A Connectionist Model for Prediction of Humidity Using Artificial Neural Network

Bimal Dutta[1], Susanta Mitra[2]
[1]Hooghly Engineering & Technology College
Vivekanada Road, P.O. Hooghly
West Bengal - 712103, India
[2]Meghnad Saha Institute of Technology (Techno India Group)
Nazirabad, P.O
East Kolkata - 700107, India
bkd_hetc@yahoo.co.in, susantamit@gmail.com

ABSTRACT: *Prediction of humidity is one important and challenging task that needs lot of attention and study for analyzing atmospheric onditions, specially the warm weather. Advent of digital computers and development of data driven artificial intelligence approaches like Artificial Neural Networks (ANN) have helped in numerical prediction of humidity. However, very few works have been done till now in this area. The present study developed an ANN model based on the past observations of several meteorological parameters like temperature, humidity, atmospheric pressure and vapour pressure as an input for training the model. The novel architecture of the proposed model contains several multilayer perceptron network (MLP) to realize better performance. The model is enriched by analysis of several alternative models like online feature selection MLP (FSMLP) and self organizing feature map MLP (SOFM-MLP).The improvement of the performance in the prediction accuracy has been demonstrated by the selection of the appropriate features. The FSMLP is used as preprocessor to select good features. The results obtained after applying the model indicate that it is well suitable for humidity as well as warm weather prediction over large geographical areas.*

Keywords: Artificial Neural Networks, Multi-layer Perceptron, Backpropagation, Feature Selection

Received: 12 March 2011, Revised 18 April 2011, Accepted 21 April 2011

1. Introduction

Accurate information about weather and proper prediction of humidity is often useful for warning about natural disasters caused by abrupt change in climatic conditions. The complexity of the atmospheric processes makes quantitative forecasting of humidity an extremely difficult and challenging task. Presently, weather predictions are made by collecting quantitative data about the current state of the atmosphere and using scientific understanding of the atmospheric processes to project how the atmosphere will evolve.

Artificial Neural Networks (ANN) perform nonlinear mapping between the inputs and outputs without detailed consideration of he internal structure of the physical processes associated with humidity. This approach is essentially a data driven approach. ANN emulates the parallel distributed processing of the human nervous system and are parallel computational models, comprising closely interconnected adaptive processing units. The adaptive nature of neural networks adopts artificial intelligence (AI) learning techniques like supervised and unsupervised learning. ANN model has already proved to be very powerful in dealing with complex problems like function approximation, pattern recognition and has been applied for weather prediction, stock market prediction etc.

A number of studies which have used ANN to model complex nonlinear relation of input and output for weather forecasting [6][7][11], have been reported. However, very few works have used ANN-based connectionist methods to forecast humidity. Again, all of these works are restricted to feed forward ANN models with back propagation and that uses either linear or nonlinear time series data only.

This paper is an outcome of an ANN based humidity prediction model developed, trained and tested with continuous (daily) humidity data as input over a period of 7 years [1]. Here three distinct alternative models, namely MLP, FSMLP and SOFM-MLP have been studied and analyzed. However, mainly 2 models, MLP and FSMLP were designed and tested separately with different number of hidden nodes. The results produced by each of the models were compared and the suitability of the models justified. The model has been applied to justify that ANN is an appropriate predictor for humidity forecasting in Kolkata (Latitude = 22° 34' 10" N, Longitude = 88° 22' 10" E) as well as eastern part of India. The prediction is based on the past observations of several meteorological parameters like temperature, humidity, air pressure and vapour pressure. The data was collected daily by the meteorological department from the rain gauge stations at two different times of the day.

2. ANN Approach

In this section the basics of the 3 models as referred in Section 1 are discussed. This theoretical basis of the models has been applied during the design and implementation of the same.

2.1 Multilayer Perceptron (MLP)

MLP is one of the most widely used neural network architectures. It consists of several layers of neurons of which the first layer is known as the *input layer*, last layer is known as the *output layer* and the remaining layers are called as *hidden layer*. Every node in the hidden layers and the output layer computes the weighted sum of its inputs and apply a sigmoidal activation function to compute its output, which is then transmitted to the nodes of the next layer as input [3]. The main objective of MLP learning is to set the connection weights in such a way the error between the network output and the target output is minimized. A typical MLP network is shown in Figure 1. According to [2] under a fairly general assumption a single hidden layer is sufficient for multilayer perceptron to compute an uniform approximation of a given training set of input and output. So, the present study is restricted to three-layer network i.e. one hidden layer.

