INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

An Evaluation of Bidirectional Long Short-Term Memory Model for Estimating Monthly Rainfall in India

Chawngthu Zoremsanga[1]*, **Jamal Hussain**[2]

1 PhD Scholar, Department of Mathematics and Computer Science, Mizoram University, Aizawl, Mizoram, India
2 Professor, Department of Mathematics and Computer Science, Mizoram University, Aizawl, Mizoram, India

OPEN ACCESS

Received: 03-10-2023
Accepted: 09-04-2024
Published: 25-04-2024

Citation: Zoremsanga C, Hussain J (2024) An Evaluation of Bidirectional Long Short-Term Memory Model for Estimating Monthly Rainfall in India. Indian Journal of Science and Technology 17(18): 1828-1837. https://doi.org/10.17485/IJST/v17i18.2505

* **Corresponding author**.

zoremsanga@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Predicting the amount of rainfall is difficult due to its complexity and non-linearity. The objective of this study is to predict the average rainfall one month ahead using the all-India monthly average rainfall dataset from 1871 to 2016. **Methods:** This study proposed a Bidirectional Long Short-Term Memory (LSTM) model to predict the average monthly rainfall in India. The parameters of the models are determined using the grid search method. This study utilized the average monthly rainfall as an input, and the dataset consists of 1752 months of rainfall data prepared from thirty (30) meteorological sub-divisions in India. The model was compiled using the Mean Square Error (MSE) loss function and Adam optimizer. The models' performances were evaluated using statistical metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). **Findings:** This study discovered that the proposed Bidirectional LSTM model achieved an RMSE of 240.79 and outperformed an existing Recurrent Neural Network (RNN), Vanilla LSTM and Stacked LSTM by 8%, 4% and 2% respectively. The study also finds that increasing the input time step and increasing the number of cells in the hidden layer enhanced the prediction performance of the proposed model, and the Bidirectional LSTM converges at a lower epoch compared to RNN and LSTM models. **Novelty:** This study applied the Bidirectional LSTM for the first time in predicting all-India monthly average rainfall and provides a new benchmark for this dataset.

Keywords: Deep Learning; LSTM; Rainfall prediction; Stacked LSTM; Bidirectional LSTM

1 Introduction

Rainfall is the primary source of drinking water and affects every organism on this planet. It also affects the transportation, farming, and the administration of renewable energy. Rainfall prediction includes predicting the amount of rain expected to fall in a given area within a specified period[(1)]. It is an essential tool in science, such as agriculture, meteorology, hydrology, and climatology. Rainfall prediction helps people and businesses prepare for the effects of heavy rain, such as floods, crop damage, and

landslides. However, it is challenging to accurately predict rainfall due to the significant number of elements that influence it [2].

In general, the methods for forecasting rainfall involve physical, statistical and machine-learning methods. Physical methods use numerical weather prediction models that use various physical equations and computer simulations to predict rainfall. Using the existing weather parameters and other features, these models can predict the amount of rainfall and its timing. However, these methods are not considered practical for forecasting rainfall due to their extreme resource and data demands. Statistical methods rely on the analysis of historical data to predict future rainfall patterns. These methods use mathematical models and statistical methods such as Linear Regression and Autoregressive Integrated Moving Average (ARIMA) to detect trends and patterns in existing data and then infer these into future predictions. This method is relatively simple; however, it is restricted by the quality of the available data as they work well with linear and stationary data only.

On the other hand, machine learning-based methods can go beyond simple statistical analysis by using complex methods to learn from data and make predictions. They can capture patterns that statistical methods may not recognize and discover patterns in linear and non-linear data. Thus, they have become more and more popular among researchers. These methods have also effectively predicted trends in time-series data, image processing, and natural language processing [3]. For example, [4–6] implemented Artificial Neural Network (ANN) to predict monthly rainfall in Queensland, Australia and compared their result with statistical models. Their results showed that ANN performs better than the statistical models. Reference [7] also shows the supremacy of neural networks over the ARIMA model by predicting the summer monsoon rainfall in India. Reference [8] forecast the monsoon rainfall for Kerala in India by comparing the K-Nearest Neighbour (KNN), ANN and Extreme Learning Machine (ELM). Their experiment showed that the ELM method is superior to the KNN and ANN models. Reference [9] recommends the ANN model compared to the Multiple Regression (MR) analysis methods for predicting rainfall in Victoria, Australia and demonstrates that ANN has a better learning capability in the studied regions.

