IOSR Journal of Environmental Science, Toxicology and Food Technology (IOSR-JESTFT)
e-ISSN: 2319-2402,p- ISSN: 2319-2399.Volume 15, Issue 5 Ser. II (May 2021), PP 26-35
www.iosrjournals.org

Deep CoLSTM model for estimation of monthly meteorological parameters of Meghalaya

Vedanth Venkatakrishnan[1], Arpit Bansal[2]
[1,2]Department of Earth Sciences, Indian Institute of Technology Kanpur, India

Abstract: *In addition to the traditional time-series forecasting methods, researchers over the past few years have suggested various methods based on artificial intelligence for the prediction of meteorological conditions. We have proposed an architecture based on deep learning named CoLSTM (Convolution Stacked Long Short-Term Memory), which convolutes the raw data before feeding into the LSTM layer capable of learning complex features. Additionally, we have compared the performance of CoLSTM with other deep learning models like FNN (Feedforward Neural Network) and SANN (Seasonal Artificial Neural Network) along with the traditional time-series forecasting model ARIMA (Auto-Regressive Integrated Moving Average). We have used the above-mentioned models on monthly data of Minimum Temperature, Maximum Temperature and Vapour Pressure from the region of Meghalaya in India.*

Keywords*: deep learning, neural network, time-series, temperature, vapour pressure*

Date of Submission: 06-05-2021 Date of Acceptance: 20-05-2021

I. Introduction

Global Climate change in the past several decades has caused losses to unprecedented levels. It has caused multidimensional problems that need multidimensional solutions. The main problem lies in the prediction of its harmful consequences before time. India being a geologically diverse country has faced regional disparities for these consequences. Further the warming rate of 0.6 ^{o}C per year has been attributed to anthropogenic activities (IPCC, 2001a and 2001b). The need of predictions and forecasting of physical parameters have gained importance to avoid or nullify the far-reaching consequences of climate change from food security to tourism.

The North East Himalayan region is well known for its frail landscape that makes it vulnerable to climate change. People living in these areas are cut off from mainland India and are dependent on Jhum Agriculture technique. Abrupt climate change in these areas possess serious danger, even then very little concern has been shown for this area.

Crop yield is highly dependent on physical parameters like temperature [1], vapour pressure [2] and precipitation. These parameters are also interrelated to each other. Higher temperatures foster lower precipitation leading to droughts. Though, enough work has been done for precipitation prediction [3] but not sufficient work has been done for temperature and vapour pressure prediction. Slight increase in temperature can cause significant reduction in crop yield. Similarly, Vapour pressure deficit (VPD) also plays a very important role in estimating crop yield. Therefore, forecasting of these parameters becomes highly important, so that people are prepared for the needs of changing climate. Also, these predictions become useful for the supervision of other events like tourism, health sector, transport and providing agencies with the forecasted data that help them to make decisions for disaster management.

There are many techniques already being used for forecasting ranging from Naive techniques like Moving Average, Auto regression to more complex techniques like Artificial Intelligence (AI), Machine learning (ML) and even more complex techniques like Deep Learning. Machine learning is based on the principle of adaptive learning and making sense of the data being fed to the machine. It is well suited for the problems in which data shows some patterns. Machine learning by nature has a unique ability to map complex nonlinear functions to the given data that helps to make accurate predictions but if we go further beyond Deep learning has the ability to map even more complex patterns that sometimes are not possible with simple dense neural network. Deep learning uses bigger architectures that allows it to have more trainable parameters and yield better results. But in some cases, especially with sequential data, as we have a deep architecture, problems of gradient vanishing and gradient explosion may occur leading to unfavourable results. We have to keep checks and balances on the data range all well and use more complex techniques to ensure that this problem does not occur.

Temperature and vapour pressure data need highly complex mapping functions. That is possible with deeper architectures. In this paper we have proposed a model named CoLSTM (Convolutional Stacked Long

Short-Term Memory). This model uses convolutional as well as doubly stacked LSTM units making it highly adaptive to learn complex mapping functions along with being time efficient. We have also compared the forecasting result with three other well-known models used for forecasting – ARIMA (Auto-Regressive Integrated Moving Average), FNN (Feedforward Neural Network) and SANN (Seasonal Artificial Neural Network). We found that CoLSTM outperformed all the other three models.

II. Literature Review

Mislan et al. [4] proposed a Back Propagation Neural Network (BPNN) algorithm to predict rainfall with good accuracy in Tenggarong, East Kalimantan, Indonesia. They used monthly rainfall data from 1986 to 2003 and 2004 to 2008 as training and test data, respectively. Two different architectures of BPNN were used for comparison, with each having two hidden layers. The parameter of Mean Square Error (MSE) has been used to calculate prediction accuracy.

Paras et al. [5] have used an algorithm based on the Artificial Neural Network to forecast different weather parameters like maximum temperature, minimum temperature, and relative humidity. They have worked on weekly data from April 1996 to March 1999 collected at Pantnagar station in Uttarakhand state of India.

Sumi et al. [6] have compared the performance of different machine learning models, including Artificial Neural Network (ANN), Multivariate Adaptive Regression Splines (MARS), K-Nearest Neighbour (KNN) and Support Vector Regression (SVR) on average daily and monthly rainfall of Fukoka city in Japan. It further suggests constructing a hybrid multi-model method based on the individual predictive models. It has been shown that the hybrid method performs better than the single models for daily rainfall series while SVR performs better than the hybrid method for monthly rainfall series.

