

# A Rainfall Prediction Model using Artificial Neural Network

Kumar Abhishek<sup>1\*</sup>, Abhay Kumar<sup>2#</sup>, Rajeev Ranjan<sup>3#</sup>, Sarthak Kumar<sup>4§</sup>

<sup>\*</sup>Dept. of Computer Science and Engineering, NIT Patna-800005

<sup>#</sup>Dept. of Information Technology, NIT Patna-800005

<sup>§</sup>Deolitte India, Pune.

kumar.abhishek@nitp.ac.in, abhay.kumar@nitp.ac.in, rrrnitp@gmail.com, sar1989thak@gmail.com

**Abstract--** The multilayered artificial neural network with learning by back-propagation algorithm configuration is the most common in use, due to of its ease in training. It is estimated that over 80% of all the neural network projects in development use back-propagation. In back-propagation algorithm, there are two phases in its learning cycle, one to propagate the input patterns through the network and other to adapt the output by changing the weights in the network. The back-propagation-feed forward neural network can be used in many applications such as character recognition, weather and financial prediction, face detection etc. The paper implements one of these applications by building training and testing data sets and finding the number of hidden neurons in these layers for the best performance.

In the present research, possibility of predicting average rainfall over Udupi district of Karnataka has been analyzed through artificial neural network models. In formulating artificial neural network based predictive models three layered network has been constructed. The models under study are different in the number of hidden neurons.

**Keywords:** Monsoon rainfall, Udupi, prediction, artificial neural network, back propagation algorithm, multilayer artificial neural network.

## I. INTRODUCTION

Weather forecasting is one of the most imperative and demanding operational responsibilities carried out by meteorological services all over the world. It is a complicated procedure that includes numerous specialized fields of know-how. The task is complicated because in the field of meteorology all decisions are to be taken in the visage of uncertainty. Different scientists over the globe have developed stochastic weather models which are based on random number of generators whose output resembles the weather data to which they have been fit.

The reason is that ANN (Artificial Neural Network) model is based on 'prediction' by smartly 'analyzing' the trend from an already existing voluminous historical set of data. Apart from ANN, the other models are either mathematical or statistical. These models have been found to be very accurate in calculation, but not in prediction as they cannot adapt to the irregularly varying patterns of data which can neither be written in form of a function, or deduced from a formula. These real-life situations have been found to be better

interpreted by 'artificial neurons' which can learn from experience, i.e by back-propagation of errors in next guess and so on. This may lead to a compromise in accuracy, but give us a better advantage in 'understanding the problem', duplicating it or deriving conclusions from it.

Amongst all weather happenings, rainfall plays the most imperative part in human life. Human civilization to a great extent depends upon its frequency and amount to various scales. Several stochastic models have been attempted to forecast the occurrence of rainfall, to investigate its seasonal variability, to forecast yearly/monthly rainfall over some geographical area.

The paper endeavors to develop an ANN model to forecast average monthly rainfall in the Udupi district of Karnataka. Indian economy is standing on the summer monsoon. So prediction of rainfall is a challenging topic to Indian atmospheric scientists. Back propagation ANN to forecast the average summer monsoon rainfall over Udupi district and aroma of newness further lies in the fact that here various MANN models are attempted to find out the best fit.

## II. LITERATURE SURVEY

Hu(1964) initiated the implementation of ANN, an important soft computing methodology in weather forecasting. Since the last few decades, ANN a voluminous development in the application field of ANN has opened up new avenues to the forecasting task involving environment related phenomenon (Gardener and Dorling, 1998; Hsiesh and Tang, 1998). Michaelides et al (1995) compared the performance of ANN with multiple linear regressions in estimating missing rainfall data over Cyprus. Kalogirou et al (1997) implemented ANN to reconstruct the rainfall over the time series over Cyprus. Lee et al (1998) applied ANN in rainfall prediction by splitting the available data into homogenous subpopulations. Wong et al (1999) constructed fuzzy rules bases with the aid of SOM and back-propagation neural networks and then with the help of the rule base developed predictive model for rainfall over Switzerland using spatial interpolation. Toth et al. (2000) compared short-time rainfall prediction models for real-time flood forecasting. Different structures of auto-regressive moving average (ARMA) models, ANN and nearest-neighbors approaches were applied for forecasting storm rainfall

occurring in the Sieve River basin, Italy, in the period 1992-1996 with lead times varying from 1 to 6 h. The ANN adaptive calibration application proved to be stable for lead times longer than 3 hours, but inadequate for reproducing low rainfall events. Koizumi (1999) employed an ANN model using radar, satellite and weather-station data together with numerical products generated by the Japan Meteorological Agency (JMA) Asian Spectral Model and the model was trained using 1-year data. It was found that the ANN skills were better than the persistence forecast (after 3 h), the linear regression forecasts, and the numerical model precipitation prediction. As the ANN model was trained with only 1 year data, the results were limited. The author believed that the performance of the neural network would be improved when more training data became available. It is still unclear to what extent each predictor contributed to the forecast and to what extent recent observations might improve the forecast.

