Artificial Neural Networks for Prediction of severe weather systems over Guwahati, North – Eastern Region of India

[1]Y.Md.Riyazuddin, [2]S.Mahaboob Basha, [3]K.Krishna Reddy

[1]PP.CSE.0393, Research Scholar-CSE, Rayalaseema University. Kurnool,Andhra Pradesh India
[2]Dept of CSE,Al Habeeb Engneering College,Hyderabad,Telangana,India
[3]Dept of Physics,Yogi Vemena University, Kadapa ,Andhra Pradesh,India

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 fora large range of users in the community.In this paper, a study is conducted on artificial neural network (ANN) model (adopted Litta et al. 2012 procedure) to predict severe thunderstorms that occurred over Guwahati during pre-monsoon of 2010 by utilizing Severe thunderstorm observations and Regional Modelling (STROM) data collected by the Yogi Vemana University, Kadapa. To identity thunderstorm and its temporal and spatial resolution a Micro Rain Radar data is utilized. The capabilities of six 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. A statistical analysis based on mean absolute error, root mean square error, correlation coefficient and percentage of correctness is performed to compare the predicted and observed data. The results show that the ANN model with Levenberg Marquardt algorithm predicted the thunderstorm activities well in terms of sudden fall of temperature and intensity as compared to other five algorithms. Our results are in good agreement with Litta et al., 2013 observations and prediction of thunderstorms over Kolkata but their maximum occurrence was in early hours of the day at Guwahati whereas their occurrence in the evenings in West Bangal region.

***Keywords*: artificial neural network, thunderstorms, severe weather, Ground-based Meteorological Sensors**

1. Introduction:

Severe and extreme weather is a major natural hazard all over the world, often resulting in major natural disasters such as thunderstorms, tornados, wind storms, flash floods, and forest fires damages. In the last 50 years there has been a significant increase in the damage caused by extreme weather events, with the damage from extreme weather events increasing every decade, and the economic costs dramatically increasing during the 1990s. The cause of this large increase is debatable. On the one hand, there is evidence that the global climate is changing, with increases in extreme weather events observed over the past 50 years, and further increases are predicted for the future. On the other hand, with the growing population of the world, not only are extreme weather events better detected and reported, but more people are living in harm's way [1].

In India, thunderstorm is an important weather phenomenon particularly during pre-monsoon (April and May) and post monsoon seasons over the Indian region. The thunderstorms reach severity when continental air meets warm moist air from ocean in the lower troposphere [2,3]. The eastern and north eastern part of the country i.e. Bihar, Gangetic West Bengal, Jharkhand, Orissa, Assam and other states of North East Region of India (NERI) gets most affected by severe thunderstorms during pre-monsoon months (March-May), in particular, during April & May. In this season, a lot of thunderstorms occur over NERI, Bangladesh, Nepal, and Bhutan. They are called Nor'westers because they usually propagate from the northwest to the southeast. These severe thunderstorms cause devastative damages in this region in April and May, every year [4].

The improvement in the prediction of this severe weather phenomenon has been done in this work using empirical and dynamical approaches. The most widely used empirical approach for weather prediction is artificial neural network (ANN). ANN based approach can be used to model complex relationships between inputs and outputs or to find patterns in data. The recent advances in neural network methodology for modeling nonlinear, dynamical phenomena along with the impressive successes in a wide range of applications, motivated to investigate the application of ANNs for the prediction of thunderstorms. The techniques for predicting thunderstorms can beclassified into two groups (a) the empirical approach and (b)the dynamical approach [5]. First method is a historical treatment of thunderstorm extrapolation tech niques [knowledgebased expert systemsincluding fuzzy logic and artificialneural network (ANN)]. The second method is pr edictionusing high resolution numerical weather prediction (NWP)models. The second approach for studying thunderstorms isalr eady specified by many researchers.

Most weatherprediction systems use a combination of empirical anddynamical techniques. However, a little attent ion has beenpaid to the use of ANNs in thunderstorm forecasting. ANN- based approach can be used to model complex relationships. 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 Thomp-son [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 excel-lent 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.

In this paper, experiments are conducted with ANN model to predict severe thunderstorms that occurred over Guwahati (26.1065° N, 91.5860° E) using STROM-2010 thunderstorm affected meteorological parameters. The geographical location of the study location 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.

