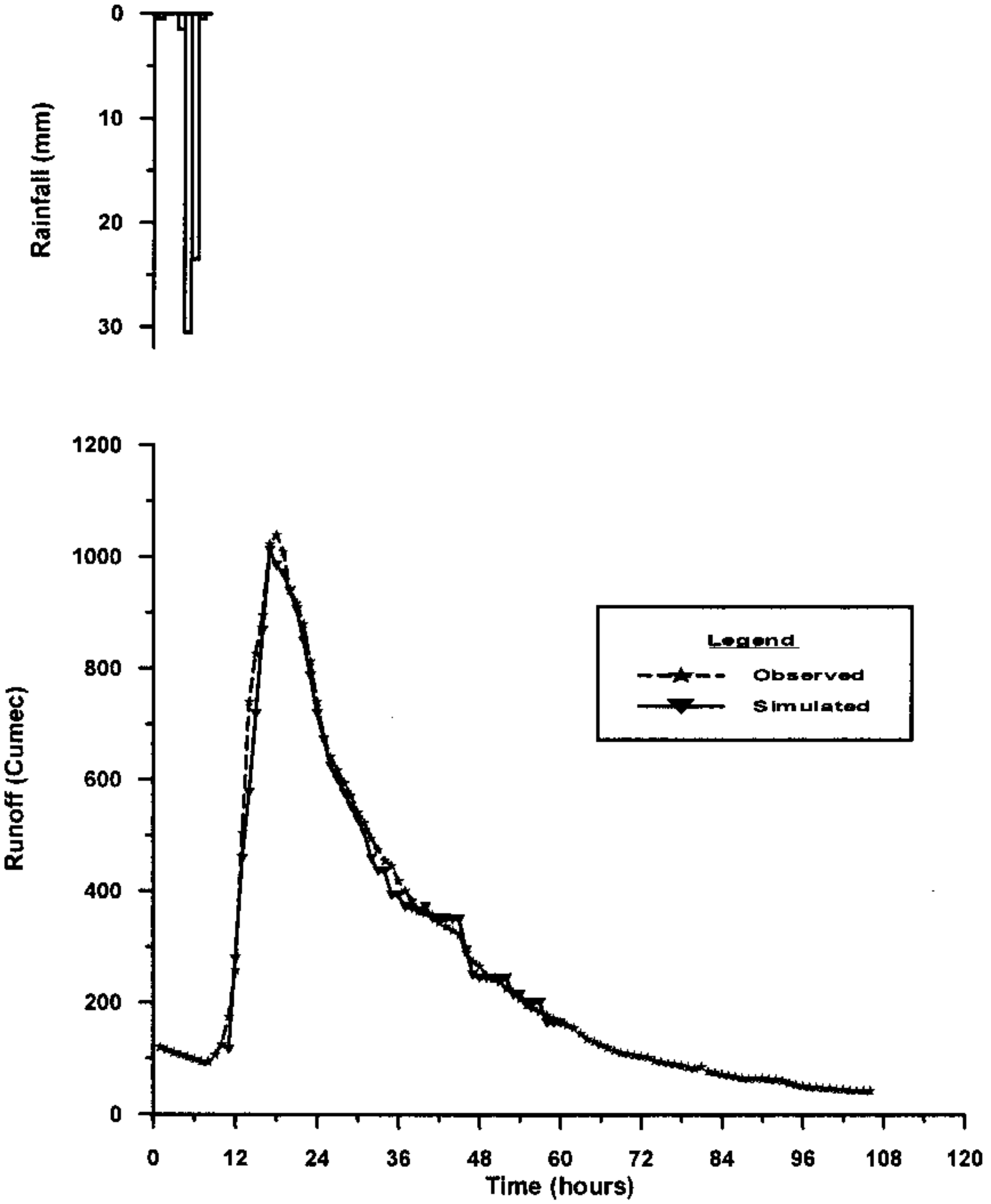


Figure 12: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -3 forTraining of the ANN model.

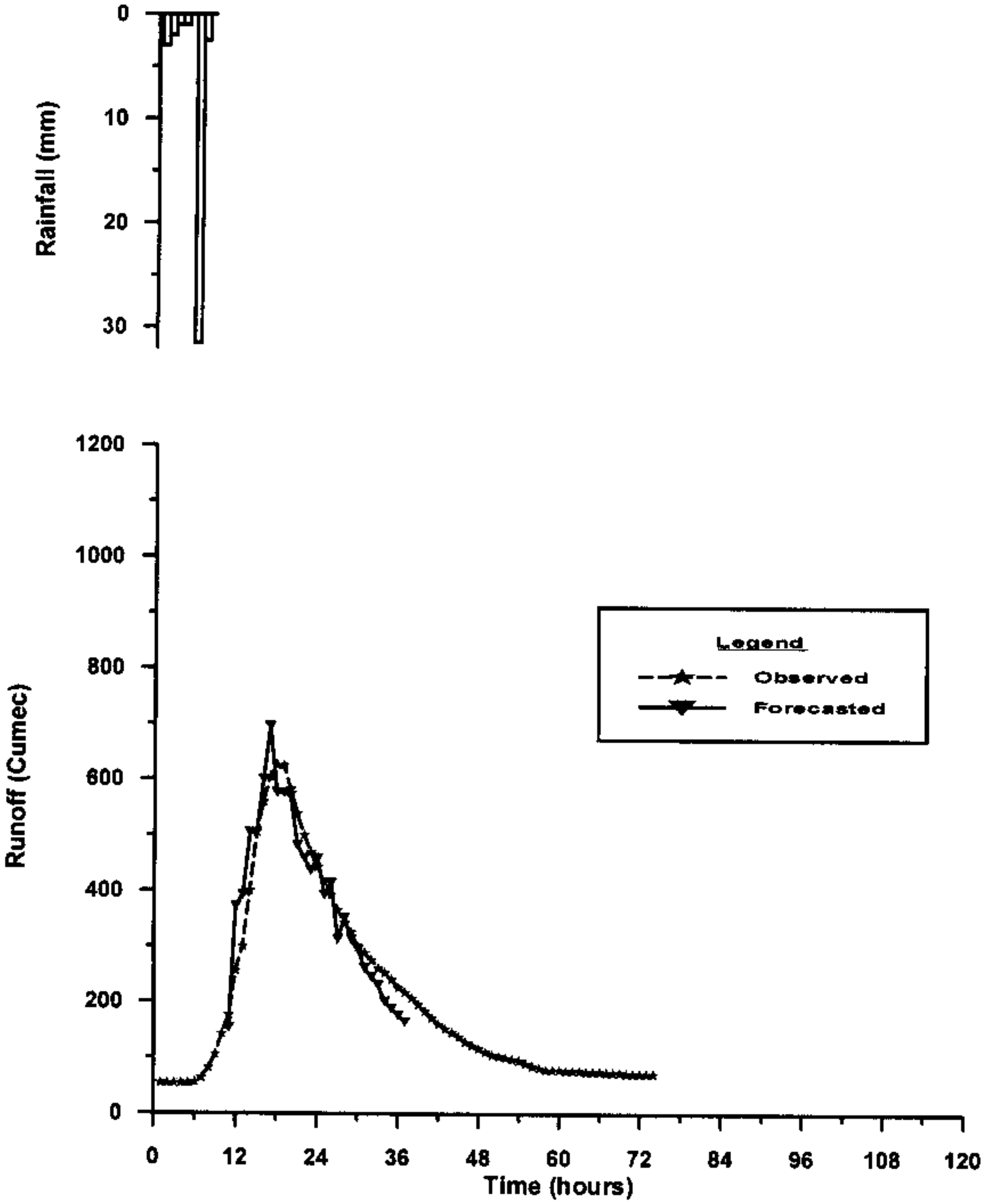


Figure 13: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -4 forTesting of the ANN model.