2.2 Clustering

Cluster analysis or clustering is the assignment of a set of observations into subsets (called *clusters*) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

2.2.1 Self Organizing Feature Map (SOFM)

A Self-Organizing Feature Map (SOFM) or Self-Organizing Map (SOM) is a neural network approach that uses competitive unsupervised learning scheme. Learning is based on the concept that the behavior of a node should impact only those nodes and arcs near it. Weights are initially assigned randomly and adjusted during the learning process to produce better results. During this learning process, hidden features or patterns in the data are uncovered and the weights are adjusted accordingly. The term selforganizing indicates the ability of these neural networks to organize the nodes into clusters based on the similarity between them. Those nodes that are closer together are more similar than those that are far apart. A well known type of SOFM is the Kohonen self-organizing map [3].

Architecture: The self-organizing feature map is an algorithmic transformation ADSOFM : $\Re p \rightarrow V(\Re q)$ that is often advocated for visualization of metric-topological relationships and distributional density properties of feature vectors (signals) $X = \{x_1, \ldots, x_N\}$ in $\Re p$. SOFM is implemented through a neurallike network architecture as shown in Figure 2 and it is believed to be similar in some ways to the biological neural network. The visual display produced by ADSOFM helps to form hypotheses about topological structure present in X. Although, here we concentrate on (m x n) displays in $\Re 2$, in principle X can be transformed onto a display lattice in $\Re q$ for any q.

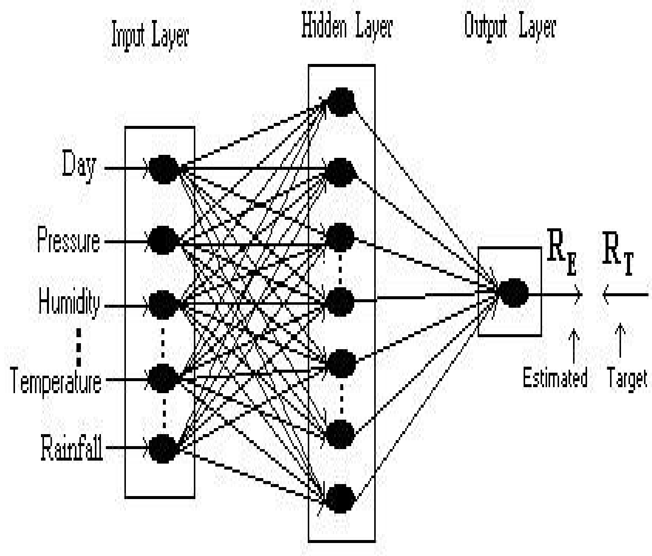

Figure 1. A typical MLP for rainfall prediction

As shown in Figure 2, input vectors $\mathbf{x} \in \Re^p$ are distributed by a fan-out layer to each of the $(m \times n)$ output nodes in the competitive layer. Each node in this layer has a weight vector (prototype) $\mathbf{v}_{ij}$ attached to it. Let $O_p = \{\mathbf{v}_{ij}\} \subset \Re^p$ denote the set of $m \times n$ weight vectors. O_p is (logically) connected to a display grid $O_2 \subset V(\Re^2)$. (i, j) in the index set $\{1,2, \ldots, m\} \times$ e $\{1,2, \ldots, n\}$ is the logical address of the cell. There is a one-to-one correspondence between the $m \times n$ p-vectors $\mathbf{v}_{ij}$ and the $m \times n$ cells $(\{i, j\})$, i.e., $O_p \leftarrow O_2$.