Fig.1: The geographical location of Guwahati in NERI..

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. Instrumentation, Data and Methodology:

2.1. Experimental Setup.

Fig. 2 Observational site showing instrumental facility installed at Guwahati

The present study is carried out by installing a vertical profiling Micro Rain Radar (MRR) and Parisivel Disdrometer (PSD) in the premises of Regional Meteorological Centre (RMC), India Meteorological Department (IMD), Guwahati, a North East (NE) region of India (**Figure 2**). During the field campaign MRR was operated with vertical resolution of 200 m, temporal resolution of 1 min and the Parisivel Disdrometer is operated with 1 min integration time. These two instruments were installed with a separation of 1.5 meters on the top of the Radiosonde/ Rawinsonde (RSRW) building of the regional meteorological center (RMC), India meteorological department (IMD), Guwahati. IMD meteorological instruments like Radiosonde/Rawinsonde (RS/RW), and X-band Radar data also used.

2.2. 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 feed forward 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 feed forward 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.

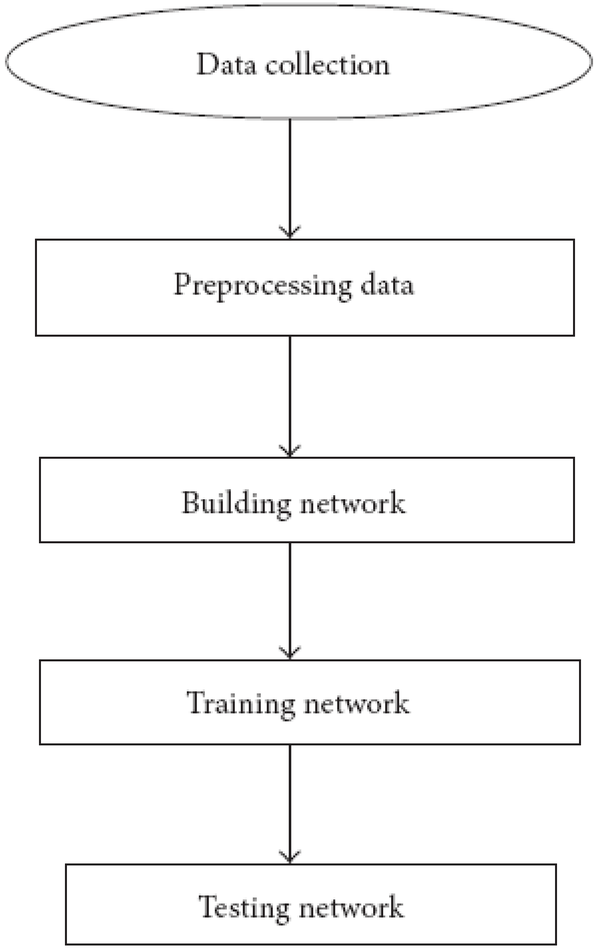


Fig. 3: Basic flow for designing ANN model.

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)pre-processing data, (3) building the network, (4) training and (5) test performance of model. The basic flow in designing ANN model is given in Figure 3. The weather data, namely, hourly mean sea level pressure, relative humidity, and wind speed of 3 years (April and May 2008 to 2010) collected from the India meteorological department (IMD) of Guwahati, were used as the input data for training and testing ANN model which will be used for the prediction of hourly temperature . Major numbers of thunderstorms are occurred over Guwahati 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 pre processing 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 neighbouring 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.

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 4 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.

