Deep Learning Models for the Prediction of Rainfall

Aswin S, Geetha P and Vinayakumar R

Abstract—**Rainfall is one of the major source of freshwater for all the organism around the world. Rainfall prediction model provides the information regarding various climatological variables on the amount of rainfall. In recent days, Deep Learning enabled the self-learning data labels which allows to create a data-driven model for a time series dataset. It allows to make the anomaly/change detection from the time series data and also predicts the future event's data with respect to the events occurred in the past. This paper deals with obtaining models of the rainfall precipitation by using Deep Learning Architectures (LSTM and ConvNet) and determining the better architecture with RMSE of LSTM as 2.55 and RMSE of ConvNet as 2.44 claiming that for any time series dataset, Deep Learning models will be effective and efficient for the modellers.**

Index Terms—**ConvNet, Deep Learning, LSTM, Precipitation, Rainfall Prediction.**

I. Introduction

RAINFALL is a form of precipitation where the water is distributed among oceans, atmosphere and surface of the Earth. It is important for life where water is evaporated from the Earth surface and rises as water vapor into the atmosphere where heat is carried from the surface with it. Rain is responsible for storing most of the fresh water upon the globe for plants and animals. The water vapor that rises into atmosphere condenses forming cloud droplets and rainfall releases into atmosphere [1] [2].

The requirement in prediction of rainfall is important while considering the various factors involved. Climate Prediction Center's provides data on rainfall prediction such as National Centers for Environmental Prediction (NCEP), National Weather Service (NWS) in the National Oceanic and Atmospheric Administration (NOAA) by providing the assess and forecasts the impacts of changes in climate and intimating the risks caused for precaution and also to reduce economic risks for maximizing the profits1[1-5].

Many studies have been carried out in the prediction of rainfall. The data for rainfall predictions are available for many years and use of neural network models for such time-series data indicates that the complexity in processing is higher and accomplishes by predicting amount of rain in a specific area with less accuracy[6-9]. Deep-Learning technologies are different from neural networks as the models are changed with respect to the increase in the hidden layers and better performance. Deep-learning model has the benefit of implementing a single model as the features in the data's are extracted and classified as a single classifier [10-16].

In this paper, the time-series data of rainfall obtained from NCEP center which gives the information of the parameter involved i.e., Global Monthly Average Rainfall for different locations has been pre-processed and predicted using Deep Learning Architectures (LSTM and CNN) creating models in a deep neural network consisting of multiple layers of hidden units and communicating with different layers in the network with much better accuracy than the earlier methods. In earlier methods, data is modelled only by feature extraction but this method of using LSTM and CNN (ConvNet) removes the technique of feature engineering which make one of the advantages of these models. These models help in reducing the computational complexity. LSTM model and ConvNet model shows almost similar error rate as computational analysis without the use of GPU, lacks in the performance with respect to the tuning of hyper-parameters of these models[17-22].

This paper includes various sections where Section II contains the Literature Work, Section III details the Background work of the architectures used in this paper, Section IV gives detailed information regarding the dataset used, the Proposed Architecture of the predictive model, Section V discusses regarding the Results of the work done and Section VI concludes the paper with future work.

II. Literature Work

P Malhotra et.al(2015) discusses that the stacked LSTM networks learns higher level temporal patterns and may be a reliable technique to model time series data that can be used to detect/predict anomalies with results on four real-world datasets which involves small-term and long-term temporal behavior data model [7]. The Wavelet Neural Network (WNN) model proposed by R.Venkata Ramana et.al (2013) was applied to dataset of monthly rainfall obtained from ground station of Darjeeling. The analysis shows that the WNN models works relatively good comparing with ANN [4].

Aswin S is with the Centre for Computational Engineering and Networking, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidhyapeetham, India. (E-mail: shivvashwin11@gmail.com).

Geetha P is with the Centre for Computational Engineering and Networking, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidhyapeetham, India. (E-mail: p_geetha@cb.amrita.edu).

