Available online at: https://ijact.in

Date of Submission	20/07/2019
Date of Acceptance	10/09/2019
Date of Publication	05/10/2019
Page numbers	3431-3439(9 Pages)

Cite This Paper: Shobha N., Asha T. Mean Squared Error Applied in Back Propagation for Non Linear Rainfall Prediction, 8(9), COMPUSOFT, An International Journal of Advanced Computer Technology. PP. 3431-3439.

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

MEAN SQUARED ERROR APPLIED IN BACK PROPAGATION FOR NON LINEAR RAINFALL PREDICTION

Shobha N[1], Asha T[2]

[1]Dept of Information Science and Engineering, APSCE, Bengaluru, Karnataka
[2]Dept of Computer Science and Engineering, BIT, Bengaluru, Karanataka

Abstract: Analyzing global weather forecasts is challenging and expensive. Machine learning algorithms analyze trends in weather data by adopting regression model and neural network model. Our proposed model is based on two methodologies Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) to predict rainfall. MLR determines most significant parameters of rainfall data for ANN. Mean Squared Error (MSE) generated at ANN model was back propagated to get more accurate results. The model was tested on five year (2013 to 2017) meteorological data of Bengaluru station. ANN with Back Propagation Neural Network (BPNN) was applied to forecast low rainfall, average rainfall and heavy rainfall. Performance of the model was measured by R and MSE value. Experimental result shows that back propagation of MSE and data normalization yields good results.

Keywords: Meteorological Data; Multiple Linear Regression; Artificial Neural Network; Back Propagation Neural Network; Data normalization;

I. INTRODUCTION

Weather alerts are necessary because they preserve life, economic losses and wealth. Many meteorological indices such as drought index, the critical point of temperature and rainfall are important to improve quality of life. Researchers determine the structure of patterns and correlation trends in weather dataset. Climatic observations are chaotic in nature and prediction becomes less accurate as forecast data increases. These observations are then used to build model to detect rainfall, storms, flood forecast, precipitation amount and advisory to the farmer and fisherman etc. Analyst use ANN [1][2][3] and MLR [4][5] model to solve different issues related to rainfall prediction. Features in weather data have distinct range. Data normalization makes dataset to fall on common scale. In our study ANN algorithm gives better result with normalized data. In the first step, out of ten input variables, MLR finds four variables as influential parameters that cause rainfall and given as input to ANN. The input data was divided into True and False based on rainfall value nonzero and zero. Again split nonzero values into two groups True_Above(dataset) and True_Below(dataset), ANN was experimented on both the dataset. To improve prediction results, MSE was back propagated and ANN was tested on modified dataset. The proposed model was able to predict less rainfall, average rainfall and heavy rainfall. Model evaluation was done by p value, R and MSE value.

Related work was explained in section II. Methodological description was given in section III. Graphical representation of results was outlined in section IV followed by concluding remarks in section V.

II. Related Work

An artificial neural network is an information processing technique. ANN can apply for regression of continuous attributes to predict target variable.

A model consisting of artificial neural network and wavelet decomposition was proposed to predict daily rainfall by (Wassamon et al., 2009) [1]. The rainfall data was collected from five stations of south Thailand over the period 1995-2006. One set of wavelet pattern representing the rainfall data and another wavelet pattern acting as smoothing filter were given as input to artificial neural network. Multilayer feed forward neural network was applied to predict rainfall data for n successive days. Performance of the network was measured by coefficient of determination (R^2) and root mean square (RMSE) values.

A case study on monthly rainfall prediction using artificial neural network was given by (Mislan et al., 2015) [2]. A back propagation neural network algorithm with two hidden layers and three different epochs were applied on rainfall data which was collected from the year 1986-2008. Predictive accuracy was calculated using mean square error (MSE).

Application of Artificial neural network for monthly and seasonal rainfall forecasting in Queensland, Australia was given by (John Abbot et al., 2012) [3]. The model was built with inputs climatic indices, monthly rainfall, atmospheric temperature, solar data and derived output as monthly rainfall forecast, three months in advance. The experiment was conducted using neural network software called Synapse, where weights of neurons were incrementally adjusted based on calculated error to achieve target output. Model validation was done by comparing RMSE and Pearson correlation coefficient values with predictive ocean atmosphere model (POAMA) and general circulation model (GCM).

Author (Adil et al., 2016) [4] has combined two methods called clustering and regression to predict monthly rainfall in Victoria of Australia. The performance of the model was compared with other prediction models such as ANN, MLR and support vector machine for regression (SVMreg) with measures called root mean square, mean absolute scaled error and mean absolute error.

A novel method to discover information from general groups of records was proposed by (Harsha et al., 2012) [5]. The learning algorithm uses prior experience and current data to estimate outcome of day-to-day occurring accidents. DIgGER algorithm uses genetic algorithm to solve the limitations of linear prediction system. Algorithm finds next outcome using predicted output and actual output, these values were called as delta values. The delta values were used for error correction and algorithm study genetically.

Imbalanced classification techniques was applied to check the climate indices and to forecast western north pacific summer monsoon was proposed by (Troncoso et al., 2017) [6]. Initially monthly index of monsoon intensity was developed and models based on trees, rules and black box models were built to predict forecast. The results were shown graphically.

III. Methodology

A. The dataset

The dataset comprises of meteoric scrutiny of five years (2013 to 2017) from the Indian Meteorological Department, Bengaluru [15]. The meteorological variables temperature1, humidity1, cloud1, wind1, slp1 recorded at 0700 LMT (7:20 AM or Indian Standard Time 7:20) temperature2, humidity2, cloud2, wind2, slp2 recorded at 1400LMT(2:20 PM or Indian Standard Time 14:20)

B. Multiple Linear Regression

There exists relationship between two or more variables in the dataset. Regression model predicts specific value of one variable in terms of another. Multiple Linear Regression learns the relationship existing between target variable and explanatory variable.