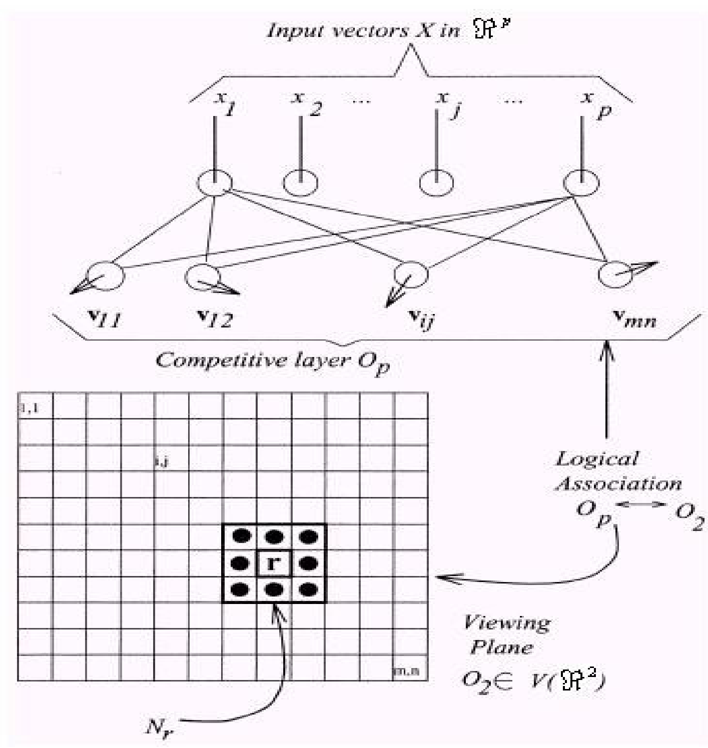

Figure 2. The SOFM network architecture

2.2.2 Online Feature Selection

An online feature selection network selects the good features while learning the estimation task. The performance of the network can further be improved and this can also tell one about various important features respond for this prediction. The technique for online feature selection is discussed below.

In a standard multilayer perceptron network, the effect of some features (inputs) can be eliminated by not allowing them into the network. The "partially useful" features can be identified and attenuated according to their relative usefulness [4] [5] [8]. This can be realized by associating an adaptive gate to each input node. The gate should be modeled in such a manner that for a good feature, it is completely opened and the feature is passed unattenuated into the net; while for a bad feature, the gate should be closed tightly. On the other hand for a partially important feature, the gate could be partially open. Mathematically, the gate is modeled by a function F with a tunable parameter. The degree to which the gate is opened determines the goodness of the feature. The input feature value is multiplied by its attenuation value and the modulated feature value is passed into the network. The gate functions attenuate the features before they propagate through the net so we may call these gate functions as attenuator functions. A simple way of identifying useful gate Functions is to use s-type (or sigmoidal) functions with a tunable parameter which can be learnt using training data.

Let $F: R \rightarrow [0,1]$ be an attenuation function associated with each of the p input nodes. If x is the node input then $xF(\gamma)$ is the node output. Thus, $xF(\gamma_i)$ can be viewed as the activation function of the i-th input layer node, where γ_i is a parameter (not a connection weight) of the activation function. Thus, the input layer nodes act as "neurons" (i.e., have internal calculations). Notice that $F(\gamma_i)$ acts as a *fixed* multiplier for *all* input values of the i-th feature once γ_i is known. The function F can have various forms. In the experiments described below the attenuation function used is

$$F(\gamma) = \frac{1}{1+e^{-\gamma}}$$

Thus, the i-th input node attenuates x_i by an amount $F(\gamma_i) \in (0, 1)$, where the input "weight" γ_i is a parameter to be learnt during training. If $F(\gamma_i)$ is close to 0, input feature x_i may be chosen to be eliminated: this is how the FSMLP accomplishes feature selection. How do we learn γ_i? If the input layer of the FSMLP with nodes is regarded as in Figure 1, it can be viewed as the "first"' hidden layer in a standard MLP. Then the back-propagation formulae for the MLP are simply extended backwards into this new first layer to adjust the p γ_is during training. Let

- q = number of nodes in the first hidden (notinput) layer;
- μ = learning rate for the parameters of the attenuator membership functions;
- η = learning rate for the connection weights;
- $w_{ih}^{ij}(t)$ = weight connecting i-th node of the input layer to the j-th node of the first hidden layer for the t-th iteration;
- δ_j^1 =error term for the j-th node of the first hidden layer.
- $F'(\gamma_i)$ = derivative of F at γ_i;
- $F: R \rightarrow (0,1)$ = attenuator function with argument γ_i for input node i;