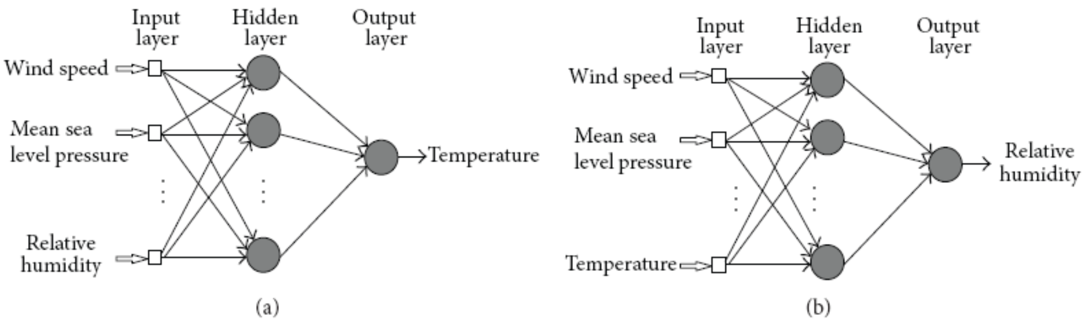


Fig.4: Architecture of multilayer perceptron network for the prediction of (a) temperature and (b) relative humidity.

The 24 h ANN model outputs of surface temperature and relative humidity at Guwhati during two thunderstorm days of May 2010 (0n 22nd May and 27th, 2010) 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.

