Weather Forecasting using Deep Feed Forward Neural Network (DFFNN) and Fuzzy Outlier Removal

M. Nagaselvi and Dr.T. Deepa

Abstract--- Weather forecasting is significant designed for people who facilitate to create additional informed daily decisions, and to keep out of danger. Precise Weather forecasting is becomes one of most difficult problems approximately the world. Unlike conventional methods, modern Weather forecasting consists of a combination of system models, examination. Data mining uses various technologies to forecast weather, predict rainfall, wind pressure, humidity, etc. Classification in data mining varied the restriction to view the clear information. The prediction of weather should be precise and moreover the weather be supposed to be forecasted previous at least a month before which determination be cooperative designed for many applications like agriculture, military, etc. Artificial Neural Networks (ANNs) have been applied comprehensively to together regress and classify weather experience. So in this work proposed Deep Feed Forward Neural Network (DFFNN) is rendering accurate predictions with noisy datasets, there is currently not a significant amount of research focusing on whether DFFNN are capable of producing accurate forecasts of relevant weather variables from small-scale, imperfect datasets. In addition the proposed work shows the outliers in the dataset is also removed by using fuzzy technique during the classification task. There is no important quantity of investigate focusing on the forecasting performance of Neural Networks realistic to weather datasets with the purpose of have been temporally rolled-up from a base dataset.

Keywords--- Data Mining, Classification, Weather Forecasting, Classification Methods, Soft Computing, Artificial Neural Network (ANN), Deep Feed Forward Neural Network (DFFNN) and Fuzzy Outlier Removal

M. Nagaselvi, MCA,(Mphil)., Mphil Scholar, Sri Ramakrishna College of Arts and Science, Coimbatore-641044. E-mail:nagaselvi11@gmail.com
Dr.T. Deepa, Msc, MBA, Mphil, Phd, Sri Ramakrishna College of Arts and Science, Coimbatore-641044

I. INTRODUCTION

Weather forecasts is collecting the quantitative data about the current state of the atmosphere on a given place and using scientific understanding of atmospheric processes to project how the atmosphere will react on that place. Weather warnings are important forecasts to protect life and property. Forecast based on temperature and prediction are important to agriculture, traders within commodity markets and companies which estimate demand over coming days. Outdoor activities may suppressed by heavy rain, snow and the wind chill at that situation forecasting can be used to plan activities and survive them.

The massive computational power is required to solve the equations that describe the atmosphere, error involved in measuring the initial conditions and an incomplete understanding of atmospheric processes. Forecasts become less accurate as the variation in current time and the time for which the forecast is increased. The use of ensembles and model consensus reduces the error and uses the most likely outcome.

Soft Computing [1-2] deals with impression, uncertainty, partial truth, and approximation to achieve practibility, robustness and low solution cost. Soft Computing is an automatic approach for forecasting, in many applications. Weather Forecasting is the main goal of atmospheric research. It is described as the most advanced area in meteorology. The weather forecasting is not only highly complex and highly quantitative. The various methods used in forecasting the weather are as follows:

1. Numerical Weather Prediction (NWP): uses the computer’s power to predict a forecast. Complex computer programs are forecast models, which runs on supercomputers and provide predictions on many atmospheric variables such as temperature, pressure, wind, and rainfall. A forecaster examines how the features predicted by the computer will interact to produce the day's weather. The NWP method is flawed in that the equations used by the models to simulate the atmosphere are not precise. The computer's prediction of how that initial state will evolve will not be entirely accurate without the initial state.
2. Statistical methods are used with the numerical weather prediction. Statistical methods use the previous records of weather data based on the assumption future data will be assumed as a repetition of the past weather. The main purpose of studying the past weather data is to find out the aspects of the weather that are good indicators of the future events. After establishing these relationships, correct data can be safely used to predict the future conditions. Only overall weather can be predicted in this way. However, such systems are capable of providing only such information, which is usually generalized over a larger geographical area. The variables defining weather conditions like temperature (maximum or minimum), relative humidity, pressure etc., vary continuously with time, forming time series of each parameter and can be used to develop a forecasting model either statistically or using some other classification methods .

Data mining is used to find the unknown properties in the data. Classification is the problem of identifying the set of classes of new observation. The individual observations are analyzed into various explanatory variables, features, etc. In order to avoid the problem of error propagation partially or completely probabilistic classifiers can be more effectively incorporated into larger machine-learning tasks.

