International Journal of Computer Applications Technology and Research
Volume 3– Issue 7, 473 - 478, 2014, ISSN: 2319–8656

Flood Prediction Model using Artificial Neural Network

Abhijit Paul
Assam University
Silchar, Assam, India

Prodipto Das
Assam University
Silchar, Assam, India

Abstract: This paper presents a Flood Prediction Model (FPM) to predict flood in rivers using Artificial Neural Network (ANN) approach. This model predicts river water level from rainfall and present river water level data. Though numbers of factors are responsible for changes in water level, only two of them are considered. Flood prediction problem is a non-linear problem and to solve this nonlinear problem, ANN approach is used. Multi Linear Perceptron (MLP) based ANN's Feed Forward (FF) and Back Propagation (BP) algorithm is used to predict flood. Statistical analysis shows that data fit well in the model. We present our simulation results for the predicted water level compared to the actual water level. Results show that our model successfully predicts the flood water level 24 hours ahead of time.

Keywords: FPM; ANN; MLP; FF; BP

1. INTRODUCTION

Flood occurs when river bursts its banks and the water spills on top of the floodplain. Flooding tends to be caused by heavy rainfall, when absorption of water is low and overflows are not controllable by river channels. The faster the rainwater reaches the river channel, the more likely it is to flood. Floods can cause damage to lives and property and possessions as well as disruption to communications. There is no mechanism to avoid flood but only a prediction can secure the life of inhabitants and also can reduce damages. To predict possible flood, most of the factors such as amount of rainfall, present river water level, degree of ground saturation, degree of permeable soil etc. need to be determined. If a forecast is issued after the prediction, then a flood warning can be communicated to warn the public about the possible extent of the flood, and to give people time to move out of the area. If forecasts can be made with long lag time between the storm and peak discharge, damages can be reduced in great scale.

The effective implementation of flood monitoring and forecasting system is non-trivial, since it requires the reliability coupled with the availability of related information. Over the years, flooding has been studied under various considerations and methodologies such as wireless sensors network, embedded system with a middleware, internet-based real-time data acquisition, and flood modeling and forecasting [1-3]. In addition to sensor technologies, space and satellite data technologies have been used to improve the accuracy [4] [5]. These papers provide great insights into the development of flood forecasting and modeling using data from satellite, image processing, and GIS. Different Mathematical and Statistical models are also used for flood forecasting [6]. Mathematical models are based on physical consideration and statistical models are based on analysis. To build flood monitoring and warning system, one of the widely used infrastructures is ad-hoc wireless sensor network [7]. In ad-hoc network, set of remote wireless are deployed in monitoring area to monitor water condition data, and these data are broadcasted in the form of web, sms or email technology to build a real time flood monitoring and forecasting system.

The wide variety of available forecasting techniques used by the hydrologists today, include physically based rainfall-runoff modeling techniques, data-driven techniques, and combination of the both, with forecasts ranging from short-term to long-term [8] [9]. Although hydrologists have used many models to predict flooding, the problem remains. Some of the models find difficulties with dynamic changes inside the watersheds. Some models are too difficult to implement and need to have robust optimization tools and some models require an understanding of the physical processes inside the basin. These problems have lead to exploration of a more data driven approach.

Therefore, to improve the accuracy of flood models and to deal with some of the above limitations, in recent years, several hydrological studies have used new techniques such as ANN, fuzzy logic and neuro-fuzzy to make flood predictions [10] [11]. These techniques are capable of dealing with uncertainties in the inputs and can extract information from incomplete or contradictory datasets. These new methods are frequently developed for hydrological and flood modeling only with rainfall and runoff as input and output, usually without taking into consideration of other flood causative factors.

Difficulty in river flood prediction is river water level fluctuation in highly nonlinear way. To solve this nonlinear problem, MLP based ANN approach is used. The input and output parameters used in this model are based on real-time data obtained from Flood Forecasting and Warning Centre, Bangladesh Water Development Board [12]. FPM is tested by the statistical fit functions- SST, SSE, MSE, RMSE, MAPE and R2 [13]. Then we present our simulation results of FPM for the predicted water level compared to the actual water level.