Vinayakumar R is with the Centre for Computational Engineering and Networking, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidhyapeetham, India. (E-mail: vinayakumarr77@gmail.com).

978-1-5386-3521-6/18/$31.00 ©2018 IEEE

Mohini.P.Darji et.al (2015) discusses and surveyed the problems while applying Neural Networks for rainfall forecasting. The results from the survey shows that neural networks applied are compatible to predict rainfall and feed-forward neural network works well for monthly rainfall data and time-delay neural network works well for yearly rainfall data [2].

S.Narejo et.al (2016) implemented a predictive model for weather forecasting for a multivariate time-series dataset using Deep Belief Network (DBN) and Restricted Boltzmann Machines (RBM) that resulted in an outstanding performance and higher accuracy [3][5]. R.Senthil Kumar et.al analyzed the prediction of rainfall using different data mining techniques and compared those different techniques such that Decision Trees and K-mean clustering shows the better results for limited data. When the dataset in increased, the accuracy is decreased later.

Pallavi et.al (2016) presented the survey of rainfall prediction using ANN and Fuzzy Logic methods. From the survey, it been stated that the Adaptive Neuro Fuzzy Inference System (ANFIS) method results to show a satisfactory level of results compared to other techniques [9]. Mislan et.al (2015) experimented the prediction of rainfall time-series dataset using ANN with Back Propagation Neural Network (BPNN) algorithm that performs with higher accuracy with a very least MSE [8].

Ankit Sharma et.al (2015) developed ANN using BPNN for predicting the rainfall in regions of Delhi. The results concludes that it gives moderate level of accuracy but when there is an increase in the neurons/data, MSE decreases eventually [10].

III. Background

A. Long Short Term Memory (LSTM)

LSTM network is a well-known method for predicting time-series data for any size and duration [14]. LSTM is a building block in the layer of Recurrent Neural Network (RNN) which is preferred to solve the gradient problem that forces the constant error flow in it [16]. A LSTM unit consists of three gates where all three gates acts as controllers for the values going through the network in such a way that it is a multi-layer neural network. The block diagram of LSTM architecture is shown in Fig.1. This Network is trained by back propagating the values through time by iterating for 1000 times to minimize the error by changing the weight of each neuron proportion to its derivative. The activation functions used in this network are Linear [15].

In general, the input data to the LSTM architecture is $x = (x_1, x_2, x_3, \ldots, x_{t-1}, x_t)$ and estimates an output $Y = (Y_1, Y_2, Y_3, \ldots, Y_{t-1}, Y_t)$ by updating the three multiplication unit i.e., Input Gate, i , Output Gate, op, Forget Gate, fr on a memory cell with continuous write, read and reset operations on memory cell, mc from time $t = 1 \; to \; T$ in the hidden layer of this network. The function of this layer is generally formulated for the time step, T as follows:

$$i_t = \sigma(w_i x_t + w_{hi} h_{t-1} + w_{mci} mc_{t-1} + b_i) \quad (1)$$

$$fr_t = \sigma(w_{xfr} x_t + w_{hfr} h_{t-1} + w_{mcfr} mc_{t-1} + b_{fr}) \quad (2)$$

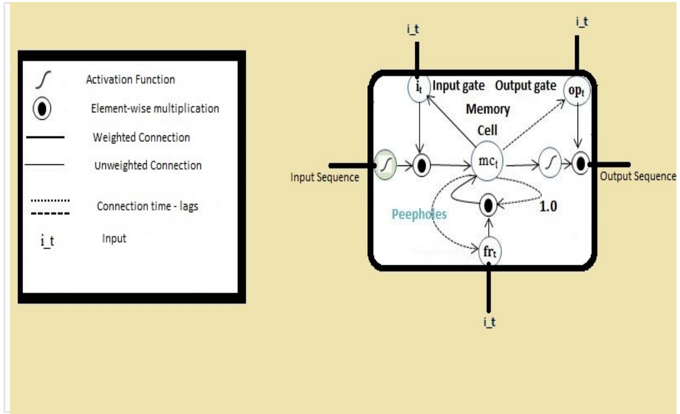

Fig. 1. LSTM – Block Diagram

$$mc_t = fr_t . mc_{t-1} + i_t . \tanh(w_{xmc} x_t + w_{hmc} h_{t-1} + b_{mc}) \quad (3)$$