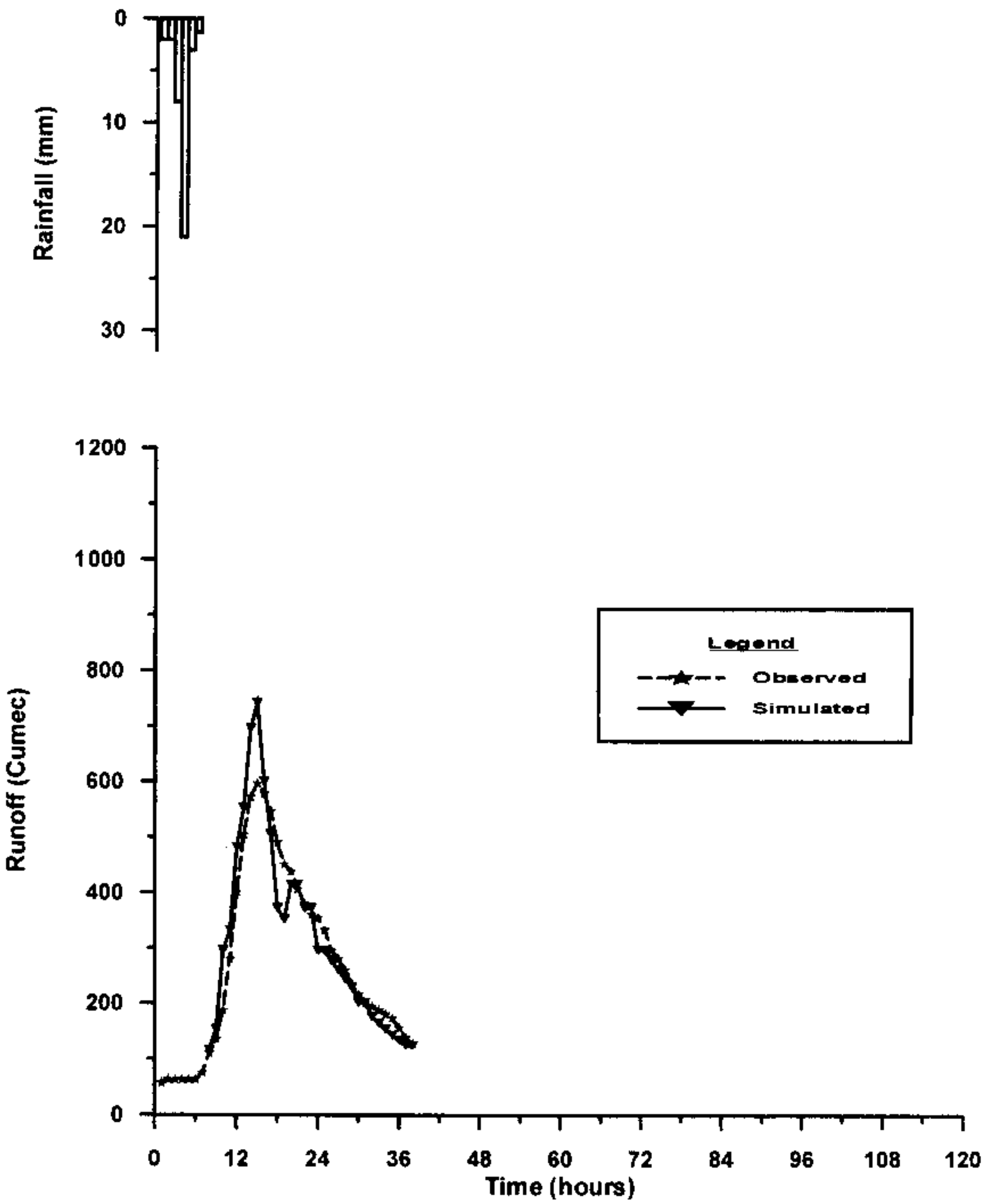


Figure 14: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -5 forTesting of the ANN model.

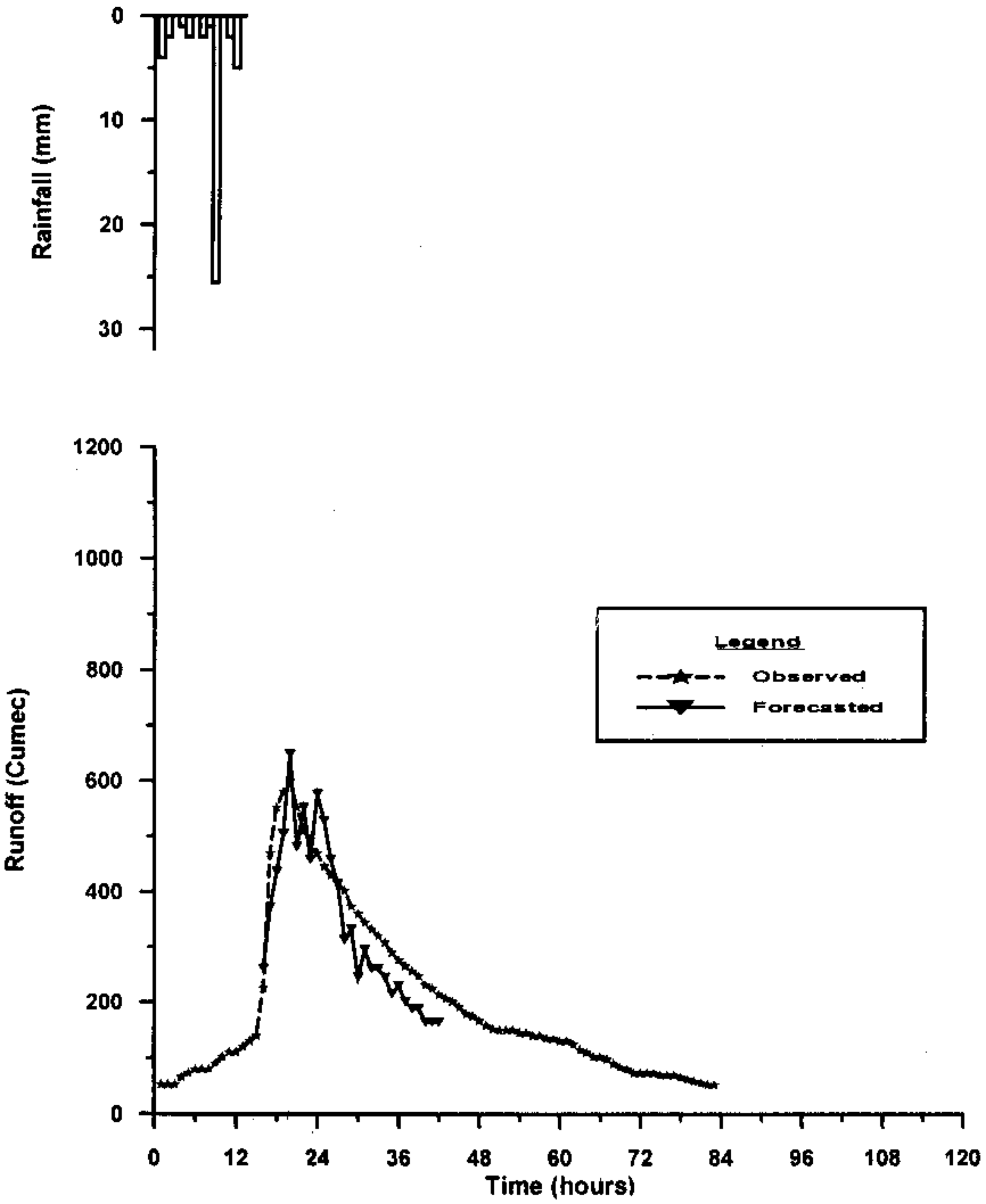


Figure 15: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph - 6 forTesting of the ANN model.

Case 1:

In this case the flood hydrograph FH1, FH2, FH3, FH4, FH5 were used to find out the weights between different layers. The initial random weights between different layers were updated using backpropagation algorithm. The updated weights were used to simulate the flood hydrograph FH6 and comparison was made with the observed flood hydrograph. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table 11 and Table 13. Table 11 represents the weight matrix between input layer and hidden layer whereas Table 13 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table 12 whereas for the output neuron its value is 0.2882. The r.m.s.e. in forecasting the event FH6 is 73.75. Figure 16 shows the forecasted and observed event FH6.

Case 2

As shown in Table 10 the flood hydrograph FH1, FH2, FH3, FH4, FH6 were used in this case to find out the weights between different layers. The initial random weights between different layers were updated using backpropagation algorithm. The updated weights were used to simulate the flood FH5 and comparison was made with the observed flood hydrograph. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table14 and Table 16. Table 14 represents the weight matrix between input layer and hidden layer whereas Table 16 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table 15 whereas for the output neuron its value is 0.1139. The r.m.s.e. in forecasting the event FH5 is 55.40. The Figure 17 shows the forecasted and observed event FH5.

Case1

Table11 Weight matrix between input layer (LI) and hidden layer (Lh) for case 1

-1.1952	-1.0911	-0.2430	-0.3086	-0.6424	-0.1549	-0.5729	-0.0003	-0.6038	-0.1514	-0.2439	0.0897	0.3441	-0.0421	0.3939	0.2261	0.3065	0.3395
-1.4252	-1.2688	-0.2660	-0.5174	-0.3198	-0.1338	-0.5154	0.0258	-0.1552	0.0690	-0.1561	0.0054	-0.1649	-0.0708	0.0538	0.1082	0.1197	-0.3041
-1.2500	-0.5970	-0.4984	-0.2889	-0.9213	-0.3231	-0.3260	-0.3745	-0.4416	-0.4784	-0.2721	0.5579	0.4348	0.3098	0.6158	0.2175	0.3852	0.3432
0.0479	-0.7737	0.2004	1.2439	1.0090	1.0312	0.8885	0.9766	1.0295	1.0643	0.9659	1.4378	0.9696	0.8299	0.8946	0.5787	0.4715	-0.0369
-1.0827	-0.6162	-0.3407	-0.4678	-0.8258	-0.5941	-0.5136	-0.3877	-0.5477	-0.8317	-0.3766	0.3405	0.2380	0.5202	0.5974	0.8229	0.3456	0.0234
-0.8853	-0.5212	-0.4784	-0.6005	-0.7552	-0.8361	-0.4759	-0.6625	-0.5325	-1.0712	-0.6875	0.0877	0.0604	0.3146	0.6298	0.3588	0.4457	-0.0533
-1.5744	-1.0585	-0.2722	-0.2856	-0.3609	-0.3204	-0.1904	-0.2393	-0.1801	-0.1293	-0.0441	-0.1151	0.3364	-0.1937	0.0220	0.2061	0.2909	0.1550
-1.1461	-1.3535	-0.2245	0.0948	-0.3994	-0.3823	-0.2154	0.0259	0.3020	0.3318	-0.3542	-0.6100	-0.0083	-0.2427	-0.4183	-0.2678	-0.1923	-0.0266
-2.4885	-2.4979	0.0254	-1.0279	-1.0847	0.2106	0.9385	1.5304	1.2420	-0.6760	-1.4615	-0.9398	-0.4110	-0.0359	0.1026	-0.1402	-0.1644	-0.3950