2. STUDY AREA

The river Manu originates from Sakhan range, Tripura, India and flows northerly via Kailashahar, Tripura, India to Bangladesh [14]. It joins the Kushiyara River at Manumukh in Maulvi Bazar district of Bangladesh. Basin area of Manu is 1979 sq. km. and its annual flow is 170034 in thousand m3 [15]. Its highest water level is 20.42 m and danger level is 18.0 m [12]. (Data collected online at http://www.ffwc.gov.bd/).

3. DATA SET

Daily water levels and rainfall data are collected online at Manu RB gauging station of river Manu from the Bangladesh Water Development Board (BWDB) website [12]. Approximate 200000 data at Manu RB gauging stations of river Manu under Meghna basin are utilized for training and testing.

4. FLOOD SIMULATION

4.1 ANN

ANNs are mathematical models of human perception that can be trained for performing a particular task based on available empirical data [16] [17]. When the relationships between input data and output data are unknown, they can make a powerful tool for modeling. The theory and mathematical basis of ANNs are explained in detail by many researchers. The model is based on a Feed Forward Multilayer Perceptron (FFMLP). As shown in Fig-1, an FFMLP includes a number of neurons or nodes that work in parallel to transform the input data into output categories. Typically, an FFMLP consists of three layers namely input, hidden layers and output. Each layer, depending on the specific application in a network, has some neurons. Each neuron is connected to other neurons in the next consecutive layer by direct links. These links have a weight that represents the strength of outgoing signal.

The input layer receives the data from different sources. Hence, the number of neurons in the input layer depends on the number of input data sources. The data are processed in hidden and output layers actively. The number of hidden layers and number of neurons in each layer are often defined by trial and error [18]. The number of neurons in output layers is fixed according to the application. Each hidden neuron responds to the weighted inputs it receives from the connected neurons from the preceding input layer. Once the combined effect on each hidden neuron is determined, the activation at this neuron is determined via a transfer function. Many differentiable nonlinear functions are available as a transfer function. Since the sigmoid function enables a network to map any nonlinear process, most networks of practical interest make use of it [19].

A typical Feed Forward ANN is shown in Fig-1.

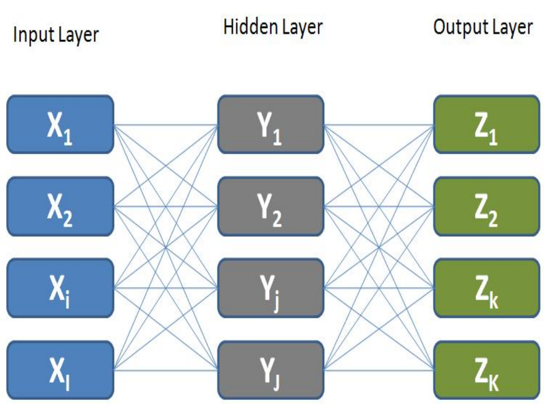

Figure. 1 Three Layer Feed Forward ANN

The first layer, called the input layer, consists of I nodes and connects with the input variables. This layer performs no computation but is used to distribute I inputs into the network. The last layer connects to the output variables and is called the output layer. This layer consists of K output nodes. The one or more layers of processing units located between the input and output layers have no direct connections to the outside world and are called hidden layers. An MLP may have a number of hidden layers. However, we have considered only two hidden layers. This layer consists of J hidden nodes.

In general, all connections are 'feed forward'; that is, they allow information transfer only from an earlier layer to the next consecutive layers. Nodes within a layer are not interconnected, and nodes in nonadjacent layers are not connected.

Each hidden node j receives I incoming signals (x_i), from every node i in the previous layer (for example from Input layer to Hidden layer). Associated with each incoming signal (x_i), there is a weight (w_{ij}) connected between layer I and J. The effective incoming signal (net_j) to node j is the weighted sum of all the incoming signals as follows:

$$\mathrm{net_j} = \sum_{i=1}^{I} \mathrm{w_{ij} x_i} \tag{1}$$

This effective incoming signal (net_j) passes through a nonlinear activation function to produce the outgoing signal, called the activation or activity level, (y_j) of the node.

The delta learning rule or back propagation algorithm is used for learning the network. The purpose of this algorithm is to adjust the weights w_{ij} and w_{jk} which connects Input layers to Hidden layer and Hidden layers to Output layers to assure a minimization of the error function (E_k).

$$E_k = \frac{1}{2}(z_k - t_k)^2 \tag{2}$$

where z_k, is the output from output layer and t_k is the target value.