Deshpande [7] found that the results of Multilayer Perceptron (MLP) Neural Network and Jordon Elman Neural Network were the closest to the actual output in comparison to other predictive models. Data was collected from the Government Rainfall Monitoring Agency in Yavatmal in Maharashtra state of India. The performance of the models was calculated from parameters like Mean Square Error (MSE) and Normalized Mean Square Error (NMSE). 'Neurosolution 5.0', which is an object-oriented environment for designing, prototyping, simulating, and deploying Artificial Neural Network (ANN) solution, has been used to measure the performance parameters.

Barde and Patole [8] have performed a comparative study of different machine learning techniques for regression and classification of weather attributes. It has been shown that K-Nearest Neighbour (KNN) performs better for classification while Naïve Bayes performs better for regression.

Nanda et al. [9] adopted the complex statistical model ARIMA along with three different kinds of Artificial Neural Network (ANN), namely Multilayer Perceptron (MLP), Legendre Polynomial Equation (LPE) and Functional Link Artificial Neural Network (FLANN) to predict rainfall. Based on Absolute Average Percentage Error (AAPE), it has been found that FLANN gives a very close and better prediction result in comparison to the ARIMA model.

Dikshit et al. [10] have proposed a stacked Long Short-Term Memory (LSTM) architecture to forecast a commonly used drought measure, that is, Standard Precipitation Evaporation Index. The challenge of forecasting at long lead times has been a challenge due to climate change and complexities involved in drought assessment, but the authors believe deep learning techniques like stacked LSTM can solve this issue. The model was applied in the New South Wales region of Australia. It was trained using data from the period 1901-2000 and tested on data from the period 2001-2018 with the results being forecasted at lead times ranging from 1 month to 12 months. The model was based on two statistical measures, namely the Coefficient of Determination and Root Mean Squared Error.

III. Dataset Description and Workflow

The dataset is revived from MC (Meteorological Center) Shillong in the state of Meghalaya in India. The data was accessed through the official website of IMD (Indian Meteorological Department). The dataset contains monthly data for physical parameters like minimum temperature, maximum temperature and vapour pressure for 102 years from 1901 to 2002.

Before using this data into our models for training, we need to clean and process the data. For any missing observation, we need to substitute a value for the missing place. We have used the technique of linear interpolation for this substitution. Also, if there are any outliers, we need to take care of them as they can affect the training of the model. We have used K-means clustering algorithm [11] for removing the outliers.

After pre-processing, we need to scale the data for better training. We have used a Min-Max Scaler for this purpose, which scales the data between 0 and 1. It is represented in the equation form by *(1)*.

$$v_i = \frac{a_i - \min(a)}{\max(a) - \min(a)} \quad (1)$$

Here v_i denotes the corresponding scaled value of a_i, $\min(a)$ denotes the minimum value in the series, and $\max(a)$ denotes the maximum value in the series. We need scaling to ensure that the trainable parameters are able to completely capture the complex functions. If we do not scale the data, large values will be given in the training leading to large gradients and thus the parameters may not be trained suitably.

After scaling the data, the next step is to split the data in train and test sets. We have used a 90:10 split for training and testing. Lastly, we have quantified the model evaluation with Root Mean Squared Error (RMSE). We have used Keras for modelling, which is a top-level API for Tensorflow, which in turn is a GPU-based Python library for deep learning.

The flowchart in *Figure 1* summarizes the procedure followed from pre-processing the dataset to training different models on it:

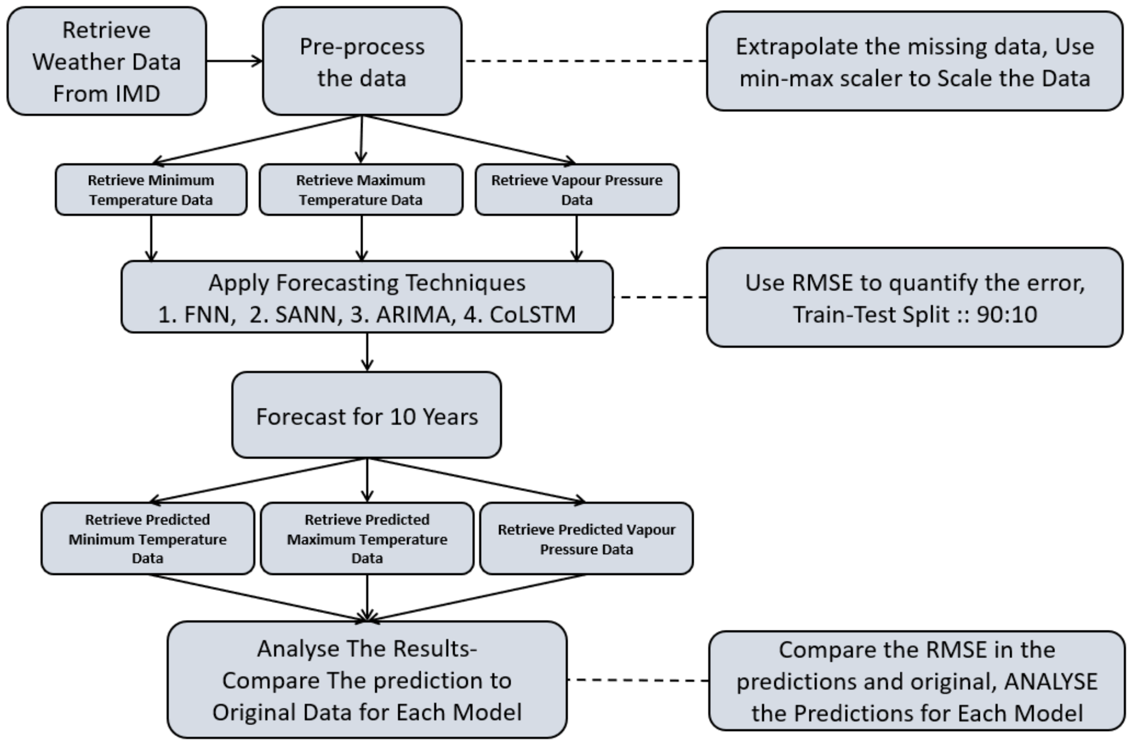
Retrieve Weather Data From IMD
Pre-process the data
Extrapolate the missing data, Use min-max scaler to Scale the Data
Retrieve Minimum Temperature Data
Retrieve Maximum Temperature Data
Retrieve Vapour Pressure Data
Apply Forecasting Techniques 1. FNN, 2. SANN, 3. ARIMA, 4. CoLSTM
Use RMSE to quantify the error, Train-Test Split :: 90:10
Forecast for 10 Years
Retrieve Predicted Minimum Temperature Data
Retrieve Predicted Maximum Temperature Data
Retrieve Predicted Vapour Pressure Data
Analyse The Results- Compare The prediction to Original Data for Each Model
Compare the RMSE in the predictions and original, ANALYSE the Predictions for Each Model