This paper aims to both explore current applications of Neural Networks [3] to weather variable prediction and also to apply neural networks to a custom weather dataset. For better understanding how this research stands apart from existing research, it is helpful to note some major themes in existing research. In the first step, most documented experiments have used in neural networks to predict weather occurrences in large-scale settings or environments. For example, neural networks have been used to predict quantitative rainfall amounts for the Dallas-Ft. Worth area [4]. In the Second step, even in research focused on employing Neural Networks to account for local weather differences that cannot be predicted by large scale weather models, the local differences usually still apply to larger regions being monitored at several different points. For instance, in order to give more accurate in four separate regions in the mid-Atlantic United States, neural networks have been used to process output from Numerical Weather Prediction (NWP) models.

The goal is to determine the feasibility of using a rather imperfect dataset obtained from a single collection unit as input to neural networks in order to obtain regression and Deep Forward Neural Network (DFFNN) and Artificial Neural Network (ANN) classification predictions [5-6] methods for various weather variables. The variables defining weather conditions like temperature (maximum or minimum), relative humidity, pressure etc., vary continuously with time. Time series of each parameter can be used to develop a forecasting model either statistically or using some other classification methods . In addition the proposed work, outliers in the dataset is also removed by using fuzzy technique during the classification task.

This research is differentiated from existing ANN research primarily through the choice of datasets. Compared to large datasets built over decades from a network of collection stations, a dirty, real-world dataset obtained from a single, commercially available, solar-powered weather collection station is used. If it can be shown that DFFNN is trained from this dataset are capable of predicting even a few useful variables with reasonable accuracy, then further research into predicting a wider range of regression and classification variables from the dataset is warranted.

II. BACKGROUND STUDY

In [5] a Feed forward multi-layered artificial neural network model is designed to estimate the maximum surface temperature and relative humidity needed for the genesis of severe thunderstorms over Calcutta. Using single layer network and one hidden layer artificial neural network prediction error is calculated and compared. Result shows the efficiency of the one hidden layer ANN. This estimation can help in predicting a probable thunderstorm day with one day or 24 hrs in advance. In [6] described a weather forecasting problem-rain fall using different neural network architectures namely Electronic Neural Network (ENN) model and optoelectronic neural network model. They did experiments using these two models and the percentage of correctness of the rainfall estimation of the neural network models and the meteorological experts are compared. The accuracy of ENN and opto-electronic neural network models, is compared with two metrological experts.

In [7] shows, how ANN can be used for Forecasting Weather for Iran city. They implements ANN for one day ahead prediction of weather parameter i.e. temperature of Iran. Their study was based on most common neural network model Multilayer Perceptron (MLP) which is trained and tested using ten years past metrological data. Minimum error is achieved using MLP between exact and predicted values at each day and has a good performance, reasonable prediction accuracy and minimum prediction error in general.

In [8] a new technique is proposed for Weather classification and forecasting using Levenberg Marquardt Back Propagation Feed Forward Neural Network. Levenberg BP is the fastest and best among many BP. The classification and Prediction of Weather using BPNN is basically a forecasting kit which aims to gather data i.e. weather parameters like temperature, pressure, humidity, wind direction. These predictors are taken as input neuron to BP. In [10] the past data like rainfall, wind speed, dew point, and temperature is used to predict weather using k nearest neighbor algorithm. It generates accurate result in advance in prediction of weather. The results are in Boolean attributes and numeric values. The changes can be recognized by using the patterns. Pattern recognition can be used.

In [11] data mining techniques used meteorological data for prediction and decision making. The Weather forecasting parameter's relationships are found using data mining techniques. Since meteorological data are vast and time constrained, it not only need to modify by traditional data mining. But also can be modified using some other techniques. In [12] by using decision trees the stored data in past are used to predict the upcoming climate. By using all parameters the prediction can be improved and perfect. And also the limit for prediction will not limit itself. In [13] self organizing data mining technique called enhanced Group Method of Data Handling (e-GMDH) is employed to predict and forecast weather. e- GMDH works efficiently when compared with older data mining techniques. Graphical User Interface (GUI) which is partly placed in the algorithm must be updated to include the current functionalities.

