Hindawi Publishing Corporation
International Journal of Atmospheric Sciences
Volume 2013, Article ID 525383, 14 pages
http://dx.doi.org/10.1155/2013/525383

*Research Article*

# Artificial Neural Network Model in Prediction of Meteorological Parameters during Premonsoon Thunderstorms

**A. J. Litta,[1] Sumam Mary Idicula,[1] and U. C. Mohanty[2]**

[1] *Department of Computer Science, Cochin University of Science and Technology, Cochin, Kerala 682 022, India*
[2] *School of Earth Ocean and Climate Sciences, Indian Institute of Technology Bhubaneswar, A2-707, Toshali Bhawan, Satyanagar, Bhubaneswar, Odisha 751007, India*

Correspondence should be addressed to A. J. Litta; littaaj@gmail.com

Received 21 May 2013; Revised 6 October 2013; Accepted 10 October 2013

Academic Editor: Hui Wang



Forecasting thunderstorm is one of the most difficult tasks in weather prediction, due to their rather small spatial and temporal extension and the inherent nonlinearity of their dynamics and physics. Accurate forecasting of severe thunderstorms is critical for a large range of users in the community. In this paper, experiments are conducted with artificial neural network model to predict severe thunderstorms that occurred over Kolkata during May 3, 11, and 15, 2009, using thunderstorm affected meteorological parameters. The capabilities of six learning algorithms, namely, Step, Momentum, Conjugate Gradient, Quick Propagation, Levenberg-Marquardt, and Delta-Bar-Delta, in predicting thunderstorms and the usefulness for the advanced prediction were studied and their performances were evaluated by a number of statistical measures. The results indicate that Levenberg-Marquardt algorithm well predicted thunderstorm affected surface parameters and 1, 3, and 24 h advanced prediction models are able to predict hourly temperature and relative humidity adequately with sudden fall and rise during thunderstorm hour. This demonstrates its distinct capability and advantages in identifying meteorological time series comprising nonlinear characteristics. The developed model can be useful in decision making for meteorologists and others who work with real-time thunderstorm forecast.

## 1. Introduction

Thunderstorm, resulting from vigorous convective activity, is one of the most spectacular weather phenomena in the atmosphere. It is one of the global phenomena that can occur anywhere in the world at any time. It is also known as lightning storm or hailstorm. This storm is a form of weather characteristic containing strong wind, lightning, heavy rain, and sometimes snow or hail. Although thunderstorm is generally very short-lived phenomena, it has great potential to produce serious damage to human life and property such as lightning, damaging straight-line wind, large sized hail, heavy precipitation, and flooding. Many parts over the Indian region experience thunderstorms at higher frequency during premonsoon months (March–May), when the atmosphere is highly unstable because of high temperatures prevailing at lower levels. Severe thunderstorms form and move generally from northwest to southeast over the eastern and northeastern states of India during the premonsoon season. These severe thunderstorms associated with thunder, squall lines, lightning, torrential rain, and hail cause extensive loss in agriculture, damage to property, and also loss of life. The casualties reported due to lightning associated with thunderstorms in this region are the highest in the world. The strong wind produced by the thunderstorm is a real threat to aviation. The highest numbers of aviation hazards are reported during occurrence of these thunderstorms. These severe thunderstorms have significant socioeconomic impact in the eastern and northeastern parts of the country. An accurate location specific and timely prediction are required to avoid loss of lives and property due to strong winds and heavy precipitation associated with these severe local storms [1].

Accurate forecasting of thunderstorms and severe thunderstorms are critical for a large range of users in the community. The general public can benefit from timely forecasts and warnings of impending severe thunderstorms. Thunderstorm forecasting typically has proved to be one of the most difficult tasks, due to their rather small spatial and temporal extension and the inherent nonlinearity of their dynamics and physics [2]. The techniques for predicting thunderstorms can be classified into two groups (a) the empirical approach and (b) the dynamical approach [3]. First method is a historical treatment of thunderstorm extrapolation techniques (knowledge-based expert systems including fuzzy logic and artificial neural network (ANN)). The second method is prediction using high resolution numerical weather prediction (NWP) models. The second approach for studying thunderstorms is already specified by many researchers [4–7]. Most weather prediction systems use a combination of empirical and dynamical techniques. However, a little attention has been paid to the use of ANNs in thunderstorm forecasting. ANN-based approach can be used to model complex relationships between inputs and outputs or to find patterns in data. ANN can be viewed as a mathematical model or computational model that is inspired by the structure or functional aspects of biological neural networks. Neural networks are designed to extract existing patterns from noisy data. The procedure involves training a network (training phase) with a large sample of representative data, after which one exposes the network to data not included in the training set (validation or prediction phase) with the aim of predicting the new outcomes [8]. The interest in neural networks comes from the networks' ability to mimic human brain as well as its ability to learn and respond. As a result, neural networks have been used in a large number of applications and have proven to be effective in performing complex functions in a variety of fields [9].

ANN has proven to be powerful and general technique for machine learning (ML) [10]. Most successful application of neural networks involved pattern recognition, statistical mapping, or modeling [11]. According to Bailey and Thompson [12], successful application can include signal validation, process monitoring, diagnostics, signal and information processing, and control of complex system. James et al. [13] mentioned that ANNs have the ability to tackle the problem of complex relationships among variables that cannot be accomplished by more traditional methods. ANNs are excellent tools for complex manufacturing processes that have many variable and complex processes. According to Palade et al. [14], ANNs represent an excellent tool that has been used to develop a wide range of real world applications, especially in case when traditional solving methods fail. The advantages of ANNs such as ideal learning ability from data, classification capabilities, and generalization for situation do not contain training data set, computationally fastness once trained due to parallel processing, noise tolerance. There were these advantages that make ANNs to be successfully applied to various real world problems, including medical diagnosis, image computing, speech recognition, and weather forecasting [15–19].

Bodri and Čermák [15] developed an ANN using 38 years of rainfall data to predict monthly and yearly precipitation levels for multiple sites in the Czech Republic. Using spatial and temporal data of rainfall, Luk et al. [16] developed an ANN for short-term precipitation prediction focused on predicting flash flood rainfall amounts for 15 min ahead for various areas of western Sydney, Australia. Maqsood et al. [17] used an ensemble of ANNs to provide 24 h predictions for air temperature, wind speed, and humidity at the Regina Airport in Canada. Chaudhuri and Chattopadhyay [18] designed a feedforward multilayered artificial neural network model to estimate the maximum surface temperature and relative humidity. Chattopadhyay [19] implemented a feedforward ANN with one hidden layer to forecast average summer monsoon over India and established that it gave a better forecast than a forecast based on multiple linear regression and persistence. The growing development of computer-aided analysis has facilitated the application of various ML techniques in hydrological modeling [20–25]. ANN have been applied successfully for time series modeling in many hydrological contexts such as river flow [20], flood forecasting [21], and water quality modeling [22]. The recent advances in neural network methodology for modeling nonlinear, dynamical phenomena along with the impressive successes in a wide range of applications are motivated to investigate the application of ANNs for the prediction of hourly temperature and relative humidity needed for the genesis of severe thunderstorms over Kolkata.

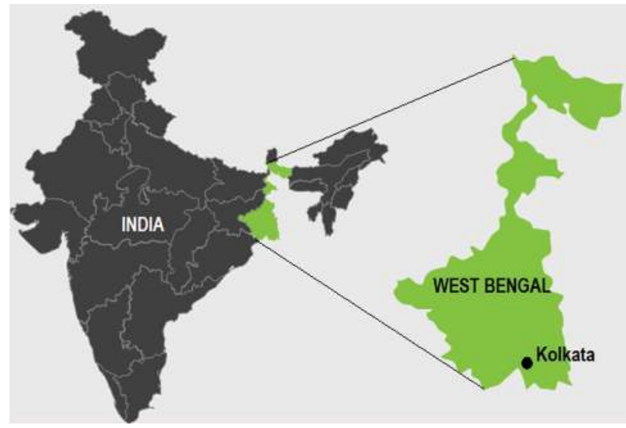



Figure 1: The geographical location of Kolkata in West Bengal.

In this paper, experiments are conducted with ANN model to predict severe thunderstorms that occurred over Kolkata (22.52°N, 88.37°E) using thunderstorm affected meteorological parameters. The geographical location of the study area is given in Figure 1. The performance of six learning algorithms, namely, Step (STP), Momentum (MOM), Conjugate Gradient (CG), Quick Propagation (QKP), Levenberg-Marquardt (LM), and Delta-Bar-Delta (DBD), is evaluated using predicted hourly surface temperature and relative humidity during these thunderstorm days. The accuracy of the predictions was evaluated by the correlation coefficient (CC), the root mean-square error (RMSE), the mean absolute error (MAE), and percent correct (PC) between the measured and predicted values. The developed ANN model with LM algorithm was applied to derive thunderstorm forecast from 1 to 24 hour (h) ahead at Kolkata. The goal of this study was to use ANNs to predict hourly temperature and relative humidity during thunderstorm days from 1 to 24 h ahead using prior weather data as inputs. This study is presented in the following manner. Section 2 presents the ANN model configurations and a brief description about learning algorithms used for the present study. The case descriptions of thunderstorm events are given in Section 3. The results and discussions are described in Section 4 and the conclusions in Section 5.

## 2. Data and Methodology

*2.1. ANN Experimental Setup.* The developed ANN model is based on one of the neural network architecture called multilayer perceptron network (MLPN) model (also known as *multilayer feedforward network*). This is the most popular network architecture in use today. This is the type of network which the units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output, and the units are arranged in a layered feedforward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases), the free parameters of the model. Such networks can model functions of almost arbitrary complexity with the number of layers and the number of units in each layer, determining the function complexity. Important issues in MLPN design include specification of the number of hidden layers and the number of units in these layers [9]. Once the number of layers and number of units in each layer have been selected, the network's weights and thresholds must be set so as to minimize the prediction error made by the network. This is the role of the training algorithms.