$$op_t = \sigma(w_{xop} x_t + w_{hmc} h_{t-1} + b_{op}) \quad (4)$$

$$h_t = op . \tanh(mc_t) \quad (5)$$

$b_i, b_{fr}, b_{mc}, b_{op}$ are the bias units of the input, forget, memory and output gate respectively. m, w and h are memory cell, weight matrix and output of the hidden layer respectively. Sigma (σ) and $tanh$ are the activation function in the range [0, 1] and [-1, 1] respectively.

B. Convolutional Neural Network (ConvNet)

ConvNet is a form of Artificial Neural Network (ANN) with neurons having weights and biases. The inputs are received by each neuron and performs non-linear dot product operation. The ConvNet Architecture differs from other neural networks in the way that it also works with multi-dimensional data. The three Operations that is involved in this architecture in form of different layers are Convolution with Non-Linearity (ReLU), Pooling, and Fully Connected Layer [11].

1) *Convolutional Layer:* The main purpose of using Convolutional Layer is to extract the features in the data. Convolution helps to pertain the information of the pixels by understanding the features a matrix smaller than the data. A filter is slided over the input data and the resulting output obtained from this layer is known to be as the Feature Map or Convolved Feature. Generally, the size of this layer is controlled by three parameters: Depth, Stride and Zero-padding [11]. The Activation Function (ReLU) is abbreviated as Rectified Linear Unit which is a non-linear and element-wise operation that replaces all negative pixel values in the output of convolutional layer by zero. The purpose of this function is introducing the non-linearity in ConvNet as most real-time data requires this architecture to learn would be non-linear. Tanh or Sigmoid (σ) can also be used, but ReLU performs well in most cases. The output of this layer is referred to as the 'Rectified' feature map.

2) *Max-Pooling Layer:* Pooling reduces the dimensionality of each output from convolutional layer but preserves the most important information. Max Pooling is

defined by a spatial neighborhood that take the biggest value from the rectified feature map. The problem of over-fitting is controlled due to the reduction in the size and computation.

3) Fully-Connected (FC) Layer: This layer is a MLP using a *Softmax* function in the output layer indicating that every neuron on this layer is connected. The purpose of this layer is to use the features for predicting the input test data based on the train data. The probability of the sum of the output of this will always be 1. [11][17]

IV. EXPERIMENTS

All experiments performed runs on GPU enabled with Tensor Flow [18] in single NVidia GK110BGL Tesla k40 in conjunction with Keras [19]. All deep learning architectures were trained using the back propagation through time (BPTT) [20] technique.

A. Description of the Dataset

The Global Precipitation Climatology Project (GPCP) launched the first version of the dataset which is a monthly precipitation dataset that covers the area globally from the period July 1979 to January 2018. The outcome of the dataset incorporates precipitation estimating from microwave data, infrared data and rain gauge observations which is an analysis by merging the data's by also containing field data from satellite estimation of infrared and microwave wavelengths with error estimates for all fields. The data provides 2.5 degree x 2.5 degree lat-long global grids consisting the matrix size of 144 x 72 x 468. At the regional scale, the differences are systematic with standard climatologies. The Parameter considered in the dataset is the Global Average Monthly Rainfall in the units of millimeters (mm). The objective is to train this Global Average Monthly Rainfall data using Deep Learning architectures (LSTM and ConvNet) and obtain accurate prediction results in the test data with minimal error [21]. The Distribution map of the average rainfall from the data from NCEP for the month of Jan 2018 is shown in Fig. 2 whereas the its general plot is shown in Fig. 3.