First, many experiments have used with neural networks to predict quantitative rainfall amounts at various locations and look-ahead ranges. For instance, researchers in Thailand were obtained highly accurate forecasts to predict quantitative rainfall amounts in the one to three hour look-ahead range using feed-forward neural networks in order to predict possible flooding dangers [14]. Additionally, neural networks have been used in research to generate probabilities of precipitation and quantitative precipitation forecasts using data from the Eta atmospheric model and upper air soundings. As shown next, neural networks have also been used to predict other less common weather phenomena. Neural networks have also been used to predict weather phenomena besides the traditional forecast values, such as probability/amount of rainfall, wind speed, barometric pressure, etc. Tornadoes are predicted successfully[15]. Identification of fog at various forecast ranges ranging from 3 hours to 18 hours around Canberra International Airport [15], Australia. Hopefully, this survey provides in depth idea of the depth and variety of neural network-based weather forecasting.

III. Proposed Deep Feed Forward Neural Network Methodology and Fuzzy Methods for Data Preprocessing

The goal of weather prediction is to determine the feasibility of using a rather imperfect dataset obtained from a single collection unit as input to neural networks in order to obtain regression and Deep Forward Neural Network (DFFNN) and Artificial Neural Network (ANN) classification predictions methods for various weather variables of interest. The variables defining weather conditions like temperature (maximum or minimum), relative humidity, pressure etc., vary continuously with time, forming time series of each parameter and can be used to develop a forecasting model either statistically or using some other classification methods . This research is distinguished from existing ANN research primarily through the choice of datasets. Rather than using large datasets built over decades from a network of collection stations, a dirty, real-world dataset obtained from a single, commercially available, solar-powered weather collection station is used. If it can be shown that DFFNN is trained from this dataset are capable of predicting even a few useful variables with reasonable accuracy, then further research into predicting a wider range of regression and classification variables from the dataset is warranted.

The practical applications of a system make this research worthwhile. First, the experiment uses low-cost or free software and hardware, which minimizes the production cost of a system built around the DFFNN model. Second, let's assume that experimentation reveals that it is possible to develop a regression and classification model with strong predictive capabilities from the source dataset through the use of neural networks. If a collection station is then prepositioned in a crop field and allowed to gather data over time, then a neural network can be trained to predict weather variables of interest for that small geographical point. This information would be very useful in remote areas, where radio and network connectivity to existing weather services is limited. In addition the proposed work, outliers in the dataset is also removed by using fuzzy technique during the preprocessing task.

Incomplete dataset samples are removed by data preprocessing is a common step in many disciplines, including data mining, data warehousing, and optimization problems. As this section explains, data preprocessing also plays a very important role in neural networking. First, data normalization is examined by using the fuzzy technique. Data normalizing is the process of scaling data "to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0" [19]. To perform the scaling task in this work data normalization is carreiedout using fuzzy technique which converts the original dataset samples into fuzzified values 0-1. Fuzzy systems are created using membership functions (MFs) which is modeled based on dataset. Therefore, there is relation between uncertainty of input data and fuzziness expressed by MFs. Outliers and noisy data are kinds of uncertainty which doesn't affect on membership function.

Data normalization is used to remove the dependence on measurement units using fuzzy technique, which is directly relevant to weather forecasting since predictor variables are measured using a wide variety of units (miles per hour, degrees Fahrenheit, inches of Mercury, etc.). Data normalization using fuzzy technique has direct implications to neural network performance. In fact, "normalizing the input values for each attribute measured in the training tuples will help speed up the learning phase" [18].

Furthermore, it is important to normalize Neural Network training data in order to prevent weights. Next, the issue of missing values in training data is explored.

Real world data won't be a perfect one because of noisy and missing values. Data mining research has produced several methods of dealing with such dirty data, including interpolation of missing values, binning, clustering for outlier analysis, etc [18]. In the context of neural network-based weather prediction systems however, research has established that removing training tuples with missing values is usually the wisest approach. The forecasting skill of a neural network trained using replacement values for missing values is at the mercy of the quality of the estimated values [19]. In other words, if the estimated values are highly inaccurate, then the predictive capability of a neural network trained with this data will suffer. Furthermore, by comparing Neural Networks trained using tuples with estimated values and neural networks trained using only complete data, researchers found that there was no significant benefit to using estimated values in training tuples [19].

