Assessing Rainfall Prediction Model Using Artificial Neural Network

N. M. Refat Nasher[1] Md. Aziz Rahman[2] Dr. Morsheda Begum[3]

Abstract: ***Artificial Neural Network (ANN) models are developed to predict the rainfall in Dhaka city using Multilayer Prerceptrons (MLP) and Radial Basis Function (RBF) models. Forecasting rainfall is important for many management applications, in particular for flood warning systems. As rainfall is generated due to the atmospheric processes, the necessary data on the temporal and spatial scales are not feasible generally to forecast it using physically based process model. Recent developments in artificial intelligence provide an alternative approach of rainfall forecasting. This study offers a description and comparison of the main models of Artificial Neural Networks (ANN) which have proved to be useful in time series forecasting and also an application of standard procedure for forecasting rainfall data series. A comparative study establishes that the error made by the two neural network models Multilayer Prerceptrons (MLP) and Radial Basis Function (RBF) analyzed is less than a threshold value. The model with the best performance is the RBF compared to MLP.***

Key Words: Artificial Neural Network (ANN), Multilayer Perceptron (MLP), Radial Basis Funciton (RBF), rainfall forecasting.

Introduction

The weather plays a significance role in the agriculture of a country. Due to rapid growth of the density of the population and the development of industrialization, the management of water resources is increasingly in demand for a country. Thus, relevant and useful information of daily, weekly or monthly rainfall analysis could be provided to ensure that the water system management works efficiently. As the rainfall, which can be considered as one of the most important of weather factors in Bangladesh, is roughly variable from place to place and month to month and hence sometimes unpredictable.

From the point of view of the meteorological records, the significance of rainfall analysis has been highlighted by hydrological and climatologically studies because of its influence in all human activities such as agriculture, industrial and domestic [1]. Inference problem regarding variability of rainfall such as estimation

1 Assistant Professor, Department of Geography and Environment, Jagannath University, Email: refat@geography.jnu.ac.bd

2 Lecturer, Department of Statistics, Jahngirnagar University, Savar

3 Assistant Professor, Department of Statistics, Jagannath University.

and hypothesis testing, involving Markov Chain was considered by several authors [2],[3],[4].

Rahman and Mian (2002) introduced a rainfall simulation model based on first order Markov chain for annual variation in rainfall amount that is observed in Bangladesh. Although a lot of work on forecasting of rainfall using time series modeling (ARIMA) has been done, but there has been little attention on forecasting rainfall data using Artificial Neural Networks (ANN) approach. There have been a number of reported studies that have used ANNs to solve problems in meteorology [5]. For example, K. C. Luk et al.,(2001) used three types of ANNs (MLFN, PRNN ad TDNN) for forecasting rainfall near an urban catchment in Western Sydney, Australia, while Hsu et al., (1995) applied an ANN to model the rainfall runoff process [6],[7].

Rainfall forecasting is important for many catchment management applications particularly for flood warning system. The variability of rainfall in space and time however renders quantitative forecasting of rainfall extremely difficult. The depth of rainfall and its distribution in the temporal and spatial dimensions depends on many variables such as pressure, temperature and wind speed direction. Due to the complexity of the atmospheric process by which rainfall is generated and the lack of available data on the necessary temporal and spatial scales, it is not feasible generally to forecast rainfall using a physically based process model. There are two possible approaches to forecast rainfall [6]. The first approach involves the study of the rainfall process in order to model the underlying physical laws, However, this physically based process modeling approach may not be feasible because:

- Rainfall is an end product of a number of complex atmospheric process which vary both in space and time
- Even if the rainfall process can be described concisely and completely, the volume of calculations involve may be prohibitive and
- The data that is available to assist in definition of control variables for the process mode.

A second approach to forecasting rainfall is based on the pattern recognition methodology which attempts to recognize rainfall patterns based on their feature .

This study will reveal a comparison of the main ANN models in forecasting total monthly rainfall of Dhaka city. The models analyzed here are: the Multilayer Perceptron (MLP) and Radial Basis Function (RBF). The performance of the model evaluated by Mean absolute percentage errors along with robust Huber M estimator were used for the comparison of goodness of fit criterion.

MATERIALS AND METHODS

Data

For this study, monthly rainfall data for Dhaka city, the study area, was collected for a period of 52 years (1960-2012) from Bangladesh Meteorological

Development (BMD) station at Agargaon, Dhaka. The total monthly rainfall amount from several successive observations are computed and feed into the neural network. Missing values were estimated by a simple arithmetic average procedure. SPSS v20 were used for neural network forecasting. Total 636 rainfall data were analyzed. Moreover the software automatically slit data into three groups; training, validation and testing set. The training set was used to train the network whereas the validation set was used to monitor or test the network performance. The entire data are positively skewed in nature.