This study evaluates the utility of MLPN for estimating hourly surface temperature and relative humidity. Designing ANN model follows a number of systemic procedures. In general, there are five basics steps: (1) collecting data, (2) preprocessing data, (3) building the network, (4) training and (5) test performance of model. The basic flow in designing ANN model is given in Figure 2. The weather data, namely, hourly mean sea level pressure, relative humidity, and wind speed of 3 years (April and May 2007 to 2009) collected from the India meteorological department (IMD) of Kolkata, were used as the input data for training and testing ANN model which will be used for the prediction of hourly temperature. The weather data, namely, hourly mean sea level pressure, temperature, and wind speed of 3 years (April and May 2007 to 2009) of Kolkata, were used as the input data for training and testing ANN model which will be used for the prediction of hourly relative humidity. Major numbers of thunderstorms are occurred over Kolkata in April and May. Thus the hourly data sets of these two months are selected for training and testing. The other additional input parameters for each model are month, day and hour of the observation. After data collection, two data preprocessing procedures are conducted to train the ANNs more efficiently. These procedures are the following: (1) solve the problem of missing data and (2) normalize data. The missing data are replaced by the average of neighboring values. Neural networks generally provide improved performance with the normalized data. The use of original data as input to neural network may cause a convergence problem [26]. All the weather data sets were therefore transformed into values between −1 and 1 through dividing the difference of actual and minimum values by the difference of maximum and minimum values. At the end of each algorithm, outputs were denormalized into the original data format for achieving the desired result. Separate models with same configuration have been built to predict both surface parameters, namely, temperature and relative humidity.

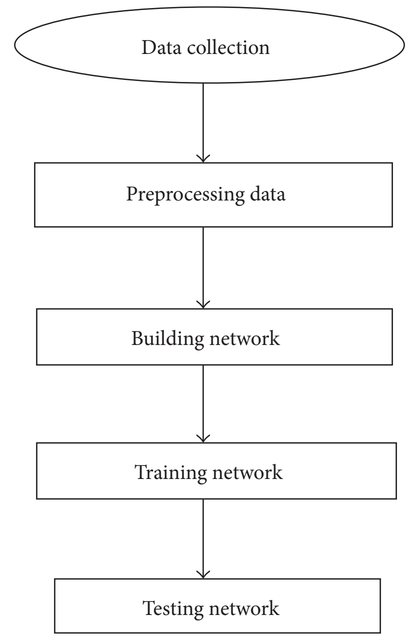
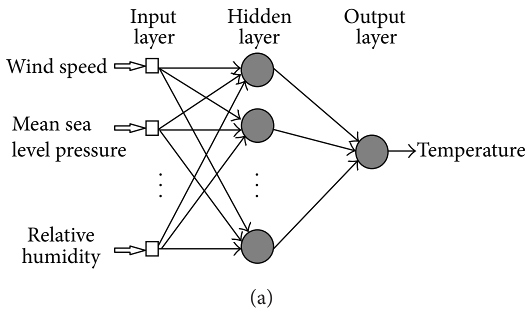



Figure 2: Basic flow for designing ANN model.

A three-layer structure (one input layer, one hidden layer, and one output layer) was selected with hyperbolic tangent (tanh) transfer function for hidden layer and linear transfer function for output layer. Figure 3 provides an overview of the structure of MLPN model for the prediction of temperature and relative humidity. The chosen weather data were divided into two randomly selected groups, the training group, corresponding to 67% of the patterns, and the test group, corresponding to 33% of patterns, so that the generalization capacity of network could be checked after training phase. Networks were trained for a fixed number of epochs. The error level was set to a relatively small value ($10^{-4}$). The optimal number of hidden neurons was obtained experimentally by changing the network design and running the training process several times until a good performance was obtained. A random number generator was used to assign the initial values of weights and thresholds with a small bias as a difference between each weight connecting two neurons together since similar weights for different connections may lead to a network that will never learn.

The 24 h ANN model outputs of surface temperature and relative humidity at Kolkata (22.52°N, 88.37°E) during three thunderstorm days of May 2009 (May 3, 11, and 15, 2009) were used to evaluate these models. The performance of six learning algorithms, STP, MOM, CG, QKP, LM, and DBD is evaluated using predicted hourly surface temperature and relative humidity during thunderstorm days and found LM algorithm for future thunderstorm studies. The accuracy of the predictions was evaluated by CC, RMSE,



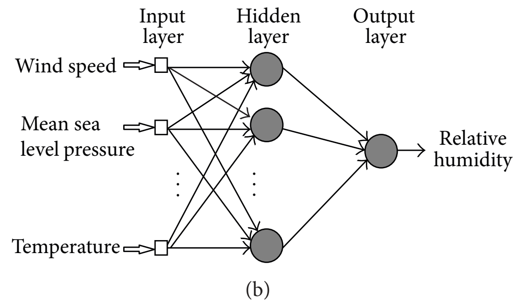




Figure 3: Architecture of multilayer perceptron network for the prediction of (a) temperature and (b) relative humidity.

MAE, and PC between the measured and predicted values. The developed ANN model with LM algorithm was applied to derive thunderstorm forecast from 1 to 24 h ahead at Kolkata from the data of 3 consecutive years (April and May 2007–2009). Models were created to predict temperature and relative humidity at hourly intervals with 1, 3, 6, 12, and 24 h ahead. The results are evaluated using MAE, RMSE, CC, and PC. The ANN model simulations are carried out using the Neurosolutions software developed by Neuro Dimensions Inc. of Florida [27].

*2.2. Learning Algorithms.* In neural network, the learning algorithms play quite important role in the process. An appropriate topology may still fail to give a better model, unless trained by a suitable learning algorithm. A good learning algorithm will shorten the training time while achieving a better accuracy. Therefore, training process is an important characteristic of the ANNs, whereby representative examples of the knowledge are iteratively presented to the network, so that it can integrate this knowledge within its structure. There are a number of training algorithms used to train a MLPN, and a frequently used one is called the backpropagation (BP) learning algorithm [28]. The BP algorithm, which is based on searching an error surface using gradient descent for points with minimum error, is relatively easy to implement. However, the BP algorithm has some problems for many applications. The algorithm is not guaranteed to find the global minimum of the error function since gradient descent may get stuck in local minima, where it may remain indefinitely. In addition to this, long training sessions are often required in order to find an acceptable weight solution because of the well-known difficulties inherent in gradient descent optimization. Therefore, a lot of variations to improve the convergence of the BP were proposed such as DBD, QKP [29–31]. Optimization methods such as second-order methods (CG, Quasi-Newton (QN), LM) have also been used for ANN learning in recent years. The LM algorithm combines the best features of the Gauss-Newton technique and the steepest-descent algorithm but avoids many of their limitations. In particular, it generally does not suffer from the problem of slow convergence [32].

A number of researchers have carried out comparative studies of MLPN learning algorithms. Kişi and Uncuoğlu [33] compared LM, CG, and resilient algorithm for streamflow forecasting and determination of lateral stress in cohesionless soils. They found that LM algorithm was faster and achieved better performance than the other algorithms in learning. Esugasini et al. [34] considered the problem of breast cancer diagnosis and compared the classification accuracy of the standard steepest descent against the classification accuracy of the gradient descent with momentum and adaptive learning, resilient BP, QN, and LM algorithm. The simulations show that the neural network using the LM algorithm achieved the best classification performance. Raju et al. [35] demonstrated the application of ANNs in predicting the weekly spring discharge with three different learning algorithms. The learning algorithms considered by the authors were QKP algorithm, batch BP algorithm, and LM algorithm. They conclude that the QKP algorithm had a better performance to the application. Übeyli and Güler [36] compared BP, DBD, extended DBD, QKP, and LM algorithms to compute the quasistatic parameters, the characteristic impedance, and the effective dielectric constant of the asymmetric coplanar waveguides (ACPWs). The results of the LM algorithm for the quasistatic parameters of the ACPWs were in very good agreement with the results available in the literature. Cheng et al. [20] made sensitivity study with 3 training algorithms, namely, the gradient descent, LM, and Scaled Conjugate Gradient (SCG) algorithms to forecast daily and monthly river flow discharges in Manwan Reservoir. The sensitivity analysis of the training algorithms shows that the SCG algorithm can enhance the accuracy of model prediction results effectively. The results of above studies have illustrated that the relative performance of algorithms depends on the problem being tackled. Six learning algorithms were applied in this study, in order to identify the one which trains a given network more efficiently. Brief descriptions of these learning algorithms are as follows.

*2.2.1. Step (STP) Algorithm.* Gradient descent (GD) learning rules provide first-order gradient information about the network's performance surface (e.g., backpropagation and real-time recurrent learning). The most straightforward way of reaching the bottom (the minima) given which way is up is to move in the opposite direction. With this scenario, the only variable is the step size (i.e., how far should it move before obtaining another directional estimate). If the steps are too small, then it will take too long to get there. If the steps are too large, then it may overshoot the bottom, causing it to rattle or even diverge. The step uses this procedure to adapt the weights of the activation component that it is stacked on.

*2.2.2. Momentum (MOM) Algorithm.* Step components try to find the bottom of a performance surface by taking steps in the direction estimated by the attached backpropagation component. Network learning can be very slow if the step size is small. It can oscillate or diverge if it is chosen too large. For further complicate matters, a step size that works well for one location in weight space may be unstable in another. The momentum provides the gradient descent with some inertia, so that it tends to move along a direction that is the average estimate for down. The amount of inertia (i.e., how much of the past to average over) is imposed by the momentum parameter. The higher the momentum is, the more it smoothes the gradient estimate, and the less effect a single change in the gradient has on the weight change. The major benefit is the added ability to breakout of local minima that a step component might otherwise get caught in. Note that oscillations may occur if the momentum is set too high. The momentum parameter is the same for all weights of the attached component. An access point has been provided for the step size and momentum allowing access for adaptive and scheduled learning rate procedures.