It can be easily shown that the learning rule for connection weights remains the same for all layers except for) $w_{ih}^{ij}(t)$. The update rule for) $w_{ih}^{ij}(t)$ and γ_i are:

$$w_{ij}^{ih}(t) = w_{ij}^{ih}(t-1) + \eta x_i \delta_j^1 F(\gamma_i(t-1))$$

$$\gamma_i(t) = \gamma_i(t-1) + \mu x_i \left(\sum_{j=1}^{q} w_{ij}^{h} \delta_j^1 F'(\gamma_i(t-1)) \right)$$

The p weights γ_i are initialized with values that make $F(\gamma_i) = \frac{1}{1+e^{-\gamma}}$ close to 0 for all i.

Consequently, $x_i F(\gamma_i)$is small at the beginning of training, so the FSMLP *allows only a very small "fraction" of each input feature value to pass into the standard part of the MLP*. As the network trains, it selectively allows only important features to be active by increasing their attenuator weights (and hence, increasing the multipliers of x_i associated with these weights) as dictated by the gradient descent. The training can be stopped when the network has learnt satisfactorily, i.e., the mean squared error is low or the number of iteration reaches a maximum limit. Features with *low* attenuator weights are eliminated from the feature set.

3. Datasets And Experiments

The present study developed an ANN model based on the past observations of several meteorological parameters like temperature, humidity, air pressure and vapour pressure as an input for training the model. The developed model overcomes the difficulties in training ANN models with continuous data. The architecture of the proposed model contains several multilayer perceptron network (MLP). The model is enriched by analysis of several alternative models like online feature selection MLP (FSMLP) and self organizing feature map MLP (SOFM-MLP) for better prediction. FSMLP and SOFMMLP can produce good predictions with fewer inputs.

The experiments were carried out in the following sequence. First, the effectiveness of multilayer perceptron networks was studied for prediction of humidity. Next, in FSMLP model some good features were selected online while producing good prediction.

3.1 Data Acquisition

The meteorological data captured through devices are compiled in a file. This input file has day wise record of parameters like Minimum Temperature, Maximum Temperature, Relative Humidity, Minimum Air Pressure, Maximum Air Pressure, Minimum Vapour Pressure, Maximum Vapour Pressure, and Date. The file contains data of seven years. So, there is an observation of 8 variables on a particular day, say *t*. In the MLP model the humidity for the *t* th day is determined by the atmospheric parameters for the past 2 days i.e. days *(t-1)*, *(t-2)*. To enable the selection of the best model, the training data set should cover the high, medium and low humidity periods. So the data for entire years were chosen as the training data sets. The data is preprocessed before training. The detail of the pre-processing is discussed in the next section.

3.2 Data Preprocessing

Neural network training can be made more efficient if certain preprocessing steps are performed on the network inputs and targets. Before training, it is often useful to scale the inputs and targets so that they always fall within a specified range. The input metrics were *normalized* using min-max normalization. Min-max normalization performs a linear transformation on the original data. Suppose that $\min_A$ and $\max_A$ are the minimum and maximum values of an attribute A. It maps value v of A to v' in the range -1 to 1 using the formula below :

$$\mathbf{v'} = \frac{\mathbf{v - min_A}}{\mathbf{max_A - min_A}}$$

A function in MATLAB is used to scale inputs and targets so that they fall in the range [-1,1]. The trained network is simulated with the normalized input, and then the network output is converted back into the original units. The weights and biases are initialized before training using suitable commands of MATLAB.

3.3 Methodologies

The data was organized for training and testing. The records of the input file were partitioned into two separate files. One input file had first 80% of the total records and was used for network training. This is considered as 'training data set'. The other input file contain 20% of the remaining records was considered as 'test data set'. This was used for testing the network for unknown outputs once the network is trained. The resulting output predictions $y_j(t)$ are compared with a corresponding desired or actual output, $d_j(t)$. The mean squared error at any time *t*, $E(t)$, is calculated using the formula

$$MSE(t) = \tfrac{1}{2} \Sigma (y_j(t) - d_j(t))^2 \text{ for j = 1 to n.}$$

For the MLP model, transfer function is the well known sigmoid function. There was a single hidden layer and several runs of MLP were made with different number of hidden nodes in the hidden layer. The number of nodes was 5, 10, 15 and 20.