The Basics of Artificial Neural Network Approach

An ANN is a computational approach inspired by studies of the brain and nervous systems in living organisms. It is believed that the powerful functionality of a biological neural system is attributed to the parallel distributed processing nature of a network of cells; known as neurons An ANN emulates this structure by distributing the computation of small and simple processing units, called artificial neurons, or nodes. With this architecture, an ANN has proven to be a powerful mathematical model which excels at function approximation and pattern recognition. In recent years the study of artificial neural networks (ANN) has aroused great interest. The main reason underlying this interest lies in the fact that ANN are general, flexible, nonlinear tools capable of approximating any sort of arbitrary function. Due to their flexibility as function appropriators, ANN are robust methods in tasks related with pattern classification, the estimate of continuous variables and time series forecasting [9]. In this later case, ANN offer several potential advantages with respect to alternative methods-mainly ARIMA time series models-when it comes to dealing with problems concerning non linear data which do not follow a normal distribution [9].

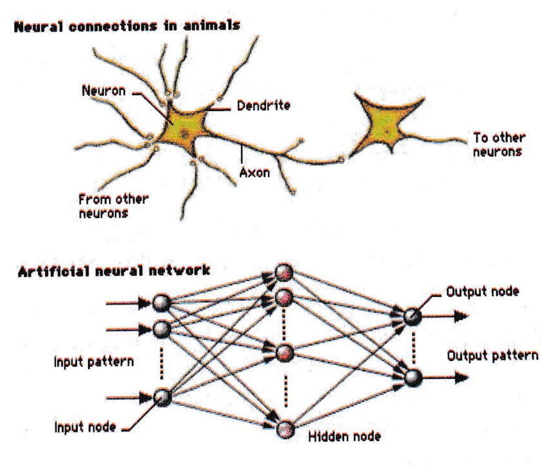


Figure 1: The Neural connection in animals' biological neuron (in top) and the counterpart Artificial Neural network structure (in bottom) [Eldon Y. Li, 1994]

An ANN is formed by nodes connected together. Nodes with similar characteristics are arranged into a layer. A layer can be seen as a group of nodes which have connections to other layers or the external environment, but which have no interconnections. Shown in figure 1 is a simple ANN which consists of three layers of nodes along with biological neuron systems. There are basically three types of layers. The first layer connecting to the input variables is called the input layer. The last layer connecting to the output variables is called the output layer. A single output node is shown in Figure 1 for clarity of illustration. It is straightforward to extend to multiple output nodes. Layers between the input and output layers are called hidden layers; there can be more than one hidden layer. Information is transmitted through the connections between nodes. In a simple situation, information is passed forward only, as shown in Figure 1. This type of network is called a feed forward network or multilayer feed forward neural network (MLFN). The main parameters of the MLFN are the connection weights. The estimation of parameters is known as "training" in which optimal connection weights are determined by minimizing as objective function.

ANN for Rainfall Forecasting

The purpose of this study is to forecast rainfall one time –step ahead. For the development of the proposed rainfall forecasting models, the rainfall process was assumed to be a Markovian process, which means that the rainfall value at a given location in space and time is a function of a finite set of previous realizations. With this assumption, a model structure can be expressed as

$$X(t+1) = g(X(t), X(t-1), X(t-2, \cdots, X(t-k+1)) + e(t), \quad (1)$$

Where,

$X(t)$ represents a vector of rainfall values $x_{1t}, x_{2t}, \cdots, x_{Nt}$ at N different locations at time t,

$g()$ is a nonlinear mapping function, which will be approximated using an ANN,

$e(t)$ is a mapping error (to be minimized) and

k is the (unknown) number of past rainfall realizations contributing to rainfall at the next time-step; usually, k refers to the lag of the network; if $k=1$, the rainfall at the next time-step is related only to the present rainfall, thus giving a lag-1 network.

Alternative Networks

There are a number of alternative ANNs which can be used to represent the Markovian model expressed in equation (1), where the continuous process of rainfall is being represented by a discrete process. Based on a review of these alternatives, two suitable ANN model configurations were identified and adopted in this study for comparison, They are

- Multilayer perceptron
- Radial Basis Functions

Both of the model SPSS software keeps, 60% data for training, 20% data for training and 20% data for validation purposes. Holdout validation techniques used for data validation.

