Accurate forecasting of thunderstorms and severe thunderstorms is 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 non-linearity of their dynamics and physics (Orlanski 1975). 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 (Bishop 1995). 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 (Haykin 1994).

ANN have proven to be powerful and general technique for machine learning (Shavlik et al. 1991). Most successful application of neural networks involved pattern recognition, statistical mapping or modeling (Sentiono et al. 2002). According to Bailey and Thompson (1990), successful application can include signal validation, process monitoring, diagnostics, signal and information processing and control of complex system. James et al. (2000) 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. (2001), 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 not contain training data set, computationally fastness once trained due to parallel processing, noise tolerance. There were these advantages that make
Artificial Neural Network Model for Thunderstorm Prediction

ANNs to be successfully applied to various real world problems, including medical diagnosis (Hayashi et al. 2000), image computing, speech recognition, process control and modeling (Palade et al. 2001). However, a little attention was paid to the use of ANNs in weather forecasting (Maqsood et al. 2000 and 2004; Bodri and Cermak 2000; Luk et al. 2000; Wedge et al. 2005; Steidley et al. 2005; Chaudhury and Chattopadhyay 2005).

Bodri and Cermak (2000) developed an ANN using 38 year of rainfall data to predict monthly and yearly precipitation levels for multiple sites in the Czech Republic. Using spatial and temporal data of recent rainfall, Luk et al. (2000) 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. (2004) used an ensemble of ANNs to provide 24 hour (h) predictions for air temperature, wind speed, and humidity at the Regina Airport in Canada. Wedge et al. (2005) developed an ANN for prediction of waves spilling over sea walls in using sea conditions and wall properties as inputs. Steidley et al. (2005) used ANNs to predict tidal water levels for periods of 3–48 h ahead for a shallow embayment on the coast of Texas in the United States. Chaudhury and Chattopadhyay (2005) designed a feed-forward multi-layered ANN model to estimate the maximum surface temperature and relative humidity. The recent advances in neural network methodology for modeling non-linear, dynamical phenomena along with the
impressive successes in a wide range of applications, 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.

In this chapter, 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 3.1.

![Figure 3.1: The geographical location of Kolkata in West Bengal.](image)

The performance of six learning algorithms namely Step (STP), Momentum (MOM), Quick Propagation (QKP), Delta-Bar-Delta (DBD), CG and LM are evaluated using predicted hourly surface temperature and relative humidity during 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 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.

3.1 Data and Methodology

3.1.1 ANN experimental setup

The developed ANN model is based on one of the neural network architecture named 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 multilayer perceptron design include specification of the number of hidden layers and the number of units in these layers (Hagan et al. 1996). 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 learning algorithms. The best known example of a neural network learning algorithm is BP. Modern second-order algorithm such as CG and LM are substantially faster for many problems (Bishop 1995; Shepherd 1997). There are also heuristic modifications of BP which work well for some problem domains, such as QKP (Fahlman 1988) and DBD (Jacobs 1988).

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) train and (5) test performance of model. The basic flow in designing ANN model is given in Figure 3.2. The hourly surface weather parameters namely mean sea level pressure (hPa), relative humidity (%) and wind speed (ms\(^{-1}\)) 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 the ANN model which will be used for the prediction of hourly temperature. The hourly surface weather parameters namely hourly mean sea level pressure (hPa), temperature (\(^{\circ}\)C) and wind speed (ms\(^{-1}\)) of 3 years (April and May 2007 to 2009) of Kolkata were used as the input data for training and testing the ANN model which will be used for the prediction of hourly relative humidity. The other
additional input parameters for each model are month, day and hour of the observation. The total length of the data record included for ANN model is more than 4000. Fragment of the data used for this study are attached in the appendix.

**Figure 3.2:** Basic flow for designing ANN model.

After data collection, two data preprocessing procedures are conducted to train the ANNs more efficiently. These procedures are: (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 (Khan and Ondrusek 2000). 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 de-normalized into the original data format for achieving the desired result.

The configuration of the neural network depends highly on the problem. In this study, 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 3.3 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 and test group. The hourly surface parameters during April and May 2007-2008 and April 2009 correspond to training group and the hourly surface parameters of May 2009 correspond to test group. 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. 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. Hourly surface temperature is used as test data for the first model and hourly relative humidity for second model.