*2.2.3. Conjugate Gradient (CG) Algorithm.* The GD algorithms (like "step" and "momentum") use only the local approximation of the slope of the performance surface (error versus weights) to determine the best direction to move the weights in order to lower the error. Second-order methods use or approximate second derivatives (the curvature instead of just the slope) of the performance surface to determine the weight update. This information is very important for determining the optimal update direction. Since this method makes use of the second derivatives of the function to be optimized, it is typically referred to as the second-order methods.

*2.2.4. Levenberg-Marquardt (LM) Algorithm.* The LM algorithm is one of the most appropriate higher-order adaptive algorithms known for minimizing the MSE of a neural network. It is a member of a class of learning algorithms called "pseudosecond order methods." Standard gradient descent algorithms use only the local approximation of the slope of the performance surface (error versus weights) to determine the best direction to move the weights in order to lower the error. Second-order methods use the Hessian or the matrix of second derivatives (the curvature instead of just the slope) of the performance surface to determine the weight update, while pseudosecond order methods approximate the Hessian. In particular the LM utilizes the so-called Gauss-Newton approximation that keeps the Jacobian matrix and discards second-order derivatives of the error. If the performance surface is quadratic (which is only true in general for linear systems), then using a second-order method can find the exact minimum in one step. A key advantage of the LM approach is that it defaults to the gradient search when the local curvature of the performance surface deviates from a parabola, which may happen often in neural computing.

*2.2.5. Quick Propagation (QKP) Algorithm.* The QKP uses information about curvature of the error surface. This requires the computation of the second-order derivatives of the error function during training. The QKP assumes the error surface, a function of connection weights, to be locally quadratic (i.e., a parabola) and attempts to jump in one step from the current position directly into the minimum of the parabola. The QKP computes the derivatives in the direction of each weight. After computing the first gradient as in regular backpropagation, a direct step to the error is attempted by changing the weight.

*2.2.6. Delta-Bar-Delta (DBD) Algorithm.* The DBD is an adaptive step-size procedure for searching a performance surface. The step size and momentum are adapted according to the previous values of the error at the neurons. If the current and past weight updates are both of the same sign, it increases the learning rate linearly. The reasoning is that if the weight is being moved in the same direction to decrease the error, then it will get there faster with a larger step size. If the updates have different signs, this is an indication that the weight has been moved too far. When this happens, the learning rate decreases geometrically to avoid divergence [27].

## 3. Case Description

For the present study three severe thunderstorm cases, May 3, 11, and 15, 2009, have been taken and the description of each case is as follows.

Case 1 was a severe thunderstorm, which was reported on May 03, 2009 over Kolkata (Figure 1) with a maximum speed of 61.2 kilometer per hour (kmph) lasting for a few minutes. This intense convective event produced 31.4 mm (millimeter) rainfall over Kolkata. In the synoptic charts at 0000 UTC (Coordinated Universal Time) a low pressure area was found at the surface over north Chattisgarh and adjoining Jharkhand, and a trough from this extending southward up to interior Tamilnadu across Andhra Pradesh is found. At 1.5 km (kilometer) above sea level (a.s.l), cyclonic circulation is seen over west Uttar Pradesh, and a trough from this extends southeastwards up to south peninsula across east Madhya Pradesh and Andhra Pradesh. There was no significant trough in midtroposphere. No subtropical westerly jet maxima were seen over the region. A few places recorded moderate rainfall over Gangetic West Bengal (GWB) and isolated rainfall over Orissa, Chattisgarh, and Bihar. Bankura recorded 24.9 mm and Sriniketan 38.2 mm of rainfall.

Case 2 was a severe thunderstorm, which was reported on May 11, 2009, over Kolkata with squally winds of the order of 87 kmph. Rainfall of 33.3 mm was reported over Kolkata. The synoptic charts show a trough at sea level chart from east Uttar Pradesh to north Tamilnadu across east Madhya Pradesh and Andhra Pradesh. Cyclonic circulation in lower levels is found over Bihar and neighborhood. Trough from this extends up to extreme south peninsula across Chattisgarh, Telangana, and Rayalaseema. Another cyclonic circulation existed over Arunachal Pradesh adjoining Assam and Meghalaya. A trough from Arunachal Pradesh to northwest Bay of Bengal was found in middle troposphere. Subtropical westerly jet maxima were found over the region. Light to moderate rain occurred at few places over Orissa and GWB with Midnapore and Alipore reporting 17.8 mm and 21.9 mm, respectively.

Case 3 was a severe thunderstorm, which was reported on May 15, 2009. A squall passed over Kolkata at 1230 UTC on May 15, 2009, with a maximum speed of 68.4 kmph. This intense convective event produced 16.9 mm rainfall over Kolkata. The synoptic charts show a trough at sea level from east Madhya Pradesh to south coastal Tamilnadu across Telangana and another trough to northeast Bay of Bengal across Orissa. Cyclonic circulation is seen in lower levels over west Uttar Pradesh and a trough from this extends up to coastal Andhra Pradesh across Vidarbha with embedded cyclonic circulation over Telangana. Trough in midtroposphere is found from Arunachal Pradesh to north Bay of Bengal. Subtropical westerly jet maxima were found over the region. A few places of GWB recorded moderate rainfall and isolated rainfall over Orissa and Bihar. Bankura recorded 34.0 mm and Midnapore 51.6 mm of rainfall [37].

## 4. Results and Discussion

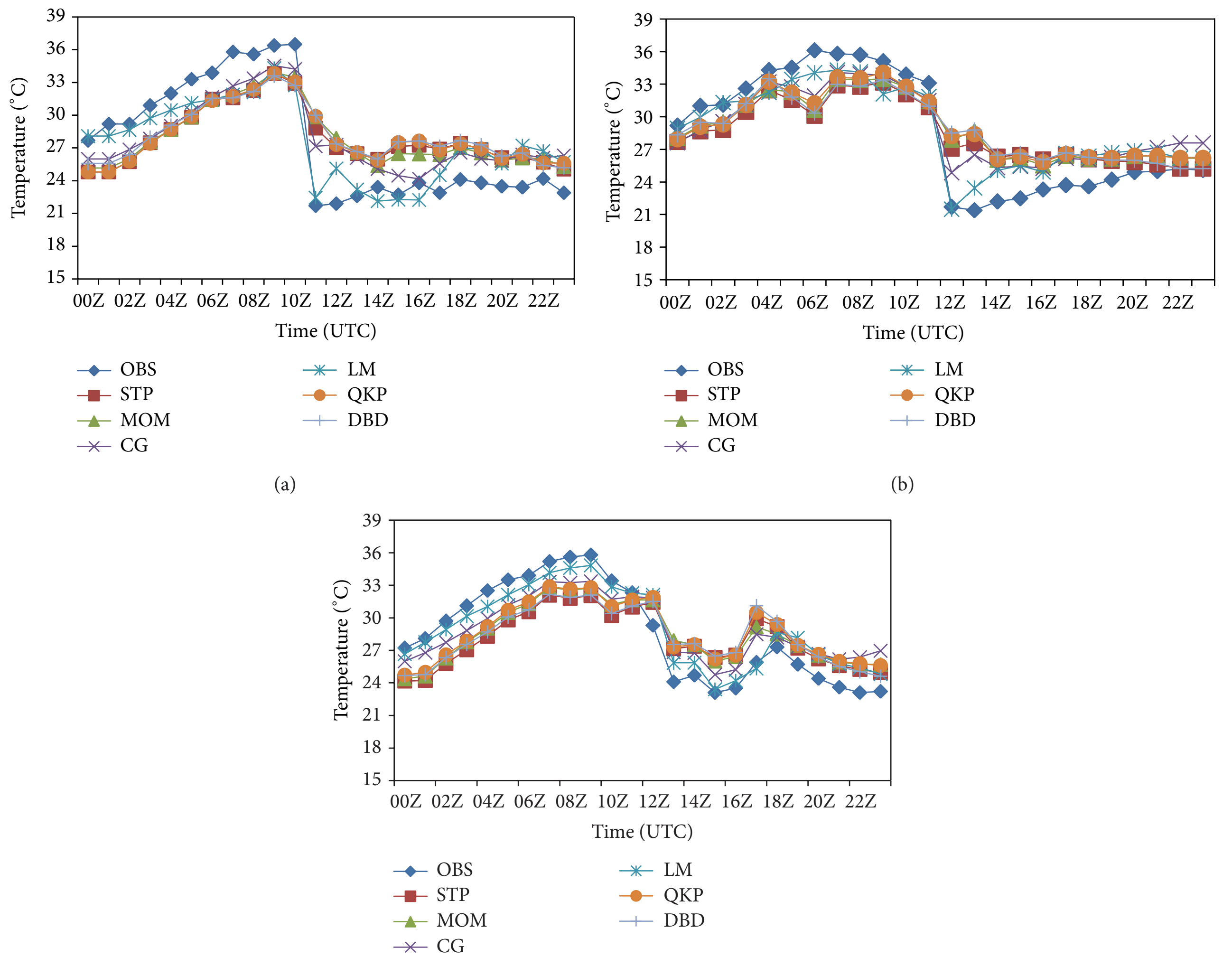
According to the previous studies of [38–40], the general preconditions for the initiation of thunderstorms are conditional instability, a sufficiently deep humid layer in the lower and midtroposphere, and an uplifting mechanism to initiate convection. The formation of thunderstorms is an interaction between these conditions on different scales. The surface parameters play a significant role in the genesis, whereas the strength of the upper air pull is required to assess the growth of the thunderstorm [41]. The greater the density differences between air masses (temperature and humidity), the greater the atmospheric instabilities that develop, and the greater the intensity of these thunderstorms [42]. Recent studies show a high positive correlation between surface temperature and lightning activity [43]. Temperature and relative humidity on the surface are useful tool in forecasting the likelihood occurrence of a thunderstorm [44]. A sudden drop in temperature or sudden increase in relative humidity during the day indicates for the occurrence of thunderstorm [41]. The occurrence and intensity of three severe thunderstorms are examined in the following sections by the analysis of observed and ANN model predicted surface relative humidity and temperature.