Multilayer Perceptron

A multilayer perceptron or MLP model is made up of a layer N of input neurons, a layer K of a output neurons and one or more hidden layers; although it has been show that for most problems it would be enough to have only one layer L of hidden neurons. Figure 2 shows a typical multilayer perceptron along with the outline for input-output pattern. In this type of framework, the connections between neurons always feed forwards, that is, the connections feed from the neurons in a certain layer towards the neurons in the next layer.

The mathematical representation of the function applied by the hidden neurons in order to obtain an output value b_{pj}, when faced with the presentation of an input vector $X_p: x_{p1}, x_{p2}, \dots, x_{pi}, \dots, x_{pN}$, is defined by:

$$b_{pj} = f_L\left(\theta_j + \sum_{i=1}^{N} w_{ij} \cdot x_{pi}\right) \qquad (2)$$

Where, f_L is the activation function of hidden neurons L, θ_j is the threshold of hidden neuron j, w_{ij} is the weight of the connection between input neurons and hidden neuron and finally, x_{pi} is the input signal received by input neuron i for pattern p.

In a general way, a sigmoid function is used in the hidden layer neurons in order to give the neural network the capacity of learning possible nonlinear function, whereas the linear function is used in the output neuron in the event of an estimation of a continuous variable.

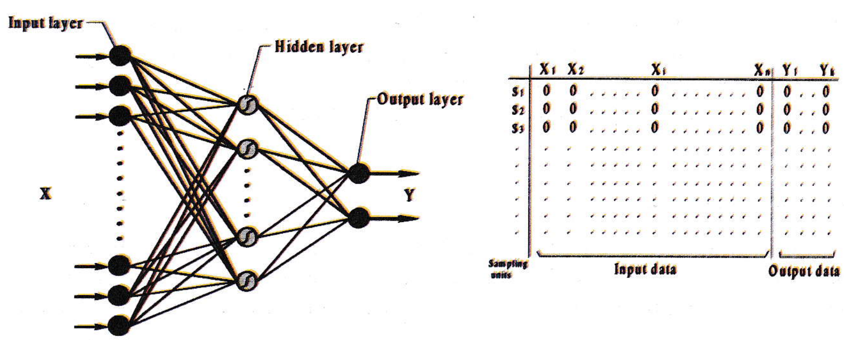


Figure 2: Semantic illustration of a three layered feed-forward neural network also called MLP with one input layer, one hidden layer and one output layer. The right hand side of the figure shows the data set to be used in backpropagation network models. $X_1,\ldots,X_N$ are the N input neurons, $Y_1,\ldots,Y_K$ are the output neurons and S_1, $S_2,\ldots$ are the observation data.

Radial Basis Function

Radial Basis Function or RBF models are made up of three layers just like the MLP network [10]. The peculiarity of RBF lies in the fact that the hidden neurons operate on the basis of the Euclidean distance that separate the input vector from the weight vector which is stored by each one (the so-called centroid), a quantity to which a Gaussian radial function is applied, in a similar way to the kernel function in the kernel regression model [11].

Out of the most widely used radial functions (Gaussian, quadratic, inverse quadratic, spline), in this study the Gaussian was applied as the activation function of the hidden neurons on input vector, in order to obtain an output value b_{pj} :

$$b_{pj} = exp\left[\frac{-\sum_{i=1}^{N}(x_{pi} - w_{ij})^2}{2\sigma^2}\right] \tag{3}$$

If input vector coincides with the centroid of neuron *j*, this responds with a maximum output (the unit0. That is to say, when the input vector is located in a region near the centroid of a neuron, this is activated, indicating that it recognizes the input pattern; if the input pattern is very different to the centroid, the response will tend towards zero. The normalization parameter i.e. scale factor measures the Gaussian width, and would equal the radius of influence of the neuron in the space of the inputs; the greater the scale parameter the larger the region dominated by the neuron around the centroid [12].

Measure of Fit

The most widely used measure of fit in the field of time series forecasting is the Mean Absolute Percentage Error (or MAPE) due to the fact that it is easy to interpret-it is interpreted in terms of percentage error-and does not depend on the measure scale of the variable [13].

$$MAPE = \frac{\sum_{p=1}^{P}\left|\frac{y_{pk} - \hat{y}_{pk}}{y_{pk}}\right| . 100}{P}$$

(4)

Where y_{pk} is the desired value for output neuron *k* belonging to pattern , $\hat{y}_{pk}$ is the output signal of output neuron *k* for pattern *p*, and P is the number of total pattern analyzed. As far as the MAPE is concerned, it does not constitute a resistant location index as it is based on the calculation of the arithmetic mean of the percentage errors and therefore, in sensitive to the presence of outlier vales. With the purpose of overcoming the limitations presented by the MAPE as a measure of fit in time series forecasting, in this study not only did we calculated the arithmetic mean but we also obtained a resistant location estimator, the M-estimator of Huber [14].