Input dataset consist of ten explanatory variables temparature1, temparature2, humidity1, humidity2, cloud1, cloud2, wind1, wind2, surface pressure1, surface pressure2 and one target variable rainfall. This dataset fed as input to MLR model to find parameters that have significant relation existing with target variable rainfall.

C. Artificial neural network

Artificial Neural Network (ANN) is a program that learns how to do things. Brain contains 100 billion cells called neurons. These neurons have dendrites and axons which are used to send signals to other neuron. Neural network is computerized in similar way. It has an input layer, an output layer and hidden layer existing between input and output layer. Each connection is associated with weights and bias unit is connected with hidden and output neuron.

Training a neural network is adjusting all the weights based on the input which spit out the correct outputs. Back propagation algorithm propagate one way through the network from input to output then propagate back from output to input. If target value is not matching with output values then error will be calculated using the formula:

$$E_{error} = \sum 1/2(Target - Output)^2$$

Based on the error value weight at each layer is updated. Forward propagate and calculate output values then check with the target values until error is minimized.

ANN with back propagation has been used for nonlinear rainfall prediction. Essential patterns obtained from MLR method were considered as significant and given as input to ANN model. The input data has 1826 instances with five essential trends such as humidity1, humidity2, cloud1, wind1 and predicted variable rainfall. The objective is to predict Rainfall=low, average and high by considering this dataset. The model was built in mat lab and steps were described below.

1. The first step is to set the input and output dataset, then we use nnstart() function and four hidden neurons were selected to design the network. The ANN network is composed of an input layer of four neurons, an output layer with one neuron and four hidden layer.
2. In the second step 70% of sample data was taken as training, 15% of sample data was considered as testing and remaining 15% data was reserved for validation.
3. In third step the model was trained using Levenberg Marquardt method and performance of an algorithm on raw data, 0 to 1 and -1 to 1 normalized data was observed.
4. Data division procedure was explained in step wise and diagrammatical representation was shown in the Fig. 1. Input data has 1826 instances, zero and non zero values of rainfall were assigned to False class and True class respectively. ANN algorithm was applied on raw data, 0 to 1 and -1 to 1 normalized data.
5. In the initial step, rainfall = True instances was chosen, out of 1826 records 400 records belongs to True class. True class instances were again divided into two groups "True_Below"(dataset) and "True_Above"(dataset). The "True_Below" dataset have rainfall values min to (min+max)/2 and "True_Above" dataset have rainfall values ((min+max)/2)+1 to max. First 2/3 percent of "True_Below" instances were classified as low and remaining 1/3 percent of instances were classified as high. ANN method was executed on the classified instances of raw data, normalized data and results were tabulated.
6. In the next step, instances belongs to "True_Above" (dataset) were considered. First 1/3 percent of "True_Above" dataset instances were split into low and remaining 2/3 ratio of records were split into high. The Above class records were normalized to 0 to 1 and -1 to 1 and ANN algorithm was compiled on the dataset and results were outlined.
7. In the final step mean squared error is used as error correction. In back propagation, mean squared error for training, testing and validation was added with corresponding value of attribute in the "True_Below" dataset called "True_Below_MSE" and "True_Above" dataset called "True_Above_MSE". The modified data fed as input to ANN. Results of an algorithm on the dataset were analyzed with respect to performance plot and regression plot.
8. An algorithm for proposed model was summarized below

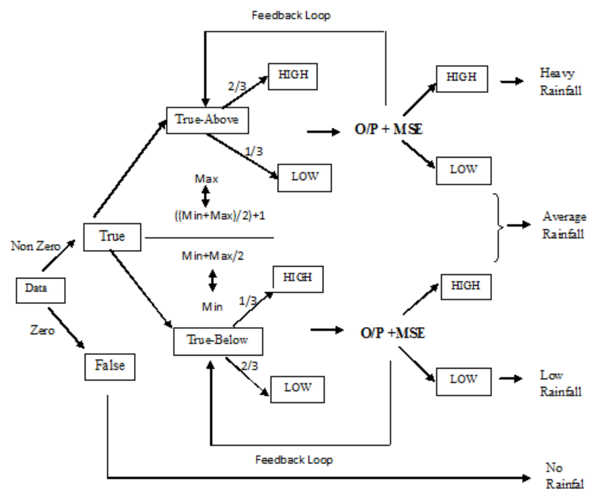

Fig.1. Pictorial representation of data division

Algorithm: Proposed model MLR Step 1: Data preprocessing Step 2: Compute multiple linear regression function on the data set. Step 3: predicting the output result Data division Step 4: Select number of significant patterns from step2 write into 'weatherdata'. Step 5: call ANN( )

```
Step 6: sequence = [ ]
Step 7: for item in weatherdata.rainfall:
Step 8:      if(item>0):
Step 9:                  sequence.append(item)
Step 10: min = sequence[0]
Step 11: for item in sequence:
Step 12:   if(item<min)
Step 13:                  min=item
Step 14: max = sequence[0]
Step 15: for item in sequence:
Step 16:   if(item>max)
Step 17:                  max=item
Step 18: avg = (min+max)/2
Step 19: avg1 = ((min+max)/2) + 1
Step 20: bol = [ ]
Step 21: bol1 = [ ]
Step 22: for item in weatherdata.rainfall:
Step 23:   if(item < = avg):
Step 24:                  bol.append(True_Below)
Step 25:   else:
Step 26:                  bol1.append(True_Above)
Step 27: avg2 = avg*2/3
Step 28: for item in True_Below.rainfall:
Step 29:   if(item < = avg2)
Step 30:                  rainfall.item = low
Step 31:   else:
Step 32:                  rainfall.item = high
Step 33: is_long = pd.series(bol)
Step 34: print (True_Below[is_long])
Step 35: call ANN( )
Step 36: a = avg1*1/3
Step 37: avg3 = a+avg1
Step 38: for item in True_Above.rainfall:
Step 39:   if(item < = avg3)
Step 40:                  rainfall.item = low
Step 41:   else:
Step 42:                  rainfall.item = high
Step 43: is_long = pd.series(bol1)
Step 44: print(True_Above[is_long])
Step 45: call ANN( )
```

IV. EXPERIMENTAL RESULTS

The MLR model was built using R software. The model has ten inputs and four outputs. The result of the model was tabulated in Table I.