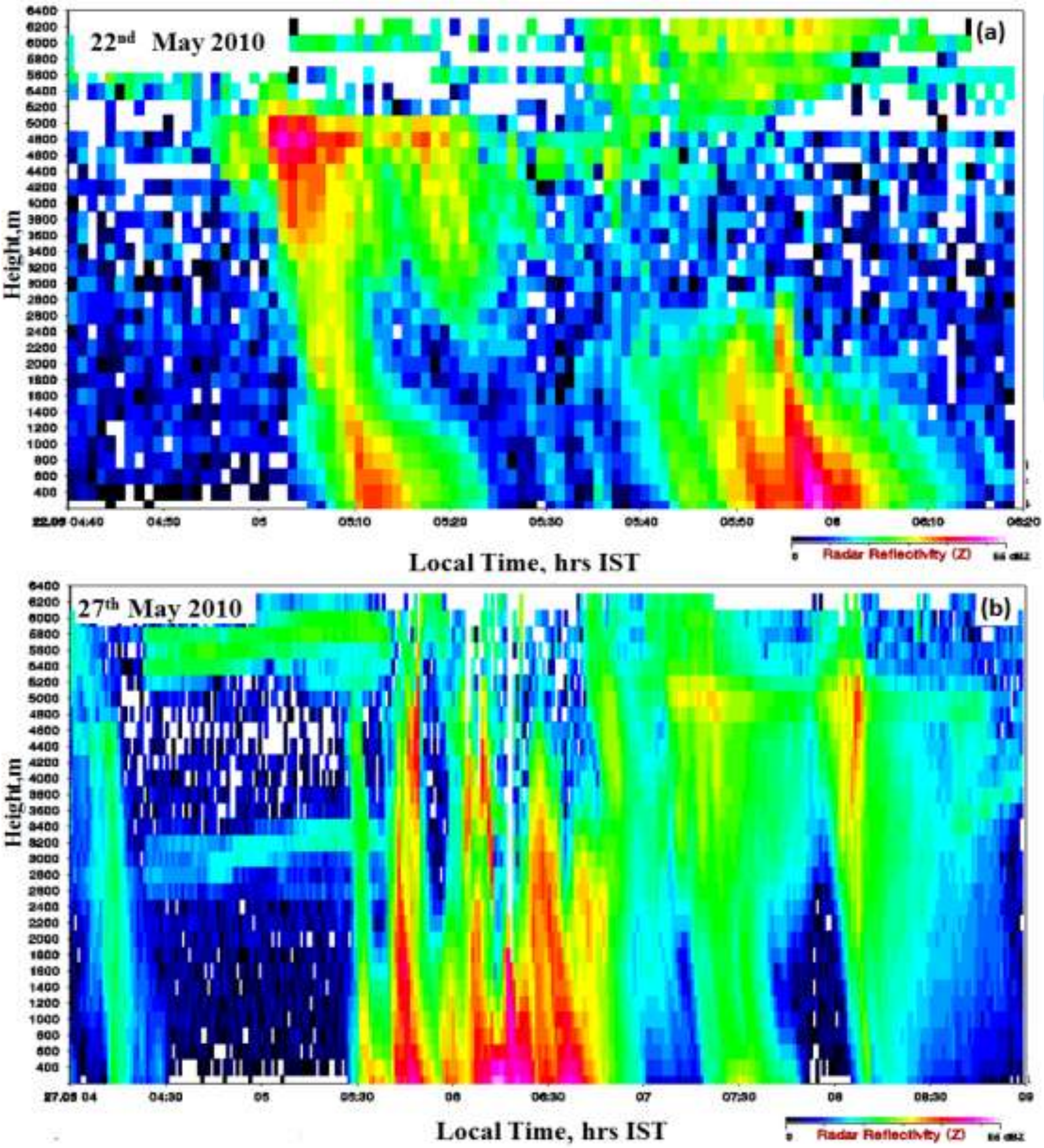


Fig.5: Time-height-cross-section of vertical structure of thunderstorm precipitating clouds observed on (a) 22nd May 2010 and (b) 27h May 2010.

The vertical profile of radar reflectivity obtained from MRR is given in fig. 5. The figure was generated based on1-min temporal and 200-m vertical resolutions of Radar reflectivity, dBZ, during the passage of two-thunderstorms (a) on 22nd May 2010 between 04:40 to 06:20 hrs, IST and 03:00 hrs to 09:00 hours and (b) on 27th may 2010. From this figure we can clearly identify different cloud fractions such as shallow, deep, mixed (Convective/Stratifrom) and Strati form precipitating rainfall. This informs is very much essential to identify initiation, propagation and dissipation of thunderstorms.

3. LEARNING ALGORITHMS

In neural network, the learning algorithms plays quite important role in the process. Throughout the process, the learning algorithm is used to adjust the weight, bias and other input parameters in such a manner that the model is able to count its best fit with the environment in a minimum amount of time. Conventional back propagation learning algorithms are often too slow for practical problems, so high performance algorithms that can converge from ten to one hundred times faster than back propagation algorithms were used. These faster algorithms fall into two main categories: The first category uses heuristic techniques developed from an analysis of the performance of the standard steepest descent algorithm. The second category uses standard numerical optimization techniques. The first category includes the gradient descent with adaptive learning rate, gradient descent with momentum, gradient descent with momentum and adaptive learning rate, and the resilient algorithm. In the standard steepest descent, the learning rate is fixed and its optimal value is always hard to find. The heuristic techniques allow the optimal learning rate to adaptively change during the learning process as the algorithm moves across the performance surface. Therefore, the performance could be improved. The second category includes CG, Quasi Newton (QN), and LM algorithm. In the CG algorithm, a search is performed along conjugate directions; therefore the convergence is faster than steepest descent directions. The QN method often converges faster than CG methods since it does not require calculation of second derivatives. For instance, it updates an approximate Hessian matrix at each iteration. The LM method combines the best features of the Gauss-Newton technique and the steepest-descent method. It also converges faster than CG methods since the Hessian Matrix is not computed but only
approximated. For instance, it uses the Jacobian that requires less computation than the Hessian matrix [13].

In science and engineering problems, there are many papers in the literature that examined the effectiveness of each category of algorithms on the performance of the MLPN. For instance, authors in [14] compared the performance of LM, back propagation (BP) with momentum and BP with momentum and adaptive learning rate to classify the transformer oil dielectric and cooling state. They found that the BP with momentum and adaptive learning rate improves the accuracy of the BP with momentum and also gives a fast convergence to the network. The authors in [15] compared LM, CG and resilient algorithm for stream-flow forecasting and determination of lateral stress in cohesion less soils. They found that LM algorithm was faster and achieved better performance than the other algorithms in learning. The authors in [16] 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. The authors in [17] demonstrated the application of ANNs in predicting the weekly spring discharge with three different learning algorithms. The learning algorithms considered by the authors were QP algorithm, batch BP algorithm, and LM algorithm. They conclude that the QP algorithm had a better performance to the application. Finally, authors in [18] compared BP, DBD, extended DBD, QP, 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. In this study, some of these algorithms namely STP, MOM, CG, QP, LM and DBD were tried in MLPN to predict hourly surface temperature and the results are discussed.

3.1. Step (STP) Algorithm. Gradient descent (GD) learning rules provide first-order gradient information about the network's performance surface (e.g., back propagation and real-time recurrent learning). The most straightforward way of reaching the bottom(the minima) given which way is up is tomoveinthe 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.