*4.1. Comparison of Learning Algorithms.* The ANN model predicted surface temperature and relative humidity with different learning algorithms during severe thunderstorm cases are explored in the following section. Analysis of the results of these experiments is helpful to understand the impact of learning algorithms on the prediction of severe thunderstorm events and assist in the customization of model for future severe thunderstorm predictions over east and northeast Indian region.

Figure 4 shows the interomparison of observed and ANN model predicted diurnal variation of surface temperature (°C) with different learning algorithms over Kolkata valid for May 3, 11, and 15, 2009. From the figures, it is clearly visible that the observed data (OBS) show a sudden drop in temperature in all three thunderstorm days. The ANN model with different learning algorithms captured the temperature drop during the thunderstorm hour for all the three cases. But the predicted intensity is different for different algorithms. For the first case (Figure 4(a)), the observed temperature showed a sudden drop of 15°C from 36.7°C to 21.7°C at 1000 UTC. The ANN model prediction with LM showed a drop from 33°C to 22°C (11°C) at 1000 UTC, whereas CG presented a drop from 34°C to 27°C (7°C) at 1000 UTC. All other algorithms show a difference less than 4°C during thunderstorm hour. The DBD has a least performance than other algorithms. In the second thunderstorm case (Figure 4(b)), observed temperature fall is from 33.1°C to 21.7°C (11°C) at 1200 UTC, whereas LM indicated a drop from 32°C to 21°C (11°C) at the same thunderstorm hour. CG showed only 6°C difference between predicted and observed values. The other algorithms presented less intensity in difference between predicted and observed values. For the third case (Figure 4(c)), observed temperature showed a drop from 29°C to 24°C (6°C) at 1300 UTC, whereas LM showed a drop from 32°C to 27°C (5°C). All other algorithms are also captured the sudden fall with almost same intensity of observation and LM algorithm for this thunderstorm case.

Relative humidity at surface level has been taken into account, as it is an essential factor in intense convection. Storm days require a sufficiently humid and deep layer in the lower and middle atmosphere [36]. Figure 5 shows the intercomparison of observed and ANN model predicted relative humidity (%) with different learning algorithms over Kolkata for severe thunderstorm days. For all the thunderstorm cases, ANN model with different algorithms have captured the increase in relative humidity during thunderstorm hour as in the observation. But the predicted intensity is different for different learning algorithms. In the first case (Figure 5(a)), the observed relative humidity showed a rise of 48% from 52% to 100% at 1000 UTC. The ANN model prediction with LM showed a rise from 53% to 95% (42%) at 1000 UTC. All other algorithms except CG show same change in intensity (32%) at 1000 UTC, whereas CG presented a rise from 55% to 81% (25%) at 1100 UTC. The performance of CG algorithm is poor than all other algorithms during first thunderstorm case. In the second case (Figure 5(b)), observed relative humidity rise is from 66% to 100% (34%) at 1200 UTC, whereas LM indicated a rise from 68% to 100% (32%) at the same time. As in the previous case, the CG shows increase in relative humidity at 1400 UTC with 16% change in intensity. The changes in intensity predicted by other algorithms are also same and the intensity of sudden increase is 23%. For the third case (Figure 5(c)), observed relative humidity showed a rise from 63% to 100% (37%) at 1300 UTC, whereas LM showed a rise from 73% to 95% (22%). The other algorithms showed an intensity rise around 10%. From these analyses of temperature and relative humidity, we can see that ANN model with LM algorithm well predicted diurnal variation during thunderstorm days and captured the sudden drop and rise with almost same intensity of observation as compared to other algorithms.

The results of statistical analysis based on MAE, RMSE, and CC to evaluate forecasted temperature are shown in Table 1. The results of Table 1 indicated that LM algorithm has less MAE and RMSE as compared to all other algorithms for




(a) (b) (c)

Figure 4: Comparison of ANN predicted hourly surface temperature using different learning algorithms with observation on (a) May 3, 2009, (b) May 11, 2009, and (c) May 15, 2009.

Table 1: Performance comparison of different learning algorithms in hourly temperature prediction.

| Statistical analysis | Dates | STP | MOM | CG | LM | QKP | DBD |
|---|---|---|---|---|---|---|---|
| MAE | May 3, 09 | 3.36 | 3.24 | 2.69 | 2.08 | 3.48 | 3.41 |
| | May 11, 09 | 2.69 | 2.57 | 2.27 | 1.72 | 2.54 | 2.62 |
| | May 15, 09 | 2.93 | 2.66 | 2.08 | 1.21 | 2.69 | 2.90 |
| | Mean | **2.99** | **2.82** | **2.35** | **1.67** | **2.90** | **2.98** |
| RMSE | May 3, 09 | 3.54 | 3.50 | 2.90 | 2.35 | 3.70 | 3.69 |
| | May 11, 09 | 3.07 | 3.02 | 2.44 | 1.90 | 2.99 | 3.19 |
| | May 15, 09 | 3.07 | 2.76 | 2.20 | 1.41 | 2.78 | 3.04 |
| | Mean | **3.23** | **3.09** | **2.51** | **1.89** | **3.16** | **3.31** |
| CC | May 3, 09 | 0.82 | 0.82 | 0.90 | 0.93 | 0.79 | 0.80 |
| | May 11, 09 | 0.89 | 0.89 | 0.94 | 0.97 | 0.89 | 0.86 |
| | May 15, 09 | 0.74 | 0.80 | 0.91 | 0.96 | 0.80 | 0.74 |
| | Mean | **0.82** | **0.84** | **0.92** | **0.95** | **0.83** | **0.80** |

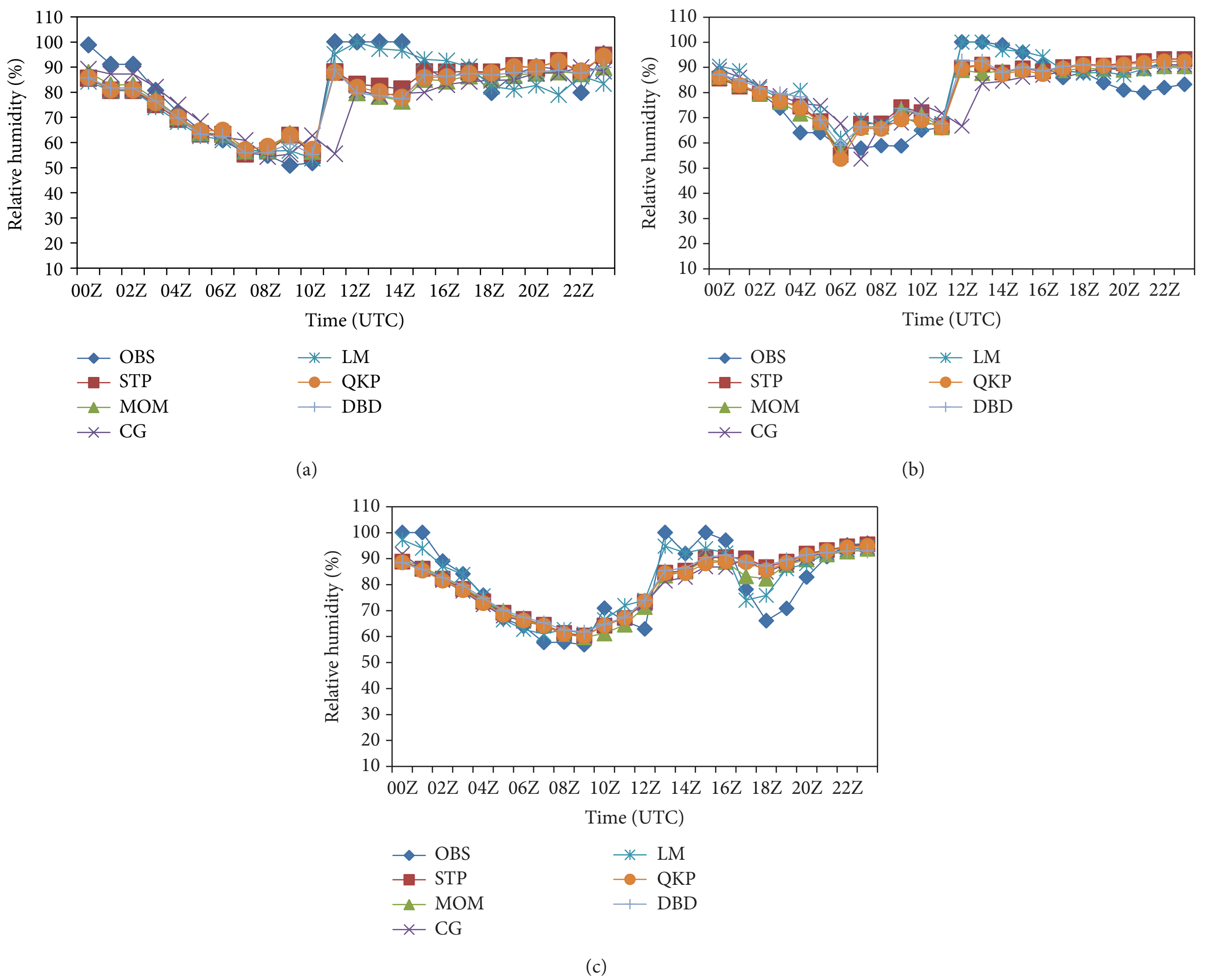



Figure 5: Comparison of ANN predicted hourly relative humidity using different learning algorithms with observation on (a) May 3, 2009, (b) May 11, 2009, and (c) May 15, 2009.

these 3 thunderstorm cases. The CG algorithm has also given moderate results. All other algorithms displayed more error in all thunderstorm cases as compared to LM and CG algorithms. The average MAE and RMSE from these 3 cases are also less for LM algorithm than other 5 algorithms. Another verification method used for this study is correlation coefficient. From the table we can clearly see that all the algorithms are positively correlated. The LM algorithm has the highest CC in all three cases as compared to all other algorithms. The average CC of LM and CG algorithms is more than 0.9. The CC of other algorithms is less than 0.85. The performance of DBD algorithm is less efficient than other algorithms. The analysis shows that LM algorithm is best for hourly temperature prediction over Kolkata during thunderstorm days.