The outgoing signal (z_k) is a function of the activation as follows:

$$z_k = f(net_k) \tag{3}$$

The effective incoming signal net_k, comprising weighted sum signals from the hidden layer (y_j) is calculated as follows:

$$net_k = \sum_{j=1}^{J} \mathrm{w_{jk}\, y_j} \tag{4}$$

The outgoing signal (y_j) is the result of the incoming signal net_j, being passed through the activation function, as follows:

$$y_j = f(net_j) \tag{5}$$

The weight adjustment of both w_{ij} and w_{jk} are based on the gradient descent search, which changes the weights in the direction in which the error surface goes down most steeply, as follows:

From input layer to hidden layer:

$$\Delta w_{ij} = -\eta \frac{\partial E_k}{\partial w_{ij}} \tag{6}$$

where η is a learning parameter.

From hidden layer to output layer:

$$\Delta w_{jk} = -\eta \frac{\partial E_k}{\partial w_{jk}} \tag{7}$$

Finally, the updated weights are

$$w_{ij} = w_{ij} + \Delta w_{jk} \tag{8}$$

$$w_{jk} = w_{jk} + \Delta w_{jk} \tag{9}$$

There are various activation functions employed in ANNs, the most commonly used ones being the following the unipolar binary function or sigmoid function (S). The sigmoid function is defined as

$$S(net_j) = \frac{1}{1 + e^{-\lambda net_j}} \tag{10}$$

where $\lambda > 0$. λ is proportional to the neuron gain determining the steepness of the function.

The term λnet_j can vary on the range $-\infty$ to $+\infty$, but $S(net_j)$ is bounded between 0 and 1.

4.2 ANN Architecture

The ANN architecture refers to the number of layers and connection weights. It also defines the flow of information in the ANN. In ANN, design of suitable structure is the most important and also the most difficult part. There are no strict rules to define the number of hidden layers and neurons in the literature.

In this research, three-interconnection ANN architecture comprises an input layer, two hidden layers, and an output layer is used. The input layer contains two neurons (one for rainfall and another for water level) each representing a causative factor that contributes to the occurrence of the flood in the catchment. The output layer contains a single neuron representing river water level after 24 hours. The hidden layers and their number of neurons are used to define the complex relationship between the input and output variables.

4.3 Data Normalization

Neural network training can be made more efficient by performing certain pre-processing steps on the network inputs and targets [20]. Network input processing functions transforms inputs into better form for the network use. The normalization process for the raw inputs has great effect on preparing the data to be suitable for the training. Without this normalization, training the neural networks would have been very slow. There are several types of data normalization.

To normalize input and output dataset, the requirements are-

(i) Dataset minimum and maximum value.

(ii) Normalized scale minimum and maximum value.

$$X_{nor} = a + \frac{(X - A)(b - a)}{(B - A)} \tag{11}$$

where X_{nor} is the normalized value of X, A is minimum value in the dataset, B is maximum value in the dataset, a is minimum value in the normalized scale and b is maximum value in the normalized scale.

4.4 Training and Testing the network

The aim of training process is to decrease the error between the ANN output and the real data by changing the weight values based on a BP algorithm.

A successful ANN model can predict target data from a given set of input data. Once the minimal error is achieved and training is completed, the FF algorithm is applied by ANN to generate a classification of the whole data set.

To train the ANN, a 2-N–N-1 format is used in this study, where N represents number of nodes in the hidden layer. By varying the number of neurons in both hidden layers, the neural networks are run several times to identify the most appropriate neural network architecture based on training and testing accuracies. Most appropriate ANN is selected based on minimum mean square error.