3.2. Momentum (MOM) Algorithm. Step components try to find the bottom of a performance surface by taking steps in the direction estimated by the attached back propagation 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.

3.3. 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
.

3.*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 "pseudo second 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 pseudo second 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.

3.5. 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 back propagation, a direct step to the error is attempted by changing the weight.

3.6. 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].

4. RESULTS AND DISCUSSION

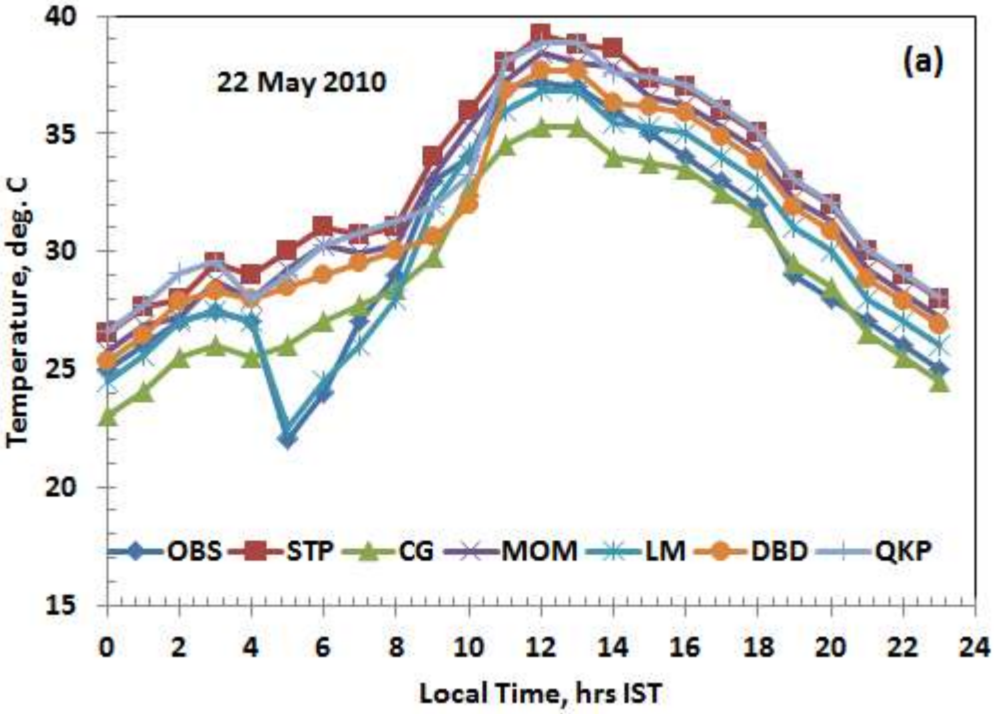
Today there are a number of parameters available that may be used to characterize pre-convective conditions and predict the beginning of convection. Johns and Doswell [19] and McNulty [20] have reviewed severe thunderstorms forecasting in detail. According to them, three of the most important factors to examine in determining occurrence of severe thunderstorm events are intense instability, a sufficiently deep humid layer in the lower and middle troposphere and an updraft to initiate convection. The formation of thunderstorms is an interaction between these conditions on different scales. The ANN model predicted results with different learning algorithms of 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.

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 [21]. 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 [22]. Recent studies show a high positive correlation between surface temperature and lightning activity [23]. The hourly temperatures on the surface are useful tool in forecasting the likelihood occurrence of a thunderstorm [24]. Meteorologists warn that a sudden drop in temperature during the day indicate for thunderstorm [21].

Figure 6 shows the inter-comparison of observed and ANN model predicted diurnal variation of surface temperature (0C) with different learning algorithms over Kolkata valid for 22nd May 2010 and 27th May 2010. From the figures, it is clearly visible that the observed data show a sudden drop in temperature in all three thunderstorm days. The ANN model with different learning algorithms captured the sudden 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 6a), the observed temperature showed a sudden drop of 140C from 360C to 220C at 10 UTC. The ANN model prediction with LM showed a drop from 330C to 220C (110C) at 10 UTC, whereas CG presented a drop from 340C to 270C (70C) at 10 UTC. All other algorithms show a difference less than 40C during thunderstorm hour. The DBD has a least performance than other algorithms. In the second thunderstorm case (Figure 3b), observed temperature fall is from 330C to 220C (110C) at 13 UTC, whereas LM indicated a drop from 320C to 210C (110C) at