The results of statistical analysis based on MAE, RMSE, and CC to evaluate forecasted relative humidity are shown in Table 2. The results of Table 2 indicated that LM algorithm has less error as compared to all other algorithms for these 3 thunderstorm cases as in temperature study. All other algorithms have also given moderate results except CG algorithm. The CG algorithm displayed more error in all cases. But in temperature prediction (Table 1), CG algorithm is performed well than other 4 algorithms, namely, STP, MOM, DBD, and QKP. The average MAE and RMSE of LM algorithm have least value than other 5 algorithms. From Table 2, we can clearly see that all the algorithms are positively correlated. The LM algorithm has the highest CC in all three cases as compared to all other algorithms. The average CC of 3 thunderstorm cases is more for LM algorithm which is more than 0.9. The CC of other algorithms except CG is 0.8 which is strong correlation. The performance of CG algorithm is less efficient than other algorithms for the prediction of hourly surface relative humidity during thunderstorm days. The results show that LM algorithm is best for hourly relative humidity prediction over Kolkata during thunderstorm days.

Figure 6 gives the performance accuracy of learning algorithms for hourly temperature and relative humidity prediction. The percent correct (PC) of temperature presented a percentage number of the times when the forecast is accurate to within $\pm 2^{\circ}$C. The results (Figure 6(a)) clearly indicated

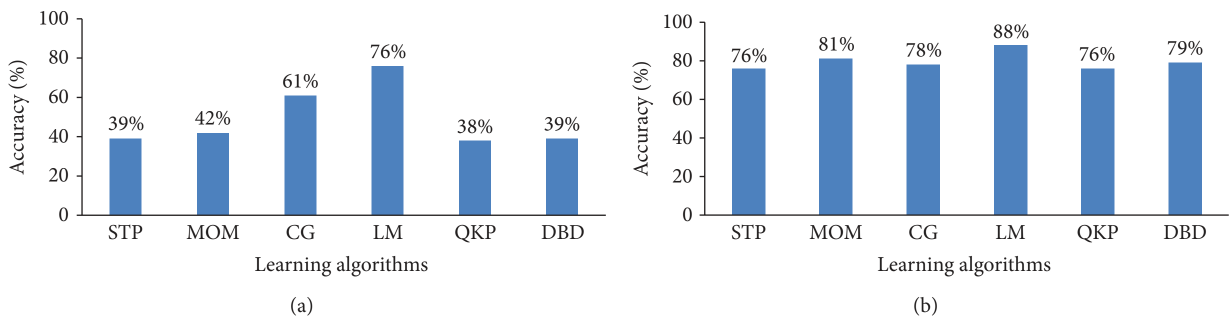



Figure 6: Performance accuracy of learning algorithms for the prediction of (a) temperature and (b) relative humidity during thunderstorm days.

Table 2: Performance comparison of different learning algorithms in hourly relative hunidity prediction.

| Statistical analysis | Dates | STP | MOM | CG | LM | QKP | DBD |
|---|---|---|---|---|---|---|---|
| MAE | May 3, 09 | 6.62 | 6.78 | 8.21 | 5.24 | 7.06 | 6.65 |
| | May 11, 09 | 6.80 | 5.77 | 8.46 | 5.20 | 6.19 | 6.26 |
| | May 15, 09 | 9.23 | 9.20 | 9.86 | 3.02 | 9.53 | 8.98 |
| | Mean | **7.55** | **7.25** | **8.84** | **4.49** | **7.60** | **7.30** |
| RMSE | May 3, 09 | 8.77 | 9.50 | 12.64 | 6.55 | 9.43 | 9.42 |
| | May 11, 09 | 7.99 | 7.06 | 10.71 | 6.76 | 7.25 | 7.30 |
| | May 15, 09 | 10.33 | 9.94 | 10.93 | 3.71 | 10.56 | 9.89 |
| | Mean | **9.03** | **8.83** | **11.43** | **5.67** | **9.08** | **8.87** |
| CC | May 3, 09 | 0.86 | 0.86 | 0.69 | 0.93 | 0.84 | 0.86 |
| | May 11, 09 | 0.84 | 0.88 | 0.64 | 0.95 | 0.86 | 0.89 |
| | May 15, 09 | 0.68 | 0.76 | 0.67 | 0.95 | 0.68 | 0.69 |
| | Mean | **0.80** | **0.83** | **0.67** | **0.95** | **0.80** | **0.82** |

that overall accuracy of LM algorithm for three events is 76%. CG gave a moderate accuracy of 61%. Other algorithms displayed less accuracy. The PC of relative humidity presented a percentage number of the times when the forecast is accurate to with ±10% confidence range. The results (Figure 6(b)) indicated that overall accuracy of LM algorithm for three events is 88%. MOM algorithm also shows a good accuracy of 81%. The other algorithms displayed a moderate accuracy, which is more than 75%. LM algorithm shows more accuracy for the prediction of hourly temperature and relative humidity.

The time-series plots and statistical analysis of temperature and relative humidity revealed that LM algorithm well predicted the occurrence and intensity of all 3 thunderstorm cases as in the observation. The results suggest that the ANN model with LM algorithm holds promise for prediction of surface weather parameters with reasonable accuracy in severe thunderstorm cases.

*4.2. Comparison of Different Advanced Predictions.* The developed ANN model was applied to derive thunderstorm forecast from 1 to 24 h ahead at Kolkata from the data of 3 consecutive years (2007–2009). The ANN models were created to predict surface temperature and relative humidity at hourly intervals with 1 h, 3 h, 6 h, 12 h, and 24 h ahead during severe thunderstorm cases, and the results are evaluated in the following section. Analysis of the results of these experiments is helpful to understand the efficiency of ANN model to predict severe thunderstorm events in advance and can apply operationally over east and northeast Indian region.

The comparison between observed and predicted surface temperature for 1 to 24 h advanced forecasting on May 3, 11, and 15, 2009, is shown in Figure 7. As seen in the figure, 1 h advanced forecast could forecast quite accurately. It was captured the sudden fall in temperature during thunderstorm hours for all 3 thunderstorm days. The 3 h forecast was in the next position and very close to the observation for the first case. Some deviations are there in the second and third case. The 6 and 12 h forecast failed to capture the entire pattern. For the first case (Figure 7(a)), the observed temperature showed a sudden drop of 15°C from 36.7°C to 21.7°C at 1000 UTC. The 1 h ahead forecast model showed a drop from 33°C to 22°C (11°C) at 1000 UTC, whereas 3 h presented a drop from 32°C to 25°C (7°C) at 1000 UTC. In the second thunderstorm case (Figure 7(b)), observed temperature fall is from 33.1°C to 21.7°C (11°C) at 1200 UTC, whereas 1 h ahead forecast model indicated a drop from 31°C to 21°C (10°C) at the same thunderstorm hour. The other models failed to capture a sudden fall during thunderstorm hour. For the third case (Figure 7(c)), observed temperature showed a drop from 29°C to 24°C (6°C) at 1300 UTC, whereas 1 h advanced prediction model showed a drop from 32°C to 27°C (5°C).