TABLE I. COEFFICIENT VALUES FOR REGRESSION MODEL

	Estimate	Std. Error	P-value
(Intercept)	88.99952	115.2696	0.440156
temperature 1	-0.199840	0.123471	0.105719
temperature2	0.125645	0.134289	0.349586
humidity1	0.085905	0.042329	0.042559
humidity2	0.080012	0.025621	0.001819
cloud1	0.869381	0.154990	2.34473E-08
cloud2	0.154437	0.253046	0.541732
wind1	0.013736	0.003359	4.53346E-05
wind2	-0.003707	0.004826	0.442593
surface pressure1	0.273294	0.158566	0.084961
surface pressure2	-0.38199	0.197582	0.053349

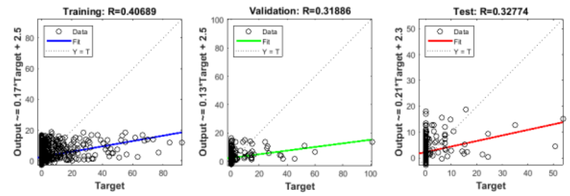

Fig.2. (a) Regression plot of raw data

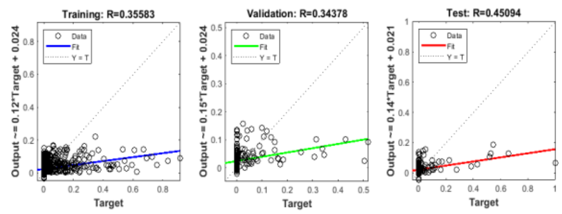

Fig.2. (b) Regression plot of 0 to 1normalized data

As shown in Table I, p-value of humidity1 is 0.04, humidity2 is 0.0018, cloud1 is 2.34e-08 and wind1 is 4.53e-05, these variables were considered as significant towards predicted variable rainfall. Remaining variables were considered as not significant because their p-value greater than or equal to 0.05 and also coefficient values were lower than standard error. Intercept is the y intercept, coefficient value was around 88.99, slope for humidity1 is 0.085, humidity2 is 0.080, cloud1 is 0.86 and wind1 is 0.013. The regression equation was given by

Estimated rainfall= 88.9995+0.08590*humidity1+ 0.0800 * humidity2 + 0.8693*cloud1 + 0.0137*wind1

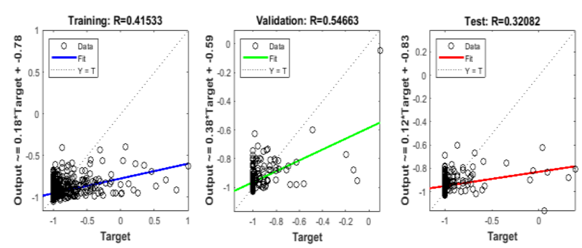

Fig.2. (c) Regression plot of -1 to 1 normalized data

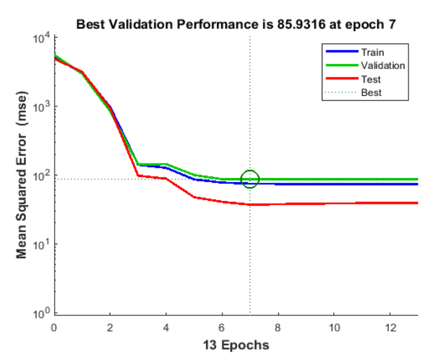

Fig.2. (d) performance plot of raw data

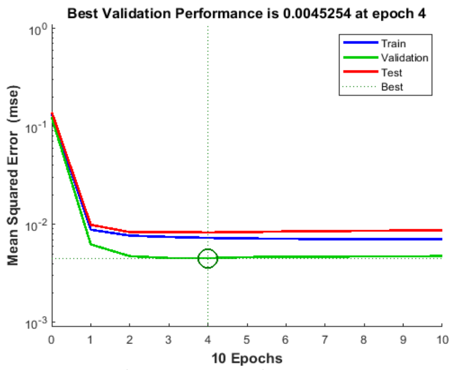

Fig.2. (e) Performance plot of 0 to 1 normalized data

The ANN model was built by using MATLAB. Model has 4 input layers, 4 hidden layers and 1 output layer. The function fitnet() was used to define number of hidden layers and trainlm() function was used to call Levenberg Marquardt algorithm to train the data. Input file consisted of 1826 observations and five columns of attributes of daily rainfall data. Results of ANN on the dataset were shown in the Fig. 2(a).