Abraham et al. (2001) used an ANN with scaled conjugate gradient algorithm (ANN-SCGA) and evolving fuzzy neural network (EfuNN) for predicting the rainfall time series. In the study, monthly rainfall was used as input data for training model. The authors analyzed 87 years of rainfall data in Kerala, a state in the southern part of the Indian Peninsula. The empirical results showed that neuro-fuzzy systems were efficient in terms of having better performance time and lower error rates 5 compared to the pure neural network approach. Nevertheless, rainfall is one of the 20 most complex and difficult elements of the hydrology cycle to understand and to model due to the tremendous range of variation over a wide range of scales both in space and time (French et al., 1992).

### III. ARTIFICIAL NEURAL NETWORK

A neural network is a computational structure inspired by the study of biological neural processing. There are many different types of neural networks, from relatively simple to very complex, just as there are many theories on how biological neural processing takes place.

#### A) Feed Forward Network

A layered feed forward neural network has layers, or subgroups of processing elements. A layer of processing elements makes independent computations on data that it receives and passes the result to another layer. The next layer may in turn make its independent computations and pass on the result to yet another layer. Finally, a subgroup of one or more processing elements determines the output of the network. Each processing element makes its computation based upon a weighted sum of its inputs. The first layer in the input layer and the last layer is the output layer. The layers that are in between these two layers are the hidden layers. The processing elements are seen units that are similar to neurons working in the brain, and hence, they are referred to as cells, neuromines, or artificial neurons. A threshold function is sometimes used to qualify the output of a neuron in the output layer. Even though our subject matter deals with artificial neurons, we will simplify them as neurons. Synapses between

neurons are referred to as connections, which are represented by edges of a directed graph in which the nodes are the artificial neurons.

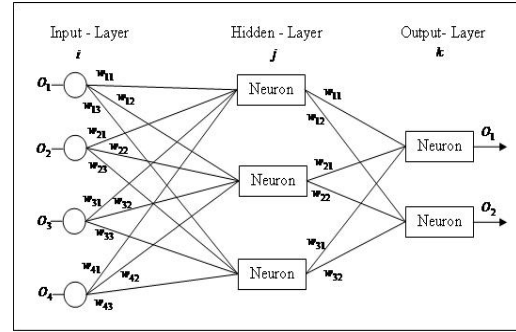


Fig.1—Layout of feed forward neural network

### IV. EXPERIMENTAL SETUP

The model building process consists of four sequential steps:

1. Selection of the input and the output data for the supervised BP learning.
2. Normalization of the input and the output data.
3. Training of the normalized data using BP learning.
4. Testing the goodness of fit of the model.
5. Comparing the predicted output with the desired output.

#### A) Selection Of The Input And The Output Data

In Udupi, Karnataka, the months of April to November are identified as the rainfall season with May, June, July, August, and October as the main monsoon seasons. Thus the present study explores the data of these 8 months from 1960 to 2010. There will be 400 entries in the output and the input files. The Input parameters are the average Humidity and the average Wind Speed for the 8 months of 50 years from 1960-2010 making it a 2\*400 matrix. In the output parameter is average rainfall in the 8 months of every year from 1960 -2010. The input file consists of 2 rows and 400 columns while the output file consists of 1\*400 matrix. The data stated above was retrieved from [www.Indiastat.com](http://www.Indiastat.com) and the IMD website. The unknown values were randomized keeping in mind the average value of the data.

#### B) Normalization of Data:

The input and the output data obtained have to be normalized because they are of different units and otherwise there will be no correlation between the input and the output values. First the mean of all the data separately were taken for humidity, wind speed and rainfall. Let the mean be M.

$$M = \text{sum of all entries} / \text{number of entries}$$

Then the standard deviation, SD, for each of these parameters individually were calculated.