In case of SOFM-MLP model, training data was partitioned into homogenous subgroup using self organizing feature map (SOFM) and applied on the trained network.

For FSMLP model, *online feature selection* was done by selecting the good features (inputs) that will improve the rainfall prediction while learning the estimation task. In other words, some inputs can be eliminated by not allowing them into network. During the initial part of the training FSMLP allows only a very small "fraction" of each input feature value to pass into the standard part of the MLP. As the network trains, it selectively allows only important features to be active by increasing their attenuator weights as dictated by the gradient descent. Features with low attenuator weights were eliminated from the feature set.

4. Results And Observations

After the input file is prepared, the training is done taking into consideration all the parameters .Thus the humidity, **H(t)** of tth day is the function of the following parameters.

(D(t- 2), Min Temp(t-2), Max Temp(t-2), Min Vpr Prs (t-2), Max Vpr Prs(t-2), Max Prs(t-2), Min Prs(t- 2), Rel Hmdity(t-2), D(t- 1), Min Temp(t-1), Max Temp(t-1), Min Vpr Prs (t-1), Max Vpr Prs(t-1), Max Prs(t-1), Min Prs(t- 1), Rel Hmdity(t-1)

In functional form, **H(t)** can be defined as

H(t) = ***f*** ((D(t- 2), Min Temp(t-2), Max Temp(t-2), Min Vpr Prs (t-2), Max Vpr Prs(t-2), Max Prs(t-2), Min Prs(t- 2), Rel Hmdity(t-2),D(t- 1), Min Temp(t-1), Max Temp(t-1), Min Vpr Prs (t-1), Max Vpr Prs(t-1), Max Prs(t-1), Min Prs(t- 1), Rel Hmdity(t-1)).

Input Parameters	**Exp1**	**Exp2**	**Exp3**	**Exp4**	**Frequency**	**Decision**
D(t- 2)	0.27	0.22	0.32	0.31	4	**Rejected**
Min Temp(t-2)	0.35	0.35	0.37	0.36	4	**Rejected**
Max Temp(t-2)	0.50	0.51	0.54	0.53	4	Selected
Min Vpr Prs (t-2)	0.55	0.565	0.53	0.56	4	Selected
Max Vpr Prs(t-2)	0.24	0.35	0.29	0.28	4	**Rejected**
Max Prs(t-2)	0.58	0.50	0.50	0.50	4	Selected
Min Prs(t- 2)	0.57	0.52	0.52	0.31	3	**Rejected**
Rel Hmdity(t-2)	0.56	0.56	0.50	0.47	3	Selected
D(t- 1)	0.57	0.58	0.51	0.50	3	Selected
Min Temp(t-1)	0.58	0.59	0.47	0.56	4	Selected
Max Temp(t-1)	0.34	0.21	0.29	0.28	3	**Rejected**
Min Vpr Prs (t-1)	0.59	0.51	0.51	0.51	4	Selected
Max Vpr Prs(t-1)	0.34	0.20	0.30	0.29	4	Selected
Max Prs(t-1)	0.56	0.57	0.59	0.48	3	Selected
Min Prs(t- 1)	0.57	0.58	0.51	0.51	4	Selected
Rel Hmdity(t-1)	0.58	0.49	0.57	0.56	3	Selected

Table1. Humidity Estimation Using MLP

After training the testing is done. The result is shown in Table 1.

The Table 1 indicates that the result is not good. The graphical representation of the target and computed humidity is shown in the following Figure 3.

The feature selection technique was used to remove some irrelevant features. All the input parameters are taken as input to the feature selection method. The following table (Table 2) shows the result of feature selection technique.