RESULT AND DISCUSSION

In accordance with the description made, a set of MLP and RBF network models were designed, form the manipulation of a series of parameters. In the case of the MLP models, different seed values were used for the initialization of the weights, as well as a number (between one and four) of hidden neurons. The learning algorithm used was the conjugated gradients one and as an activation function of the hidden and output neurons, the sigmoid and linear ones, respectively [15]. As far as the different RBF models are concerned, they were constructed by varying the number of centroids, between 5 and 25 and the values of the normalization parameter, between 0.2 and 1.5. For each of the two types of network model designed, the one that showed the lowest mean absolute percentage error (MAPE, calculated using the M-estimator of Huber) with the validation group is chosen. Figure 3 shows the time series plot of observed monthly total rainfall series for test data set. Figure 4 and 5 indicates the relative performance of MLP and RBF outputs with the actual data series. Table 1 show the mean percentage error calculated using the M-estimator of Huber for each of the models selected in the validation phase, for the three sets of data. The values of the corresponding arithmetic mean appear in the parenthesis.

Table 1: Absolute Percentage Error for MLP and RBF for Training, Validation and Test three sets of data

	Training	**Validation**	**Test**
MLP	0.502(0.832)	0.874(0.978)	0.447(0.687)
RBF	0.452(0.489)	0.563(0.674)	0.348(0.356)

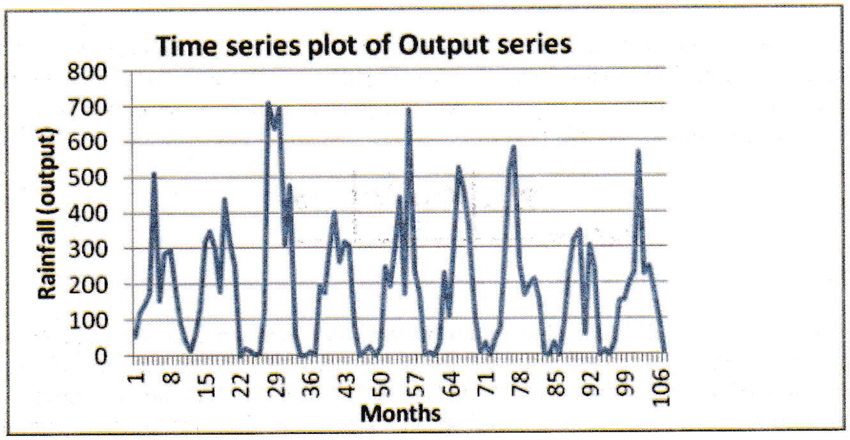


Figure 3 Time series plot of monthly total rainfall in Dhaka city for test data

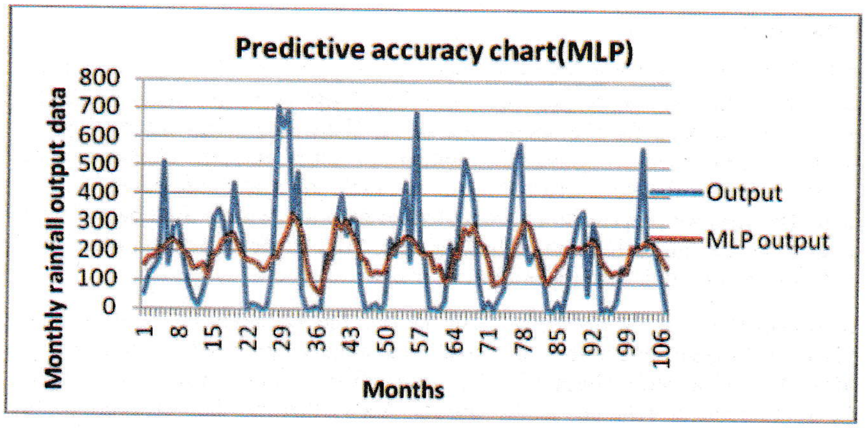
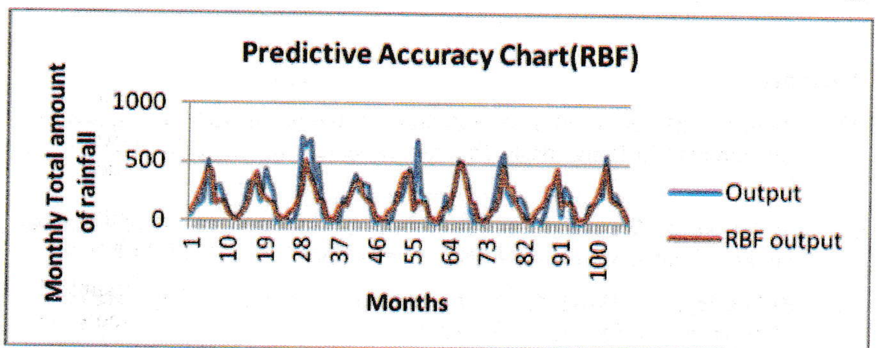