The Dataset is preprocessed in the way that the original data which is in a form a 3-Dimensional Array (144 x72 x468) consisting of latitude, longitude and months is converted into a 2-Dimensional Array (10368 x 468) of locations and months. This is done by converting from the array size of (144x72x1) for one month data into (10368 x1) and followed the same for 468 months. The converted dataset with array size of (10368 x 468) is split into four different files for training and testing: X-train and X-test with array size of (10368 x 164) and Y-train and Y-test with array size of (10368 x 70) by having precipitation as features and the data for different months are taken as label.

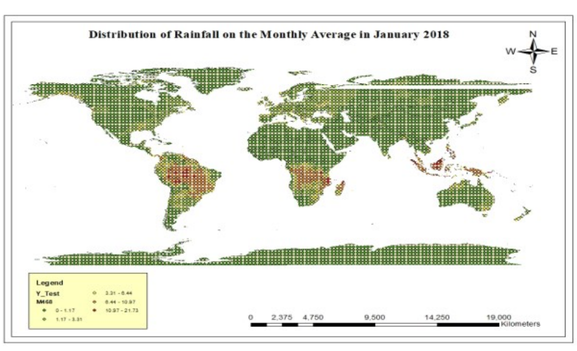

Fig. 2. Geographic Distribution of Monthly Average Rainfall Data Points for one month among 468 months. (Jan 2018)

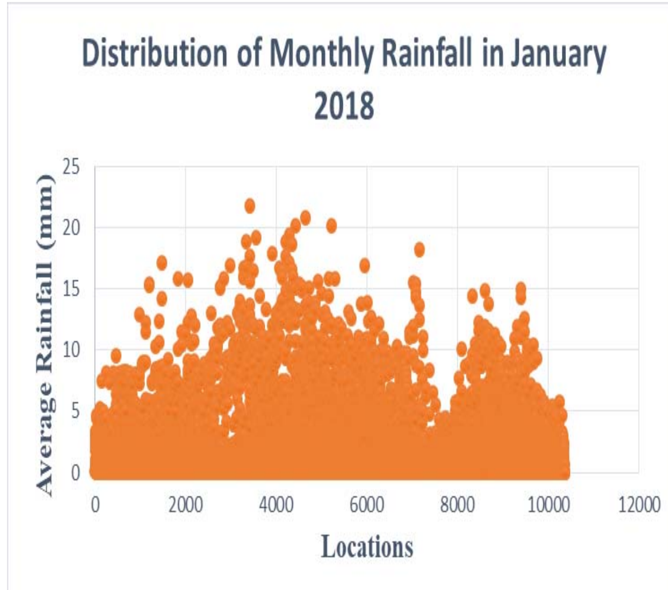

Fig. 3. General Plot of Monthly Average Rainfall Data Points for the Month of Jan 2018

B. Hyper-parameter Selection

Deep learning algorithms such as ConvNet and LSTM are parameterized because the optimal performance depends on the right parameters. To select the number of memory blocks for LSTM, 2 trails of experiments are conducted for the memory blocks of 4, 8, 16, 32 and 64. All trails of experiments are performed in LSTM are conducted for 1000 epochs. For the LSTM layer with 32 memory blocks performed well in comparison to the other memory blocks. Moreover the performance of memory blocks with 64 is closer. So, the number of memory blocks is set to 32 for the rest of the experiments. Similarly in ConvNet, for selecting the number of filters, 2 trails of experiments are conducted for the filter size of 4, 8, 16, 32, and 64 with filter length 3 for 100 epochs. In ConvNet, filter size 32 has performed well and the performance of filter size of 64 is almost comparable to 32. Thus, 32 filters is set for the rest of the experiments. To select the learning rate, three trails of experiments are run in the range [0.01 to 0.05]. Experiments with lower learning rate has performed well comparing to the higher learning rate. Based on the training time and computational cost, we decided to set the learning rate 0.05 for the rest of the experiments.