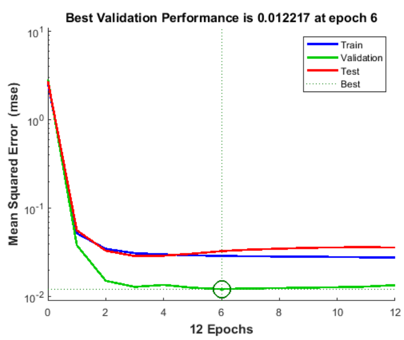

Fig.2. (f) Performance plot of -1 to 1 normalized data

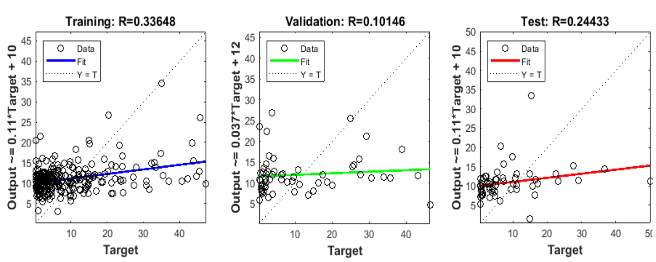

Fig.3. (a) Regression plot of True_Below raw data

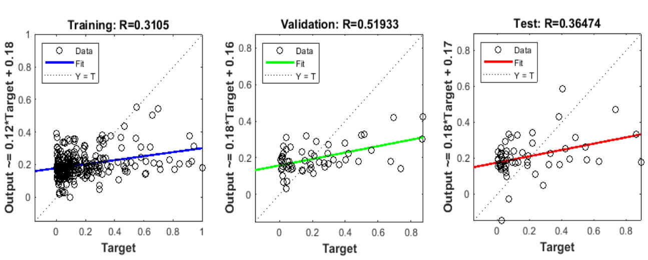

Fig.3. (b) Regression plot of True_Below 0 to 1 normalized data

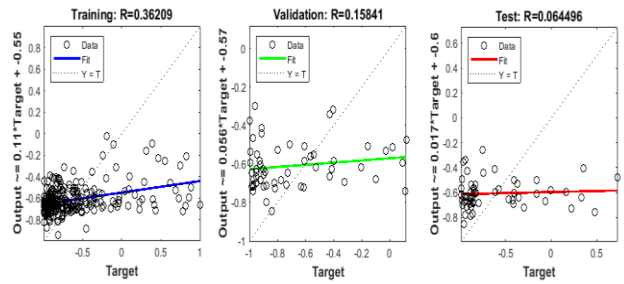

Fig.3. (c) Regression plot of True_Below -1 to 1 normalized data

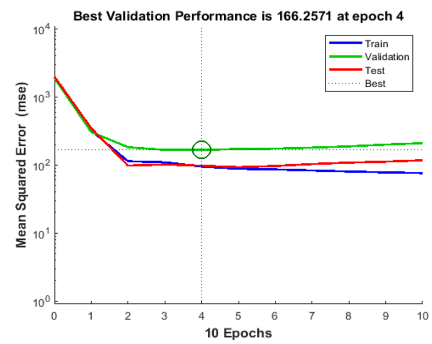

Fig.3. (d) Performance plot of True_Below raw data

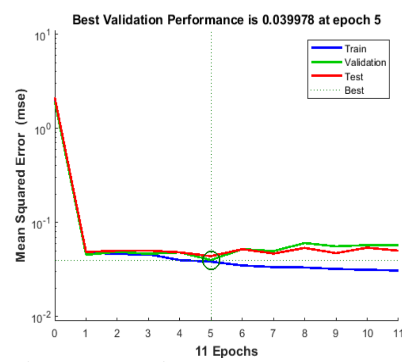

Fig.3. (e) Performance plot of True_Below 0 to 1 normalized data

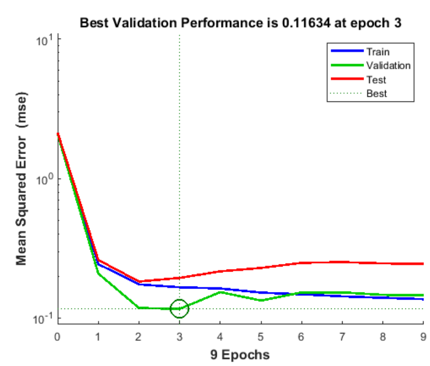

Fig.3. (f) Performance plot of True_Below -1 to 1 normalized data

The model produces R value of 0.4068, 0.3188 and 0.3277 for training, validation and testing respectively. Since the predicted variable Rainfall is continuous attribute the value may vary from 0(minimum) to 100.8(maximum) hence the estimated outcome was nearly equal to 40%. The same dataset was normalized to 0 to 1, -1 to 1and ANN results on the dataset produces R value of 0.35, 0.34, 0.45 and 0.41, 0.54, 0.32 for training, validation and testing respectively and shown in the Fig. 2(b), 2(c).