However, the traditional machine learning models can consider only the present input for processing and thus have a limitation in learning hidden patterns within the input data. A deep learning model such as RNN is beneficial compared to traditional machine learning models as they can learn dependencies in the data. But they have a short memory and cannot learn long-term dependencies due to the vanishing gradient problem. LSTM, introduced by [10], is becoming more popular as it has overcome the limitations of the RNN and can capture long-term dependencies in the input data. Though RNN and LSTM have the advantage of learning the dependencies within the data, the information is transferred only in the forward direction. On the other hand, the Bidirectional LSTM model introduced by [11] is implemented using a bidirectional network to learn the past and future hidden states. This bidirectional method for handling the input significantly improved the performance and facilitated the model's learning of long-term dependencies within the input data.

India, whose economy greatly depends on farming, relies on rainfall for irrigation [12]. Thus, researchers have understood and studied the importance of modelling and predicting rainfall for many years. Reference [13] studied the stacked autoencoder to decide which weather observations are the best indicators for predicting the summer monsoon in India. They combined the Regression Tree and Decision Tree along with Bagging to forecast the long-term monsoon of India and proved that their models were better than the IMD model. Reference [14] compared Intensified- LSTM and Holt-Winters, RNN, Extreme Learning Machine (ELM), LSTM and ARIMA models for forecasting rainfall in India using multiple weather parameters in the Hyderabad area.

[15] proposed an LSTM deep learning technique and a Sequence-to-Sequence (Seq2Seq) approach to distinguish between the rainy and dry seasons in the central part of India. The authors examined June to September daily rainfall from 1948 to 2014 and compared their proposed models against SVM and K-Nearest Neighbor (KNN). Their results show that the LSTM and Sequence-to-Sequence systems outperformed the SVM and KNN methods.

[16] employed ConvLSTM in combination with Salp-Stochastic Gradient Descent (S-SGD) to forecast precipitation in India, using rainfall data from the all-India region and Tamil Nadu state between 1901 and 2015. To evaluate S-SGD-based ConvLSTM, they compared it to ConvLSTM, cluster-wise linear regression (CLR) technique, multilayer perception (MLP) classification algorithm and dynamic self-organizing multilayer network inspired by the immune algorithm (DSMIA).

Numerous investigations have been conducted to forecast rainfall in India. However, less research has been conducted to predict monthly rainfall in India using the all-India monthly average rainfall dataset. [17] Proposed a benchmark model using single-cell Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models to forecast the average rainfall in all of India. [18] Utilized the same dataset by comparing the Vanilla LSTM and Stacked LSTM, and they demonstrated that the stacked LSTM model outperformed the models proposed by [17]. Thus, the objective of this study is to analyze the prediction performance of the Bidirectional LSTM model to predict the monthly average rainfall using the all-India monthly average rainfall dataset from 1871 to 2016. This dataset consists of 1752 average monthly rainfall data, which was prepared using thirty (30) sub-divisional rainfall in India. The study also attempts to find the parameters of the model, such as the number of input time steps, the number of epochs and the number of cells in the hidden layer using the grid search method. The performance

of the proposed model is analyzed using the MAE and RMSE metrics, and they are compared with the results of the existing study[13].