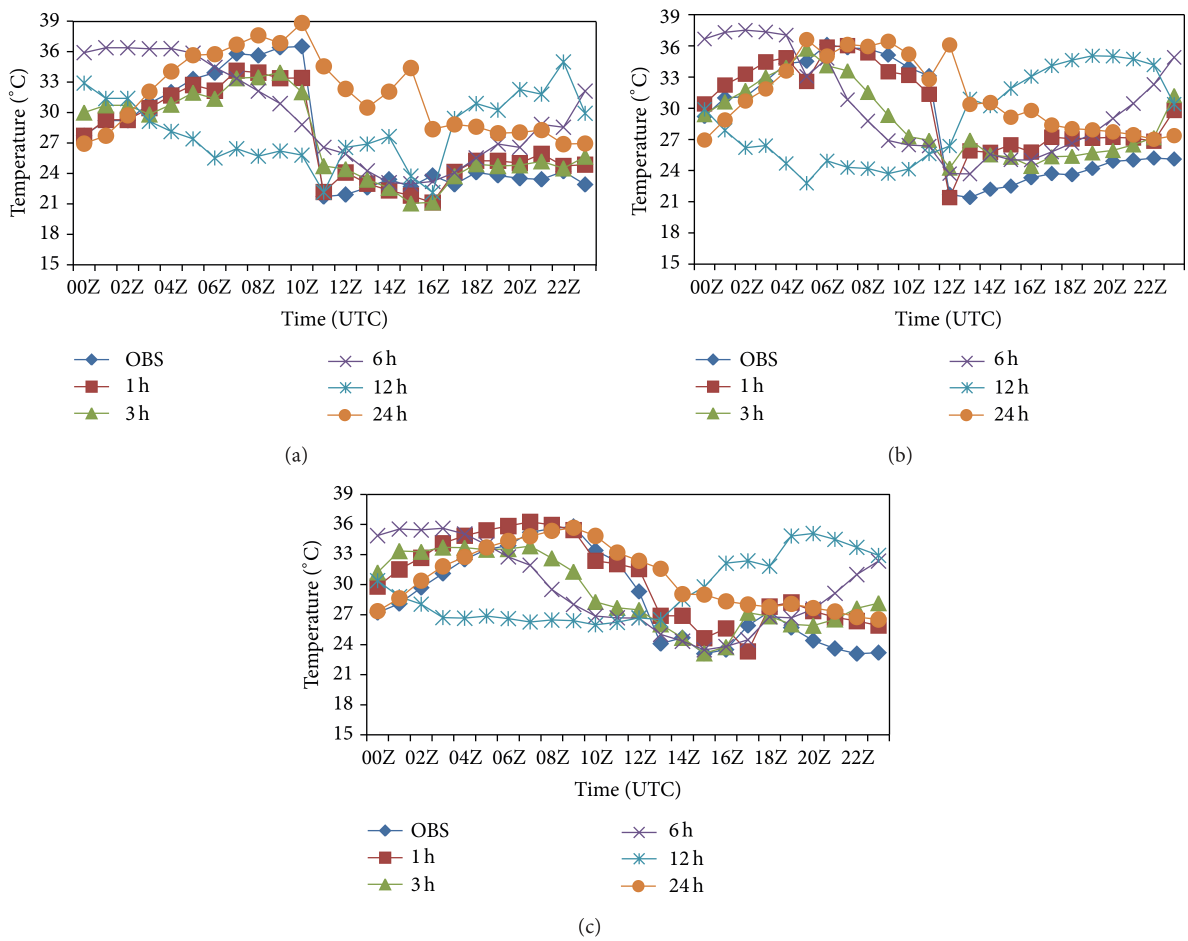



Figure 7: Comparison of ANN predicted hourly temperature using different advanced prediction models with observation on (a) May 3, 2009, (b) May 11, 2009, and (c) May 15, 2009.

The temperature forecast at 24 h ahead model also gave good results as compared to 6 and 12 h ahead forecasts. The hourly temperature variation of 24 h ahead model shows a small fall with an intensity of 6°C at 1600 and 1300 UTC for the first two thunderstorm cases and 3°C at 1400 UTC for the third thunderstorm case. This model captured sudden fall with 6-hour time lag for the first cases and 1 hour for second and third cases.

The statistical analyses of the ANN model performance for the advanced prediction of surface temperature during thunderstorm days are given in Table 3. Both MAE and RMSE are less for 1 and 3 h advanced prediction and also have a high positive correlation. The errors are high and have low correlation (+ve or −ve correlation) for 6 and 12 h advanced predictions. The 24 h ahead forecast models are better performed than 6 and 12 h ahead prediction and the average correlation is 0.70. The results were highly satisfactory for temperature forecast with 1 to 3 h ahead. The 6 and 12 h ahead forecast accuracy were very poor as compared to 1, 3, and 24 h. The percent correct of these 5 models is given in Figure 9(a). The figure clearly indicates that overall accuracy of 1 h ahead forecast for three events is 70%. The 3 and 24 h ahead forecast models are also close to this with 65% and 54%. The other two models (6 and 12 h) displayed less accuracy.

The comparison between observed and predicted relative humidity for 1 to 24 h ahead forecasting on May 3, 11, and 15, 2009, is shown in Figure 8. The results show 1 h ahead forecast captured sudden increase in relative humidity at thunderstorm hours during all three thunderstorm cases. The 3 h advanced prediction model was able to predict the rise in relative humidity during thunderstorm hour in the first two cases. The 24 h forecast was also close to the observation for the second thunderstorm case, even though one hour time lag exists. The 6 and 12 h forecast failed to capture the entire pattern for all 3 cases as in temperature prediction. In the first case (Figure 8(a)), the observed relative humidity showed a rise of 48% from 52% to 100% at 1000 UTC. The 1 h advanced ANN model prediction shows a rise from 60% to 97% (37%) at 1000 UTC. The 3 h ahead model showed a rise with an intensity of 20% at 1000 UTC. The other models failed to capture sudden rise as in 1 and 3 h ahead forecast. In the second case (Figure 8(b)), observed relative humidity rise is from 66% to 100% (34%) at 1200 UTC, whereas 1 h ahead model indicated a rise from 70% to 88% (18%) at the same time. The intensity of increase is less (12%) for 3 h ahead model and the other models failed to capture the sudden rise during thunderstorm hour. The 24 h ahead model showed a sudden rise from 61% to 92% (31%) with one-hour time lag

Table 3: Performance comparison of different advanced predictions for hourly temperature during thunderstorm days.

| Statistical analysis | Date | 1 h | 3 h | 6 h | 12 h | 24 h |
|---|---|---|---|---|---|---|
| MAE | May 3, 09 | 1.34 | 1.88 | 4.04 | 6.06 | 4.61 |
| | May 11, 09 | 2.10 | 2.44 | 4.62 | 8.47 | 3.21 |
| | May 15, 09 | 2.07 | 2.32 | 3.84 | 6.54 | 2.11 |
| | Mean | **1.84** | **2.22** | **4.17** | **7.02** | **3.31** |
| RMSE | May 3, 09 | 2.36 | 2.13 | 4.94 | 6.81 | 6.35 |
| | May 11, 09 | 2.42 | 3.06 | 5.17 | 8.97 | 4.71 |
| | May 15, 09 | 2.29 | 2.95 | 4.82 | 7.19 | 2.93 |
| | Mean | **2.35** | **2.71** | **4.98** | **7.66** | **4.66** |
| CC | May 3, 09 | 0.97 | 0.93 | 0.63 | −0.19 | 0.53 |
| | May 11, 09 | 0.95 | 0.82 | 0.48 | −0.80 | 0.70 |
| | May 15, 09 | 0.94 | 0.76 | 0.38 | −0.77 | 0.91 |
| | Mean | **0.95** | **0.84** | **0.50** | **−0.59** | **0.71** |

Table 4: Performance comparison of different advanced predictions for hourly relative humidity during thunderstorm days.

| Statistical analysis | Date | 1 h | 3 h | 6 h | 12 h | 24 h |
|---|---|---|---|---|---|---|
| MAE | May 3, 09 | 9.08 | 15.72 | 19.26 | 21.39 | 20.06 |
| | May 11, 09 | 5.96 | 7.43 | 13.26 | 20.04 | 11.65 |
| | May 15, 09 | 8.03 | 11.02 | 14.54 | 17.40 | 7.31 |
| | **Mean** | **7.69** | **11.39** | **15.69** | **19.61** | **13.00** |
| RMSE | May 3, 09 | 10.63 | 18.73 | 25.01 | 24.59 | 27.65 |
| | May 11, 09 | 7.73 | 10.58 | 15.47 | 21.63 | 14.05 |
| | May 15, 09 | 9.98 | 12.55 | 16.94 | 19.67 | 8.71 |
| | **Mean** | **9.45** | **13.95** | **19.14** | **21.96** | **16.80** |
| CC | May 3, 09 | 0.92 | 0.77 | 0.40 | −0.39 | 0.45 |
| | May 11, 09 | 0.85 | 0.70 | 0.19 | −0.65 | 0.61 |
| | May 15, 09 | 0.76 | 0.61 | 0.08 | −0.40 | 0.72 |
| | **Mean** | **0.84** | **0.69** | **0.22** | **−0.48** | **0.59** |

for this thunderstorm event. For the third case (Figure 8(c)), observed relative humidity showed a rise from 63% to 100% (37%) at 1300 UTC, whereas 1 h ahead model showed a rise from 70% to 92% (22%). The 3 and 24 h ahead models showed an intensity of rise around 3% and 12%, respectively. From these analyses of temperature and relative humidity, we can see that 1 h advanced prediction model well predicted diurnal variation during thunderstorm days and captured the sudden drop and rise with almost same intensity of observation as compared to other models.

The statistical analyses of the ANN model performance for the advanced prediction of hourly relative humidity during thunderstorm days are given in Table 4. Both MAE and RMSE are less for 1, 3, and 24 h advanced prediction. The highest correlation coefficient is for 1 h advanced prediction model (0.80). The 3 and 24 h model have also a positive correlation. The errors are high and have low correlation for 6 and 12 h advanced prediction models as in the temperature forecast case. The results were satisfactory for relative humidity forecast with 1, 3, and 24 h ahead as in temperature forecast. The percent correct (PC) of these 5 models is given in Figure 9(b). The figure clearly indicates that overall accuracy of 1 h ahead forecast for three events is 74%. The 3 and 24 h ahead forecast models are also close to this with 61% and 54%. The 6 and 12 h ahead model forecast accuracy were very poor as compared to other three advanced prediction models.

The models developed in this section show how surface temperature and relative humidity can be predicted for 1 to 24 h ahead with an ANN model. Although the results varied, the 1 and 3 h ahead ANN models were able to predict hourly temperature and relative humidity adequately with sudden fall and rise. Even 24 h advanced prediction model can able to predict features of thunderstorm with reasonable accuracy. Although the model performance of 6 and 12 h forecasting was low and the forecasting was not as accurate as expected, the developed model can still useful in decision making for meteorologists and others who work with real-time thunderstorm forecast.