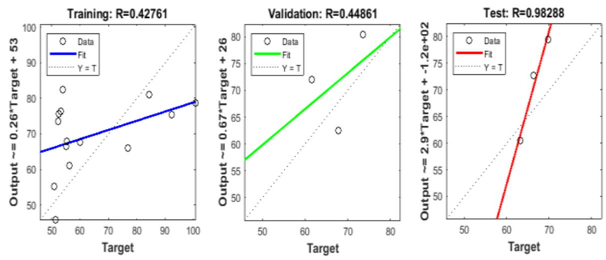

FIg.4. (a) Regression plot of True_Above raw data

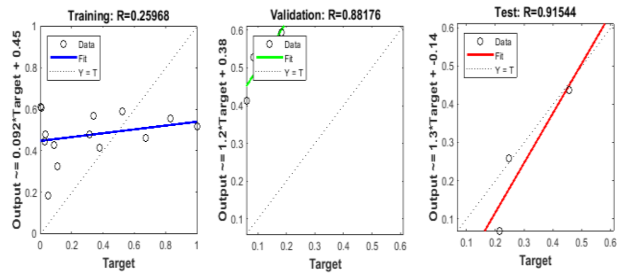

Fig.4. (b) Regression plot of True_Above 0 to 1 normalized data

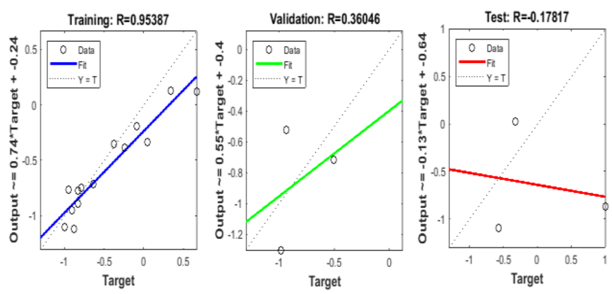

Fig.4. (c) Regression plot of True_Above -1 to 1 normalized data

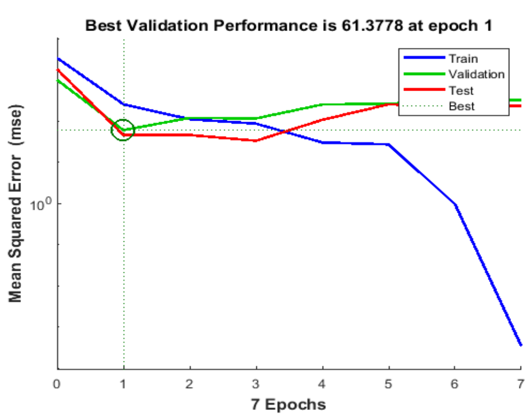

Fig.4. (d) Performance plot of True_Above raw data

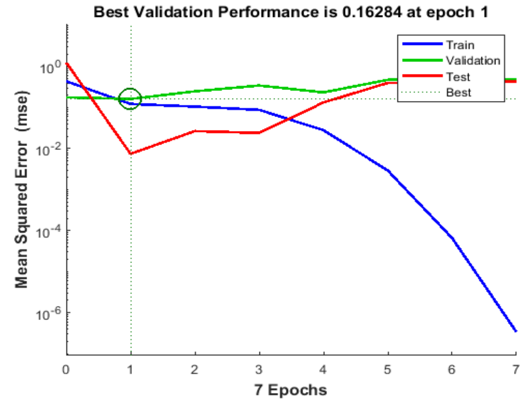

Fig.4. (e) Performance plot of True_Above 0 to 1 normalized data

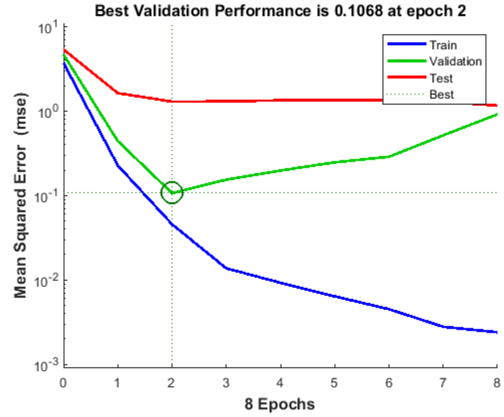

Fig.4. (f) Performance plot of True_Above -1 to 1 normalized data

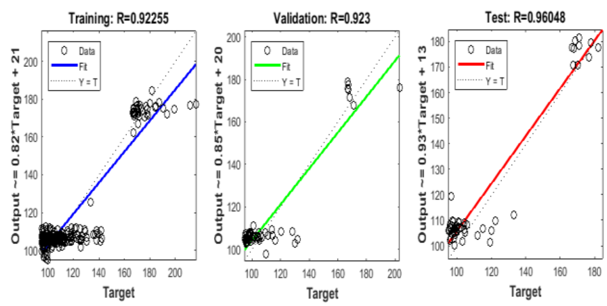

Fig.5. (a) Regression plot of True_Below_mse raw data

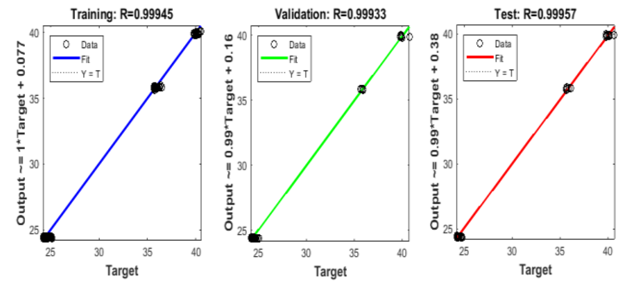

Fig.5. (b) Regression plot of True_Below_mse 0 to 1normalized data

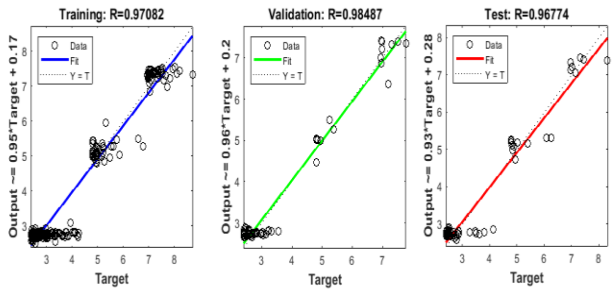

Fig.5. (c) Regression plot of True_Below_MSE -1 to 1normalized data

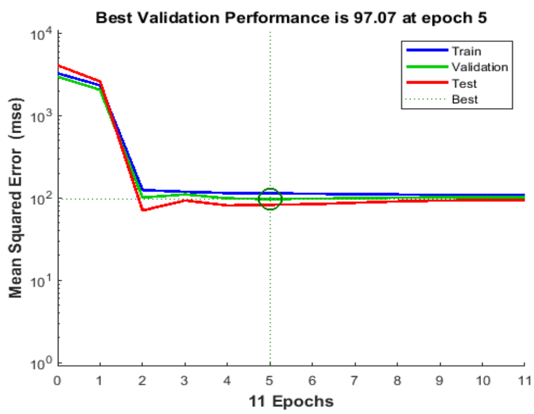

Fig.5. (d) Performance plot of True_Below_MSE raw data

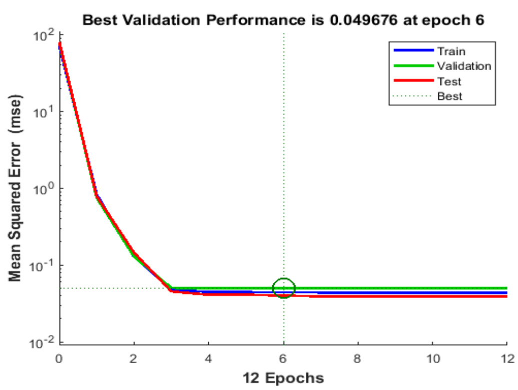

Fig.5. (e) Performance plot of True_Below_MSE 0 to 1normalized data

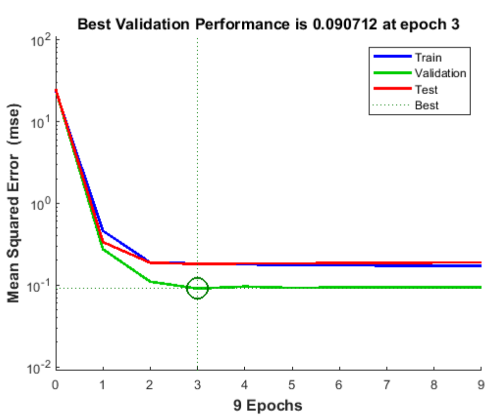

Fig.5. (f) Performance plot of True_Below_MSE -1 to 1 normalized data

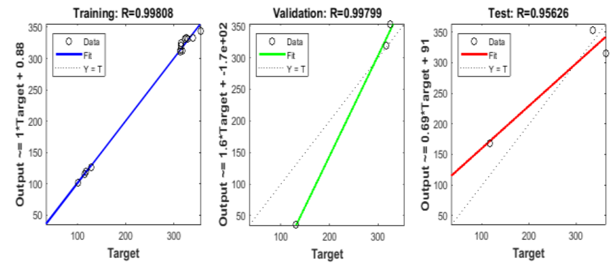

Fig.6. (a) Regression plot of True_Above_MSE raw data

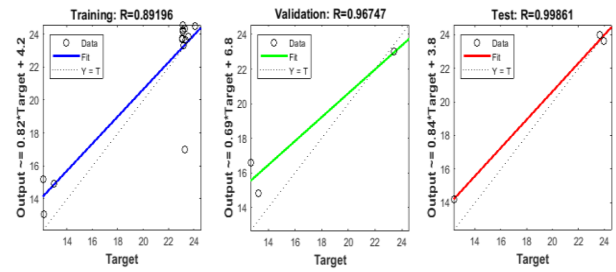

Fig.6. (b) Regression plot of True_Above_MSE 0 to 1 normalized data

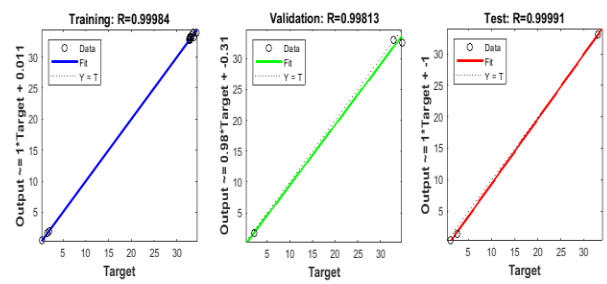

Fig.6. (c) Regression plot of True_Above_MSE -1 to 1 normalized data