**Figure 3.3:** Architecture of MLPN for the prediction of (a) temperature and (b) relative humidity.
The 24 h ANN model outputs of surface temperature and relative humidity at Kolkata (22.52°N, 88.37°E) during three severe thunderstorm days of May 2009 (3, 11 and 15 May 2009) were used to evaluate these models. The capability of six different learning algorithms in predicting thunderstorms were studied and their performances were compared for the prediction of surface temperature and relative humidity. The learning algorithms took for these studies are STP, MOM, CG, QKP, LM and DBD. Performance and reliabilities of the models were then evaluated by a number of statistical measures like RMSE, MAE, CC and PC. 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 NeuroDimension Inc. of Florida (NeuroDimension 2005). The details of the NeuroSolutions software are given in appendix.

3.1.2 Statistical analysis

Verifying forecasts of continuous variables measures how the values of the forecasts differ from the values of the observations. The performances of the models developed in this study were assessed using various standard statistical performance evaluation criteria. The following
statistical parameters were calculated for the comparisons between the forecast and observation data.

**MAE:** This is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. The MAE is a linear score which means that all the individual differences are weighted equally in the average. The MAE is a common measure of forecast error in time series analysis. It measures accuracy (the level of agreement between the forecast and the observations) for continuous variables. The MAE is given by

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |F_i - O_i| \tag{3.1}
\]

where \(F_i\) is the forecast and \(O_i\) is the observation. The difference between the forecast and the observation is the error. The lower the errors, the greater the accuracy. The range is 0 to infinity. The perfect score is 0.

**RMSE:** It is a frequently used measure of the differences between values predicted by a model and the values actually observed. It measures average error, weighted according to the square of the error. It does not indicate the direction of the deviation. The RMSE puts greater influence on large errors than smaller errors, which may be a good thing if large errors are especially undesirable, but may also conservative forecasting. The RMSE is given by
\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (F_i - O_i)^2} \] (3.2)

**CC:** This indicates the strength and direction of a linear relationship between two random variables. That means it measures the strength of the linear relationship between the forecasts and observations. The \( r \) is given by

\[ r = \frac{\sum (F - \bar{F})(O - \bar{O})}{\sqrt{\sum (F - \bar{F})^2} \sqrt{\sum (O - \bar{O})^2}} \] (3.3)

The value of \( r \) is such that \(-1 \leq r \leq +1\). The + and – signs are used for positive linear correlations and negative linear correlations, respectively. If \( F \) and \( O \) have a strong positive linear correlation, \( r \) is close to +1. An \( r \) value of exactly +1 indicates a perfect positive fit. A correlation greater than 0.8 is generally described as strong, whereas a correlation less than 0.5 is generally described as weak.

**PC:** It is used to represent the number of times out of 100 a task is performed correctly. Percentages represent the portion of 100 outcomes that are successful or meet certain criteria. The temperature forecast is ‘Correct’ when the difference between the forecasted temperature and the actual is ±2°C or less. The relative humidity forecast is ‘Correct’ when the difference between the forecasted relative humidity and the actual is ±10% or less.
3.2 Learning Algorithms for the Present Study

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 is a number of learning algorithms used to train a MLPN and a frequently used one is called the BP learning algorithm. The BP algorithm, which is based on searching an error surface using gradient descent for points with minimum error, is relatively easy to implement. The pseudo code for BP algorithm (Mitchell 1997) is given as follows:

**Pseudo Coding**

Assign all network inputs and output
Initialize all weights with small random numbers, typically between -1 and 1
repeat
    for every pattern in the training set
        Present the pattern to the network
        // Propagated the input forward through the network:
        for each layer in the network
for every node in the layer
  1. Calculate the weight sum of the inputs to the node
  2. Add the threshold to the sum
  3. Calculate the activation for the node
end
end

// Propagate the errors backward through the network
for every node in the output layer
  calculate the error signal
end
for all hidden layers
  for every node in the layer
    1. Calculate the node's signal error
    2. Update each node's weight in the network
  end
end

// Calculate Global Error
  Calculate the Error Function
end

while ((maximum number of iterations < than specified) AND
  (Error Function is > than specified))

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 (Fahlman 1988; Rumelhart et al. 1986; Jacobs 1988). Optimization methods such as second-order methods (CG, QN and 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 (Hagan and Menhaj 1994). Six learning algorithms were applied in this study, in order to identify the one which trains a given network more efficiently. All of them are variations of basic BP algorithms. Variations of these learning algorithms are given in the following sections:

3.2.1 STP algorithm

Gradient descent (GD) learning rules provide first order gradient information about the network’s performance surface (e.g. BP 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 STP uses this procedure to adapt the weights of the activation component that it is stacked on (NeuroSolutions Manual 2003).