Figure 1: *Experiment Workflow*

IV. Proposed Architecture

Feedforward Neural Network (FNN)

Feedforward Neural Network (FNN) is one of the most widely used neural network models. It finds application in various fields due to its ability to model non-linear time series data [12],[13]. FNN is characterized by an architecture of an input layer, one or more hidden layers and an output layer.

The formula governing the relationship between the inputs y_{t-i} (i = 1, 2, ... , p) and the output y_t in an FNN with p input nodes, h hidden nodes and 1 output node is given in *(2)*

$$y_t = G(a_0 + \sum_{j=1}^{h} a_j F\,(b_{0\,j} + \sum_{i=1}^{p} b_{ij} y_{t-i})\,)$$

(2)

Where a_j, b_{ij} (i = 1, 2, ... , p; j = 1, 2, ... , h) are the connection weights, a_0, b_{0j} are the bias terms, F, G are activation functions of hidden and output layer, respectively.

We have proposed a FNN model with a three-layered architecture to perform one-step ahead forecasting. The number of nodes in each layer for forecasting the three different parameters has been decided on the basis of capturing the maximum information and minimized the root mean squared error (RMSE). In the implemented model, ReLU activation function is used, and the model is optimized using Adam Optimizer. *Figure 2* shows the FNN architecture implemented for modelling maximum temperature.

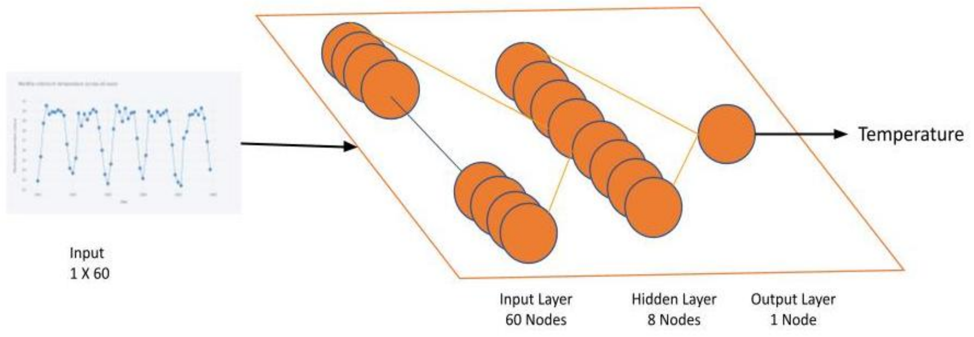

Figure 2: *FNN Architecture for Maximum Temperature*

Seasonal Artificial Neural Network (SANN)

Hamzaçebi [14] has proposed an alternative model, namely Seasonal Artificial Neural Network (SANN) which has proved to be quite efficient in dealing with data that is strongly seasonal in nature. The structure is similar to a regular three-layered FNN model with s input nodes, h hidden nodes and s output nodes, where s is the seasonality of the time series data. Thus, we are predicting the next ‘s’ data points by implementing a neural network that takes the previous ‘s’ data points as the input.

As the seasonality of rainfall data is 12, we have proposed an SANN model with 12 input nodes and 12 output nodes to forecast the next 12 data points. The number of hidden nodes vary according to the forecasted parameter. The loss metric to choose the number of nodes along with the activation function and model optimizer remain the same as the FNN model. *Figure 3* shows the SANN architecture implemented for modelling maximum temperature.

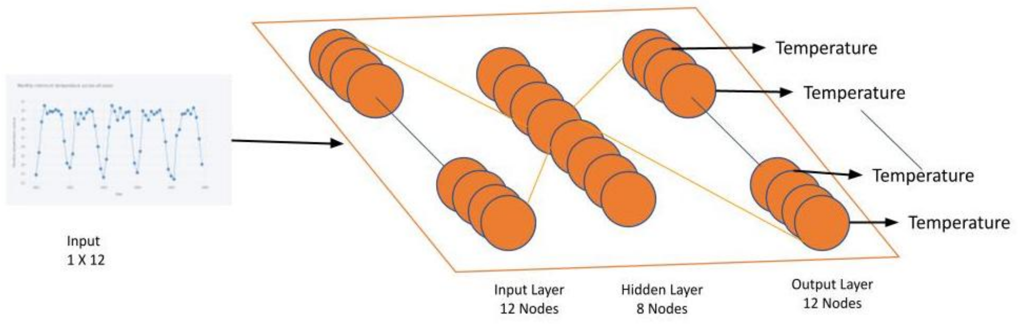

Figure 3: *SANN Architecture for Maximum Temperature*

Auto-Regressive Integrated Moving Average (ARIMA)

Auto-Regressive Integrated Moving Average (ARIMA) [15],[16] is a common model used for time series modelling and forecasting. The ARIMA model consists of three parts. The Auto-Regressive (AR) part signifies the weighted moving average over the past observation. The Integrated (I) part signifies the linear trend or the polynomial trend. The Moving Average (MA) part signifies the weighted moving average over the past errors. Thus, to describe a ARIMA model, we need 3 parameters, namely p (related to AR) = order of autocorrelation; d (related to I) = order of integration; q (related to MA) = order of moving average.