Deep Feed Forward Neural Network (DFFNN)

A DFNN is a feed-forward, artificial neural network that has more than one layer of hidden units between its inputs and its outputs. Each hidden unit, j, typically uses the logistic function (the closely related hyberbolic tangent is also often used and any function with a well-behaved derivative can be used) to map its total input from the layer below, x_j, to the scalar state, y_j that it sends to the layer above.

$$y_j = logistic(x_j) = \frac{1}{1 + e^{-x_j}} \quad (1)$$

$$x_j = b_j + \sum_i y_i w_{ij} \quad (2)$$

where b_j is the j^{th} unit bias, i is an layer index, and w_{ij} is the weight on a connection to unit j from unit i in the layer below. For multiclass classification, output unit j converts its total input weather prediction preprocessed dataset samples into x_j and into a class probability, p_j , by using the "softmax" nonlinearity

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)} \quad (3)$$

where k is an class index. DNNs can be Discriminatively Trained (DT) by backpropagating derivatives with cost function that measures the discrepancy between the target outputs and the actual outputs produced for each training case [14]. The natural cost function C is the cross entropy between the target probabilities d and the outputs of the softmax, p

$$C = -\sum_j d_j \log p_j \quad (4)$$

where the target probabilities will be one or zero, that are the supervised information provided to train the DNN classifier. For large training sets, it is typically more efficient to compute the derivatives on a small, random "minibatch" of training cases, rather than the whole training set, before updating the weights in proportion to the gradient. This stochastic gradient descent method can be further improved by using a "momentum" coefficient, which has the value $0 < \alpha < 1$, that smooths the gradient computed for minibatch t, thereby damping oscillations across ravines and speeding progress down ravines. The bias update rule can be derived by treating them as weights on connections coming from units that always have a state of one.

$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) - \in \frac{\partial C}{\partial w_{ij}(t)} \quad (5)$$

Overfitting is reduced by large weights that can be penalized in proportion to their squared magnitude, or the learning can simply be terminated at the point at which performance on a held-out validation set starts getting worse [20]. In DNNs with full connectivity between adjacent layers, the initial weights are given small random values to prevent all of the hidden units in a layer from getting exactly the same gradient.

It is very difficult to optimize DNNs with many hidden layers. Gradient descent from a random starting point near the origin is not the best way to find a good weights, and unless the initial scales of the weights are carefully chosen [20]. The backpropagated gradients will have very different magnitudes in different layers. In addition to the optimization issues, DNNs may generalize poorly to held-out test data. DNNs with many hidden layers and many units per layer are very flexible models with a very large number of parameters. Very complex and highly nonlinear relationships between inputs and outputs are made. This ability is important for high quality weather prediction, but it also allows them to model spurious regularities that are an accidental property of the particular examples in the training set, which will cause overfitting. The overfitting is reduced by weight penalties or early stopping.

There are many ways to determine the number of hidden neurons. In this experiment, a specific forecast from a specific dataset is viewed as a distinct neural network problem. For instance, across all datasets, a distinct set of neural networks was trained to predict minimum temperature at the 1-hour lookahead range, while a different distinct set of neural networks was trained to predict minimum temperature at the 3-hour look-ahead range. This process was repeated for each look-ahead forecast in every dataset. The combination of partitioning each forecast problem into a set of neural network problems and performing 10-fold cross validation for each Neural Network in each set made it unfeasible to adopt a trial and error method to determine the optimum number of hidden layers and hidden layer neurons. The trial an d error method seems more appropriate for fine-tuning a neural network used for a specific forecasting problem, such as determining the optimum neural network structure for predicting maximum gust values at the 12-hour look-ahead range from the 1-hour dataset. In order to generate a variety of neural network topologies, this experiment used an approach based off the number of inputs and outputs in the given forecasting problem. An existing strategy suggests that the number of hidden layer neurons should be in the range $[2^{\frac{n}{2}}, 2n + m]$, where n is the number of input neurons and m is the number of output neurons [14]. This suggestion formed the basis for topology generation in this experiment.

IV. EXPERIMENTATION RESULTS

The base dataset used for this experiment contains 15,893 weather records collected from December 21st, 2011 to January 9th, 2013 via a personal weather collection station at 15 minute intervals. Each tuple contains measurements for the following 14 variables: observation date, indoor humidity, indoor temperature, outdoor humidity, outdoor temperature, absolute pressure, wind, gust, wind direction, relative pressure, dew point, wind chill, wind level, and gust level. But for simplification purpose in this work they consider only following attributes: Minimum temperature, Maximum Temperature , humidity and pressure , were the indoor and outdoor humidity values are converted into single range value , absolute and relative pressure are converted into single pressure. While there are certainly much richer datasets available from meteorological databases, this dataset was intentionally chosen because it is imperfect. The specific reasons for using this dataset in the experiment are explained next.