2 Methodology

2.1 Study area and data source

The all-India monthly rainfall dataset prepared by[19] was utilized in this study. It consists of area-weighted all-India average monthly, seasonal and annual rainfall from 1871 through 2016. There are Thirty-six (36) meteorological sub-divisions in India, as shown in Figure 1, and thirty (30) sub-divisional rainfall was used for preparing this dataset by considering each rain-gauge station within the sub-divisions. In this dataset, the four subdivisions of the Himalayas are not taken into account due to the lack of rain gauges and limited coverage of rain gauges in hilly regions. Additionally, the islands in the Bay of Bengal and the Arabian Sea are not considered in the dataset construction process to maintain a continuous flow[19].

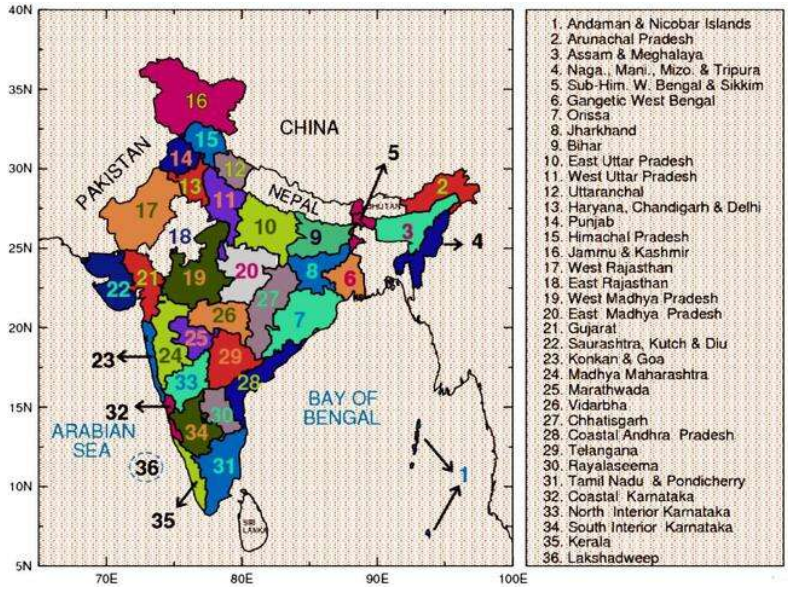

Fig 1. Meteorological Sub-Divisions of India[19]

Figure 2 illustrates the average monthly precipitation in India from 1871 to 2016 in tenths of millimetres. The data compilation consists of 1752 months of precipitation information. The mean monthly precipitation for all of India is 904.94, with a standard deviation of 951.71. The least amount of rainfall is 3, with the highest amount at 3460. For this research, three LSTM models—Vanilla LSTM, Stacked LSTM, and Bidirectional LSTM—were trained using the all-India average monthly rainfall dataset.

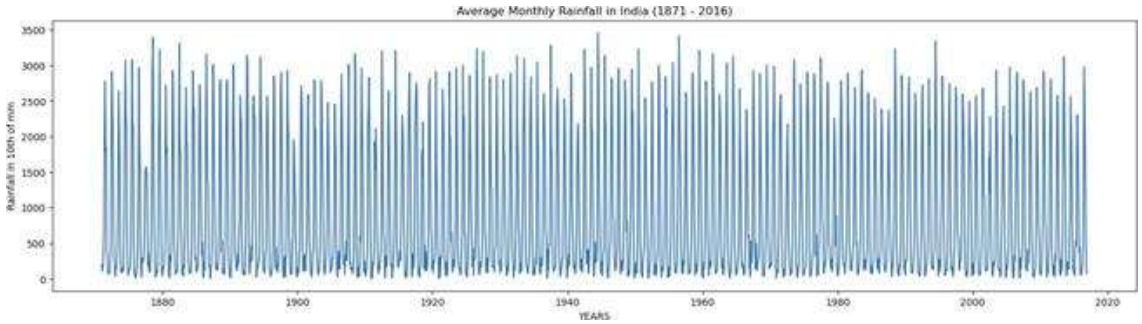

Fig 2. Average Monthly Rainfall in India (1871 - 2016)

2.2 Data preprocessing

2.2.1 Training and testing data

For this study, we split the rainfall data into training and testing sets. 70% of the data was utilized to train the models, while the remaining 30% was employed to assess the models' effectiveness in predicting the monthly rainfall.