## 5. Conclusions

The severe thunderstorms have significant socio-economic impact over eastern and northeastern parts of India. The improvement in prediction of these important weather phenomena is highly handicapped due to lack of mesoscale observations and insufficient understanding. The recent

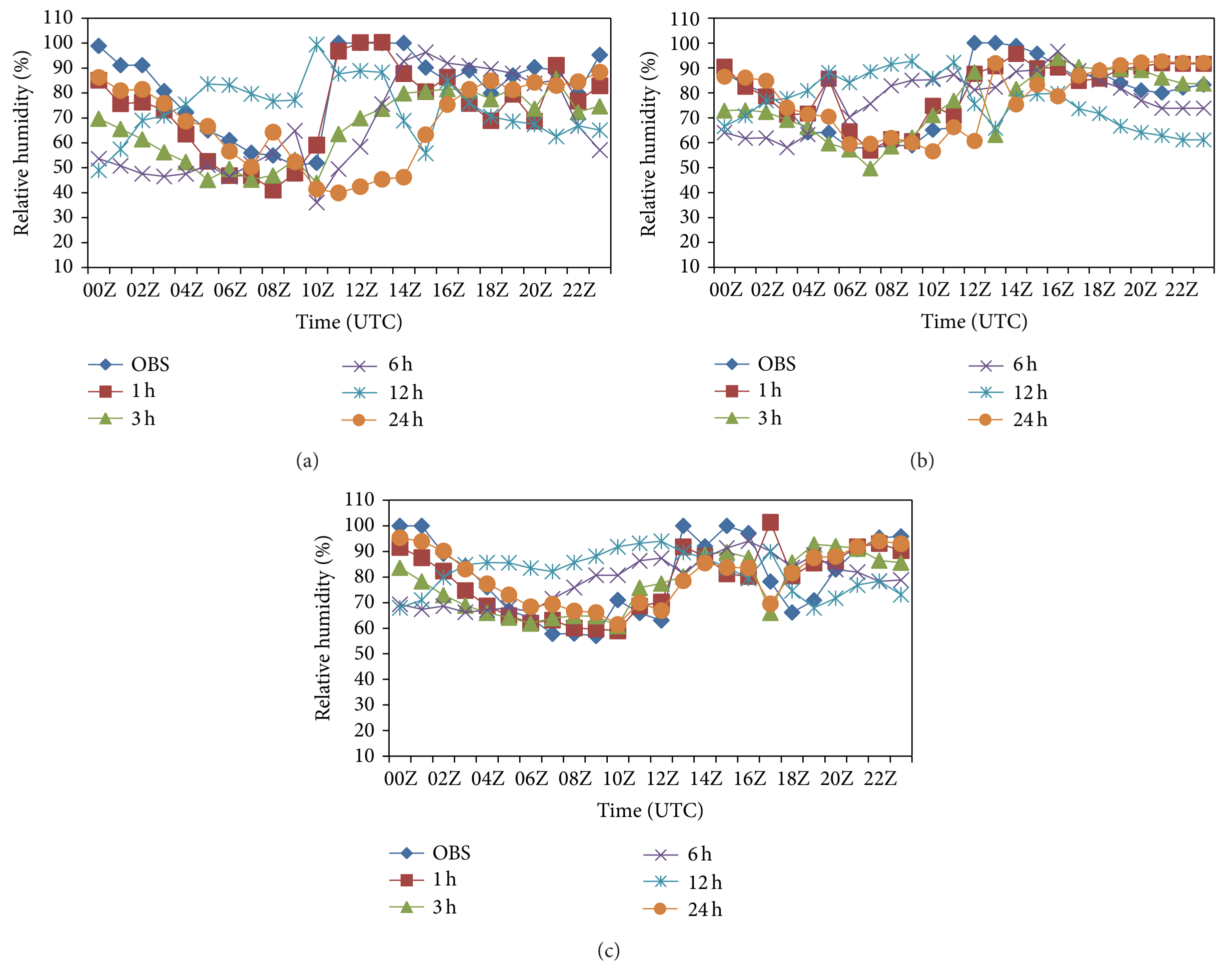



Figure 8: Comparison of ANN predicted hourly relative humidity using different advanced prediction models with observation on (a) May 3, 2009, (b) May 11, 2009, and (c) May 15, 2009.

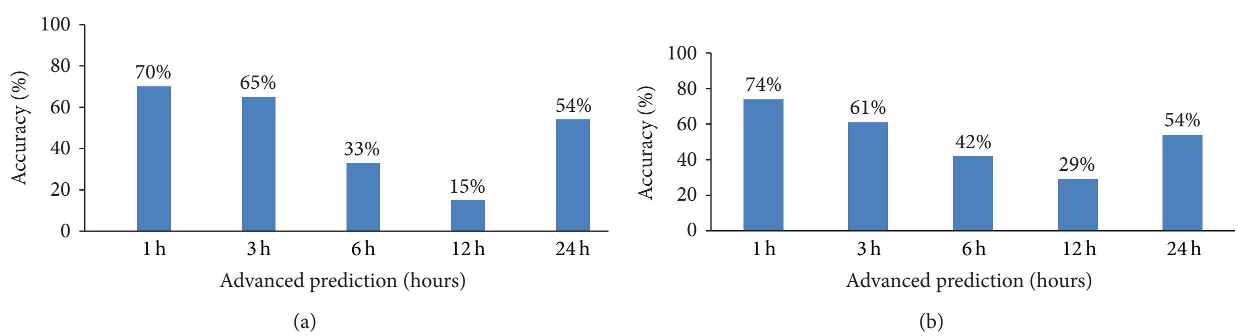



Figure 9: Performance accuracy of different advanced prediction models for the prediction of (a) temperature and (b) relative humidity during thunderstorm days.

advances in neural network methodology for modeling nonlinear, dynamical phenomena along with the impressive successes in a wide range of applications are motivated to investigate the application of ANNs for the prediction of thunderstorms. ANN has capability to extract the relationship between the inputs and outputs of a process, without the physics being explicitly provided [45]. This study evaluates the utility of ANN for estimating hourly surface temperature and relative humidity.

In this paper, sensitivity experiments have been conducted with ANN model to test the impact of learning algorithms on severe thunderstorm prediction that occurred over Kolkata on May 3, 11, and 15, 2009, and selected LM algorithm for further studies. The developed ANN model with LM algorithm was applied to derive thunderstorm forecasts from 1 to 24 h ahead at Kolkata. The objective of this study was to use ANNs to predict temperature and relative humidity during thunderstorm days from 1 to 24 h ahead using prior weather data as inputs. A statistical analysis based on MAE, RMSE, CC, and PC is also performed for comparison among predicted and observed data with different learning algorithms and advanced predictions.

The model setups were identical except for the use of different learning algorithms for the sensitivity experiments of learning algorithms. Hence the differences in the prediction results attributed to the sensitivity of learning algorithms. LM algorithm appears to be the best learning algorithm for mapping the different chaotic relationships. After analyzing the results, we can conclude that the ANN model with LM algorithm has well predicted the hourly temperature and relative humidity in terms of sudden fall of temperature and rise of humidity during thunderstorm hours. The developed ANN model with LM algorithm was used to predict surface temperature and relative humidity at hourly intervals with 1, 3, 6, 12, and 24 h ahead during same severe thunderstorm cases. Analysis of the results reveals that the 1, 3, and 24 h ANN models were able to predict hourly temperature and relative humidity adequately with sudden fall and rise. The efficiency of ANN models is reduced as the forecast lead time increased from 6 to 12 h. It can be inferred that ANN could yield more accurate results, if good data selection strategies, training paradigms, and network input and output representations are determined properly. In future, we would like to use more networks like an elman recurrent neural network (ERNN) and radial basis function network (RBFN) to examine their applicability in thunderstorm forecasting.

## Acknowledgments

The authors would like to express their sincere gratitude to the data providers specially India Meteorological Department (IMD). The authors express sincere thanks to the reviewer for providing invaluable suggestions to improve this paper.

## References

[1] A. J. Litta, U. C. Mohanty, S. Das, and S. M. Idicula, "Numerical simulation of severe local storms over East India using WRF-NMM mesoscale model," *Atmospheric Research*, vol. 116, pp. 161–184, 2012.

[2] I. Orlanski, "A rational subdivision of scales for atmospheric processes," *Bulletin of the American Meteorological Society*, vol. 56, pp. 527–530, 1975.

[3] J. W. Wilson, N. A. Crook, C. K. Mueller, J. Sun, and M. Dixon, "Nowcasting thunderstorms: a status report," *Bulletin of the American Meteorological Society*, vol. 79, no. 10, pp. 2079–2099, 1998.

[4] P. Chatterjee, D. Pradhan, and U. K. De, "Simulation of hailstorm event using mesoscale model MM5 with modified cloud microphysics scheme," *Annales Geophysicae*, vol. 26, no. 11, pp. 3545–3555, 2008.

[5] M. Rajeevan, A. Kesarkar, S. B. Thampi, T. N. Rao, B. Radhakrishna, and M. Rajasekhar, "Sensitivity of WRF cloud microphysics to simulations of a severe thunderstorm event over Southeast India," *Annales Geophysicae*, vol. 28, no. 2, pp. 603–619, 2010.

[6] A. J. Litta, U. C. Mohanty, and S. M. Idicula, "The diagnosis of severe thunderstorms with high-resolution WRF model," *Journal of Earth System Science*, vol. 121, no. 2, pp. 297–316, 2012.

[7] A. J. Litta, S. M. Ididcula, U. C. Mohanty, and S. K. Prasad, "Comparison of thunderstorm simulations from WRF-NMM and WRF-ARW over East Indian region," *The Scientific World Journal*, vol. 2012, Article ID 951870, 20 pages, 2012.

[8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan Publishing Company, New York, NY, USA, 1994.

[10] J. W. Shavlik, R. J. Mooney, and G. G. Towell, "Symbolic and neural learning algorithms: an experimental comparison," *Machine Learning*, vol. 6, no. 2, pp. 111–143, 1991.

[11] R. Setiono, W. K. Leow, and J. M. Zurada, "Extraction of rules from artificial neural networks for nonlinear regression," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 564–577, 2002.

[12] D. L. Bailey and D. M. Thompson, "Developing neural network applications," *AI Expert*, no. 9, pp. 34–41, 51990.

[13] C. D. James, R. Davis, M. Meyer et al., "Aligned microcontact printing of micrometer-scale poly-L-lysine structures for controlled growth of cultured neurons on planar microelectrode arrays," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 1, pp. 17–21, 2000.

[14] V. Palade, D. Neagu, and R. J. Patton, "Interpretation of trained neural networks by rule extraction," in *Computational Intelligence: Theory and Applications*, vol. 2206 of *Lecture Notes in Computer Science*, pp. 152–161, Springer, New York, NY, USA, 2001.