The following reasons indicates choosing the poor dataset. First, the dataset provides an opportunity to see how neural networks handle datasets with large date gaps. For instance, there is a large gap in collected data from October 12th 2012 to December 29th, 2012. Second, the dataset pertains to a geographic location of interest (Statesboro, Georgia). Finally, the dataset contains many tuples (506 to be exact) with null values. Even high end weather collection systems occasionally fail to collect measurements due to power outages, failed sensors, etc. Since it is useful to assess how neural networks would perform over periods of time in real environments where measurements may occasionally be flawed, this dataset is considered valuable.

Roll-up Dataset Generation; Before describing the dataset, it is helpful to know the concept of rolling up as it pertains to this experiment. The traditional data warehousing concept of roll-up usually refers to data aggregation from a lower granularity to a higher granularity [19]. In this experiment, temporally rolling up the base dataset will generate the derived attributes which are not available in the base dataset that can be used as neural network inputs. For example, a 24-hour roll-up contains a temperature attribute which indicates the minimum temperature observed over the past 24 hours. This information is not available in any tuple in the base 15-minute interval dataset. Time functions were applied to the base dataset described above to generate 1-hour, 6-hour, and 24-hour datasets containing 3991, 672, and 173 tuples, respectively. In this experiment, quantitative precipitation forecasts and rain/non-rain event classifications were not generated for several reasons. First, the original dataset does not contain enough rain events to allow a neural network to train effectively. Second, rainfall is traditionally a very difficult weather phenomenon to predict, even with rich datasets that span decades [21].

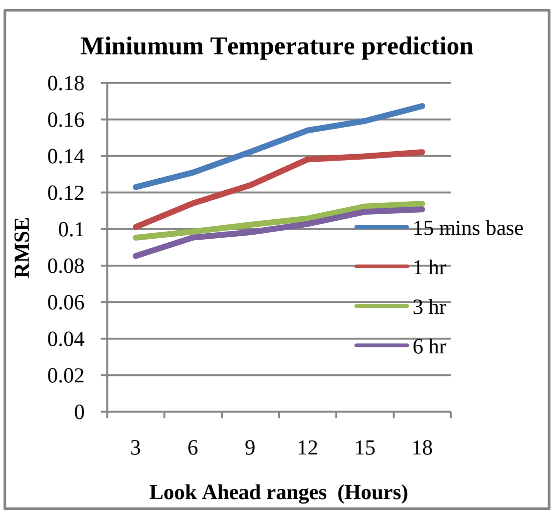

Figure 1: Minimum Temperature Prediction using the Artificial Neural Network (ANN)

Figure1 shows the results for applying ANN on the experimental datasets to conclude minimum temperature predictions more specified look-ahead ranges. Each figure shows the performance of a given Neural Network type across various datasets as measured by Root Mean Square Error (RMSE) values calculated at the 15-minute, 1-hour, 3-hour, 6-hour look-ahead ranges. A RMSE prediction threshold line is drawn at 0.1, although this value was arbitrarily chosen.

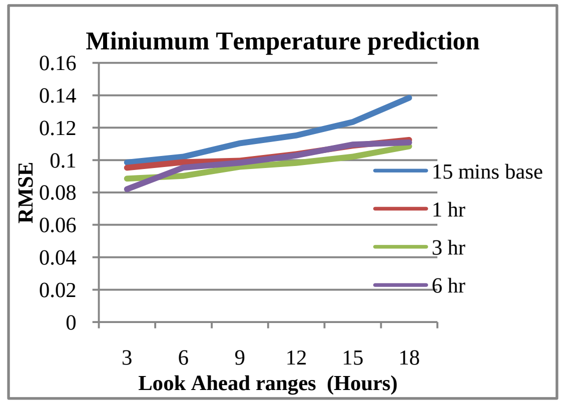

Figure 2: Minimum Temperature Prediction using the Deep Feed Forward Neural Network (DFFNN)

Figure 2 shows the results of applying DFFNN on the experimental datasets to determine minimum temperature predictions over specified look-ahead ranges. Each Figure 2 shows the performance of a given neural network type across various datasets as measured by Root Mean Square Error (RMSE) values calculated at the 15-minute, 1-hour, 3-hour, 6-hour look-ahead ranges. A RMSE prediction threshold line is drawn at 0.1, although this value was arbitrarily chosen