For modelling any time series with ARIMA, the first step is to make the time series stationary, that is, we need to remove all the trends and seasonality. To check whether the series is stationary, we have used the Dickey Fuller test. Here the null hypothesis is that the time series is non-stationary. The test results consist of a test statistic and some critical values for different confidence levels. If the ‘test statistic’ is less than the ‘critical value’, we can reject the null hypothesis and say that the series is stationary.

Second step is to find the autocorrelation and the partial autocorrelation between the values in any sequential data. Auto-correlation is defined as the correlation between the current observation and the observation after k periods. The value of ACF lies between -1 and 1.

Partial autocorrelation is the conditional correlation between the current observation and the observation after k periods. The value of PACF lies between -1 and 1. The values of ACF and PACF are found from the ACF and PACF plots shown below:

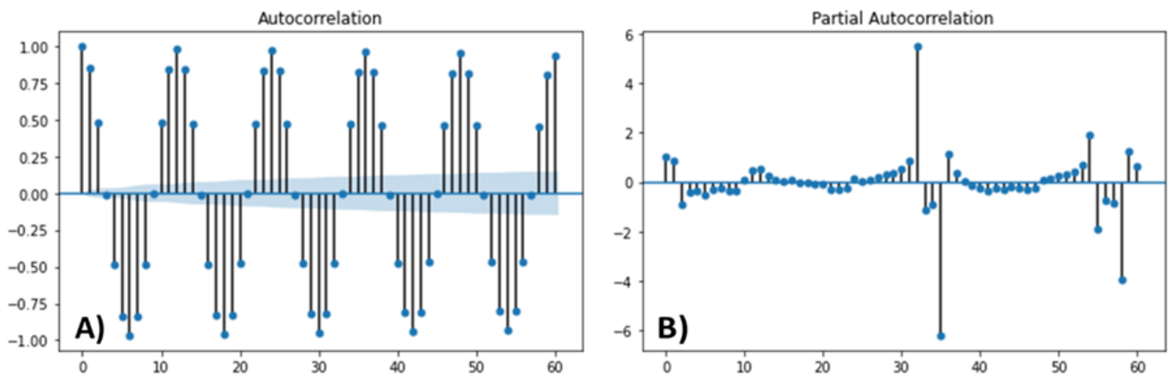

Figure 4: ***A)*** *ACF Plot and* ***B)*** *PACF Plot for Maximum Temperature*

Convolutional Stacked Long Short-Term Memory (CoLSTM)

Recurrent neural networks (RNN) [17] are well suited for modelling sequential data which are correlated to their previous values. Data series like temperature, stock market prices [18] and vapour pressures can be forecasted within a small error range using RNN as they have a correlation with their previous values. In addition to this correlation, temperature and pressure also show cyclical trends.

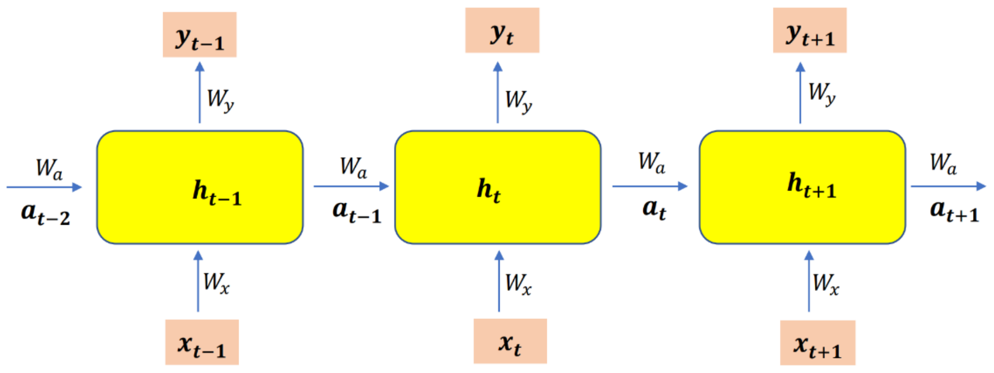

Figure 5: *Basic Architecture of RNN*

Figure 5 shows the basic architecture of RNN. Each node is connected to its previous and succeeding node. The output of each node is dependent on the input of the current node and the activation function of the previous node. The weights associated with each connector are trainable parameters and are trained as to minimize the error in predicted output. The relation between different parameter is given by (3),(4),(5).

$$h_t = W_a a_{t-1} + W_x x_t \quad (3)$$

$$a_t = g(h_t) \quad (4)$$

$$y_t = W_y a_t \quad (5)$$

Where y_t denotes the output vector, x_t denotes the input vector, h_t denotes the hidden layer vector, W_x, W_y, W_a denotes the weighing matrices, g denotes a real valued activation function. In our model we have used tanh as the activation function.

For the calculation of h_t, we multiply the W_x and W_a matrices recursively. Thus, the resulting values can be very high and can vanish which leads to an exploding and vanishing gradient problem. We use LSTM nodes that take care of this problem, which solves the long-term dependencies by storing data unlike other nodes [19]. RNNs having such long short-term memory nodes are called LSTM network. Even though LSTM is able to solve the problem of long-term dependencies, it fails to learn complex functions which ultimately affect the accuracy of the predictions.