Table12 Biases for the neurons of hidden layer for case 1

-0.7076	-0.9223	-0.1978	-2.2460	0.1652	0.5575	-0.8806	-0.8329	-1.7202

Table13 Weights between hidden layer (LH) and output layer (Lo) for case1

-0.6540	-0.0700	-1.2351	1.7812	-1.7001	-1.4437	-0.2909	0.6170	2.0714

Case 2

Table14 Weight matrix between input layer (LI) and hidden layer (Lh) for case 2

-2.5000	-2.5000	0.6390	0.0236	-0.6206	0.1611	0.4204	1.1216	0.7344	-0.9344	-2.5000	-1.7105	0.4245	0.7621	0.3074	0.1046	-0.0771	-0.1664
-0.9436	-1.0452	-0.3363	-0.2657	-0.3582	-0.0420	0.0420	0.5072	1.3195	0.6183	-0.6869	-1.0947	-0.6244	0.0740	-0.0209	-0.2363	-0.1825	-0.4517
-0.5642	-0.6913	0.3109	-1.2504	-0.3352	-0.6634	-0.5034	-0.4325	-0.6905	-0.6512	-0.4457	-0.1851	0.1251	0.0304	0.4412	0.0855	0.0766	-0.0278
-1.3678	-2.1805	0.1410	-0.4436	-0.4898	-0.7476	-0.8527	-0.1985	-0.4998	-0.6358	0.2656	0.3830	0.5369	0.3454	0.5850	0.0867	0.4455	0.1108
-0.7927	-0.5186	-0.4332	-0.5284	-0.6623	-0.5138	-0.5932	-0.9189	-0.3143	-0.6929	-0.4931	0.1939	0.0229	0.6313	0.0175	0.0655	0.2600	0.0899
-1.0015	-0.9735	-0.9235	0.0780	-1.0026	-0.9996	-0.9856	-0.6536	-0.1991	-0.6941	-0.4142	0.4952	0.5068	0.9916	0.7486	0.5485	0.2433	0.2146
-1.0784	-0.3671	-0.7624	-0.0469	-1.0332	-0.5688	-0.5040	-0.7361	-0.2577	-0.5233	-0.5027	0.0297	0.4151	0.2932	0.3817	0.4018	0.2293	0.1365
-1.5342	-1.3431	-0.1311	-0.2861	-0.3358	-0.1428	0.0724	-0.2180	-0.1314	-0.1784	0.1396	-0 4231	-0.2906	-0.1559	-0.2917	-0.2733	-0.2881	0.0786
0.3426	-0.8222	-0.0826	0.7041	1.6255	0.7271	1.2193	0.7961	0.9149	1.2476	1.4271	1.5713	1.1058	0.8639	0.7420	0.4374	0.3424	0.1437

Table15 Biases for the neurons of hidden layer for case 2

-2.0266	-0.5906	-0.2972	0.1154	-0.4424	-0.2809	-0.5126	-0.5544	-1.7300

Table16 Weights between hidden layer (LH) and output layer (Lo) for case2

2.4156	1.2416	-1.4582	-1.3801	-0.7491	-1.3706	-1.1493	-0.2220	1.3078

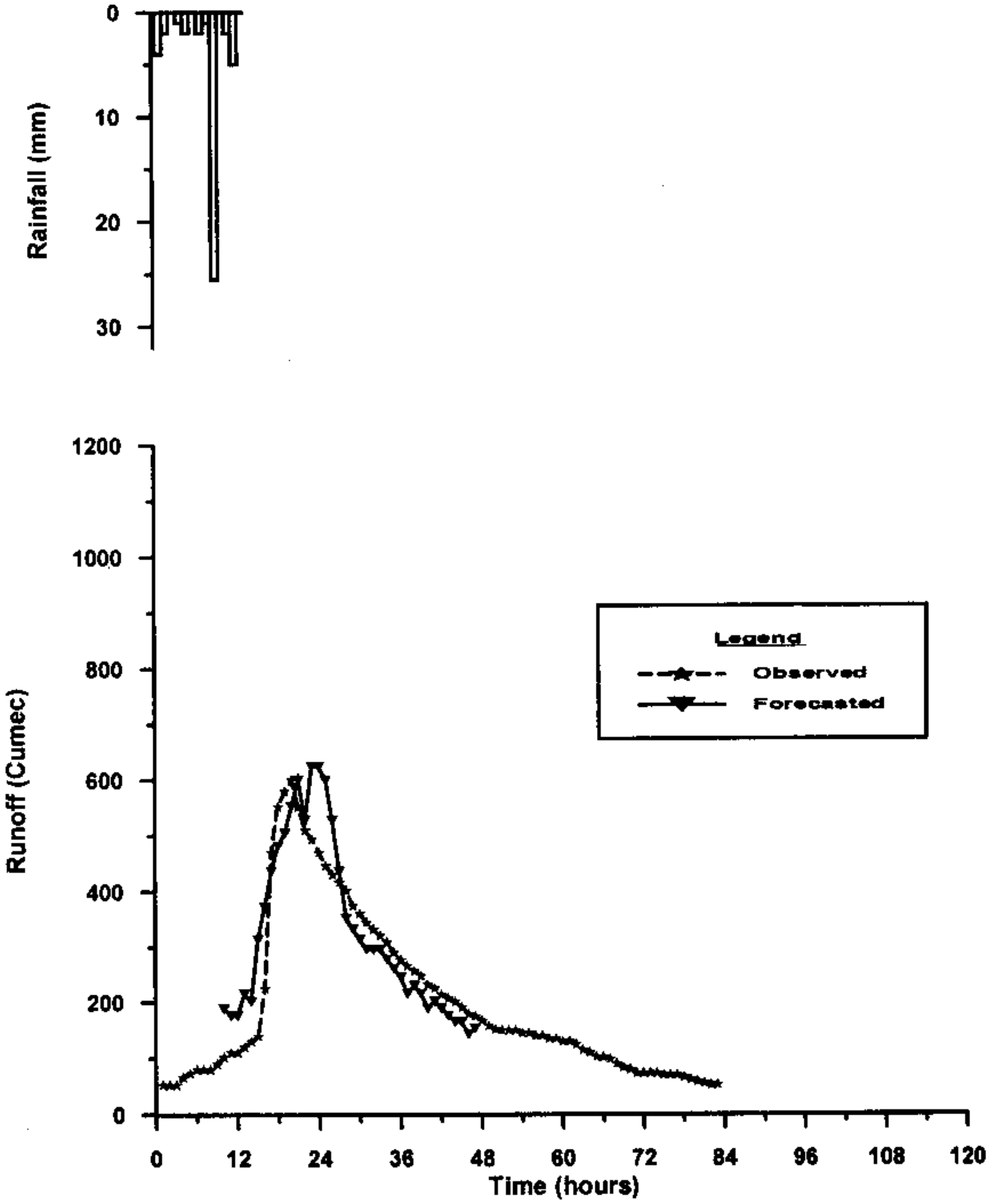


Figure 16: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph - 6 forTesting of the ANN model.