3.2.2 MOM algorithm

Step components try to find the bottom of a performance surface by taking steps in the direction estimated by the attached BP 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, 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 break out 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 (NeuroSolutions Manual 2003).
3.2.3 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 (Jalali-Heravi et al. 2008).

3.2.4 LM algorithm

The LM algorithm is one of the most appropriate higher-order adaptive algorithms known for minimizing the Mean Square Error (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 (NeuroSolutions Manual 2003).

3.2.5 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 (Chakraborty and Chakraborty 2002).

3.2.6 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 (Haciismailoglu et al. 2009).

A number of researchers have carried out comparative studies of MLPN learning algorithms. Kisi and Uncuoglu (2005) compared LM, CG and resilient algorithm for stream-flow 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. (2005) 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 algorithms. The simulations show that the neural network using the LM algorithm achieved the best classification performance. Raju et al. (2011) demonstrated the application of ANNs in predicting the weekly spring discharge with three different learning algorithms like QKP algorithm, batch BP algorithm and LM algorithm. They conclude that the QKP algorithm had a better performance to the application. Ubeyli and Guler (2004) compared BP, DBD, extended DBD, QKP, and LM algorithms to compute the quasi-static parameters, the characteristic impedance and the
effective dielectric constant, of the asymmetric coplanar waveguides (ACPWs). The results of the LM algorithm for the quasi-static parameters of the ACPWs were in very good agreement with the results available in the literature. The results of above studies have illustrated that the relative performance of algorithms depends on the problem being tackled.

3.3 Case Description

For the ANN model validation, three severe thunderstorm cases of May 2009 (3, 11 and 15 May 2009) have been taken and the description of each case is as follows:

Case 1 was a severe thunderstorm, which was reported on 3 May 2009 over Kolkata 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 above sea level (a.s.l) cyclonic circulation is seen over west Uttar Pradesh, a trough from this extends southeastwards up to south peninsula across east Madhya Pradesh and Andhra Pradesh. No significant trough in mid troposphere. 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 11 May 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 and adjoining Assam and Meghalaya. A trough from Arunachal Pradesh to northwest Bay of Bengal was found in middle troposphere. Sub-tropical 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 15 May 2009. A squall passed over Kolkata on 15 May 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 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 mid – troposphere is found from Arunachal Pradesh to north Bay of Bengal. Sub-tropical 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 (Mohanty et al. 2009).

3.4 Results and Discussion

According to the previous studies of (Doswell 1987; Johns and Doswell 1992; McNulty 1995) the general preconditions for the initiation of thunderstorms are conditional instability, a sufficiently deep humid layer in the lower and mid-troposphere 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 (Asnani 2006). 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 (Price 2006). Recent studies show a high positive correlation between surface temperature and lightning activity (William 2005). The temperature and relative humidity on the surface are useful tool in forecasting the likelihood occurrence of a thunderstorm (Lopez et al. 2007). A sudden drop in temperature or
sudden increase in relative humidity during the day indicates for the occurrence of thunderstorm (Asnani 2006).

3.4.1 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 are 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 3.4 shows the inter-comparison of observed and ANN model predicted diurnal variation of surface temperature (°C) with different learning algorithms over Kolkata valid for 3, 11 and 15 May 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 3.4a), 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
Chapter 3

Department of Computer Science

algorithms showed a difference less than $4^\circ$C during thunderstorm hour. The DBD has least performance than other algorithms. In the second thunderstorm case (Figure 3.4b), observed temperature fall is from $33.1^\circ$C to $21.7^\circ$C ($11^\circ$C) at 1200 UTC, whereas LM indicated a drop from $32^\circ$C to $21^\circ$C ($11^\circ$C) at the same thunderstorm hour. CG showed only $6^\circ$C difference between predicted and observed values. The other algorithms presented less intensity in difference between predicted and observed values during thunderstorm hour. For the third case (Figure 3.4c), observed temperature showed a drop from $29^\circ$C to $24^\circ$C ($6^\circ$C) at 1300 UTC, whereas LM showed a drop from $32^\circ$C to $27^\circ$C ($5^\circ$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 (Johns and Doswell 1992). Figure 3.5 shows the inter-comparison 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 3.5a), 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 3.5b), 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 3.5c), 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, it can be clearly seen 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 and relative humidity are shown in Table 3.1 and Table 3.2. The results of Table 3.1 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.
Figure 3.4: Comparison of ANN predicted hourly surface temperature (°C) using different learning algorithms with observation on (a) 3 May 2009 (b) 11 May 2009 (c) 15 May 2009.
Figure 3.5: Comparison of ANN predicted hourly relative humidity (%) using different learning algorithms with observation on (a) 3 May 2009 (b) 11 May 2009 (c) 15 May 2009.
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, it can be seen that all 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 are high and the values are 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.