Now after having the values of mean and SD for every parameter, the values for each parameter were normalized

$$\text{Normalized value} = (x - M) / SD$$

### C) Training of input Data-

After obtaining the normalized data, the next step is to train the input data using matlab Back-propagation Algorithm(BPA). The proposed ANN model is basically a three layered ANN back propagation learning. The algorithm takes only 70 percent of the input data for training. So out of 400 samples only 280 are taken for training and these are selected randomly from the set of data. For every attempt of training the data, the algorithm selects the training sample randomly from the whole set and not a fixed set of data and so every time you train the data, we get different values of mean square error (MSE) depending upon which 70 percent of the input data is chosen for training. Rest 60 samples are kept for validation and the remaining 60 samples are kept for testing.

### D) Testing and Validation

Testing is done after the training of the data is complete and the error is below the tolerance levels. The BPA keeps 30 % of the input data for testing and validation. So out of 400 samples, 60 are used in testing and another 60 are used for validation.

### E) Comparison Of Actual Data And Predicted Data

After the testing is done, the results are saved in the workspace and a graph is plotted between the actual output and the predicted output so that a comparison can be made. The graph is an efficient way of comparing the two types of data available with us. It can also be used to calculate the accuracy of the model.

In this paper when a graph was plotted between the actual and the predicted values , it showed high degree of similarity between them hence proving that our ANN model is quite accurate in prediction.

The following is an example of the the graph that is plotted after the testing and validation part is over

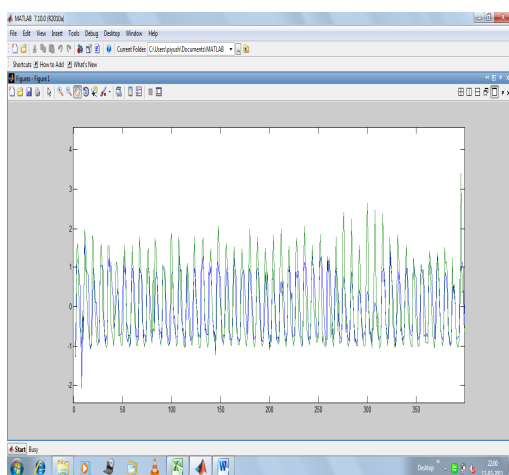


Fig. 2- Snapshot Of a comparison graph between the actual data and the predicted data

The experiments were performed on the following Neural Network :

- Feed Forward with Back –Propagation
- Layer Recurrent
- Cascaded feed Forward back Propagation

## V. RESULT AND DISCUSSION

There are two tools for implementing the algorithms in matlab. They are-

- Nntool – open network/data manager.** The single layer and the multi layer algorithms are implemented in the *nntool- open network/ data manager*.
- Nftool – Neural network fitting tool.** Only back propagation algorithm is implemented in this matlab tool.

Back Propagation Algorithm (BPA) was implemented in the nftools and a minimum MSE was obtained and a graph was plotted between the predicted values and the target values.

The following are the values recorded using the nftools→  
MSE = 3.6456

The regression can be plotted as follows→

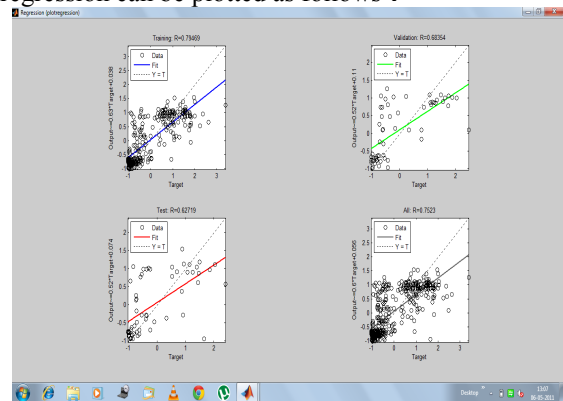


Fig. 3 Regression plot in nftools of Back-Propagation Algorithm (Single Layer)

The performance can be plotted as→

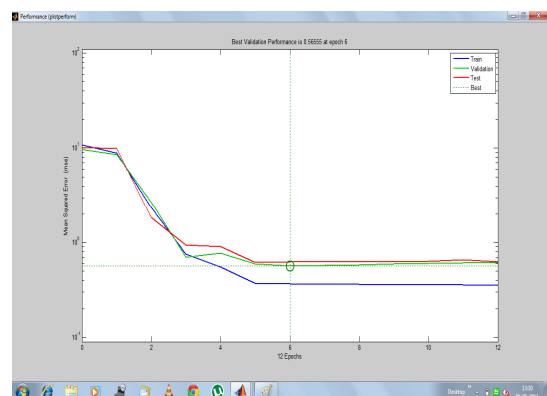


Fig. 4 Performance of Back-Propagation Algorithm (Single Layer)