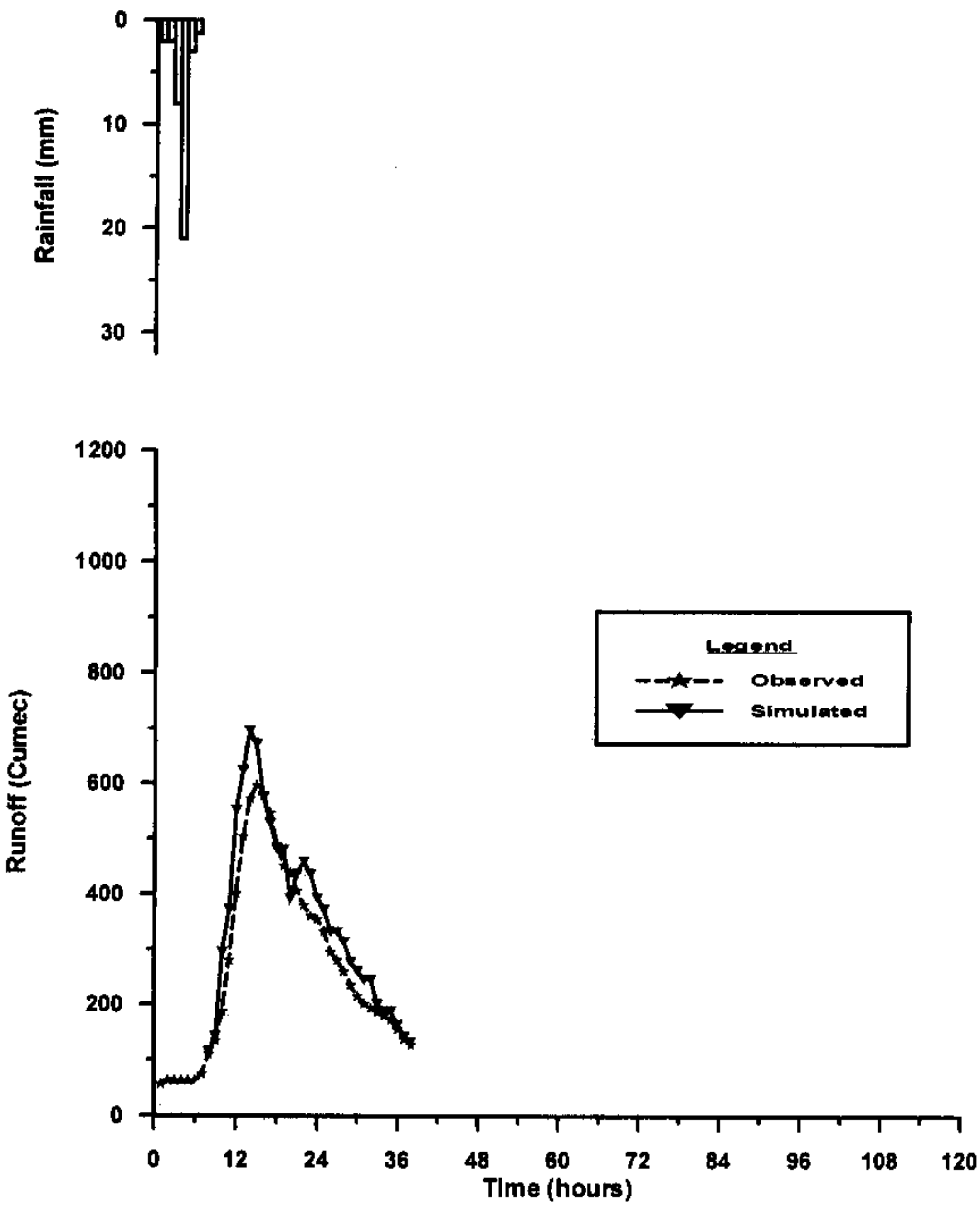


Figure 17: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -5 forTesting of the ANN model.

Case 3

In this case the flood hydrograph FH1, FH2, FH3, FH5, FH6 were used to find out the weights between different layers. The initial random weights between different layers were updated using backpropagation algorithm. The updated weights were used to simulate the flood FH4 and comparison was made with the observed flood hydrograph. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table 17 and Table 19. Table 17 represents the weight matrix between input layer and hidden layer whereas Table 19 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table 18, whereas for the output neuron its value is 0.5075.The r.m.s.e. in forecasting the event FH4 is 64.81. The Figure 18 shows the forecasted and observed event FH6.

Case 4

As shown in Table 10 the flood hydrograph FH1, FH2, FH4, FH5, FH6 were used in this case to find out the weights between different layers. The initial random weights between different layers were updated using backpropagation algorithm. The updated weights were used to simulate the flood FH3 and comparison was made with the observed flood hydrograph. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table 20 and Table 22. Table 20 represents the weight matrix between input layer and hidden layer whereas Table 22 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table21, whereas for the output neuron its value is 0.1600. The r.m.s.e. in forecasting the event FH3 is 78.99. The Figure 19 shows the forecasted and observed event FH3.

Case 3

Table17 Weight matrix between input layer (LI) and hidden layer (Lh) for case 3

-0.6805	-0.6978	0.0889	-0.5405	-0.5036	-0.5977	-0.4750	-0.4043	-0.6155	-0.3163	-0.0636	0.0022	0.4924	0.4217	0.2095	-0.0989	0.2348	0.0644
-1.0302	-1.6573	-0.6875	-0.0525	-0.8226	-0.7530	-0.8711	-0.4153	-0.6622	-0.8274	0.2244	0.7079	-0.2217	0.1112	0.8556	0.3348	0.5268	0.1468
-0.9356	-0.9264	-0.0184	-0.5713	-0.4741	-0.5562	-0.5268	-0.4384	-0.4023	-0.1791	0.0769	0.1552	0.3511	0.4401	0.2363	0.3923	0.3944	0.2517
-1.7881	-1.7717	-0.1244	-0.3775	-0.1053	-0.0353	-0.2964	-0.2656	0.1059	-0.2477	-0.0238	0.2141	-0.3694	-0.4587	-0.2124	0.0091	-0.2361	0.1974
-1.1215	-1.1628	-0.1741	0.2788	0.5169	0.1533	0.5565	0.7244	1.5362	1.2691	-0.3711	-0.8832	-0.2807	0.1407	-0.0486	-0.6295	-0.1179	-0.4416
-2.4865	-2.4894	1.0061	0.0699	-1.2664	-0.2417	-0.1644	0.6513	0.1553	-0.8966	-2.0015	-0.8686	0.7362	0.4912	0.0512	0.0712	-0.1159	-0.1855
0.5548	-0.9684	-0.0225	0.2377	1.1792	0.6989	0.8151	0.8526	0.3682	0.9880	1.7585	1.7740	0.8850	0.8254	0.5418	0.2851	0.2515	-0.2204
-0.5116	-1.0456	0.0867	-0.4540	-0.5914	-0.6165	-0.6047	-0.6513	-0.2907	-0.1729	-0.2505	0.6694	0.3244	0.2537	0.6056	0.1097	0.1775	-0.0064
-0.9162	-0.0621	-0.3329	-0.3401	-1.1135	-1.2874	-1.2560	-0.9890	-0.5832	-0.6467	-0.6623	0.3143	-0.1973	0.3790	0.1816	0.0785	-0.1214	-0.5237

Table18 Biases for the neurons of hidden layer for case 3

30.7742	0.1306	-0.3183	-0.4481	-0.8360	-1.9379	-1.7049	-0.5752	0.0646

Table19 Weights between hidden layer (LH) and output layer (Lo) for case3

-1.0518	-1.5134	-1.1510	-0.7492	1.6925	2.1911	1.2393	-0.8228	-1.1948

Case 4

Table20 Weight matrix between input layer (LI) and hidden layer (Lh) for case 4

-1.8197	-1.4138	-0.2145	0.1255	0.0686	0.0778	-0.1166	0.0766	-0.3048	0.1669	0.1187	0.1697	-0.1159	-0.2696	-0.0821	0.0979	-0.1168	0.0195
-0.7128	-0.8140	-0.0919	0.0435	0.5652	0.6214	0.6265	0.7008	0.7002	0.6875	0.8641	0.7301	0.3051	0.1393	0.1488	-0.0474	-0.3235	-0.1964
-2.4993	-2.5000	0.8190	0.7052	-0.0976	-0.1808	-0.0598	0.0778	0.3549	0.2003	-1.1363	-0.1887	0.8420	0.5648	0.3983	0.1158	-0.0321	-0.3939
-0.4696	-0.3364	0.2979	-0.3556	-0.7645	-1.4491	-1.0516	-1.2150	-0.9812	-0.8019	-0.5345	-0.0641	0.0274	0.1706	0.2677	0.1675	0.1755	-0.1910
-1.2466	-1.2507	0.0487	-0.4377	-0.3844	-0.2253	-0.3596	-0.5403	-0.5733	-0.3227	-0.2720	0.0125	-0.0664	-0.2614	0.2654	-0.1106	0.3254	0.0984
0.3301	-0.4774	-0.0477	0.0605	0.3411	0.5060	0.9651	0.9866	0.8415	1.1533	1.0473	0.7365	0.4557	0.5775	0.6665	0.5078	0.2276	-0.1599
-0.3572	-0.7312	-0.5128	0.2823	-0.7797	-0.9667	-1.3080	-1.1856	-0.8587	-0.8908	-0.4265	-0.0674	-0.1812	0.2493	0.3129	0.6009	0.0198	0.0676
-1.3649	-1.3490	-0.2266	-0.1342	-0.3152	-0.2290	-0.4136	-0.4173	-0.3371	-0.4004	0.2085	0.0239	0.1469	-0.2422	0.1331	-0.0341	-0.0420	0.1278
-0.9206	-0.8028	0.0659	-0.2433	-0.8445	-0.9617	-0.6900	-0.6814	-0.7750	-0.6236	-0.1924	0.1112	-0.0798	0.3693	0.6862	0.1660	0.3413	0.0862