Table 3.1: Performance comparison of different learning algorithms in hourly temperature prediction.

<table>
<thead>
<tr>
<th>Statistical Analysis</th>
<th>Dates</th>
<th>STP</th>
<th>MOM</th>
<th>CG</th>
<th>LM</th>
<th>QKP</th>
<th>DBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>3-May-09</td>
<td>3.36</td>
<td>3.24</td>
<td>2.69</td>
<td>2.08</td>
<td>3.48</td>
<td>3.41</td>
</tr>
<tr>
<td></td>
<td>11-May-09</td>
<td>2.69</td>
<td>2.57</td>
<td>2.27</td>
<td>1.72</td>
<td>2.54</td>
<td>2.62</td>
</tr>
<tr>
<td></td>
<td>15-May-09</td>
<td>2.93</td>
<td>2.66</td>
<td>2.08</td>
<td>1.21</td>
<td>2.69</td>
<td>2.90</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>2.99</td>
<td>2.82</td>
<td>2.35</td>
<td>1.67</td>
<td>2.90</td>
<td>2.98</td>
</tr>
<tr>
<td>RMSE</td>
<td>3-May-09</td>
<td>3.54</td>
<td>3.50</td>
<td>2.90</td>
<td>2.35</td>
<td>3.70</td>
<td>3.69</td>
</tr>
<tr>
<td></td>
<td>11-May-09</td>
<td>3.07</td>
<td>3.02</td>
<td>2.44</td>
<td>1.90</td>
<td>2.99</td>
<td>3.19</td>
</tr>
<tr>
<td></td>
<td>15-May-09</td>
<td>3.07</td>
<td>2.76</td>
<td>2.20</td>
<td>1.41</td>
<td>2.78</td>
<td>3.04</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>3.23</td>
<td>3.09</td>
<td>2.51</td>
<td>1.89</td>
<td>3.16</td>
<td>3.31</td>
</tr>
<tr>
<td>CC</td>
<td>3-May-09</td>
<td>0.82</td>
<td>0.82</td>
<td>0.90</td>
<td>0.93</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>11-May-09</td>
<td>0.89</td>
<td>0.89</td>
<td>0.94</td>
<td>0.97</td>
<td>0.89</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>15-May-09</td>
<td>0.74</td>
<td>0.80</td>
<td>0.91</td>
<td>0.96</td>
<td>0.80</td>
<td>0.74</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>0.82</td>
<td>0.84</td>
<td>0.92</td>
<td>0.95</td>
<td>0.83</td>
<td>0.80</td>
</tr>
</tbody>
</table>
The results of Table 3.2 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 3.1), CG algorithm performed well than other 4 algorithms, namely STP, MOM, DBD and QKP. The average MAE and RMSE of LM algorithm has least value than other 5 algorithms.

Table 3.2: Performance comparison of different learning algorithms in hourly relative humidity prediction.

<table>
<thead>
<tr>
<th>Statistical Analysis</th>
<th>Dates</th>
<th>STP</th>
<th>MOM</th>
<th>CG</th>
<th>LM</th>
<th>QKP</th>
<th>DBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>3-May-09</td>
<td>6.62</td>
<td>6.78</td>
<td>8.21</td>
<td>5.24</td>
<td>7.06</td>
<td>6.65</td>
</tr>
<tr>
<td></td>
<td>11-May-09</td>
<td>6.80</td>
<td>5.77</td>
<td>8.46</td>
<td>5.20</td>
<td>6.19</td>
<td>6.26</td>
</tr>
<tr>
<td>MEAN</td>
<td>7.55</td>
<td>7.25</td>
<td>8.84</td>
<td>4.49</td>
<td>7.60</td>
<td>7.30</td>
<td></td>
</tr>
<tr>
<td>RMSE</td>
<td>3-May-09</td>
<td>8.77</td>
<td>9.50</td>
<td>12.64</td>
<td>6.55</td>
<td>9.43</td>
<td>9.42</td>
</tr>
<tr>
<td></td>
<td>11-May-09</td>
<td>7.99</td>
<td>7.06</td>
<td>10.71</td>
<td>6.76</td>
<td>7.25</td>
<td>7.30</td>
</tr>
<tr>
<td></td>
<td>15-May-09</td>
<td>10.33</td>
<td>9.94</td>
<td>10.93</td>
<td>3.71</td>
<td>10.56</td>
<td>9.89</td>
</tr>
<tr>
<td>MEAN</td>
<td>9.03</td>
<td>8.83</td>
<td>11.43</td>
<td>5.67</td>
<td>9.08</td>
<td>8.87</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>3-May-09</td>
<td>0.86</td>
<td>0.86</td>
<td>0.69</td>
<td>0.93</td>
<td>0.84</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>11-May-09</td>
<td>0.84</td>
<td>0.88</td>
<td>0.64</td>
<td>0.95</td>
<td>0.86</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>15-May-09</td>
<td>0.68</td>
<td>0.76</td>
<td>0.67</td>
<td>0.95</td>
<td>0.68</td>
<td>0.69</td>
</tr>
<tr>
<td>MEAN</td>
<td>0.80</td>
<td>0.83</td>
<td>0.67</td>
<td>0.95</td>
<td>0.80</td>
<td>0.82</td>
<td></td>
</tr>
</tbody>
</table>