13 UTC. CG showed only 60C difference between predicted and observed values. The other algorithms presented less intensity in difference between predicted and observed values. For the third case (Figure 6b), observed temperature showed a drop from 300C to 240C (60C) at 13 UTC, whereas LM showed a drop from 320C to 260C (60C) and CG displayed a drop from 320C to 270C (50C) at 13 UTC. The other algorithms have also captured sudden fall and intensity. From these analyses, we can see that ANN model with LM algorithm captured the sudden temperature fall with almost same drop in intensity as compared to other algorithms.

A statistical analysis based on MAE, RMSE and CC is performed for comparisons between the predicted and observed temperature with different learning algorithms valid for 22nd May 2010 and 27th May 2010. The results indicated that, LM algorithm has less MAE and RMSE as compared to all other algorithms for these 3 thunderstorm cases. The CG algorithm has also given moderate results. All other algorithms displayed more error in all cases as compared to LM and CG algorithms. The average MAE and RMSE from these 2 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 are more than 0.9. The CC of other algorithms are less than 0.85. The performance of DBD algorithm is less efficient than other algorithms.



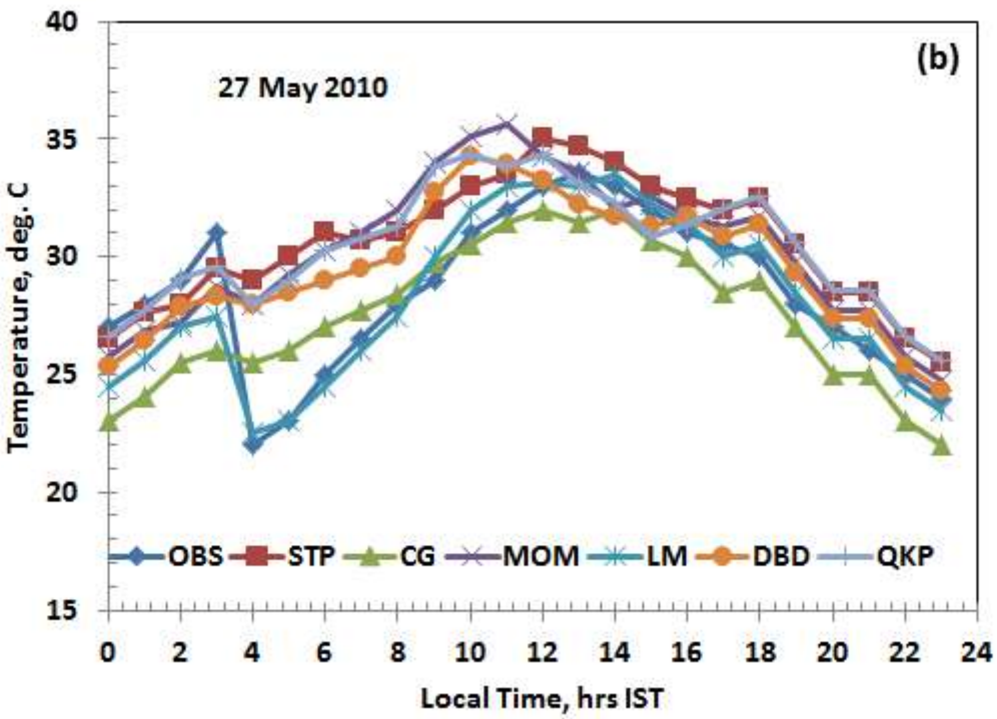


Fig 6: Comparison of ANN predicted hourly surface temperature using different learning algorithms with observation on (a) 22nd May 2010 and (b) 27th May 2009

Figure 7gives the performance accuracy of learning algorithms for hourly temperature prediction. The percentage of correctness presented a percentage number of the times when the forecast is accurate to within +/-2°C. The result clearly indicated that overall accuracy of LM algorithm for three events is 76%. CG gave a moderate accuracy of 61%. Other algorithms displayed less accuracy. The time-series plots and statistical analysis of temperature revealed that LM algorithm has well predicted sudden temperature drop for the occurrence of a severe thunderstorm on all two thunderstorm (22nd and 27th May 2010) days as in the observation.