The predicted values by BPA in nftool can be compared with the actual values in the following graph→

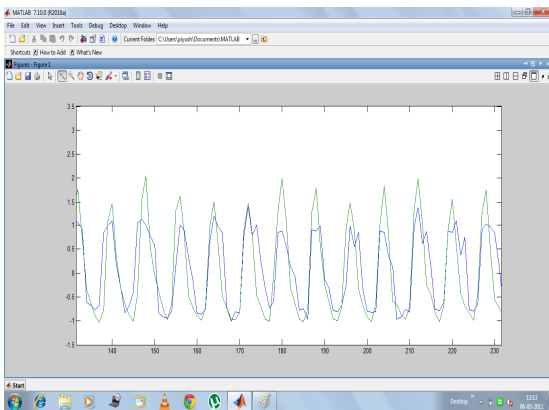


Fig 5. Back Propagation Algorithm results with target results (single-layer)

The results were fairly good and high degree of accuracy was obtained by observing the graph above. The MSE was also within the tolerance levels and the BPA algorithm turned out to be a great success.

The implementation of multi-layer architecture was done using NNTOOL in MTALAB

Three algorithm were tested in multi-layer architecture:

1. Back Propagation Algorithm (BPA)
2. Layer Recurrent Network(LRN)
3. Cascaded Back-Propagation (CBP)

The BPA implemented in the nntools showed great consistency and accuracy with the target data. The following table gives the results of different cases of BPA as observed. For multilayer architecture three hidden were used along with one input and one output layer and 10-20 neurons per layer were deployed to be used in experimental for Back-Propagation, Layer recurrent and Cascaded Back propagation. First Nntool was used for testing Back-Propagation Algorithm with the sample data and following observation were made:

TABLE-1 CASES FOR BPA

S.No.	Training function	Adaptive Learning Function	No. Of Neurons	Mean Square Error(MSE)
Case1	TRAINLM	Learngdm	10	0.47
Case2	TRAINLM	Learngd	10	0.52
Case3	TRAINLM	Learngdm	20	0.44
Case 4	TRAINLM	Learngd	20	0.48
Case 5	TRAINRP	Learngdm	10	0.46
Case 6	TRAINRP	Learngd	10	0.57
Case 7	<b>TRAINRP</b>	Learngdm	20	<b>0.42</b>
Case 8	TRAINRP	Learngd	20	0.46

Case 7 in the table above has the minimum MSE and so is the best case of BPA. Its performance can be plotted as follows→

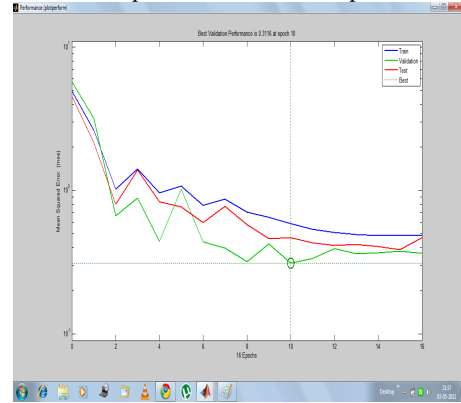


Fig 6 Back Propagation Algorithm case 7 performance

The graph is plotted between MSE and epochs. Train, validation and the test parameters are plotted against the best case. The figure clearly shows that the best validation check occurred at epoch 10.

The final result can be plotted in the following figure→

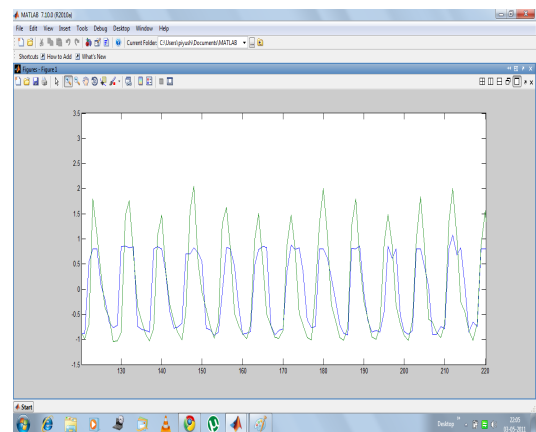


Fig 7 Back Propagation Algorithm case 7 final results with target results.