Table21 Biases for the neurons of hidden layer for case 4

-0.4611	-0.8366	-2.3864	-0.2599	-0.7145	-1.1533	-0.3489	-0.6605	-0.5228

Table22 Weights between hidden layer (LH) and output layer (Lo) for case4

-0.5503	0.6360	2.4806	-1.4262	-0.7520	1.1777	-1.2600	-0.5664	-0.9928

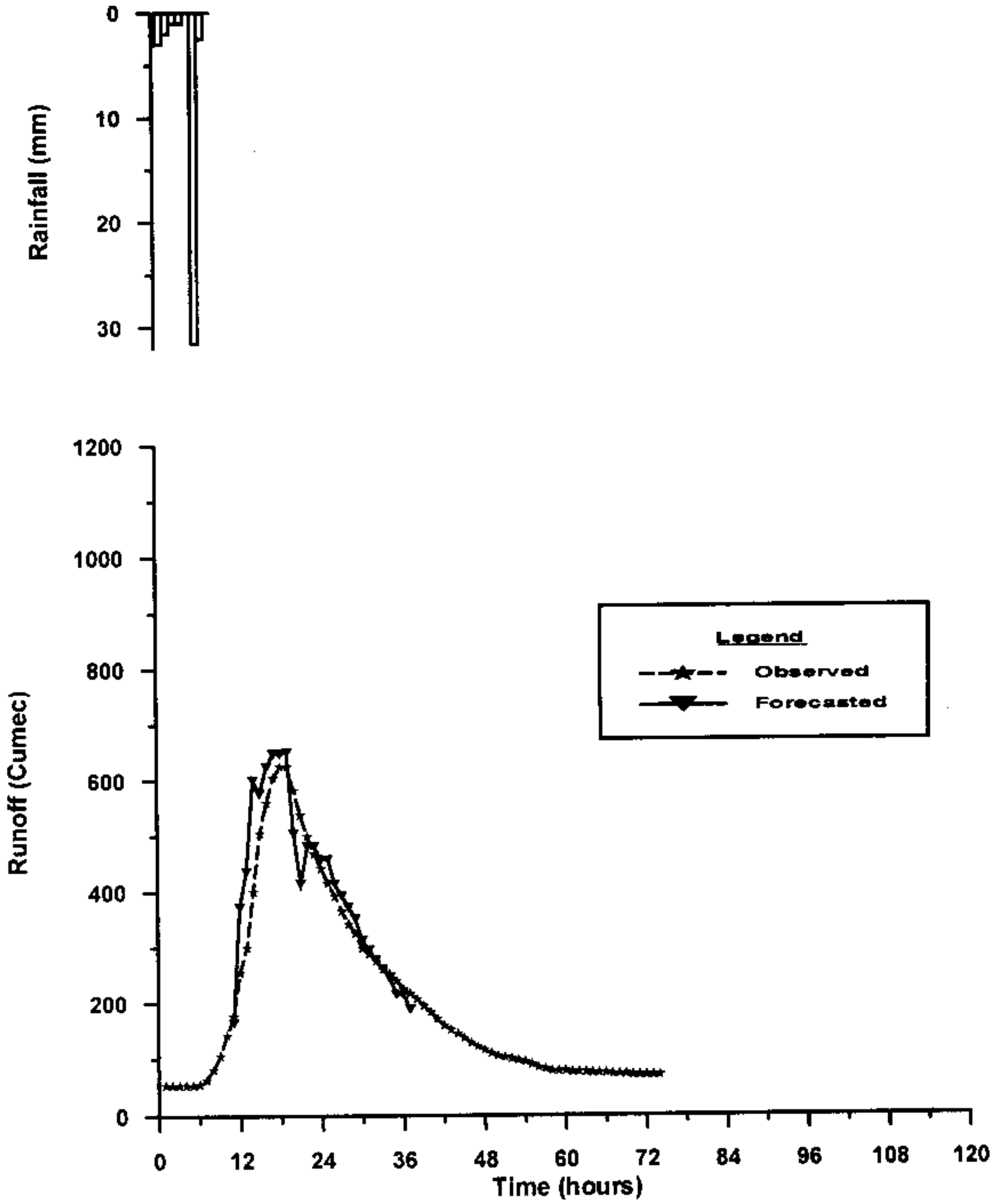


Figure 18: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -4 forTesting of the ANN model.

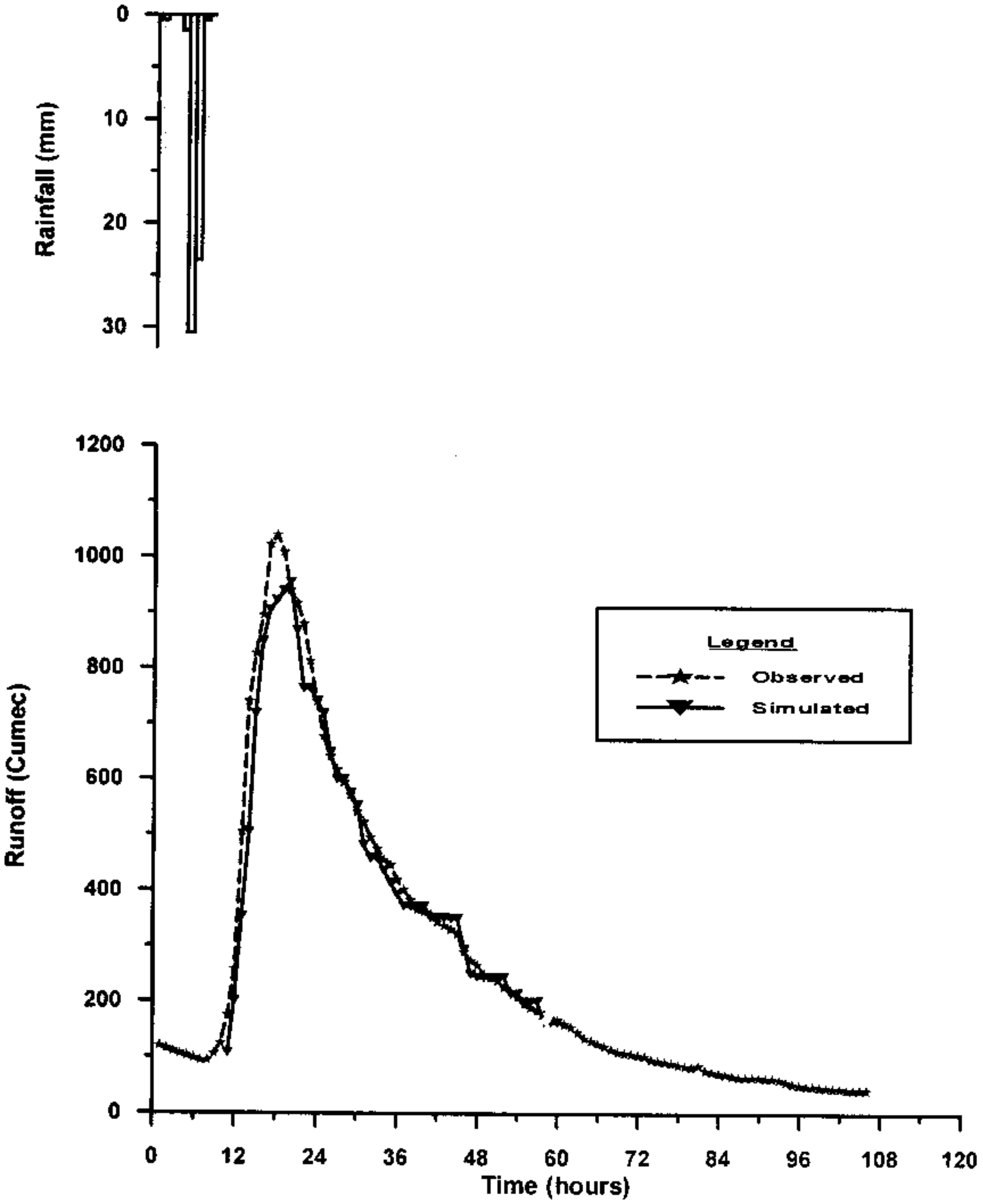


Figure 19: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -3 for Testing of the ANN model.

Case 5

In this case the flood hydrograph FH1, FH3, FH4, FH5, FH6 were used to find out the weights between different layers. The initial random weights between different layers were updated using backpropagation algorithm. The updated weights were used to simulate the flood FH2 and comparison was made with the observed flood hydrograph. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table 23 and Table 25. Table23 represents the weight matrix between input layer and hidden layer whereas, Table 25 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table 24 whereas for the output neuron its value is 0.1333. The r.m.s.e. in forecasting the event FH2 is 73.62. The Figure 20 shows the forecasted and observed event FH2.

Case 6

As shown in Table 10 the flood hydrograph FH2, FH3, FH4, FH5, FH6 were used in this case to find out the weights between different layers. The initial random weights between different layers were updated using backpropagation algorithm. The updated weights were used to simulate the flood FH1 and comparison was made with the observed flood hydrograph. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table 26 and Table 28. Table 26 represents the weight matrix between input layer and hidden layer whereas Table 28 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table 27 whereas for the output neuron its value is 0.6305. The r.m.s.e. in forecasting the event FH1 is 63.41. The Figure 21 shows the forecasted and observed event FH1.