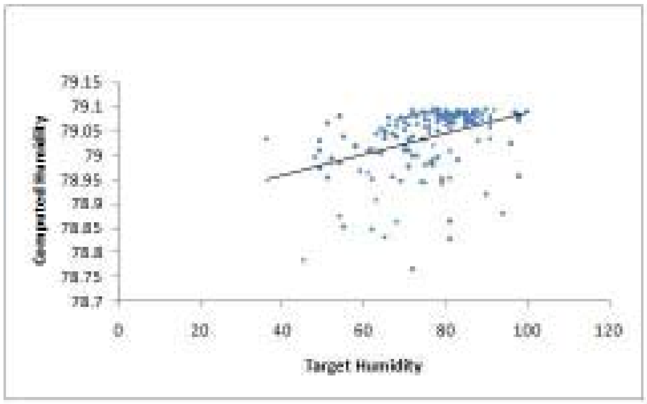

Figure 3. Graphical representation of target and computed Humidity

Target Humidity	Computed Humidity	Absolute Error
76	79.0703	3.0703
79	79.0762	0.0762
91	79.0743	11.9257
89	79.0736	9.9264
86	79.0711	6.9289
85	79.0722	5.9278
98	79.0724	18.9276
87	79.0666	7.9334
78	79.0685	1.0685
86	79.0677	6.9323
81	79.0663	1.9337
75	79.0747	4.0747
82	79.0662	2.9338
81	79.0693	1.9307
73	79.0787	6.0787
63	79.043	16.043
58	79.0177	21.0177
62	79.012	17.012
76	79.0568	3.0568

Table 2. Feature Selection By Voting Scheme

The above table shows that the irrelevant features are rejected on the basis of voting scheme. The selected features are Relative Humidity (t-2), Maximum Pressure (t-2), Minimum Vapor Pressure (t-2),Maximum Temperature (t-2), Minimum Vapor Pressure(t-1), Maximum Vapor Pressure(t-1), Maximum Pressure (t-1), Minimum Pressure (t- 1), Relative Humidity(t-1), Day (t-1), Minimum Temperature (t-1).

After feature selection the finial training and testing is done. The result is shown in table (Table 3). After feature selection technique the test result improves. The result is quite good. The graphical representation is shown in the following Figure 4.

So feature selection technique (FSMLP) can be used to increase the predicted result of neural network based prediction system. The accuracy level becomes high after incorporating feature selection technique.

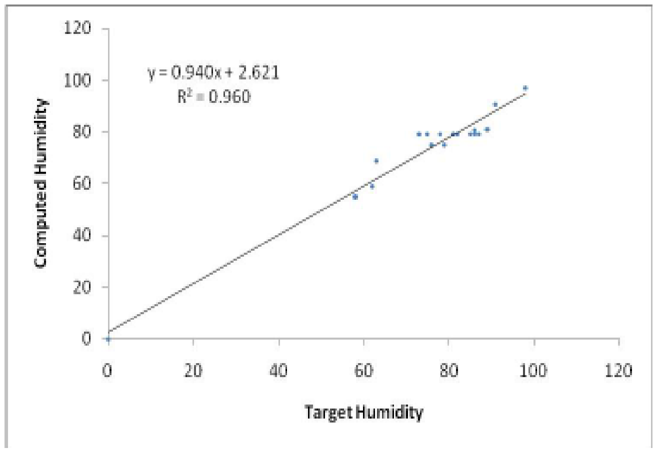

Figure 4. Graphical representation of target and computed Humidity after feature selection

Target Humidity	Computed Humidity	Absolute Error
76	75.0703	0.9297
79	75.0762	3.9238
91	90.743	0.257
89	81.0736	7.9264
86	80.711	5.289
85	79.0722	5.9278
98	97.0724	0.9276
87	79.0666	7.9334
78	79.0685	1.0685
86	79.0677	6.9323
81	79.0663	1.9337
75	79.0747	4.0747
82	79.0662	2.9338
81	79.0693	1.9307
81	79.0693	1.9307
73	79.0787	6.0787
63	69.043	6.043
58	55.0177	2.9823
62	59.012	2.988

Table 3. Humidity Estimation Using FSMLP

4.1 Observation

If a comparative study is done between Table 1 and Table 3, then it's clearly visible that in Table 1, the efficiency level of the neural network system was low. The average error in the Table 1 is 14.52%. So using feature selection, some irrelevant features are eliminated. By using significant features, the predicted result becomes quite good as shown in Table 3. The average error in Table 3 is 9.29%. The RMSE value before feature selection is **12.59**, where as the RMSE value after feature selection is **3.893869**.

5. Conclusions

Artificial neural network model discussed here has been developed to run humidity forecast for a day based on the previous days data. Humidity data from weather stations and the meteorological data from Kolkata Meteorological department were collected during 1989- 1995 to the ANN models. It consists of continuous data for the said period. Two alternative ANN models were tested with continuous humidity data and compared. Based on the testing of these models the following conclusions are made

- FSMLP turns out to be an excellent tool that can select good features while learning the prediction task.
- The neural network models proposed here can be good alternatives for traditional meteorological approaches for weather forecasting.