[15] L. Bodri and V. Čermák, "Prediction of extreme precipitation using a neural network: application to summer flood occurrence in Moravia," *Advances in Engineering Software*, vol. 31, no. 5, pp. 311–321, 2000.

[16] K. C. Luk, J. E. Ball, and A. Sharma, "A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting," *Journal of Hydrology*, vol. 227, no. 1–4, pp. 56–65, 2000.

[17] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing and Applications*, vol. 13, no. 2, pp. 112–122, 2004.

[18] S. Chaudhuri and S. Chattopadhyay, "Neuro-computing based short range prediction of some meteorological parameters during the pre-monsoon season," *Soft Computing*, vol. 9, no. 5, pp. 349–354, 2005.

[19] S. Chattopadhyay, "Feed forward artificial neural network model to predict the average summer-monsoon rainfall in India," *Acta Geophysica*, vol. 55, no. 3, pp. 369–382, 2007.

[20] C. T. Cheng, K. W. Chau, Y. G. Sun, and J. Y. Lin, "Long-term prediction of discharges in manwan reservoir using artificial neural network models," in *Advances in Neural Networks—ISNN 2005*, vol. 3498 of *Lecture Notes in Computer Science*, pp. 1040–1045, 2005.

[21] K. W. Chau, C. L. Wu, and Y. S. Li, "Comparison of several flood forecasting models in Yangtze River," *Journal of Hydrologic Engineering*, vol. 10, no. 6, pp. 485–491, 2005.

[22] N. Muttil and K. W. Chau, "Neural network and genetic programming for modelling coastal algal blooms," *International Journal of Environment and Pollution*, vol. 28, no. 3-4, pp. 223–238, 2006.

[23] W. Chen and K. W. Chau, "Intelligent manipulation and calibration of parameters for hydrological models," *International Journal of Environment and Pollution*, vol. 28, no. 3-4, pp. 432–447, 2006.

[24] R. Taormina, K. Chau, and R. Sethi, "Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1670–1676, 2012.

[25] C. L. Wu, K. W. Chau, and Y. S. Li, "Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques," *Water Resources Research*, vol. 45, no. 8, Article ID W08432, 2009.

[26] M. R. Khan and C. Ondrusek, "Short-term electric demand prognosis using artificial neural networks," *Journal of Electrical Engineering*, vol. 51, pp. 296–300, 2000.

[27] Neuro Dimension Inc., *Developers of Neuro Solutions V5.01: Neural Network Simulator*, Neuro Dimension Inc., Gainesville, Fla, USA, 2005, http://www.nd.com/.

[28] L. Fausett, *Fundamental of Neural Networks*, Prentice Hall, New York, NY, USA, 1994.

[29] S. E. Fahlman, "An empirical study of learning speed in back-propagation networks," Computer Science Technical Report CMU-CS-88-162, Carnegie Mellon University, Pittsburgh, Pa, USA, 1988.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[31] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks*, vol. 1, no. 4, pp. 295–307, 1988.

[32] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[33] Ö. Kişi and E. Uncuoğlu, "Comparison of three back-propagation training algorithms for two case studies," *Indian Journal of Engineering and Materials Sciences*, vol. 12, no. 5, pp. 434–442, 2005.

[34] S. Esugasini, M. Y. Mashor, N. A. M. Isa, and N. H. Othman, "Performance comparison for MLP networks using various back propagation algorithms for breast cancer diagnosis," in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 3682 of *Lecture Notes in Computer Science*, pp. 123–130, Springer, New York, NY, USA, 2005.

[35] M. M. Raju, R. K. Srivastava, D. C. S. Bisht, H. C. Sharma, and A. Kumar, "Development of artificial neural-network-based models for the simulation of spring discharge," *Advances in Artificial Intelligence*, vol. 2011, Article ID 686258, 11 pages, 2011.

[36] E. D. Übeyli and I. Güler, "Multilayer perceptron neural networks to compute quasistatic parameters of asymmetric coplanar waveguides," *Neurocomputing*, vol. 62, no. 1–4, pp. 349–365, 2004.

[37] U. C. Mohanty, D. R. Sikka, S. K. Prasad et al., "Weather summary, analysis and preliminary evaluation of mesoscale model during pilot experiment of severe thunderstorms," Observational and Regional Modeling (STORM) Programme, 2009.

[38] C. A. Doswell, "The distinction between large-scale and mesoscale contribution to severe convection: a case study example," *Weather and Forecasting*, vol. 2, pp. 3–16, 1987.

[39] R. H. Johns and C. A. Doswell, "Severe local storms forecasting," *Weather and Forecasting*, vol. 7, no. 4, pp. 588–612, 1992.

[40] R. P. McNulty, "Severe and convective weather: a central region forecasting challenge," *Weather and Forecasting*, vol. 10, no. 2, pp. 187–202, 1995.

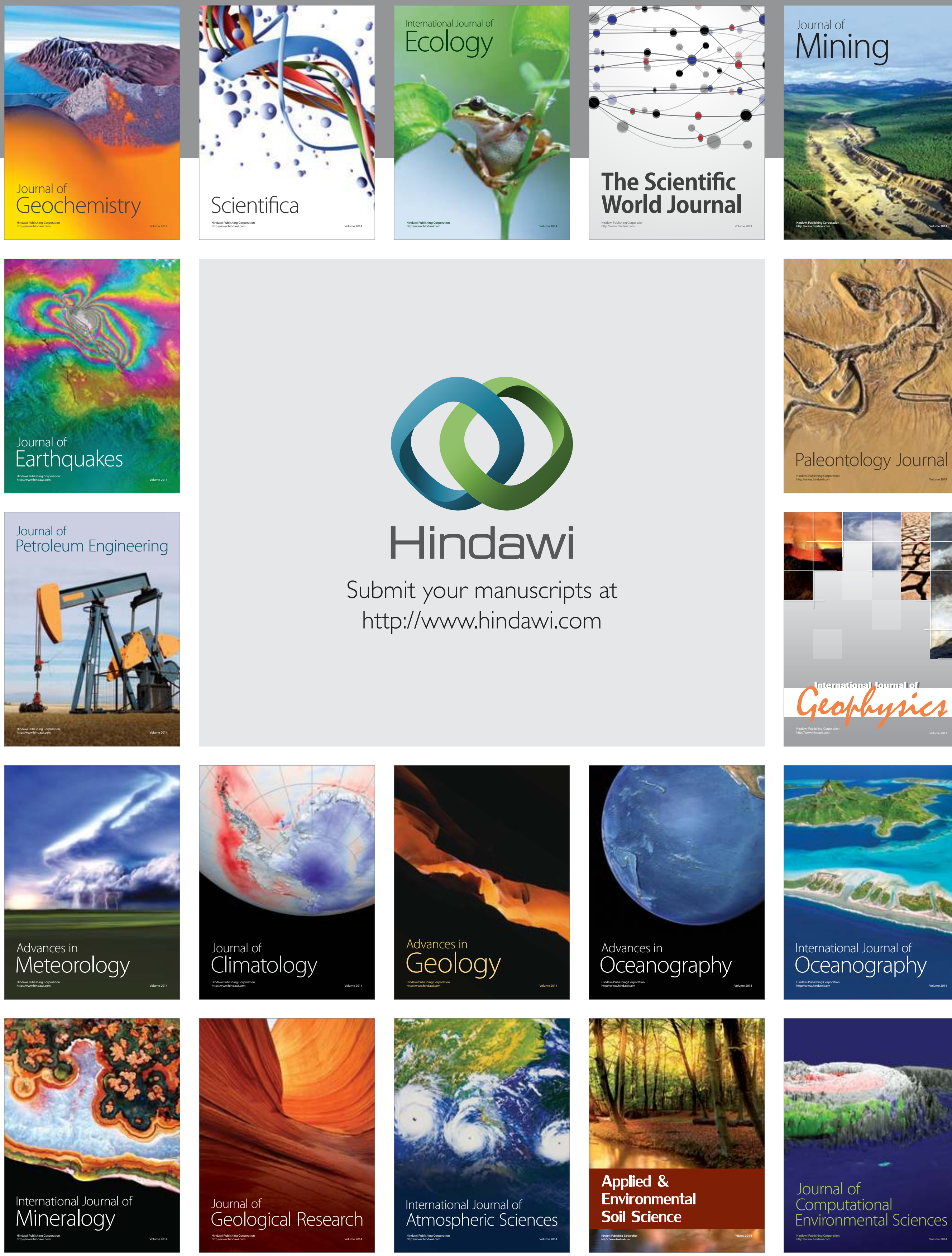
[41] G. C. Asnani, *Tropical Meteorology*, Sindh Colony, Aundh, India, 2006.

[42] C. Price, "Global thunderstorm activity," in *Sprites, Elves and Intense Lightning Discharges*, M. Fullekrug, E. A. Mareev, and M. J. Rycroft, Eds., pp. 85–99, 2006.

[43] E. R. Williams, "Lightning and climate: a review," *Atmospheric Research*, vol. 76, no. 1–4, pp. 272–287, 2005.

[44] L. López, E. García-Ortega, and J. L. Sánchez, "A short-term forecast model for hail," *Atmospheric Research*, vol. 83, no. 2–4, pp. 176–184, 2007.

[45] J. M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing Company, Saint Paul, Minn, USA, 1992.


Journal of Geochemistry
Scientifica
International Journal of Ecology
The Scientific World Journal
Journal of Mining
Journal of Earthquakes
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Paleontology Journal
Journal of Petroleum Engineering
International Journal of Geophysics
Advances in Meteorology
Journal of Climatology
Advances in Geology
Advances in Oceanography
International Journal of Oceanography
International Journal of Mineralogy
Journal of Geological Research
International Journal of Atmospheric Sciences
Applied & Environmental Soil Science
Journal of Computational Environmental Sciences