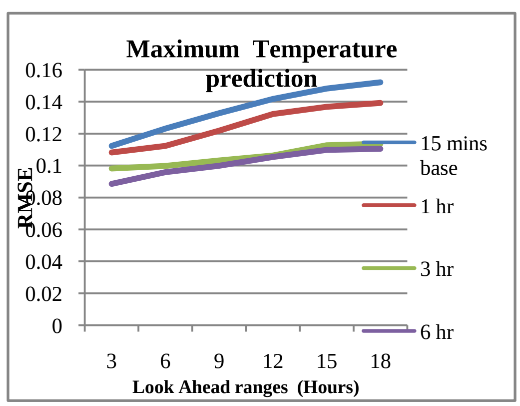

Figure 3: Maximum Temperature Prediction using the Artificial Neural Network (ANN)

Figure 3 are the results of applying ANN on the experimental datasets to determine maximum temperature predictions over specified look-ahead ranges. Each figure shows the performance of a given Neural Network type with different datasets which are measured by Root Mean Square Error (RMSE) values calculated at the 15-minute, 1-hour, 3-hour, 6-hour look-ahead ranges. A RMSE prediction threshold line is drawn at 0.1, although this value was arbitrarily chosen.

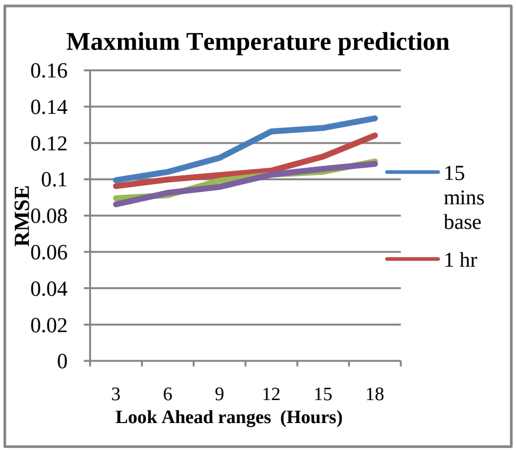

Figure 4: Maximum Temperature Prediction using the Deep Feed Forward Neural Network (DFFNN)

Figure 4 shows the results of applying DFFNN on the experimental datasets to determine maximum temperature predictions over specified look-ahead ranges. Each Figure 4 shows the performance of a given neural network type across various datasets as measured by Root Mean Square Error (RMSE) values calculated at the 15-minute, 1-hour, 3-hour, 6-hour look-ahead ranges. A RMSE prediction threshold line is drawn at 0.1, which is chosen arbitrarly. Good RMSE value will vary from application to application depending on what is acceptable to end users and there is no universally meteorological standard. For this experiment, an average variation of 10% between the squared differences of predicted and observed values is acceptable. Any points that fall above this threshold line represent poor forecasts and any points that fall below this threshold represent feasible forecasts.

F-Measure: F-measure is based on a combinatorial approach which depends on both precision and recall. Each pair can fall into one of four groups: if both objects belong to the same class and same cluster then the pair is a True Positive (TP); if objects belong to the same cluster but different classes the pair is a false positive (FP); if objects belong to the same class but different pair of order is a False Negative (FN); otherwise the objects belong to different classes and different order, and the pair is a True Negative (TN). The Rand index is simply the accuracy;

$$RI = Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \qquad (6)$$

The F-measure is another measure commonly used in the IR literature, and is defined as the harmonic mean of precision and recall; i.e.,

$$F - measure = \frac{2PR}{(P + R)} \qquad (7)$$

$$Precision(P) = \frac{TP}{(TP + FP)} \qquad (8)$$

$$Recall(R) = \frac{TP}{(TP + FN)} \qquad (9)$$

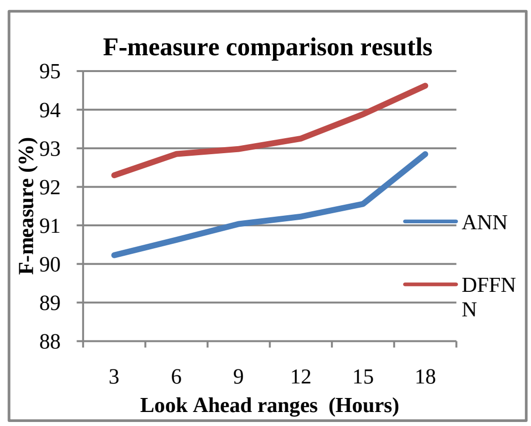

Figure 5: F measure Prediction Results using the Deep Feed Forward Neural Network (DFFNN) and Artificial Neural Network (ANN)

Figure 5 shows the F-Measure results of applying ANN and DFFNN on the experimental datasets to determine maximum temperature predictions over specified look-ahead ranges. TSM with classifier performs well when compare to ANN, and the F-Measure variation is 1.67 % (94.62 %) high when compared to ANN 92.85 % across Roll-up Dataset. It is reasonable because fuzzy outlier removal is performed to remove missing and incomplete Roll-up Dataset; it is also good F-Measure results at obtaining more Roll-up Dataset by learning from DFFNN the results are tabulated in Table 1.