To learn complex functions, two modifications have been applied to the LSTM network. Firstly, as Convolutional Neural Network (CNN) [20] are good for pattern recognition, we have added a CNN layer before the LSTM layer. The major advantage of this modification is that now the input to the LSTM layer is not a raw input since it is already processed by a convolutional layer thereby helping in deeper learning of parameters. Secondly, we have stacked LSTM layers. Stacking improves the complex feature learning because the input given to a LSTM layer is already an output from the previous LSTM layer [21].

Figure 6 shows that the final model for maximum temperature is divided into three sections. The first section is the convolution part that has sixty 1 X 1 convolutional nodes. This layer is mainly responsible for extraction of patterns from the raw input. In this layer, ReLU activation function is used. The second section is the stacked LSTM section. By stacking the LSTM nodes vertically, the network is able to improve its ability to learn complex features. This section has two stacked layers of sixty LSTM nodes each. In both the layers, tanh activation function is used. The first layer of the section returns data at each node because we need to stack another LSTM layer over it. The second layer of the section does not return data at each node, rather it returns only for the last time stamp. The third and the final section is a small fully connected traditional neural network. As the output from the last LSTM layer is a vector of dimension 60 X 1, we need to map the result on scalar quantity which in our case is minimum temperature, maximum temperature, and vapour pressure. This section has four fully connected dense layers. For each layer in this section, we have used the ReLU activation function. The entire model is compiled and optimized using Adam Optimization and is judged based on Root Mean Squared Error (RMSE).

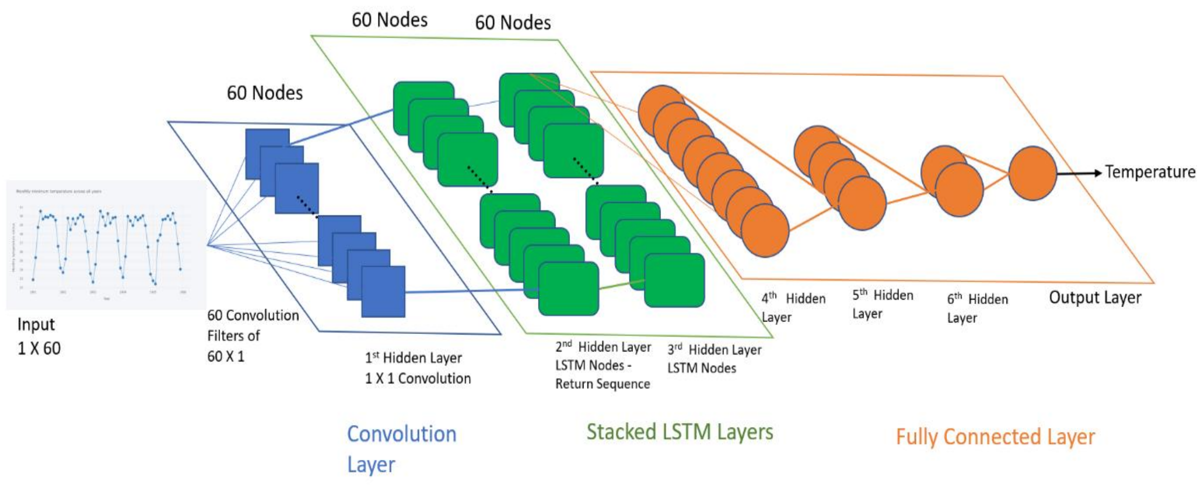

Figure 6: *CoLSTM Architecture for Maximum Temperature*

V. Results and Analysis

Each of the three climatic parameters, that is, minimum temperature, maximum temperature and vapour pressure have been predicted using CoLSTM, SANN, FNN and ARIMA. All the models were optimized by performing a grid search on the perspective parameters. The performance of these models on each parameter has been compared using the loss metric of RMSE.

Minimum Temperature

To minimize the loss metric RMSE, different permutations of the parameters were used. It was found that the three-layered FNN and SANN models performed best with an architecture of (12,4,1) and (12,7,12) respectively, with each model having a batch size of 20 and number of epochs equal to 300. The RMSE for FNN and SANN was found to be 0.86 ^{0}C and 0.63 ^{0}C, respectively.

Conventional ARIMA model works best with the parameters p = 2, d = 1, q = 1, with an AIC score of 1685.125. The RMSE for ARIMA is 0.72 ^{0}C.

For CoLSTM, we found the following architecture to give the best results: Convolutional section consists of a single layer with 54 nodes, with each node resulting because of convolution of input with a 54 X 1 filter; RNN section of the model consists of two layers with 54 nodes each, with only the first layer returning output at each timestamp; Dense section of the model works best with 3 layers with an architecture of (8,4,2,1) nodes. CoLSTM model gives the least RMSE of 0.52 ^{0}C.