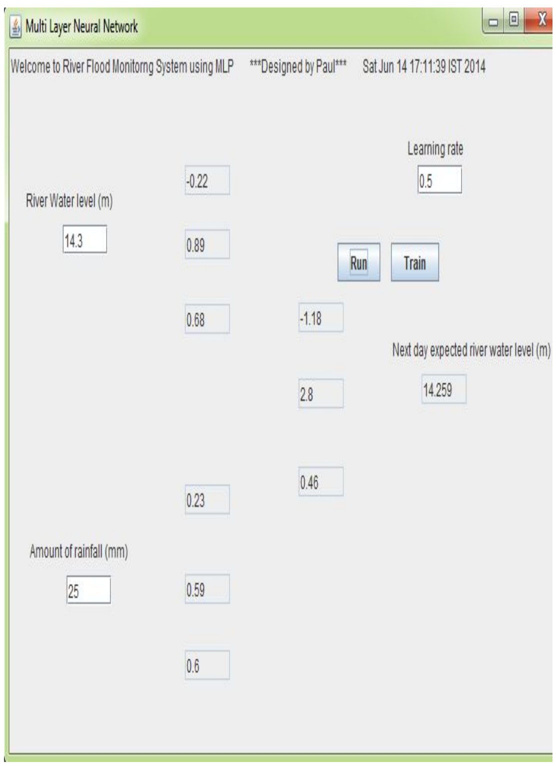

Figure. 2 Snapshot of simulation result

The number of neurons in the second and third layers is checked from 1 to 10 for each of the layer. For each ANN configuration the training procedure is repeated starting from independent initial conditions and ultimately ensuring selection of the best performing network. The decreasing trend in the minimum mean square error in the training and validation sets is used to decide the optimal learning. The training is stopped when the minimum mean square error was achieved. Here 2-6-3-1 format gives us optimal result compare to other format. Therefore this 2-6-3-1 format is used for our experiment.

Here one input layer with two nodes- river water level and amount of rainfall is taken. Two hidden layers with six and three nodes are taken. One output layer with one node is taken. Initially all the weights are supplied random value. Using back propagation algorithm all the weights are updated so that it gives minimum error. After using back propagation algorithm, the updated weights are shown in fig-2. Initially the network is trained by past database. Once training is complete, the network can predict the value for its two inputs. For 14.3 m river water level and 25 mm rainfall, the network predict next day river water level as 14.259 which is shown in fig-2. Here learning rate 0.5 is used.

An important result in testing these data was that the ANN was able to identify all values same as training stage. This result yields a R2 value of 1 which is acceptable result and it shows a high level of prediction. The simulated and ANN predicted river flow, and the regression plot are shown in figs. 4 and 5, respectively.

After ANN training process is completed, different datasets are used to extend, and to determine the model accuracy. Using new data, the network performance was evaluated. These data had the same properties as the training data but they have not been used during the training of the model.

4.5 Algorithm

Step-1: Declare two inputs (river water level as i1 and rainfall as i2), nine weights (w[9]), one learning rate (l), one output (next day water level as o).

Step-2: Initially w[0] to w[8] by random values lies between 0-1.
Initialize l=0.5.

Step-3: Calculate dataset for i1 from 0-21. So A=0, B=21.
Calculate normalized scale for i2 from 0.1-1. So a=0.1, b=1.
Calculate $i1_{nor}$ according to the formula-2.
Also calculate dataset for i2 from 0-300. So A=0, B=300.
Calculate normalized scale for i2 from 0.1-1. So a=0.1, b=1.
Calculate $i2_{nor}$ according to the formula-2.

Step-4: Train the network with the two inputs i1, i2 and measured output o from the database.
Call the function train(i1,i2,o) with different values of i1, i2 and o. Training continues until it reaches to the threshold value 0.05.

Step-5: Now o can be calculated for any given value of i1 and i2.

Step-6: The calculated output o is in normalized form. This normalized value is converted into original value according to the formula-2, which is out predicted output.

5. MODEL PERFORMANCE ASSESMENTS

Variations between the predicted and observed values are shown using gnu plot graph. Data are predicted for 24 hours, 48 hours and 72 hours lead time.