From the table (Table 3.2), it can be seen 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 and 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 3.6 gives the performance accuracy of learning algorithms for hourly temperature and relative humidity prediction. The PC of temperature presented a percentage number of the times when the forecast is accurate to within ±2°C. The results (Figure 3.6a) 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 PC of relative humidity presented a percentage number of the times when the forecast is accurate to within ±10% confidence range. The results (Figure 3.6b) 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%. The time-series plots and statistical analysis of temperature and relative humidity revealed that LM algorithm is 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.
3.4.2 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 (April and May 2007–2009). The ANN models were created to predict surface temperature and relative humidity at hourly intervals with 1, 3, 6, 12 and 24 h ahead during severe thunderstorm cases and the results are evaluated in the following section. Analysis of the results of these experiments are helpful to understand the efficiency of developed 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 3, 11 and 15 May 2009, is shown in Figure 3.7. 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 cases. The 6 and 12 h forecast failed to capture the entire pattern. For the first case, (Figure 3.7a), the observed temperature showed a sudden drop of 15°C from 36.7°C to 21.7°C (15°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 3.7b), 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 3.7c), 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). 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 show a 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 case 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 3.3. 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 was very poor as compared to 1, 3 and 24 h. The PC of these 5 models are given in Figure 3.9a. 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.
Figure 3.7: Comparison of ANN predicted hourly temperature (°C) using different advanced prediction models with observation on (a) 3 May 2009 (b) 11 May 2009 (c) 15 May 2009.
### Table 3.3: Performance comparison of different advanced predictions for hourly temperature during thunderstorm days.

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

The comparison between observed and predicted relative humidity for 1 to 24 h ahead forecasting on 3, 11 and 15 May 2009 is shown in Figure 3.8. The results show, 1 h ahead forecast captured sudden increase in relative humidity at thunderstorm hours during all three severe 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 3.8a), 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 3.8b), 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 6 and 12 h ahead 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 for this thunderstorm event. For the third case (Figure 3.8c), 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, it can be seen that 1 h advanced prediction model well predicted diurnal variation during thunderstorm days and captured the 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 3.4. 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 + correlation.
Figure 3.8: Comparison of ANN predicted hourly relative humidity (%) using different advanced prediction models with observation on (a) 3 May 2009 (b) 11 May 2009 (c) 15 May 2009.
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 PC of these 5 models are given in Figure 3.9b. 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 was very poor as compared to other three advanced prediction models.

Table 3.4: Performance comparison of different advanced predictions for hourly relative humidity during thunderstorm days.

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

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.

(a) Temperature

(b) Relative Humidity

Figure 3.9: Performance accuracy of different advanced prediction models for the prediction of (a) temperature and (b) relative humidity during thunderstorm days.
Even though the model performance of 6 and 12 h ahead 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.

### 3.5 Chapter Summary

In this chapter, sensitivity experiments have been conducted with ANN model to test the impact of learning algorithms on severe thunderstorms prediction that occurred over Kolkata on 3, 11 and 15 May 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. It is clearly demonstrated that LM algorithm performance is significantly better than other algorithms. After analyzing the results, it can be concluded 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 ANN models were created 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 and 3 h ANN models were able to predict hourly temperature and relative humidity adequately with sudden fall and rise. The efficiency of ANN models were reduced as the forecast lead time increased from 6 to 12 h. The 24 h advanced prediction model can able to predict features of thunderstorm with reasonable accuracy. The results of these analyses demonstrated the capability of ANN model in prediction of severe thunderstorm events over eastern Indian region and will helpful for real time thunderstorm forecast.

………fog…………