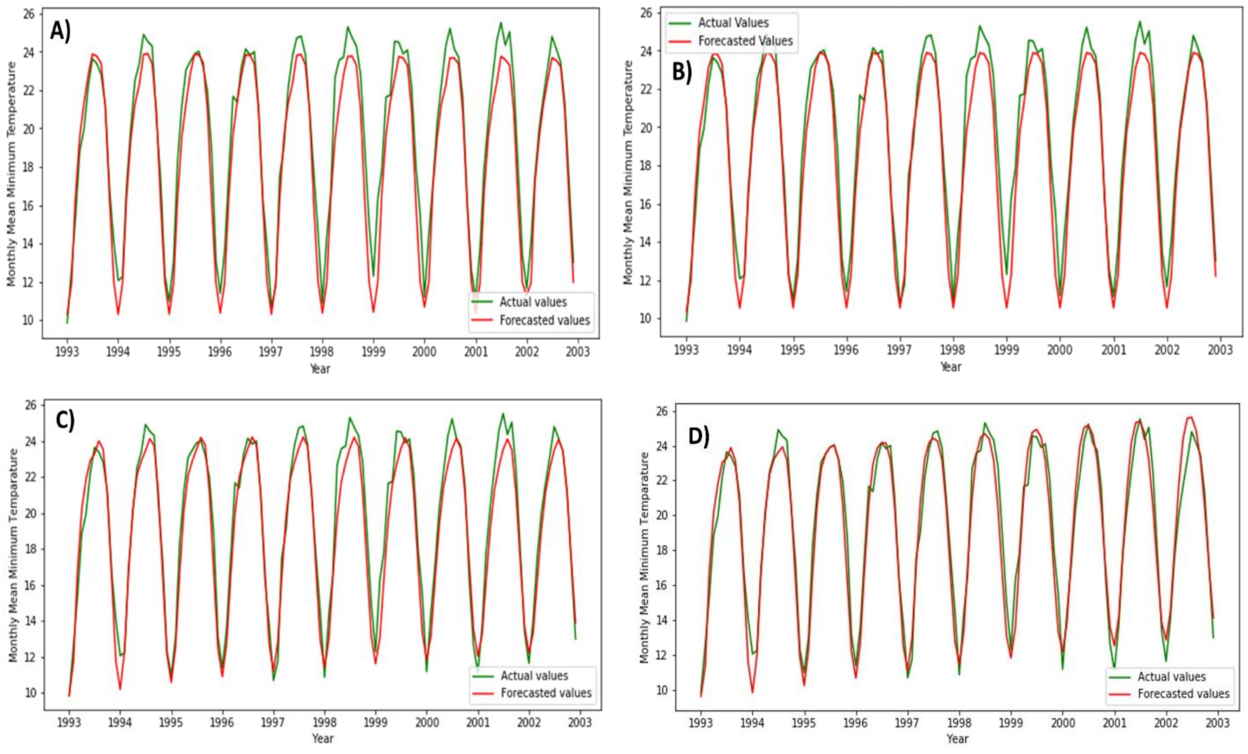

Figure 7: *Comparison of Actual vs Forecasted values of Minimum Temperature using* ***A)*** *FNN,* ***B)*** *ARIMA,* ***C)*** *SANN,* ***D)*** *CoLSTM*

Maximum Temperature

The most optimized architecture of FNN and SANN for this case too was calculated by performing a grid search on the different parameters and it was found to be (12,6,1) and (12,5,12) respectively, with a batch size of 20 and number of epochs equal to 500. The RMSE for FNN and SANN was calculated to be 3.07 ^{0}C and 2.75 ^{0}C, respectively.

Conventional ARIMA model works best with the parameters p = 2, d = 0, q = 1 giving an AIC score of 1784.538. The RMSE for ARIMA is 2.40 ^{0}C.

For CoLSTM, we found the following architecture to give the best results: Convolutional section consists of a single layer with 60 nodes, with each node resulting because of convolution of input with a 60 X 1 filter; RNN section of the model consists of two layers with 60 nodes each, with only the first layer returning output at each timestamp; Dense section of the model works best with 4 layers with an architecture of (8,4,2,1) nodes. CoLSTM model gives the least RMSE of 1.71 ^{0}C.