C. Proposed Architecture

In this paper, we proposed an architecture with two deep learning architectures - LSTM and ConvNet are used to process the data and predict the rainfall values of the Test data by training the values of Train Data and depicting the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) of the Predicted Data with respect to the Test Data. In both Architectures, the data generally passes through

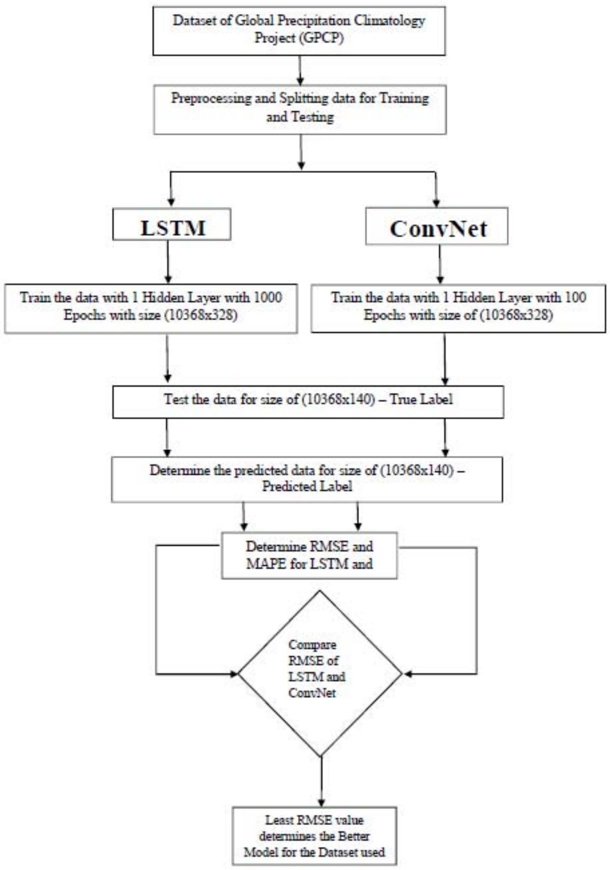

Fig. 4. Proposed Architecture

three layer which are Input layer, Hidden Layer and the Output Layer.

The proposed architecture contains the algorithm for the model of prediction of the global monthly average rainfall is shown in Fig. 4.

In CNN, Input layer receives input sequence length of 162 and contains 32 filters of length 3. It slides over the time series data and obtains the spatial feature representation. It follows the Max-pooling layer where the feature dimension of CNN layer becomes half. Max-pooling follows the dropout layer of 0.5. This is added to avoid over-fitting and acts like a regularization parameter. Then the feature data are flattened into a single dimensional vector and passed into fully connected layer where it contains 250 neurons, in which each neuron has connection to every other neurons. This is followed again by dropout layer of 0.25 and finally fully connected layer is used where it contains 70 neurons and activation function is linear. When the activation function is linear, the input will be same as output which is given as $f(x) = x$.

LSTM input layer receives an input length of 162 and it contains 32 memory blocks. This captures the rainfall data features across the given input time series data. This data is passed into fully connected layer to connect each neuron to every other neuron in the network. The loss function, $L_i = ||Predicted\ Label - True\ Label||^2$, for both the CNN and LSTM network and optimizer as Adam. The details of the

TABLE I
DETAILS OF CONVNET ARCHITECTURE

Layer(Type)	Output Shape	Parameter
Conv1D	(None, 162, 32)	128
Max-Pooling1	(None, 81, 32)	0
Dropout	(None, 81, 32)	0
Flatten	(None, 2592)	0
Dense	(None, 250)	648250
Dropout	(None, 250)	0
Activation	(None, 250)	0
Dense	(None, 70)	17570
Activation	(None, 70)	0

Total Parameters for ConvNet: 665,948

TABLE II
DETAILS OF LSTM ARCHITECTURE

Layer(Type)	Output Shape	Parameter
LSTM	(None, 32)	25216
Dense	(None, 70)	2310
Activation	(None, 70)	0

Total Parameters for LSTM: 27,526

LSTM and ConvNet Architecture used for predicting the model are listed in the Table I and Table II respectively.