In the future works, the combined use of FSMLP and SOFM-MLP may result in an excellent paradigm for prediction of humidity. Moreover, FSMLP and SOFM-MLP set may be used for prediction of other atmospheric parameters.

References

[1] Dutta, B., Ray, A., Pal, S., Patranabis, D. C. (2009). A connectionist model for rainfall prediction. *Neural, Parallel and Scientific Computations,* 17, 47-58.

[2] Haykin, S. (2001). Neural Networks : A Comprehensive Foundation, Second Edition. Pearson Education, Singapore.

[3] Kohonen, T. (1990). The self-organizing map. *In Proc. of IEEE*, 78, p. 1464-1480.

[4] Pal, N. R., Chintalapudi, K. (1997). A connectionist system for feature selection. *Neural, Parallel and Scientific Computations* 5, 359-381.

[5] Pal, N. R., Pal, S., Das J., Majumdar, K. (2003). SOFMMLP : A hybrid neural network for atmospheric temperature prediction. *In: Proc. of IEEE Trans. Geoscience Remote Sening,*. 41, 2783-2791.

[6] Paras, Mathur, S., Kumar, A. , Chandra, M. (2007). A Feature Based Neural Network Model for Weather Forecasting. *World Academy of science, Engineering and Technology*, 34, 66-73.

[7] Raodknight, C. M., Balls, G. R., Mills, G. E., Palmer- Brown, D. (1997). Modeling Complex Environmental Data. *In: Proc. of IEEE Trans. Neural Networks*, 8 (4) 852-861.

[8] Sarma, D. K., Konwar, M., Das, J., Pal, S., Sharma, S. (2005). A soft computing approach for rainfall retrieval from the TRMM microwave imager. *In: Proc. of IEEE Trans. On Geoscience and Remote Sensing*, 43, 2879-2885.

[9] Renewable energy: RD & D priorities, IEA Publications.

[10] Kalogirou, S.A. (2001). Artificial neural networks in renewable energy systems applications: a review. *Renewable and Sustainable Energy Reviews* 5 (4) 373-401

[11] Datta, B., Pal, S., Roychoudhury, R. (2011). Estimation of Solar Radiation at a particular place :comparative study between Soft Computing and Statistical Approach. *International Journal on Computer and Engineering(IJCSE),Chennai*, 3, 3027-3036.

Author Biographies

Bimal Datta is a Professor and Head of Department of Computer Science & Engineering (CSE) at Hooghly Engineering and Technology College, West Bengal, India. He has submitted his Ph.D. (Engg.). He has obtained M.Tech from IIT, Kharagpur at West Bengal, India. He has served as Guest Professor in different reputed universities of West Bengal. He has an experience of about 20 years which covers both industries and academics. He has taught a number of important papers of Computer Science, both at B.Tech and M.Tech level. His current research interests are Artificial Neural Network, Wireless computing, Image Processing, Networking Fuzzy System. He has published more than 15 papers in renowned international journals and conferences. He is associated with a number of research projects. He also acted as Head Examiner and paper setter for different university examinations.

Susanta Mitra is a Professor and Head of Department of Computer Science & Engineering (CSE) and Information Technology (IT) at Meghnad Saha Institute of Technology of Techno India Group (TIG), Kolkata, India. He has received his Ph.D. (Computer Science) from Jadavpur University, Kolkata. He is a M.Tech in Information Technology. He has an experience of about 25 years which covers both industries and academics. He also has a research experience of about 10 years. He has taught few core papers of Computer Science, both at B.Tech and M.Tech level. His current research interests include Web data management and data mining, online social networking, software engineering including design patterns, artificial neural networks. He has published several papers in renowned international journals and conferences like IEEE, ACM and few book chapters in technical books. He is a member of the editorial boards of various international journals and also member of the advisory board of renowned international book published by Springer. He is a reviewer of various reputed international journals. He has also served as a programme committee member of different international conferences. He is a member of ACM, U.S.A. and Senior Life member of Computer Society of India (CSI).