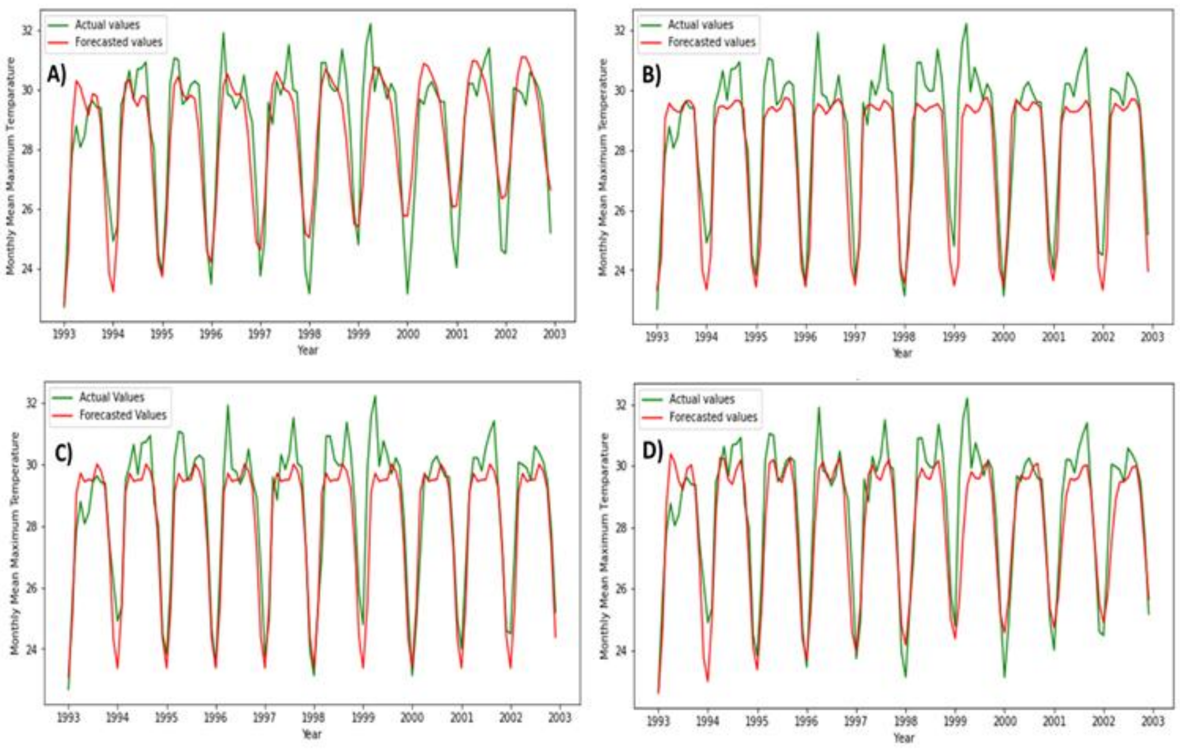

Figure 8: *Comparison of Actual vs Forecasted values of Maximum Temperature using* ***A)*** *FNN,* ***B)*** *ARIMA,* ***C)*** *SANN,* ***D)*** *CoLSTM*

Vapour Pressure

Due to its relatively less variability with respect to temperature, vapour pressure is easier to predict than the other two parameters and correspondingly has a lower RMSE. The models FNN and SANN best performed with the architecture (12,5,1) and (12,9,12) respectively, with a batch size of 20 and number of epochs equal to 500. The most optimized versions of these models gave an RMSE of 0.89 Torr and 0.57 Torr, respectively.

Conventional ARIMA model works best with the parameters p = 2, d = 1, q = 1 giving the AIC score of 1635.067. The RMSE for ARIMA is 0.77 Torr.

For CoLSTM, we found the following architecture to give the best results: Convolutional section consists of a single layer with 60 nodes, with each node resulting because of convolution of input with a 30 X 1 filter; RNN section of the model consists of two layers with 30 nodes each, with only the first layer returning output at each timestamp; Dense section of the model works best with 4 layers with an architecture of (8,4,2,1) nodes. CoLSTM model gives the least RMSE of 0.35 Torr.

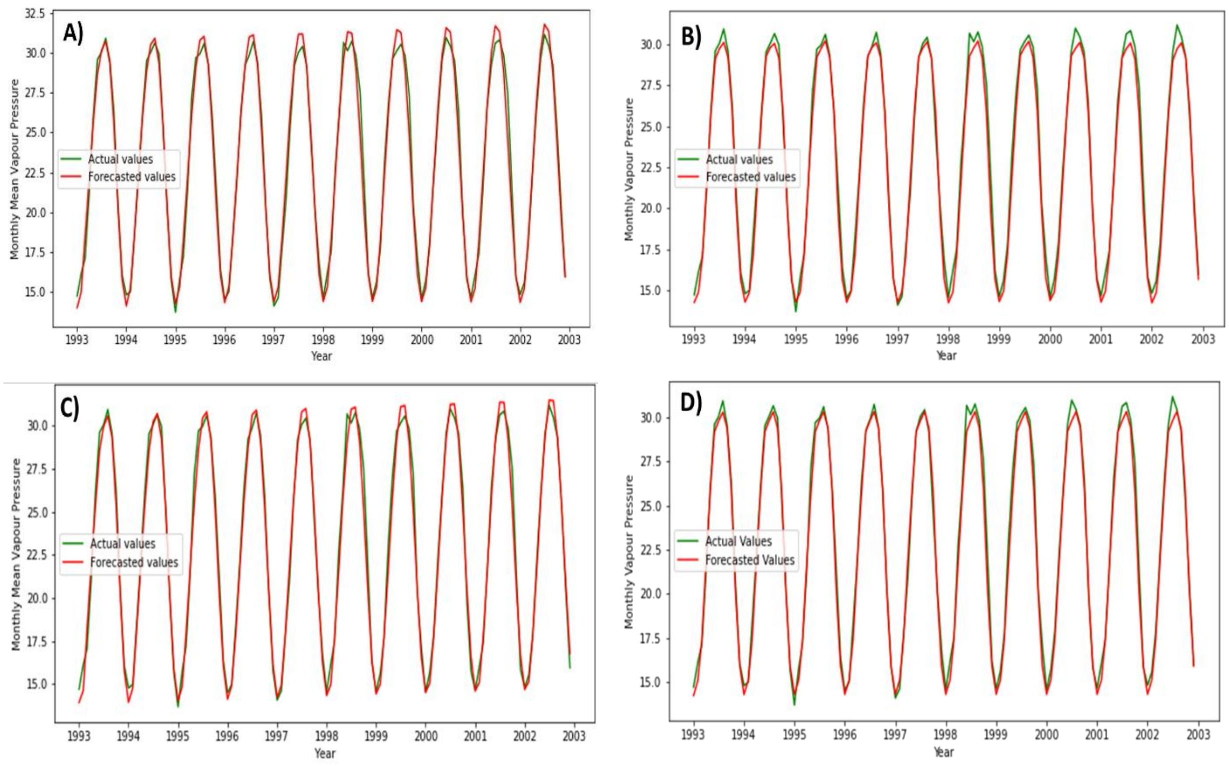

Figure 9: *Comparison of Actual vs Forecasted values of Vapour Pressure using* ***A)*** *FNN,* ***B)*** *ARIMA,* ***C)*** *SANN,* ***D)*** *CoLSTM*

Table 1 summarizes the results in terms of RMSE found by applying the four models on each of the parameters.