As shown in performance plot, blue line represents training, green line represents validation and red line represents testing. The performance plot produces the best validation performance of 85.93 at epoch 7, 0.0045 at epoch 4 and 0.0122 at epoch 6 for raw data, 0 to 1 and -1 to 1 normalized data respectively, MSE value was reduced and all three lines becomes straight line, was shown in the Fig. 2(d), 2(e) and 2(f). R value was improved and MSE value was reduced in case of normalized data as compare to raw data.

In the next step "True-Below"(dataset), instances were considered and data consisted around 380 records. ANN algorithm was applied on the dataset to find low rainfall and high rainfall, the regression graph for raw data was shown in Fig. 3(a) and it has produced R value of 0.33, 0.10, 0.24 for training, validation and testing respectively. The model produces R value of 0.31, 0.51, 0.36 and 0.36, 0.15, 0.06 for training, validation and testing with respect to 0 to 1 and -1 to 1 normalized data was shown in the Fig. 3(b) and 3(c). Performance plot as shown in Fig. 3(d), 3(e), 3(f), MSE was reduced and all three lines converges at 10^2 with best validation performance of 166.25 at epoch 4 for raw data, all three lines converges at 10^{-1} with best performance of 0.03 at epoch 5 and 0.11 at epoch 3 for 0 to 1 and -1 to 1 normalized data. To predict rainfall is low and high, the regression result was around 33%, to improve the regression results, the MSE value of training, validation and testing was back propagated and added with each attribute in the dataset and algorithm was executed on the modified dataset (True_Below_MSE).