Case 5

Table23 Weight matrix between input layer (LI) and hidden layer (Lh) for case 5

-1.3569	-1.1902	-0.1878	-0.0785	0.0652	0.1311	0.2076	-0.1191	-0.1345	0.0917	-0.2434	-0.0644	-0.1347	0.1671	-0.0624	-0.2651	0.0471	-0.0218
-0.7003	-0.7158	-0.1981	-0.3489	-0.5508	-0.7346	-0.7414	-0.4714	-0.4931	-0.4005	-0.4021	0.0963	0.2977	0.0086	0.2508	-0.1142	-0.1118	-0.0247
-0.7661	-0.6387	-0.1330	-0.0756	-0.9318	-0.6827	-0.8196	-0.4446	-0.5363	-0.4675	-0.4707	0.2264	0.0391	0.4669	0.0101	0.3700	-0.0628	0.0102
-0.7991	-0.5967	-0.6892	0.0559	-0.9127	-1.4994	-1.2523	-0.5659	-0.5098	-0.5063	-0.7339	0.0082	0.1628	0.0658	0.5429	0.0846	0.3127	-0.2646
-0.8631	-0.7727	-0.5335	-0.0873	-0.2759	-0.2963	-0.7666	-0.5086	-0.2473	-0.4040	-0.0981	-0.0697	0.0183	0.1826	0.0778	0.2515	0.2571	-0.0864
-0.7814	-0.8458	0.0329	-0.4542	-0.1474	-0.5857	-0.5818	-0.5420	-0.3463	-0.3657	-0.2625	0.2464	0.2081	0.1818	0.0845	0.1687	0.2272	-0.1217
-1.3291	-1.1392	0.1268	0.7305	0.4736	0.2465	0.3635	1.2136	1.3371	0.9876	-1.3157	-1.3551	-0.2076	0.0336	-0.2230	0.1065	-0.5181	-0.2926
-1.2856	-0.8048	-0.3140	-0.1648	-0.1283	-0.3930	-0.2438	-0.1784	-0.1594	-0.1197	-0.0455	-0.2194	0.2274	-0.1439	0.1448	-0.2054	0.0206	0.2757
0.5604	-0.8451	0.3747	0.6847	0.9284	0.6611	0.8762	0.9341	0.7532	0.9905	1.3513	1.3507	0.8878	0.4020	0.3608	0.0017	0.1085	-0.4214

Table24 Biases for the neurons of hidden layer for case 5

-0.7725	-0.2455	-0.2182	-0.0427	-0.4719	-0.5443	-0.6945	-0.7539	-1.8722

Table25 Weights between hidden layer (LH) and output layer (Lo) for case 5

-0.0746	-1.0377	-0.9844	-1.1628	-0.7994	-0.7604	1.8174	-0.4958	1.2771

Case 6

Table26 Weight matrix between input layer (LI) and hidden layer (Lh) for case 6

0.5594	-1.1747	-0.7306	0.6839	0.9196	1.0304	1.0125	0.9211	1.0420	1.3491	1.2682	1.5061	1.1599	0.9418	0.9523	0.4283	0.4896	-0.1778
-2.2612	-1.4781	-0.2841	-0.3560	-0.5814	-0.4142	-0.4613	-0.3058	-0.0495	-0.0866	0.0036	0.1467	-0.0532	-0.0768	0.0600	-0.0692	-0.3090	-0.1411
-1.0153	-2.3587	-1.0828	0.2475	-0.5000	-0.4171	-0.5613	0.1981	-0.1956	-0.5110	-0.0380	0.0486	0.1279	0.1052	0.0753	0.0365	0.1681	0.3009
-0.9824	-0.5106	-0.6647	0.0507	-0.6682	-0.7339	-0.7145	-0.3704	-0.0661	-0.4782	-0.7903	0.1283	0.7402	0.2223	0.7660	0.3395	0.4649	-0.1076
-1.9830	-1.6631	-0.4520	-0.2279	-0.3621	0.0691	0.1493	-0.0349	-0.0713	0.3995	-0.4075	-0.1467	-0.0407	-0.2747	-0.0308	-0.2469	-0.1303	-0.1634
-2.0837	-1.5903	-0.5842	-0.4642	-0.3052	-0.2646	-0.2715	-0.1925	-0.3260	-0.0001	-0.0602	-0.3494	-0.2660	-0.1652	0.0936	0.0044	-0.1829	-0.3321
-0.4506	-1.1630	-0.8624	0.2626	-0.8251	-0.6356	-0.4258	-0.5433	0.0793	-0.9421	-0.6310	0.5488	0.0030	0.4638	0.1913	0.5449	0.2361	0.1960
-0.5987	-0.7828	0.5286	-0.5820	-0.3306	-0.8926	-0.8758	-0.0600	-0.5955	-0.5058	-0.3177	-0.0928	0.2866	0.7056	0.6104	0.0150	0.1392	-0.2557
-1.5747	-1.4983	0.7271	0.9750	0.3186	0.1094	0.5571	1.6425	1.6353	0.2874	-2.4985	-1.6370	0.9554	1.0123	0.8636	0.4957	-0.3166	-0.1817

Table27 Biases for the neurons of hidden layer for case 6

-3.4217	-0.8479	-0.4001	-0.3017	-1.1053	-0.9681	-0.3492	-0.3662	-3.1063

Table28 Weights between hidden layer (LH) and output layer (Lo) for case 6

1.4848	0.2244	-1.0601	-1.4210	-0.1700	0.1016	-1.3438	-1.4939	2.4767

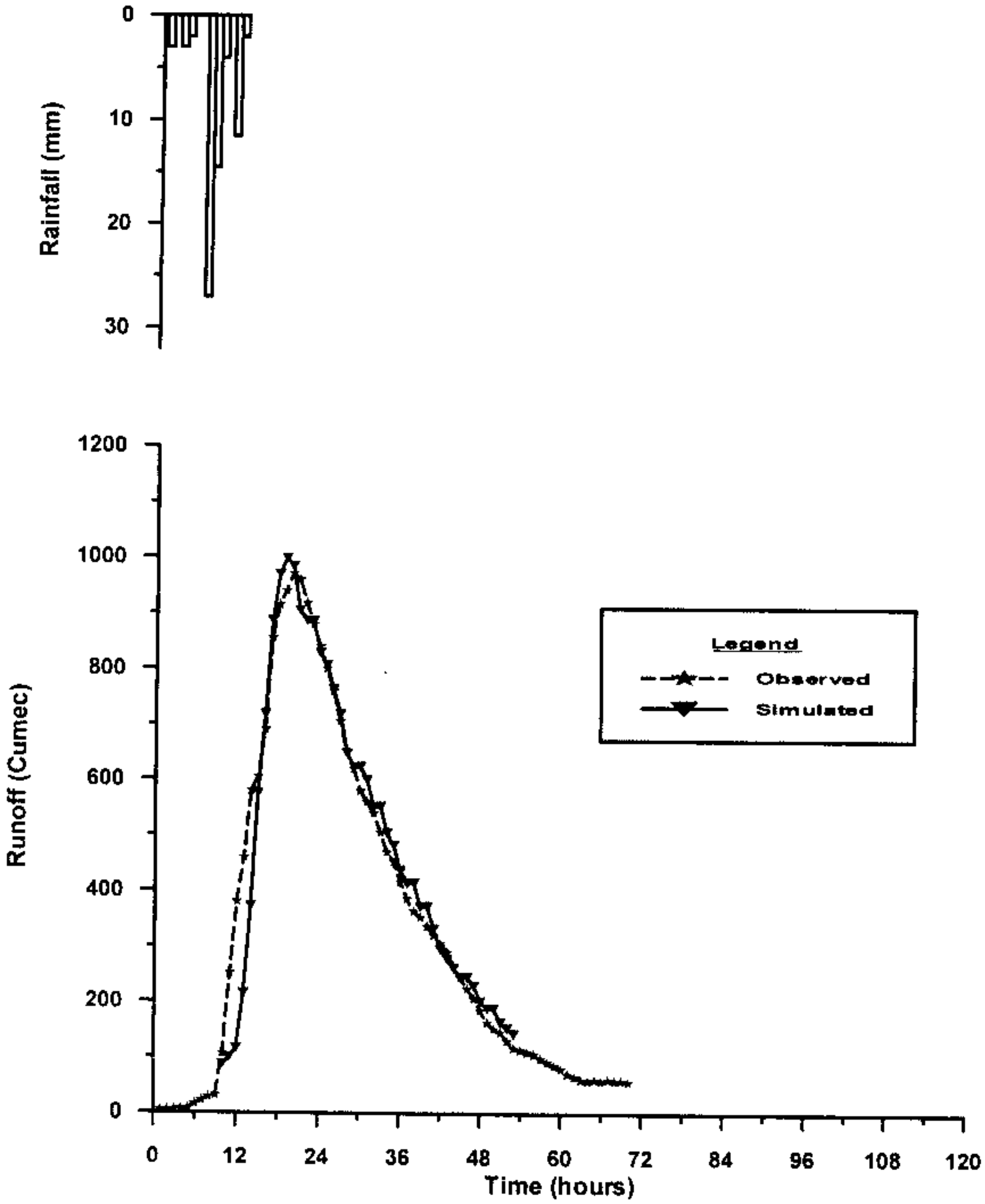


Figure 20: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -2 forTesting of the ANN model.