The Root Mean Square Error (RMSE) is the measure of the average amount of the errors which is calculated by the taking the square root of the average of the squared difference between the Predicted Label and the True Label.

$$RMSE = \sqrt{\frac{1}{N}\sum(Predicted\ Label - True\ Label)^2} \tag{6}$$

The Mean Absolute Percentage Error (MAPE) is the measure of average amount of the errors in the predicted data which is calculated by finding the absolute differences in test data between Predicted Label and True Label.

$$MAPE = \frac{1}{N}\sum|Predicted\ Label - True\ Label| \tag{7}$$

V. RESULTS AND DISCUSSIONS

The Table III shows that both ConvNet and LSTM Architectures shows similar RMSE error of 2.44 and 2.55 where 100 epochs are performed for ConvNet and 1000 epochs are performed for LSTM in predicting the global monthly average rainfall for 70 months. Among these two models, ConvNet Architecture seems to be quite promising with only 100 training epochs. The Predicted label is available for 70 months. The figures Fig.5 and Fig.6 below shows the True Label and Predicted Label of the Average Monthly

TABLE III
RMSE AND MAPE VALUES FOR THE DEEP LEARNING METHODS

Deep Learning Architectures	RMSE	MAPE
LSTM	2.55	1.6897
ConvNet	2.44	1.7281

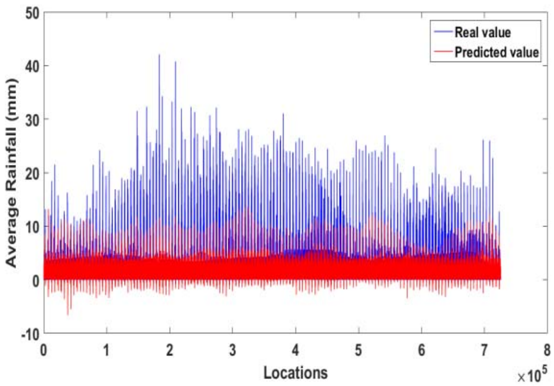

Fig. 5. True Label vs. Predicted Label of Monthly Average Rainfall Data Points using LSTM Layer 1

Rainfall for different locations around the globe.

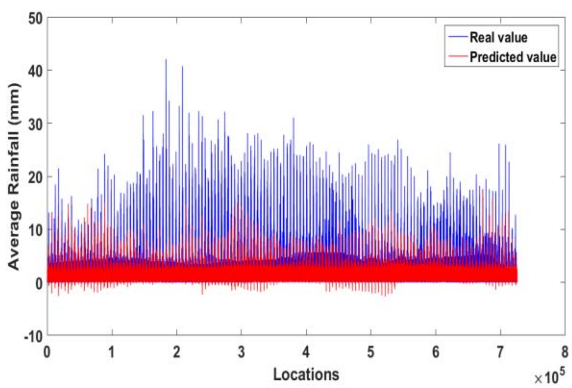

Fig. 6. True Label vs. Predicted Label of Monthly Average Rainfall Data Points using ConvNet Layer 1

VI. Conclusion and Future work

Deep learning techniques promises the gradual development in the field of Meteorology with respect to a time-series dataset. In this paper, LSTM and ConvNet Architectures are used to model and predict the Global monthly average rainfall for 10368 Geographic Locations around the globe for 468 Months. By increasing the number of hidden layers, RMSE and MAPE errors can still be reduced with fine accuracy. With fine accuracy, the data predicted for future months would be reliable for Meteorological purposes. This work can be further extended in such way that time-series dataset available for an individual country can be considered for processing with similar techniques for accurate prediction results.