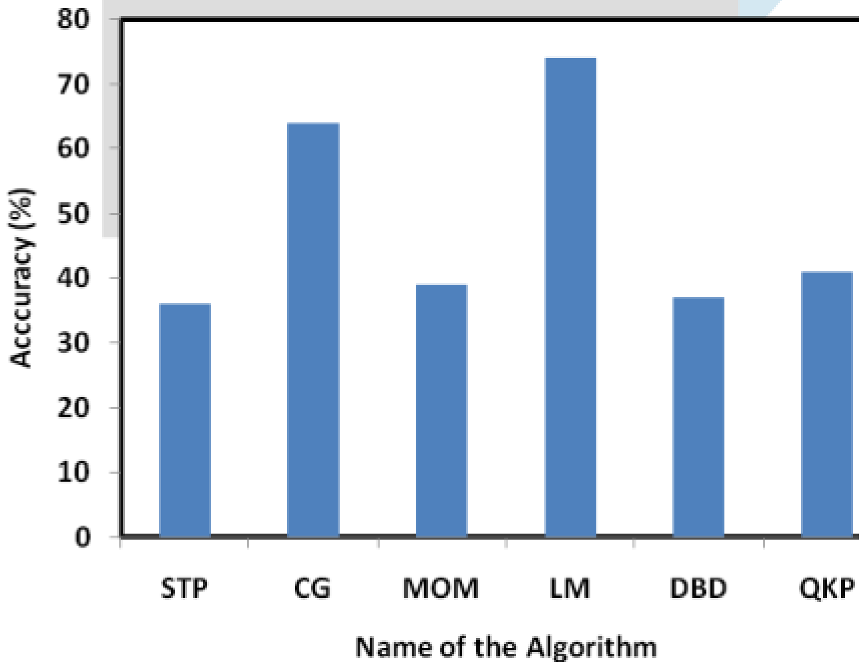


Fig 7: Performance accuracy of learning algorithms for the prediction of hourly temperature during thunderstorm days.

5. CONCLUSIONS

In this paper, sensitivity experiments have been conducted with ANN model to test the impact of learning algorithms on severe thunderstorms prediction that occurred over Guwahati on 22nd May 2010 and 27th May 2010 and validated the model results with observation. A statistical analysis based on mean absolute error, root mean square error, correlation coefficient and percentage of correctness is also performed for comparison among predicted and observed data with different learning algorithms. This is the first study conducted to investigate the sensitivity of learning algorithms with ANN model to predict thunderstorms over the eastern region of India. In all experiments, the model setups were identical except for the use of different learning algorithms. Hence the differences in the prediction results attributed to the sensitivity of learning algorithms. It is clearly demonstrated that LM algorithm performance is significantly better than other algorithms. After analyzing the results, we can conclude that the ANN model with LM algorithm has well predicted the hourly temperature in terms of sudden fall of temperature and intensity as compared to other learning algorithms. The results of these analyses demonstrated the capability of ANN model in prediction of severe thunderstorm events over eastern Indian region.

6. ACKNOWLEDGMENTS

The authors would like to express their sincere thanks to the Regional Meteorological Centre (RMC), India Meteorological Department (IMD), Guwahati for providing surface meteorological data.

REFERENCES

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

[2] JoachimKuettner,1950: Electrical and meteorological conditions inside the thunderclouds, J. Meteor, 7, 322-332.

[3] May. P.T. and Rajopadhyaya. D. K., 1995: Wind profiler observations of vertical motion and precipitation microphysics of a tropical squall line. Mon. Weather Rev, 124, 621-633.

[4] Dhawan, V.B., Tyagi, A., and Bansal, M.C. 2008. Forecasting of thunderstorms in pre monsoon season over northwest India.Mausam 59, 433–444.

[5] 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.

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

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

[8] 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.

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

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

[11] 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.

[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 micro contact 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. ˇ Cerm´ak, “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, andA. 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,” *ActaGeophysica*, 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 modeling 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 stream flow 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/.