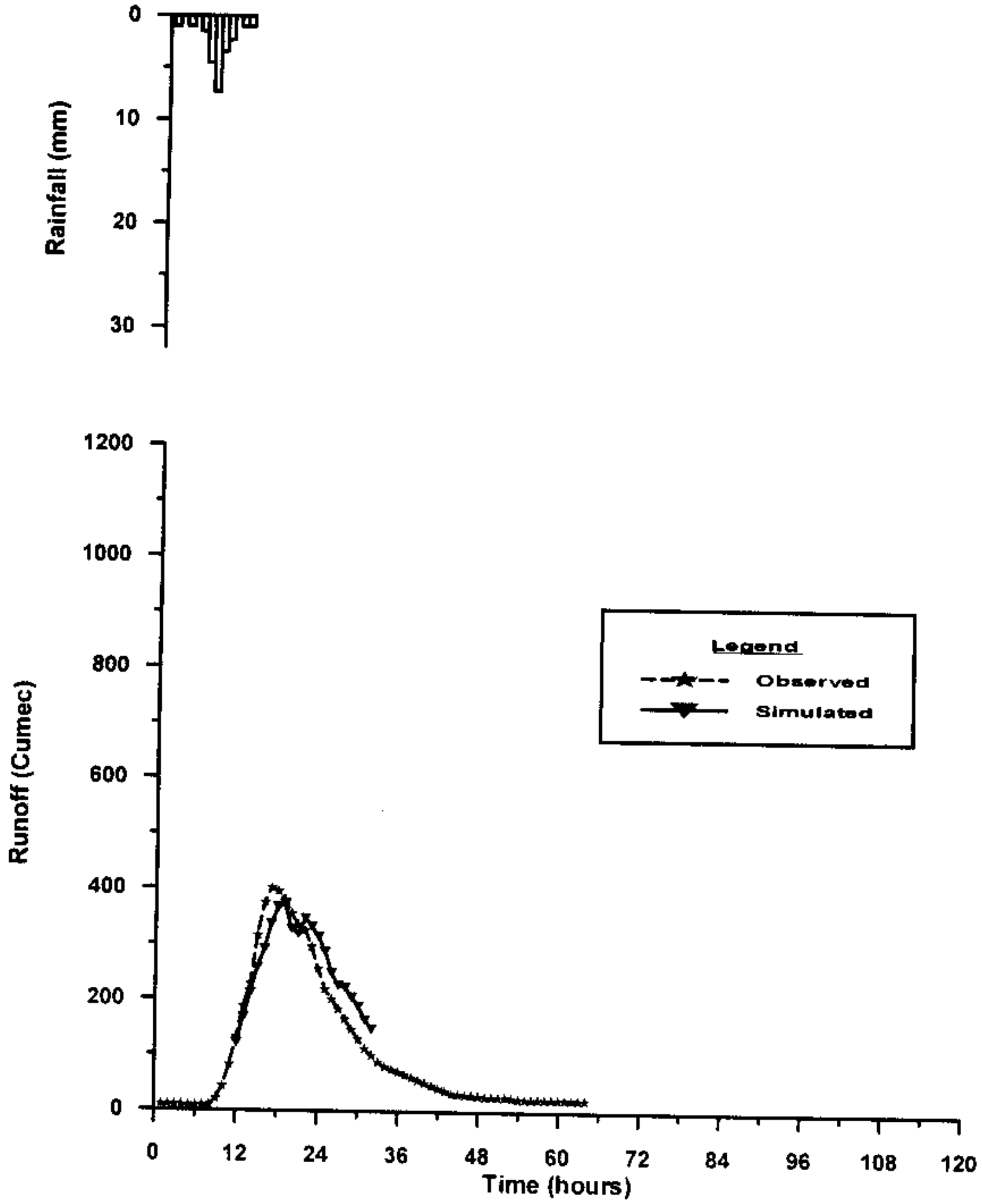


Figure 21: Observed and Simulated 6 Hours Ahead Forecast for Flood Hydrograph -1 forTesting of the ANN model.

Case 7

In this case the flood hydrographs FH1, FH2, FH3, FH4, FH5, FH6 were used to further update the weights between different layers. The initial weights were considered as in case 2 between different layers and were updated using backpropagation algorithm. In the training process the sigmoidal function is used as a transfer function. The learning rate was kept at 0.01. The total numder of training cycles were kept at 1000. The magnitude of weights were kept between –2.5 and 2.5. The weight matrixes for different layers are shown in Table 29 and Table 31. Table 29 represents the weight matrix between input layer and hidden layer whereas Table 31 represents weights between hidden layer and output layer. The biases values for neurons in hidden layer are given in Table 30, whereas for the output neuron its value is 0.1395. The comparison is made between the corresponding weight matrixes of case 2 and case 7 and difference matrixes are shown in Table 32 and Table 33. The difference of weights between input layer and hidden layer shows very small change in magnitude. Thus there is no significant change of weights between input layer and hidden layer. Similarly no significant change as shown in Table 33, is observed in weights between hidden layer and output layer. This signifies the fact that for the data available the weights and biases of case 2 can be considered as best weights which under given conditions may represents the rainfall – runoff process in the study are appropriately.

Case 7

Table29 Weight matrix between input layer (LI) and hidden layer (Lh) for case 7

-2.4952	-2.4961	0.7241	-0.0582	-0.7487	0.0661	0.3046	1.1058	0.7534	-0.9659	-2.4995	-1.6767	0.5315	0.8675	0.3821	0.1447	-0.0764	-0.1728
-0.9245	-1.0863	-0.3135	-0.2968	-0.4090	-0.0329	0.0322	0.5224	1.3734	0.6920	-0.6863	-1.1136	-0.5850	0.1108	-0.0172	-0.2450	-0.2120	-0.4671
-0.5668	-0.6873	0.3637	-1.3056	-0.2670	-0.7292	-0.5175	-0.4068	-0.6654	-0.6270	-0.4294	-0.1800	0.0795	-0.0035	0.4387	0.0873	0.1061	-0.0554
-1.4973	-2.2978	0.1562	-0.4220	-0.4327	-0.7343	-0.8038	-0.1305	-0.4890	-0.6681	0.2612	0.3369	0.4422	0.2758	0.5520	0.0624	0.4477	0.1246
-0.7865	-0.5008	-0.4394	-0.5077	-0.6608	-0.5372	-0.5935	-0.8995	-0.2928	-0.6891	-0.4948	0.2112	-0.0114	0.6243	0.0128	0.0690	0.2711	0.0761
-0.9541	-0.8852	-0.9824	0.1600	-1.0195	-0.9880	-0.9732	-0.6311	-0.1725	-0.7178	-0.4287	0.5021	0.4354	0.9701	0.7511	0.5593	0.2712	0.2114
-1.0737	-0.3318	-0.8069	0.0140	-1.0537	-0.5771	-0.5018	-0.7184	-0.2305	-0.5257	-0.5184	0.0443	0.3828	0.2694	0.3815	0.4053	0.2544	0.1178
-1.5897	-1.3910	-0.1407	-0.2885	-0.3386	-0.1518	0.0722	-0.2141	-0.1359	-0.1886	0.1337	-0.4292	-0.3061	-0.1683	-0.2985	-0.2771	-0.2863	0.0804
0.2745	-0.8385	-0.0682	0.6764	1.6054	0.7507	1.2397	0.7981	0.8425	1.1635	1.3612	1.5297	1.1428	0.9065	0.7970	0.5010	0.3821	0.2062

Table30 Biases for the neurons of hidden layer for case 7

-2.1948	-0.6052	-0.3041	0.1466	-0.4467	-0.2909	-0.5206	-0.5677	-1.8407

Table31 Weights between hidden layer (LH) and output layer (Lo) for case 7

2.4059	1.3002	-1.4682	-1.3490	-0.7391	-1.3234	-1.1466	-0.2808	1.2800

Table32 Difference of weight matrixes between input layer (L_I) and hidden layer (L_H) for case7 and case 2

-0.0048	-0.0039	-0.0851	0.0818	0.1281	0.0950	0.1158	0.0158	-0.0190	0.0305	-0.0005	-0.0338	-0.1069	-0.1054	-0.0747	-0.0401	0.0007	0.0064
-0.0191	0.0443	-0.0228	0.0311	0.0508	-0.0091	0.0098	-0.152	-0.0531	-0.0737	-0.0006	0.0189	-0.0394	-0.0368	-0.0037	0.0087	-0.0295	0.0154
0.0026	-0.0040	-0.0528	0.0552	-0.0682	0.0658	0.0141	-0.0257	-0.0251	-0.0242	-0.0163	-0.0051	0.0456	0.0339	0.0025	-0.0018	-0.0295	0.0276
0.1296	0.1173	-0.0152	-0.0216	-0.0571	-0.0133	-0.0489	-0.0680	-0.0188	0.0323	0.0044	0.0461	0.0947	0.0696	0.0330	0.0243	-0.0022	-0.0138
-0.0062	-0.0178	0.0062	-0.0207	-0.0015	0.0234	0.0003	-0.0194	-0.0215	-0.0038	0.0017	-0.0173	0.0343	0.0070	0.0047	-0.0035	-0.0111	-0.0138
-0.0474	-0.0883	0.0589	-0.0820	0.0169	-0.0116	-0.0123	-0.0225	-0.0266	0.0237	0.0145	-0.0069	0.0714	0.0215	-0.0025	-0.0108	-0.0279	0.0032
-0.0047	-0.0353	0.0445	0.0609	0.0205	0.0083	-0.0022	-0.0177	-0.0272	0.0024	0.0157	-0.0146	0.0323	-0.008	0.0002	-0.0035	-0.0251	0.0187
0.0555	0.0479	0.0096	0.0024	0.0028	0.0090	0.0002	-0.0039	0.0045	0.0102	0.0059	0.0061	0.0155	0.0124	0.1068	0.0038	-0.0018	-0.0018
0.0681	0.0163	-0.0144	0.0277	0.0201	-0.0236	-0.0204	-0.0020	0.0722	0.0841	0.0659	0.0416	-0.0362	-0.0426	-0.0550	-0.0636	-0.0397	-0.0625