The figure above clearly shows high level of accuracy and precision. Also the predicted values follow the same trend as that of the target values and the deviation or fluctuations in the graph is the least.

Hence BPA is the best algorithm for training data in the NNTOOLS.

The results of various cases of LRN are shown below. All the cases are similar to BPA in training function used and the number of neurons tested with except the MSE which has been a bit different than that of BPA.

TABLE II LRN CASES

S.No.	Training function	Adaptive Learning Function	No. Of Neurons	Mean Square Error(MSE)
Case1	<b>TRAINLM</b>	Learngdm	<b>10</b>	<b>0.44</b>
Case2	TRAINLM	Learngd	10	0.45
Case3	TRAINLM	Learngdm	20	2.04
Case4	TRAINLM	Learngd	20	0.49
Case5	TRAINRP	Learngdm	10	1.48
Case6	TRAINRP	Learngd	10	1.21
Case7	TRAINRP	Learngdm	20	0.47
Case8	TRAINRP	Learngd	20	1.30

Clearly the best case in the above figure is case 1 with the least MSE. Its performance can be plotted as follows→

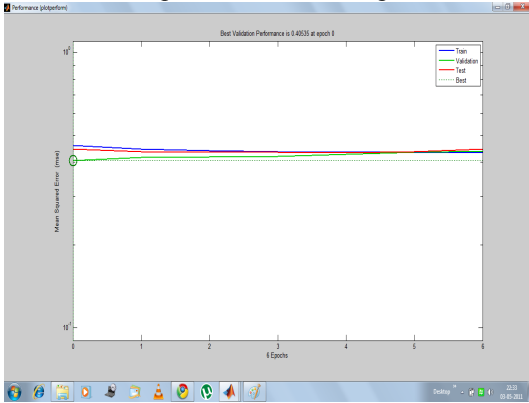


Fig 8 Layer Recurrent case 1 performance

This figure shows the best case of LRN. The best validation check occurs at epoch 0.

The final results of case 1 can be plotted as follows→

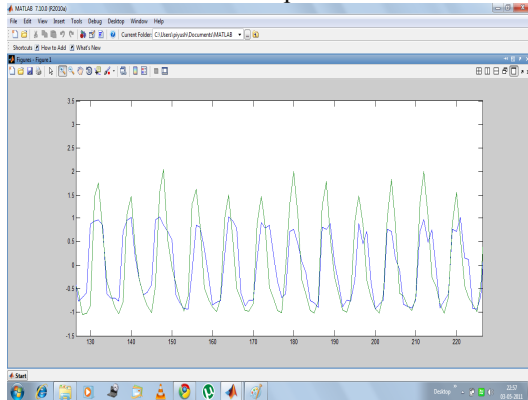


Fig 9 LRN case 1 final result with target result

This algorithm also shows high level of accuracy but in some cases the MSE was quite high as compared to BPA. The different cases of the CBP algorithm have been listed in the given table.

TABLE III CBP CASES

S.No.	Training function	Adaptive Learning Function	No. Of Neurons	Mean Square Error(MSE)
Case 1	TRAINLM	Learngdm	10	0.50
Case 2	TRAINLM	Learngd	10	0.52
<b>Case 3</b>	<b>TRAINLM</b>	<b>Learngdm</b>	<b>20</b>	<b>0.47</b>
Case 4	TRAINLM	Learngd	20	0.61
Case 5	TRAINRP	Learngdm	10	0.54
Case 6	TRAINRP	Learngd	10	1.47
Case 7	TRAINRP	Learngdm	20	0.50
Case 8	TRAINRP	Learngd	20	1.52

The best case of CBP algorithm is case 3 with the least MSE. Its performance is plotted as follows→

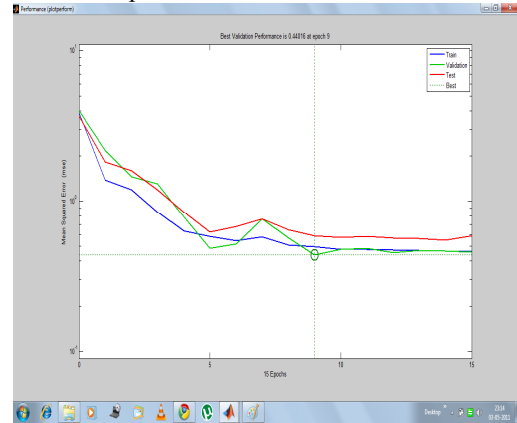


Fig 10 Cascaded Backpropagation Algorithm case 3 performance

The best validation check is at epoch 9. The final results can be plotted as follows→