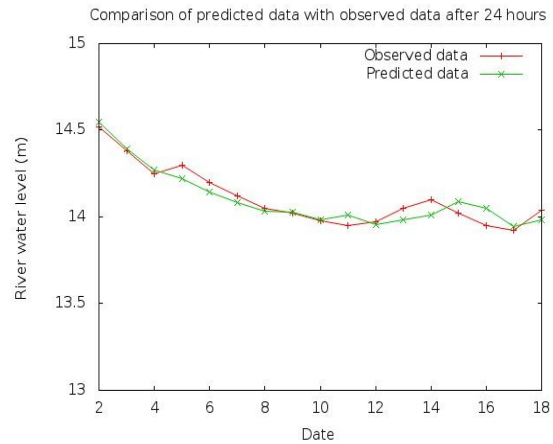

Figure. 3 Comparison of data for 24 hours lead time

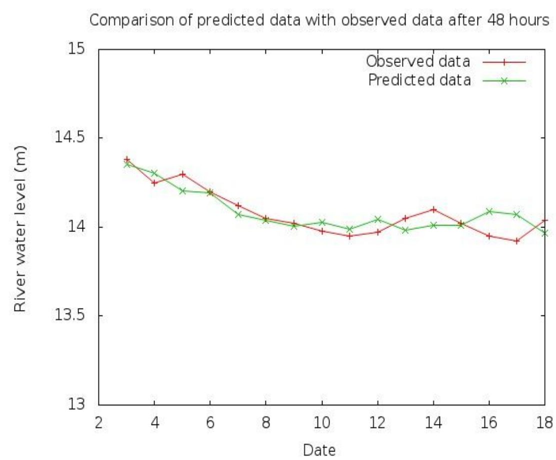

Figure. 4 Comparison of data for 48 hours lead time

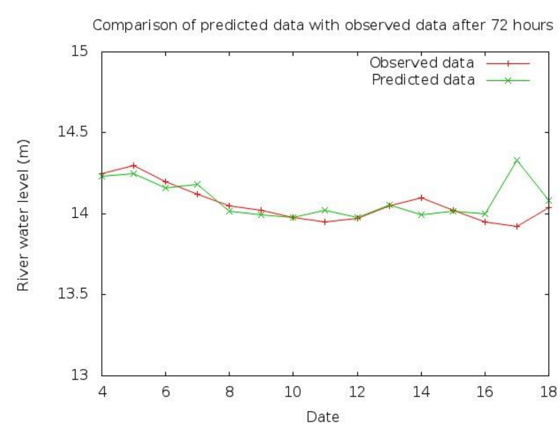

Figure. 5 Comparison of data for 72 hours lead time

The model accuracy assessment is described in terms of the error of forecasting or the variation between the observed and predicted values. In the literature, there are many performance assessment methods for measuring the accuracy and each one has advantages and limitations. Here for comparing the model, statistical measures are taken. Goodness-of-fit statistical functions measure how well data fit into the model. In this study, the most widely used methods namely coefficient of determination (R^2), total sum of squares (SST), sum square of error (SSE), mean sum of error (MSE), root mean square error (RMSE), mean absolute percentage of error (MAPE) are used to check the performance of the model. Each method is estimated from the ANN predicted values and the observed values.

Statistical formulae are given below-

Total sum of square,

$$\mathrm{SST} = \sum_{i-1}^{n} (y_i - \bar{y})^2 \tag{12}$$

Sum of square of error,

$$\mathrm{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{13}$$

Mean sum of error,

$$\mathrm{MSE} = \frac{1}{n}\ x\ \mathrm{SSE} \tag{14}$$

Root mean square of error,

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \tag{15}$$

Mean absolute percentage of error,

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n} |(y_i - \hat{y}_i)/y_i| \,x\, 100 \tag{16}$$

Coefficient of determination R square,

$$R^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} \tag{17}$$

TABLE 1.
STATISTICAL MEASURES

	For 24 hours lead time	For 48 hours lead time	For 72 hours lead time
SST	0.45695294	0.275775	0.180573
SSE	0.046602	0.083753	0.199997
MSE	0.002741	0.005235	0.013333
RMSE	0.052357	0.07235	0.115469
MAPE	0.935756	0.763017	0.62728
R^2	0.898016	0.6963	-0.10757

The results showed that the model has less SSE, MSE, and RMSE. Overall, the errors are negligible. The higher value (close to 1) of MAPE and R^2 seems the model has excellent agreement with the real data. The results show that data for 24 hours lead time has better result compare to others. Therefore our model is used to predict only water level after 24 hours which means next day river water level prediction.

6. CONCLUSION

The focus of this paper is to apply optimized ANN for next day river water level forecasting by determination of suitable input parameters and designing the best network architecture. The study reported in this article has led to the conclusion that MLP type network, consistently performed better compared to other network. Among the water level prediction after 24 hours, 48 hours and 72 hours; prediction after 24 hours performs well. Therefore our ANN model with MLP is used only for predicting next day (24 hours) water level.

7. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40 (8), pp. 102–114, 2002.