Table 1: F measure Prediction Results using the Deep Feed Forward Neural Network (DFFNN) and Artificial Neural Network (ANN)

Algorithm name		Look Ahead Ranges (Hours)					
	F-measure (%)	3	6	9	12	15	18
ANN		90.23	90.63	91.04	91.23	91.56	92.85
DFFNN		92.3	92.85	85.12	92.98	93.88	94.62

Table 2: Accuracy Prediction Results using the Deep Feed Forward Neural Network (DFFNN) and Artificial Neural Network (ANN)

Algorithm name		Look Ahead Ranges (Hours)					
	Accuracy (%)	3	6	9	12	15	18
ANN		90.86	91.2	91.56	92.14	92.87	93.21
DFFNN		93.25	93.84	94.53	95.81	93.88	96.85

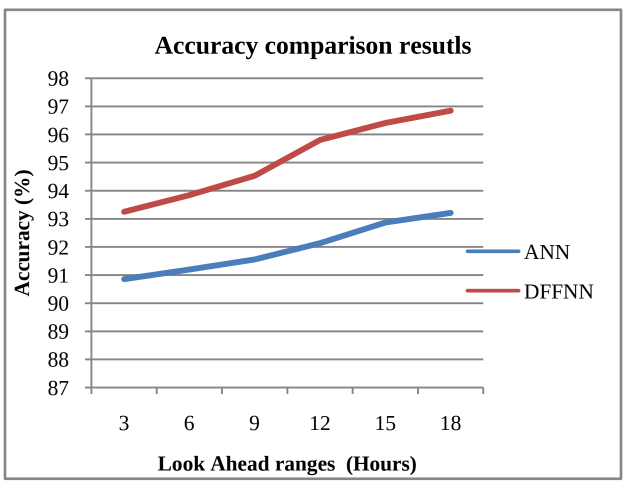

Figure 6: Accuracy Prediction Results using the Deep Feed Forward Neural Network (DFFNN) and Artificial Neural Network (ANN)

Figure 6 shows the accuracy results of applying ANN and DFFNN on the experimental datasets to determine maximum temperature predictions over specified look-ahead ranges. TSM with classifier performs well when compare to ANN, and the accuracy variation is 1.67 % (94.62 %) high when compare to ANN 92.85 % across Roll-up Dataset. It has some reasons because fuzzy outlier removal is performed to remove missing and incomplete Roll-up Dataset; it is also good accuracy results at obtaining more Roll-up Dataset by learning from DFFNN the results are tabulated in Table 2.

V. Conclusion and Future Work

In our proposed work, the effectiveness is examined using Deep Feed Forward Neural Network (DFFNN) to forecast weather variables, both in existing research and with an experimental dataset. Incomplete dataset samples is removed through data preprocessing which is a common step in many applications, such as data mining, data warehousing, and optimization problems. The data preprocessing plays a very important role in neural networking and then data normalization is examined by using the fuzzy concept. Scaling in data normalization is carried out using fuzzy technique which converts the original dataset samples into fuzzy values which is between 0 and 1. The main goal of this paper is to determine the feasibility of the variables using a rather imperfect dataset which is obtained from a single collection unit as well as given to neural networks in order to obtain good regression. Deep Forward Neural Network (DFFNN) and Artificial Neural Network (ANN) predictions methods for various weather variables of interest are also used. The weather variables depends on temperature (maximum or minimum), relative humidity, pressure etc., that vary continuously with time and forming time series of each parameter. The variables can be used to develop a forecasting model either statistically or using some other classification methods. 15-minute dataset produces forecasts more accurate than using rolled-up datasets. For rolled-up datasets, loss compensation of available training tuples are not satisfied one compared to the derived attributes in the training tuples. It is clear from this observation that the general idea of neural networking has more examples of training data of neural network processes and the more successful one will be in regressing or classifying variables. There are many possibilities to expand this experiment into future work. The effect of incorporating a distributed computing architecture is examined based on convergence times and forecast accuracy. In order to determine their forecasting capabilities, train neural networks from a more recent dataset and subsequently use them as real-world forecasting system. This is the main reason behind the formulation of an intelligent hybrid systems that overcomes the limitations of neural networks and fuzzy systems. Fuzzy systems are required to have an automatic adaption procedure which is comparable to neural networks. Hybridizing technique has both advantages and disadvantages.