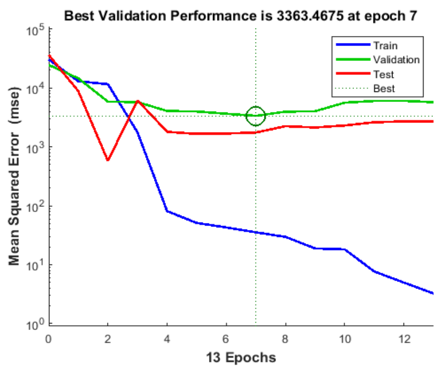

Fig.6. (d) Performance plot of True_Above_MSE raw data

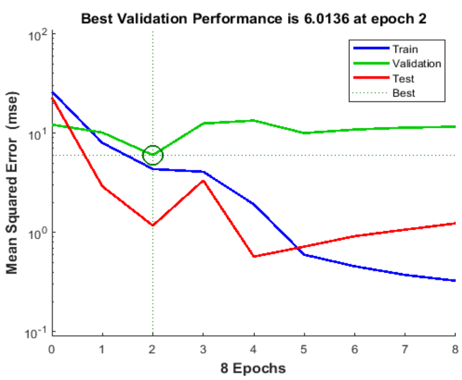

Fig.6. (e) Performance plot of True_Above_MSE 0 to 1 normalized data

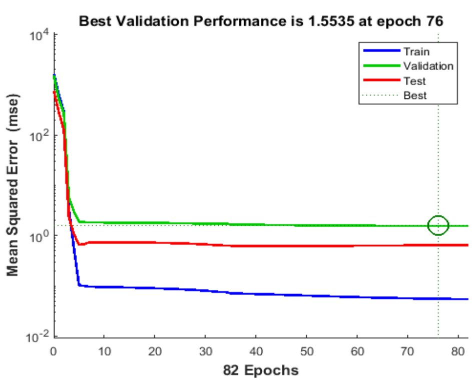

Fig.6. (f) Performance plot of True_Above_MSE -1 to 1 normalized data

The regression results on "True_Below_MSE" were shown in Fig. 5(a), 5(b) and 5(c). The model produces R value of 0.92, 0.92, 0.96 for training, validation and testing respectively with respect to raw data. For 0 to 1 and -1 to 1 normalized data, the model produces R value equal to 0.99, 0.99, 0.99 and 0.97, 0.98, 0.96 for training, validation and testing respectively. In case of raw data the prediction results was improved by 92%, in case of 0 to 1 and -1 to 1 normalized data the regression results was improved by 99% and 97% respectively.

TABLE II: REGRESSION VALUE OF DATASET

	R^2		
Original Dataset	**Training**	**Test**	**Validation**
Raw data	0.4068	0.3188	0.3277
0 to 1 normalized data	0.3558	0.3437	0.4509
-1 to 1 normalized data	0.4153	0.5466	0.3208
True_Below	**Training**	**Test**	**Validation**
Raw data	0.3364	0.1014	0.2443
0 to1 normalized data	0.3105	0.5193	0.3647
-1 to 1 normalized data	0.3620	0.1584	0.0644
True_Below_MSE	**Training**	**Test**	**Validation**
Raw data	0.9225	0.923	0.9604
0 to1 normalized data	0.9994	0.9993	0.9995
-1 to 1 normalized data	0.9708	0.9848	0.9677
True_Above	**Training**	**Test**	**Validation**
Raw data	0.4276	0.4486	0.9828
0 to1 normalized data	0.2596	0.8817	0.9154
-1 to1 normalized data	0.9538	0.3604	-0.1781
True_Above_MSE	**Training**	**Test**	**Validation**
Raw data	0.9980	0.9979	0.9562
0 to1 normalized data	0.8919	0.9674	0.9986
-1 to1 normalized data	0.9998	0.9981	0.9999

As shown in the Fig. 5(d), 5(e) and 5(f), the MSE value exponentially decaying and all the lines converging to 10^2 with validation performance 97.07 at epoch 5 for raw data, converging at 10^{-1} with validation performance 0.04 at epoch 6 and 0.09 at epoch 3 for 0 to 1 and -1 to 1 normalized data. Result shows MSE value reduces and regression value improves by 99% when MSE value fed through back propagation in the model and our objective to predict rainfall=low and rainfall=high becomes true in "True_Below_MSE" (dataset).

ANN algorithm was applied on "True_Above" (dataset) to find rainfall=low and rainfall=high, the execution results on dataset were displayed as regression graph shown in Fig. 4(a), 4(b) and 4(c). As shown in the Fig. 4(a), regression graph for raw data produces R value 0.42, 0.44, 0.98 for training, validation and testing respectively. For 0 to 1 and -1 to 1 normalized data, regression graph has R value 0.25, 0.88, 0.91 and 0.95, 0.36, -0.17 for training, validation and testing respectively. As shown in the Fig. 4(d) and 4(e), the blue line exponentially decaying where as red and green line become closer to best line with best validation performance of 61.3 at epoch 1 for raw data and 0.16 at epoch 1 for 0 to 1 normalized data. For -1 to 1 normalized data, MSE of training reduces, MSE of testing slightly reduces and becomes straight line where as validation reduced at some point and slightly increases with best validation performance 0.10 at epoch 2 was shown in Fig. 4(f). Regression value for training and testing in case of raw data was around 42% and 98%, training and testing value of 0 to 1 normalized data was 25% and 91% and regression value of testing was negative in case of -1 to 1 normalized data. To improve results of "True-Above" dataset, the MSE value of training, validation and testing was back propagated and added with each input attribute and ANN algorithm was tested on new dataset "True_Above_MSE".

The regression result of "True_Above_MSE" was shown in the Fig. 6(a), 6(b) and 6(c). Regression on raw data produces R value 0.99, 0.99, 0.95 for training, validation and testing respectively. Regression on 0 to 1 and -1 to 1 normalized data generates R value 0.89, 0.96, 0.99 and 0.99, 0.99, 0.99 for training, validation and testing respectively. MSE value of training exponentially decaying where as testing and validation converges to best line with performance of 3363 at epoch 7 for raw data as shown in fig 6(d). For 0 to 1 normalized data blue and red line randomly decaying at some instances and green line become closer to best line with best validation performance of 6.01 at epoch 2 was shown in Fig. 6(e). MSE was decreasing and at some point all three lines become straight line with best validation 1.55 at epoch 76 for -1 to 1 normalized data was shown in the Fig. 6(f).