[2] V. Seal, A. Raha, S. Maity, S. K. Mitra, A. Mukherjee, and M. K. Naskar, “A Simple Flood Forecasting Scheme using Wireless Sensor Networks,” *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, vol.3, no.1, pp. 45-60, 2012.

[3] V. Sehgal and C. Chatterjee, “Auto Updating Wavelet Based MLR Models for Monsoonal River Discharge Forecasting,” *International Journal of Civil Engineering Research*, vol. 5, no. 4, pp. 401-406, 2014.

[4] M. B. Kia, S. Pirasteh, B. Pradhan, A. R. Mahmud, W. N. A. Sulaiman, and A. Moradi, “An artificial neural network model for flood simulation using GIS:Johor River Basin, Malaysia,” *Environ Earth Sci*. Available: doi:10.1007/s12665-011-1504-z.

[5] S. Fang, L. Xu, H. Pei, Y. Liu, Z. Liu, Y. Zhu, J. Yan, and H. Zhang, "An Integrated Approach to Snowmelt Flood Forecasting in Water Resource Management," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 548-558, 2014.

[6] Y. Chakhchoukh, P. Panciatici, and L. Mili, "Electric Load Forecasting Based on Statistical Robust Methods," *Power Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 982-991, 2011.

[7] Z.J. Haas, T. Small, "A new networking model for biological applications of ad hoc sensor networks," *Networking, IEEE/ACM Transactions on,* vol. 14, no. 1, pp. 27-40, 2006.

[8] F. Hossain, E.N. Anagnostou, and T. Dinku, "Sensitivity analyses of satellite rainfall retrieval and sampling error on flood prediction uncertainty," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 42, no. 1, pp. 130-139, 2004.

[9] Y. Zhang, Y. Hong, X. Wang, J.J. Gourley, J. Gao, H.J. Vergara, and B. Yong, "Assimilation of Passive Microwave Streamflow Signals for Improving Flood Forecasting: A First Study in Cubango River Basin, Africa," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* , vol. 6, no. 6, pp. 2375-2390, 2013.

[10] F. J. Chang, J. M. Liang, and Y. C. Chen, "Flood forecasting using radial basis function neural networks," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* , vol. 31, no. 4, pp. 530-535, Nov 2001.

[11] E. Todini, "Using a Desk-Top Computer for an On-Line Flood Warning System," *IBM Journal of Research and Development*, vol.22, no.5, pp.464-471, 1978.

[12] (2014) Flood Forecasting & Warning Centre, Bangladesh Water Development Board (BWDB) website. [Online]. Available http://www.ffwc.gov.bd/

[13] S. Shakya, H. Yuan, X. Chen, and L. Song, "Application of radial basis Function Neural Network for fishery forecasting," *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on* , vol. 3, no., pp. 287-291, 10-12 June 2011.

[14] M Deb, D. Das, and M. Uddin, "Evaluation of Meandering Characteristics Using RS & GIS of Manu River," *Journal of Water Resource and Protection*, vol. 4, pp. 163-171, 2012.

[15] (2002) State of Environment Report, Tripura State Pollution Control Board. [Online]. Available http://envfor.nic.in/sites/default/files/State%20of%20Environment%20Report%20-%20Tripura%202002.pdf

[16] A.R. Gainguly, "A hybrid approach to improving rainfall forecasts," *Computing in Science & Engineering*, vol. 4, no. 4, pp. 14-21, 2002.

[17] L. C. Chang, P. A. Chen, F. J. Chang, "Reinforced Two-Step-Ahead Weight Adjustment Technique for Online Training of Recurrent Neural Networks," Neural Networks and Learning Systems, IEEE Transactions on , vol. 23, no. 8, pp. 1269-1278, 2012.

[18] Atkinson PM, Tatnall ARL, "Neural networks in remote sensing," *International Journal of Remote Sensing*, vol. 18, pp. 699–709, 1997.

[19] Bishop CM, "Neural networks and their application," *Review of Scientific Instruments*, vol. 65, no. 6, pp. 1803–1830, 1994.

[20] (2014) R. K. Biswas and A. W. Jayawardena, "Water Level Prediction By Artificial Neural Network In The Surma-Kushiyara River System of Bangladesh". [Online]. Available http://www.icharm.pwri.go.jp/training/master/publication/pdf/2009/1.synopsis_mee08177_robin.pdf