References

[1] Wolfgang Grob, Sascha Lange, Joschka Bodecker, Manuel Blum , *Predicting Time Series with Space-Time Convolutional and Recurrent Neural Networks*, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges (Belgium), 26-28 April 2017, 71-76.

[2] Mohini.P.Darji, Vipul.K.Dabhi, Harshadkumar.B.Prajapati *Rainfall Forecasting Using Neural Network: A Survey*, 2015 International Conference on Advances in Computer Engineering and Applications (ICACEA), IMS Engineering College, Ghaziabad, India,706-713.

[3] Sanam Narejo and Eros Pasero, *Meteonowcasting using Deep Learning Architecture*, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 8, 2017.

[4] R.Venkata Ramana, B. Krishna, S.R.Kumar, N.G.Pandey, *Monthly Rainfall Prediction Using Wavelet Neural Network Analysis*, Water Resource Management (2013) 27:3697–3711.

[5] S. Narejo and E. Pasero , *Time Series Forecasting for Outdoor Temperature using Nonlinear Autoregressive Neural Network Models*, Journal of Theoretical and Applied Information Technology, vol. 94(2) pp. 451-463, 2016.

[6] Rafael Hrasko, Andr´e G. C. Pacheco, Renato A. Krohling , *Time Series Prediction using Restricted Boltzmann Machines and Backpropagation*, Information Technology and Quantitative Management (ITQM 2015), Procedia Computer Science 55 (2015) , 990 – 999.

[7] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal , *Long Short Term Memory Networks for Anomaly Detection in Time Series*, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, 89-94.

[8] Mislan, Haviluddin, Sigit Hardwinarto, Sumaryono, Marlon Aipassa, *Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan – Indonesia*, International Conference on Computer Science and Computational Intelligence (ICCSCI 2015) Procedia Computer Science 59 (2015), 142 – 151.

[9] Pallavi, Garima Singh , *Review on Rainfall Forecasting Using Different Techniques and Algorithms*, International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 3, March 2016.

[10] Ankita Sharma, Geeta Nijhawan, *Rainfall Prediction Using Neural Network*, International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 3, Year: May, 2015.

[11] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[12] Jianxin Wu, *Introduction to Convolutional Neural Networks*, National Key Lab for Novel Software Technology, Nanjing University, China (2017).

[13] Emilcy Hernandez, Victor Sanchez Anguix, Vicente Julian, Javier Palanca, and Nestor Duque, *Rainfall prediction: A Deep Learning approach*, Research Gate Article (2016). DOI: 10.1007/978-3-319-32034-2-13

[14] Sepp Hochreiter and Jurgen Schmidhuber, *Long short-term memory*, Neural Computation, 9(8):1735–1780, 1997.

[15] F. A. Gers, J. Schmidhuber, and F. Cummins, *Learning to forget: Continual prediction with LSTM*, Neural Computation, Vol. 12, No. 10, pp. 2451–2471, (2000).

[16] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, *Learning precise timing with LSTM recurrent networks*, Journal of Machine Learning Research, vol. 3, pp. 115–143, Mar. 2003.

[17] LeCun, Y., Bengio, Y., Hinton, G., *Deep Learning,* Nature, 521(7553), 436.(2015)

[18] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Kudlur, M., *TensorFlow: A System for Large-Scale Machine Learning*, OSDI (Vol. 16, pp. 265-283), November, 2016.

[19] Chollet, F., *Keras*, 2015.

[20] Werbos, Paul J., *Backpropagation through time: what it does and how to do it*, Proceedings of the IEEE 78.10 (1990): 1550-1560.

[21] https://www.esrl.noaa.gov/psd/data/gridded/data.cmap.html#detail

[22] R Vinayakumar, KP Soman, Prabaharan Poornachandran, *Long short-term memory based operation log anomaly detection*, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI),236-242.