In "True_Above_MSE" (dataset) after adding MSE, the regression accuracy improved by 99% and our prediction rainfall=low and rainfall=high becomes true. By adding rainfall=high instances from "True_Below" (dataset) and rainfall=low instances from "True_Above" (dataset) will get average rainfall instances. By considering results of original data, True_Below_MSE and True_Above_MSE our model was predicting rainfall=no, rainfall=low, rainfall=average and rainfall=high.

As explained in section IV, Table II shows regression values of dataset and graph was plotted for the same. As shown in the Fig. 7, Blue line represents training data, brown line represents test data and green line represents validation. Graph shows regression value of original data, True_Below, True_Below_MSE, True_Above and True_Above_MSE. Before adding MSE the regression value of dataset was below 0.5. When MSE value was back propagated and added with the dataset then regression value was increased by 0.92 to 0.99 for True_Below_MSE and True_Above_MSE dataset. The result concludes adding MSE and use of data normalization improves regression accuracy.

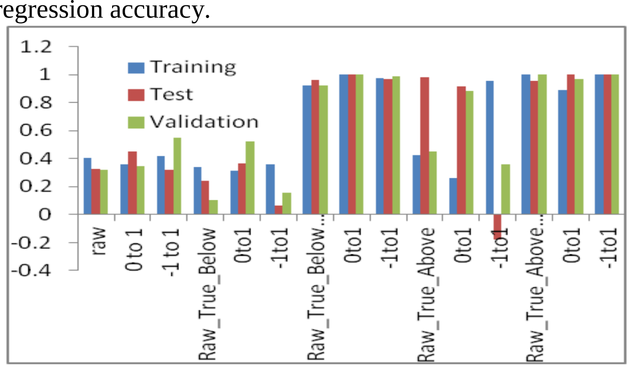

Fig.7. R value of Dataset

V. CONCLUSION

Proposed model was composed of two methods MLR and ANN. MLR were run on meteorological data to predict most significant parameters related to rainfall. ANN was applied on those significant parameters and predicts low rainfall, average rainfall and heavy rainfall. Accuracy of the model was improved by back propagating MSE in the ANN model and data normalization enhances the model performance. Finally ANN performs better against other linear method.

VI. REFERENCES

[1] Wassamon Phusakulkajorn, Chidchanok Lursinsap and Jack Asavanant, "Wavelet-Transform Based Artificial Neural Network For Daily Rainfall Prediction in Southern Thailand", *ISCIT 2009*.

[2] Mislana, Haviluddinb, Sigit Hardwinartoc, Sumaryonod and Marlon Aipassa, "Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan - Indonesia," *Elsevier, Procedia Computer Science 59 (2015) 142 – 151, International Conference on Computer Science and Computational Intelligence (ICCSCI 2015).*

[3] John ABBOT and Jennifer MAROHASY, "Application of Artificial Neural Networks to Rainfall Forecasting in Queensland, Australia," *ADVANCES IN ATMOSPHERIC SCIENCES, VOL. 29, NO. 4, 2012, 717–730, doi: 10.1007/s00376-012-1259-9.*

[4] Adil M Bagirov, et al., "Prediction of monthly rainfall in Victoria, Australia: Clusterwise linear regression approach," Elsevier, Atmospheric Research (2017), doi:10.1016/j.atmosres.2017.01.003.

[5] Harsha S, N Bhaskar, Amith Kumar, V Pujari and Nibha, "A Novel Method to Dig Information from Generic Groups of Enriched Records (DIgGER)," *ResearchGate, IJCSSEIT, Vol. 5, No. 2, December 2012, pp. 243-248.*

[6] A. Troncoso, P. Ribera, G. Asencio-Cort, I. Vega and D. Gallego, "Imbalanced classification techniques for monsoon forecasting based on a new climatic time series," *Elsevier, Environmental Modelling & Software (2017) 1-9.*

[7] Shobha N, Asha T, "Influential parameters on rainfall forecasting using multiple linear regression," *2019 JETIR February 2019, Volume 6, Issue 2 (ISSN-2349-5162).*

[8] Jiansheng Wu, Jin Long, Mingzhe Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," *ELSEVIER, Neurocomputing 148 (2015) 136–142.*

[9] F. Mekanik, M.A. Imteaz, S. Gato-Trinidad, A. Elmahdi, "Multiple regression and Artificial Neural Network for long-term rainfall forecasting using large scale climate modes", *Journal of Hydrology 503 (2013) 11–21.*

[10] Chiang, Y.M., Chang, F.J., 2009. "Integrating hydrometeorological information for rainfall–runoff modelling by artificial neural networks". Hydrological Processes 23 (11), (2009) 1650–1659.

[11] R. Shukla, K. Tripathi, A. Pandey, I. Das, "Prediction of Indian summer monsoon rainfall using nino indices: A neural network approach", *Atmospheric Research 102 (1-2) (2011) 99-109.*

[12] T. Mandal, V. Jothiprakash, "Short-term rainfall prediction using ANN and MT techniques ", *ISH J. of Hydraulic Engineering 18 (1) (2012) 20-26.*

[13] W. DeSarbo, W. Cron, "A maximum likelihood methodology for clusterwise linear regression", *Journal of Classication, 5 (2) (1988) 249-282.*

[14] L. Garcia-Escudero, A. Gordaliza, A. Mayo-Iscar, R. San Martin, "Robust clusterwise linear regression through trimming", *Computational Statistics and Data Analysis 54 (2010) 3057-3069.*

[15] http://www.imd.gov.in (Accessed on October 20, 2018)

[16] http://www.uasbangalore.edu.in/index.php/research/agrometeorology (Accessed on March 04, 2018)

[17] Pang-Ning Tan , Michael Steinbach , Vipin Kumar, Introduction to Data Mining, (First Edition), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2005

[18] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques," Second Edition, ELSEVIER Science Technology, Morgan Kaufmann, 2006, San Francisco.