REFERENCE

[1] T. Hall, H.E. Brooks, and C.A. Doswell III, “Precipitation forecasting using a neural network,” Weather and Forecasting, vol. 14, no. 3, pp. 338-345, Jun. 1999.

[2] Culclasure, Andrew. "Using Neural Networks to Provide Local Weather Forecasts." (2013).

[3] J. Harris, “Neural Networks,” class powerpoint slides for CSCI 7130, Computer Science Department, Georgia Southern University, Dec. 2011.

[4] J. Heaton, Introduction to Encog 2.5: Revision 3-November 2010, Heaton Research, Inc., 2010.

[5] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2010.

[6] I.H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Morgan Kaufmann Publishers, 2005.

[7] Sutapa Chaudhuri, Surajit Chattopadhyay, “Neurocomputing based short range prediction of some meteorological parameters during the pre-monsoon season”, Springer-Verlag 2005, Journal of Soft Computing (2005) Vol 9, pp 349–354

[8] C. Subhajini and V. Joseph Raj, “Computational Analysis of Optical Neural Network Models to Weather Forecasting”, International Journal of Computer Science Issue ISSN (Online): 1694-0814, Volume 7, Issue 5, September 2010

[9] Mohsen Hayati and Zahra Mohebi, “Temperature Forecasting Based on Neural Network Approach”, World Applied Sciences Journal ,Volume 2(6), pp 613-620,2007.

[10] R. Chaudhari, D. P. Rana, R. G. Mehta, Data Mining with Meteorological Data”, International Journal of Advanced Computer Research, Vol. 3, No. 3, Issue 11, September 2013, pp. 25-29

[11] Elia Georgiana Petre, “A Decision Tree for Weather Prediction”, Vol. LXI, No. 1/2009, 2009, pp. 77-82.

[12] Godfrey C. Onwubolu, Petr Buryan, Sitaram Garimella, Visagaperuman Ramachandran, Viti Buadromo, Ajith Abraham, “Self-Organizing Data Mining for Weather Forecasting”, IADIS European Conference Data Mining, 2007, pp. 81-88.

[13] Seema Mahajan, Dr. S. K. Vij, “Modeling and Prediction of Rainfall Data using Data Mining”, International Journal of Engineering Science and Technology, Vol. 3, No. 7, July 2011, pp. 6799-6804.

[14] N.Q. Hung, M.S. Babel, S. Weesakul, and N.K. Tripathi, “An artificial neural network model for forecasting in Bangkok, Thailand,” Hydrology and Earth System Sciences, vol. 13, no. 8, pp. 1413-1425, Aug. 2009.

[15] T. Santhanam and A.C. Subhajini, “An efficient weather forecasting system using radial basis function neural network,” Journal of Computer Science, vol. 7, no. 7, pp. 962-966, Jun. 2011.

[16] Yuval and W.W. Hsieh, “An adaptive nonlinear mos scheme for precipitation forecasts using neural networks,” Weather Forecasting, vol. 18, no. 2, pp. 303-310, Apr. 2003.

[17] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, 2012.

[18] D. Fabbian, R. De Dear, and S. Lellyett, “Application or neural network forecasts to predict fog at Canberra International Airport,” Weather and Forecasting, vol. 22, no. 2, pp. 372-381, Apr. 2007.

[19] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feed forward neural networks,” in Proc. AISTATS, 2010, pp. 249–256.

[20] A. Mohamed, G. Dahl, and G. Hinton, “Deep belief networks for phone recognition,” in Proc. NIPS Wor Workshop Deep Learning for Speech Recognition and Related Applications, 2009.

[21] H. Yuan, X. Gao, S.L. Mullen, S. Sorooshian, J. Du, and H-M. H. Juang, “Calibration of probabilistic quantitative precipitation forecasts with an artificial neural network,” Weather and Forecasting, vol. 22, no. 6, pp. 1287-1303, Dec. 2007.