Table33 Difference of weights between hidden layer (L_I) and output layer (L_o) for case 7 and case 2

0.0097	-0.0586	0.0100	-0.0311	-0.0100	-0.0472	-0.0027	0.0588	0.0278

7.0 DISCUSSION OF RESULTS

Three flood events (FH1, FH2, FH3) were considered initially to identify the ANN model, which consist of 18 input neurons in input layer, 9 hidden neurons in hidden layer and one output neuron in output layer. The fitted flood hydrograph (FH1, FH2, FH3) and 6 hour ahead forecasted flood hydrographs (FH4, FH5, FH6) were simulated using developed ANN model. The r.m.s.e for fitted and forecasted flood hydrographs are shown in Table 34. As r.m.s.e in simulating the forecasted flood hydrograph was significant, more flood events were considered to evaluate better optimized weights between neurons of different layers which may represent the rainfall runoff process in the catchment more realistically. Six different combinations of flood events (shown as S.N. 2 to 7 in Table 34) were considered for better weight optimization. The r.m.s.e for all the six cases for training and testing the ANN weights are shown in Table 34. The case2 has the least r.m.s.e in testing and considered for ANN model. Case 7 is used to check any significant change in the weight matrixes of case 2. The difference matrixes shows that there is no significant change in the weight matrixes of case 2. Thus for the given conditions and available data the weight matixes of case 2 may represent the rainfall – runoff process in the catchment more realistically and can be used for forecasting 6 hour ahead runoff at Sarath gauging site.

Table 34: Flood hydrograph considered in training and testing of ANN model and corresponding r.m.s.e.

S. N.	Events Considered for Training of ANN Model	Events Considered for Testing of ANN Model	R.M.S.E (Training)	R.M.S.E (Testing)
1	FH1, FH2, FH3	FH4, FH5, FH6	38.72	194.90
2	FH1, FH2, FH3, FH4, FH5	FH6	41.23	73.75
3	FH1, FH2, FH3, FH4, FH6	FH5	40.60	55.40
4	FH1, FH2, FH3, FH5, FH6	FH4	42.42	64.81
5	FH1, FH2, FH4, FH5, FH6	FH3	43.58	78.99
6	FH1, FH3, FH4, FH5, FH6	FH2	42.42	73.62
7	FH2, FH3, FH4, FH5, FH6	FH1	44.72	63.41
8	FH1, FH2, FH3, FH4, FH5, FH6	-	45.8	-

8.0 CONCLUSION

This study evaluated the applicability of artificial neural networks in rainfall runoff process modelling. A case study has been done for Ajay river basin to develop an ANN based rainfall runoff model for the basin to forecast 6 hour ahead runoff at Sarath. The back propagation algorithm has been used for optimisation of weights. Initially three flood events were considered for training, to find out the weights between different layers of the network. The developed ANN model was validated for rest of the flood events. The validation results suggested the updating of the existing weights of the network for better adaptation of the underlying rainfall – runoff process in the catchment. More flood events were considered for training the network, to determine weights which may represents the rainfall – runoff process in the catchment more realistically. Six sets of flood events were used in turn for training the network. The trained networks were then validated on the remaining flood event(s). The results showed that the weights determined by considering flood events FH1, FH2, FH3, FH4, FH6 (case2) in training result in least r.m.s.e in validation. A check has been performed to observe any significant change in the weight matrixes of case 2 by including one more flood event (FH5) in the training process. No significant change has been observed in the weight matrixes of case 2 for the available set of data. Thus the weights determined in case 2 can be used for issuing the 6 hour ahead forecast at Sarath gauging site of Ajay river basin for the given condition. However as the rainfall – runoff process is not stationary and every flood event has its own characteristics there is always a need for updating the existing model, whenever new flood events become available. The ANN model can perform even better if more flood events with significantly low noise component are made available for modelling the rainfall-runoff process and can be used effectively for the flood warning in the Ajay river basin.

REFERENCES

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

Buch, A.M., Mazumdar, H.S. and Pandey, R.C., "A Case Study of Runoff Simulation of A Himalayan Glacier Basin", Proceedings on International Joint Conference on Neural Networks, 1, 971-974, (1993).

Chander, S. and Prasad, T., "Forecasting and Prediction of Floods", In: *Indo-US Workshop on Flood Mitigation (New Delhi)*, (1981).

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

Chander, S., Spolia, S.K. and Kumar, A., "Flood Stage Forecasting in Rivers", In: *Monsoon Dynamics* (Ed. By Sir M.J. Lighthill), Cambridge University Press, London, UK. (1980).

Crespo, L. and Mora, E., "Drought Estimation with Neural Networks", Advances in Engineering Software, Vol. 993, pp. 167-170, (1993).

Eykhoff, P., "System Identification, Parameter and State Estimation", Wiley, New York, (1974).

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

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

French, M.N., Krajewski, W.F. and Cuykendall, R.R., "Rainfall Forecasting in Space and Time Using A Neural Network", Journal of Hydrology, Vol. 137, pp. 1-31, (1992).

GFCC, "Comprehensive Plan of Flood Control for the Ajay River System." Vol II/8, March, (1986).

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

Hecht. Nielsen, R., Neurocomputing Addison-Wesley Publication Company, New York, (1991).

Hecht. Nielsen, R., "Theory of the Backpropagation Neural Network." Proc. Int. Joint Conf. on Neural Networks, IEEE, Washington D. C., Vol. 1, 593-605, (1989)

Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problem Transportation", ASME Journal of Basic Engineering, Vol. 82, pp. 35-44, (1960).

Kang, K.W., "Evaluation of Hydrologic Forecasting System Based on Neural Network Model", Proceedings of IAHS Floods and Droughts, Japan (1993).

Karunanithi, N., Greeney, W.J., Whitley, D. and Boree, K., "Neural Network for River Flow Prediction", Journal of Computing in Civil Engineering, ASCE, 8(), 201-219, (1994).

Kashyap, R.L. and Rao, A.R., "Dynamic Stochastic Models from Empirical Data", Academic Press, New York, pp. 334, (1975).

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

Kumar, Arun, "Prediction and Real Time Hydrological Forecasting", Ph.D. Thesis, Indian Institute of Technology, Delhi, India, (1980).

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

Mutreja, K.N., Yin, A. and Martino, I., "Flood Forecasting Model for Citandy River", Flood Hydrology, Eds. V.P. Singh and Reidel Dordrecht, The Netherlands, 211-220, (1987).

O'Connell, P.E., "Real Time Hydrological Forecasting and Control", Institute of Hydrology, UK, pp. 264, (1980).

Panu, V.S. and Unny, T.E., "Extension and Application of Feature Prediction Model for Synthesis of Hydrologic Records", Water Resource Research, 16(1), 77-96, (1980).

Patrick, C., Shahab, M., Razi, G., "Performance of A Virtual Runoff Hydrograph System", Journal of Water Resources Planning and Management, Vol. 122(6), 421-427, (1996).

Raman, H. and Kumar, S., "Multivariate Modelling of Water Resources Time Series Using Artificial Neural Networks", Hydrological Sciences Journal, 40(2), 145-163, (1995).

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., "Learning Internal Representation by Error Propagation", In: Parallel Distributed Processing: Explorations in The Microstructures of Cognition, Cambridge, MIT Press, (1986).

Sage, A.P. and Husa, "Adaptive Filtering with Unknown Prior Statistics", Proceedings on Joint Automatic Control Conference, pp. 760-769, (1969).

Singh, V.P., "Hydrologic Systems – Rainfall-Runoff Modelling", Vol. 1, Prentice Hall Inc., Englewood Cliffs, New Jersey, USA, (1988).

Smith, J. and Eli, R.N., "Neural Network Models of Rainfall-Runoff Process", Journal of Water Resource Planning and Management, Vo. 121, No. 6, pp. 499-507, (1995).

Stedinger, J.R. and Taylor, M.R., "Synthetic Stream-flow Generation Model Verification and Validation", Water Resource Research, 18(4), 909-918, (1982).

Valencia, R.D. and Schaake, J.C., "Disaggregation Processes in Stochastic Hydrology", Water Resource Research, 9(3), 58-585, (1973).

DIRECTOR	K. S. Ramasastri
COORDINATOR, GPNRC	R.D.SINGH
HEAD, GPNRC	Rakesh Kumar
STUDY GROUP	Sanjay Kumar
	Rakesh Kumar
	C. Chatterjee
TECHNICAL ASSISTANCE	A. K. Sivdas