Figure 4 Predictive accuracy chart obtained from MLP for the test data set along



with observed data

Figure 5 Predictive accuracy chart obtained from RBF for the test data set along with observed data

It can be seen that the generalization error committed by each of the network models analyzed and estimated from the test set, is less than 1%. In accordance with the interpretation criteria for the MAPE value established by Lewis (1982), the two neural network models can be considered to have highly accurate forecasting [16]. The model with the best performance with the test group is the RBF than MLP model. As far as the distribution of error is concerned, it has been seen that in most cases they have outlier values, which is why the arithmetic means always provides a higher value than the M-estimator value. For this pattern recognition approach, an improvement in prediction is expected through the collection of more data for training the model. Additionally, in order to predict the occurrence of the peak rainfall rate more accurately, it may be necessary to identify more control variables, such as wind speed and direction, and incorporate them into the network input.

Conclusion

Rainfall forecasting using ANNs has been the focus herein. Two types of ANNs suitable for this task were identified, developed and compared. Concerning the procedure proposed for the application of ANN in time series forecasting, we show the convenience, against the opinion of some authors, of carrying out preprocessing of the time series so as to eliminate the systematic components in them. Besides, we propose an improvement of the most widely used performance index to assess time series forecasting models, MAPE, through the calculation of an M-estimator rather than the arithmetic mean.

With respect to overall performance of the two models analyzed, it is clear that ANN are flexible, effective tools for all researchers interested in modeling the behavior of time series. Future lines of research should be aimed at overcoming the limitations in relation to the use of ANN, that is, the selection of the parameters related with the construction of the model and the analysis of the effect or significance of the input variables in the forecast carried out. This work points out some possible directions to move in as far as this is concerned.

References

[1] Rahman, M. S. (2000) A rainfall simulation model for agriculture development in Bangladesh, *Discrete Dynamics in Nature and Society*, 5, 1-7.

[2] Anderson, T. W. and L. A. Goodman(1957) Statistical inference about Markov Chains, *Annals of Mathematics and Statistics,* 28, 89-110.

[3] Billingsley, P. (1961) Statistical methods in Markov Chain, *Annals of Mathematics and Statistics*, 32, 12-40.

[4] Lee, T. C., Judge, G.G. and Zellner, A. (1970) Estimating the parameters of the Markov probability model for aggregate time series data, North Holland, Amsterdam.

[5] Rahman, M. S. and Mian, B. M. A. (2002) Stochastic study of asymptotic behavior of rainfall distribution in Rajshahi, International Journal of Statistical Science, 1, 36-48.

[6] Luk. K. C., Ball, J. E. and Sharma A. (2001) An application of artificial neural networks for rainfall forecasting, *Mathematical and Computer Modeling,* 33, 683-693.

[7] Hsu, K. L., Gupta, V. and Cuykendal R. R. (1995) Artificial neural network modeling of the rainfall-runoff process, *Water Resources Research*, 31(10), 2517-2530.

[8] Hornik, K., Stinchcombe, M., and White, H., (1989) Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359-366.

[9] Hansen, J.V., McDonald, J.B. and Nelson, R.D. (1999) Time series prediction with genetic-algorithm design neural networks: An empirical comparison with modern statistical models, *Computational Intelligence,* 15(3), 171-184.

[10] Broomhead, D. S. and Lowe, D. (1988) Multivariable functional interpolation and adaptive networks. *Complex Systems,* 2, 321-355.

[11] Bishop, C. M. (1995) *Neural networks for pattern recognition,* Oxford: Oxford University Press.

[12] Moreno, J. J. M., Pol, P. A. and Gracia, P. M., (2011) Artificial neural networks applied to forecasting time series, *Psicothema*, 23, 322-329.

[13] Witt P. A. and Witt, C. A. (1992) *Modeling and forecasting demand in market,* Academic Press, San Diego.

[14] Huber, P. (1975) *Robust Statistics,* Nueva York: Wiley.

[15] Battiti, R. (1992) First and second order methods for learning: Between steepest descent and Newswton's method, *Neural Computation*, 4, 141-166.

[16] Lewis, C. D. (1982) *Industrial and business forecasting methods*, London, Butterworths.