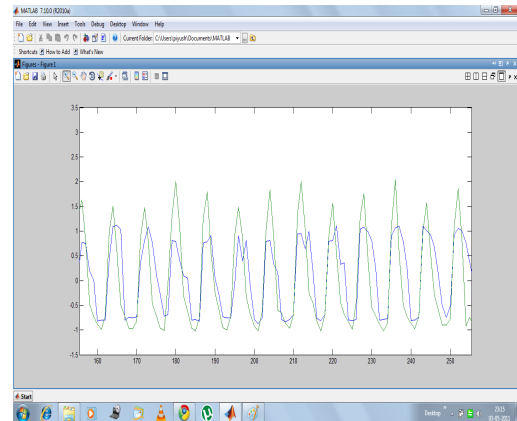


Fig 11 Cascaded Backpropagation Algorithm case 3 final results with target results

The CBP algorithm showed high MSE in almost every case as compared to BPA or LRN.

Also, its results were deviating from the fact that as number of neurons increases, the MSE decreases.

## VI. CONCLUSION

From the experimental study the following observation were made :

- As the number of neurons increases in an ANN, the MSE decreases.
- BPA is the best algorithm out of the three tested.
- LEARNGDM is the best learning function to train your data with.
- LEARNGD is a bit time consuming.
- TRAINLM is the best training function.
- Multi-layer Alogorithm is better than Single-layer algorithm terms of performance.
- NNTTOOLS should be used to implement the prediction algorithms as a it gives an option of implementing algorithms other than BPA.
- Larger the amount of input data, lower is the MSE after training.
- The input /output data should be normalized if they are of very high order.

## REFERENCES

- [1] M.J.C., Hu, Application of ADALINE system to weather forecasting, Technical Report, Stanford Electron, 1964
- [2] Michaelides, S. C., Neocleous, C. C. & Schizas, C. N. "Artificial neural networks and multiple linear regression in estimating missing rainfall data." In: Proceedings of the DSP95 International Conference on Digital Signal Processing, Limassol, Cyprus. pp 668–673 1995.
- [3] Kalogirou, S. A., Neocleous, C., Constantinos, C. N., Michaelides, S. C. & Schizas, C. N., "A time series construction of precipitation records using artificial neural networks. In: Proceedings of EUFIT '97 Conference, 8–11 September, Aachen, Germany. pp 2409–2413 1997.
- [4] Lee, S., Cho, S. & Wong, P.M., "Rainfall prediction using artificial neural network," J. Geog. Inf. Decision Anal. 2, 233–242 1998.
- [5] Wong, K. W., Wong, P. M., Gedeon, T. D. & Fung, C. C. , "Rainfall Prediction Using Neural Fuzzy Technique." 1999
- [6] E.Toth, A.Brath, A.Montanari, "Comparison of short-term rainfall prediction models for real-time flood forecasting", Journal of Hydrology 239 (2000) 132–147
- [7] Koizumi, K.: "An objective method to modify numerical model forecasts with newly given weather data using an artificial neural network", Weather Forecast., 14, 109–118, 1999.
- [8] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi "An Artificial Neural network Model for rainfall Forercastingin Bangkok,Thailand", Hydrol. Earth Syst. Sci., 13, 1413–1425, 2009
- [9] Kyaw Kyaw Htike and Othman O. Khalifa, "Research paper on ANN model using focused time delay learning", International Conference on Computer and Communication Engineering (ICCCE 2010), 11-13 May 2010, Kuala Lumpur.
- [10] Ajith Abraham, Dan Steinberg and Ninan Sajeeth Philip, "Rainfall Forecasting Using Soft Computing Models and Multivariate Adaptive Regression Splines", 2001.
- [11] Rainfall data from <http://www.Indiastat.com/karnataka/rainfall>
- [12] Ben Krose and Patrick van der Smagt , "An introduction to neural networks", Eighth edition, November 1996.
- [13] Dr S. Santosh Baboo and I. Khadar Shareef, "An efficient Weather Forecasting Model using Artificial Neural Network", International Journal of Environmental Science and Development, Vol. 1, No. 4, October 2010.
- [14] Enireddy Vamsidhar et. al., "Prediction of rainfall Using Back-propagation Neural Network Model", International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1119-1121
- [15] Paras, Sanjay Mathur, Avinash Kumar, and Mahesh Chandra, "A feature based on weather prediction using ANN" World Academy of Science, Engineering and Technology 34 2007