	RMSE		
	Minimum Temperature (^{0}C)	**Maximum Temperature (^{0}C)**	**Vapour Pressure (Torr)**
FNN	0.86	3.07	0.89
ARIMA	0.72	2.40	0.77
SANN	0.63	2.75	0.57
CoLSTM	0.52	1.71	0.35

VI. Conclusion and Future Work

In this paper, we studied a deep learning-based approach for forecasting three physical parameters affecting the meteorological conditions of a region. These parameters are minimum temperature, maximum temperature and vapour pressure and the area of study was Meghalaya in India. We proposed a deep learning architecture named CoLSTM. The results suggest that our proposed model outperformed the three other models used for comparison, which include ARIMA, ANN and SANN in terms of RMSE. The model works with a RMSE of 0.52 ^{0}C, 1.71 ^{0}C and 0.35 Torr for minimum temperature, maximum temperature, and vapour pressure, respectively. Also, the model is computationally efficient. The forecasting of these three physical parameters is an important task as it can be used in many fields ranging from agriculture to tourism.

As a part of future work, we aim to increase the performance of our model by incorporating physical parameters of nearby areas while training the model. We also aim to derive direct correlations between the agricultural yield and physical parameters in Meghalaya region.

References

[1]. Matiu M, Ankerst DP, Menzel A (2017) Interactions between temperature and drought in global and regional crop yield variability during 1961-2014. PLOS ONE 12(5): e0178339.

[2]. Wang Enli, Smith Chris J., Bond Warren J., Verburg Kirsten (2004) Estimations of vapour pressure deficit and crop water demand in APSIM and their implications for prediction of crop yield, water use, and deep drainage. Australian Journal of Agricultural Research 55, 1227-1240.

[3]. Choudhury, Burhan & Das, Anup & Ngachan, S.V. & Slong, A. & Bordoloi, L.J. & Chowdhury, Pulakabha. (2012). Trend analysis of long-term weather variables in mid-altitude Meghalaya. J. Agric. Phys.. 12. 12-22.

[4]. H. Mislan, S. Hardwinarto, and M. A. Sumaryono, ''Rainfall monthly prediction based on artificial neural network: A case study in Tenggarong Station, East Kalimantan-Indonesia,'' Procedia Comput. Sci., vol. 59, pp. 142–151, Jan. 2015.

[5]. S. M. Paras, A. Kumar, and M. Chandra, ‘‘A feature based neural network model for weather forecasting,’’ Int. J. Comput. Intell., vol. 4, no. 3, pp. 209–216, 2009.

[6]. Sirajum Monira Sumi, M. Faisal Zaman, and Hideo Hirose. “A rainfall forecasting method using machine learning models and its application to the Fukuoka city case”.

[7]. R. R. Deshpande, ‘‘On the rainfall time series prediction using multilayer perceptron artificial neural network,’’ Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 1, pp. 2250–2459, 2012.

[8]. Nishchala C Barde and Mrunalinee Patole. “Classification and Forecasting of Weather using ANN, KNN and Naive Bayes Algorithms”.

[9]. S. K. Nanda, D. P. Tripathy, S. K. Nayak, and S. Mohapatra, “Prediction of rainfall in India using Artificial Neural Network (ANN) models,” Int. J. of Intell. Syst. and Applicat., vol. 5, no. 12, pp. 1-22, 2013.

[10]. Abhirup Dikshit, Biswajeet Pradhan, Abdullah M. Alamri, “Long lead time drought forecasting using lagged climate variables and a stacked long short-term memory model”, Science of The Total Environment, Volume 755, Part 2, 2021, 142638, ISSN 0048-9697

[11]. Hartigan, J.A.; Wong, M.A. Algorithm as 136: A k-means clustering algorithm. J. R. Stat. Soc. Ser. Appl. Stat. 1979, 28, 100.

[12]. Zhang G, Patuwo B E & Hu M Y, Forecasting with artificial neural networks: The state of the art, J Forecasting 14 (1998) 35–62.

[13]. Kamruzzaman J, Begg R & Sarker R, Artificial Neural Networks in Finance and Manufacturing (Idea Group Publishing) 2006.

[14]. Hamzaçebi C, Improving artificial neural networks’ performance in seasonal time series forecasting, J Inform Sci 178 (2006) 4 550–4559.

[15]. G.Peter Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing, Volume 50, 2003, Pages 159-175, ISSN 0925-2312, https://doi.org/10.1016/S0925-2312(01)00702-0

[16]. S.L. Ho, M. Xie, The use of ARIMA models for reliability forecasting and analysis, Computers & Industrial Engineering, Volume 35, Issues 1–2, 1998, Pages 213-216, ISSN 0360-8352, https://doi.org/10.1016/S0360-8352(98)00066-7.

[17]. Lin T, Horne BG, Tino P, Giles CL. Learning long-term dependencies in NARX recurrent neural networks. IEEE Trans Neural Netw. 1996;7(6):1329-38. doi: 10.1109/72.548162. PMID: 18263528.

[18]. Shin, Dong-Ha, Kwang-Ho Choi, and Chang-Bok Kim. “Deep Learning Model for Prediction Rate Improvement of Stock Price Using RNN and LSTM.” *The Journal of Korean Institute of Information Technology* 15, no. 10 (October 31, 2017): 9–16. doi:10.14801/jkiit.2017.15.10.9.

[19]. Olah, C., 2015. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[20]. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[21]. Abhirup Dikshit, Biswajeet Pradhan, Abdullah M. Alamri, “Long lead time drought forecasting using lagged climate variables and a stacked long short-term memory model”,Science of The Total Environment,Volume 755, Part 2,2021,142638,ISSN 0048-9697 , https://doi.org/10.1016/j.scitotenv.2020.142638.

Vedanth Venkatakrishnan, et. al. “Deep CoLSTM model for estimation of monthly meteorological parameters of Meghalaya.” *IOSR Journal of Environmental Science, Toxicology and Food Technology (IOSR-JESTFT)*, 14(5), (2021): pp 26-35.