2.2.2 Normalization

The training dataset was normalized using the Min-Max method within the value of 0 to 1. The same coefficients are used to normalize the testing dataset. This prevents information leakage to the testing dataset. The formula for Min-Max normalization is given in Equation (1). To obtain the prediction results in the original scale, the prediction results are inversed-transformed, and the error between the actual values and the inversed-transformed prediction values is calculated to get the final result.

$$n' = d_{min} + (d_{max} - d_{min}) + \left(\frac{x - x_{min}}{x_{max} - x_{min}} \right) \quad (1)$$

The normalized value n' can be calculated using Equation (1) given above. In this equation, d_{min} and d_{max} represents the minimum and maximum values of the data (i.e., 0 and 1, respectively). The symbol x denotes the data to be normalized, whereas x_{min} and x_{max} are the minimum and maximum values of the data [20].

2.3 LSTM

LSTM was introduced by [10] and it is a type of RNN that is used in deep learning and natural language processing (NLP). It helps in creating powerful models that can understand complex sequences of data such as text, audio, and video. LSTM is different from other types of neural networks because it can remember long-term dependencies. This is because it has special units called "memory cells" that store and access information over time. LSTM has been applied by many researchers for time series forecasting and other tasks that require an understanding of temporal patterns. By understanding how the data changes over time, LSTM networks can make predictions about future events. LSTM has been used for classification tasks such as sentiment analysis [21] and NLP [22]. They can also be used for sequence generation tasks such as text generation [23] and time series forecasting [24].

2.4 Stacked LSTM

A stacked LSTM is an extension of the conventional LSTM structure which has several layers of LSTM memory cells. This deep learning technique is made more complex through the addition of these hidden layers and is believed to be the reason for its success in tackling various challenging prediction problems. The use of the stacked LSTM is now seen as a reliable technique for predictive sequence tasks. In the LSTM architecture, each LSTM layer produces a sequence of outputs rather than a single output for the layer beneath it. As a result, one output is generated for each input time step and, in turn, one output time step for all input time steps [25].

2.5 Bidirectional LSTM

To overcome the limitation of traditional RNN, [26] introduced the Bidirectional RNN (BRNN). This architecture was trained in both directions at once, with separate layers for each direction. [11] merged the BRNN and the LSTM cell, resulting in the Bidirectional LSTM. This type of sequence processing model features two LSTM that process the input in opposite directions, providing the network with more information and enhancing the context available to the algorithm. The bi-directionality of the LSTM helps to improve the model's performance by considering the context from both past and future time steps. This permits the model to learn the long-term dependencies in the input data and can lead to improved accuracy for modelling sentiment analysis, speech recognition, and named entity identification.

2.6 Model development

To solve the vanishing gradient problem and to capture the long-term dependencies in the input data, LSTM has been introduced by reference [10]. Even though RNN and LSTM have the advantage of capturing the relationships among the input data, the knowledge is transmitted only in the forward direction. Bidirectional LSTM model [11] on the other hand, is implemented using a bidirectional network to capture the past and future hidden states. This bidirectional approach for

processing the input improved the model's performance significantly and enabled the model to learn long-term dependencies within the input data.

This study predicted the average monthly rainfall in India one month ahead using the Bidirectional LSTM and compared the performance of the existing studies. The parameters of the models, such as the number of input time steps, the number of epochs and the number of cells in the hidden layer, were found using the grid search method. To avoid the problem of overfitting, the number of epochs is tested to the maximum until the data fits too well into the model and produces the best performances [27]. Performance metrics such as RMSE and MAE are the models' performance indicators. The graphs of the comparison between actual data and model predictions were plotted to conduct visual assessments and demonstrate the models' accuracy. The flowchart of the proposed approach for predicting the monthly rainfall in India is given in Figure 3, and the architecture of the proposed Bidirectional LSTM is presented in Figure 4.

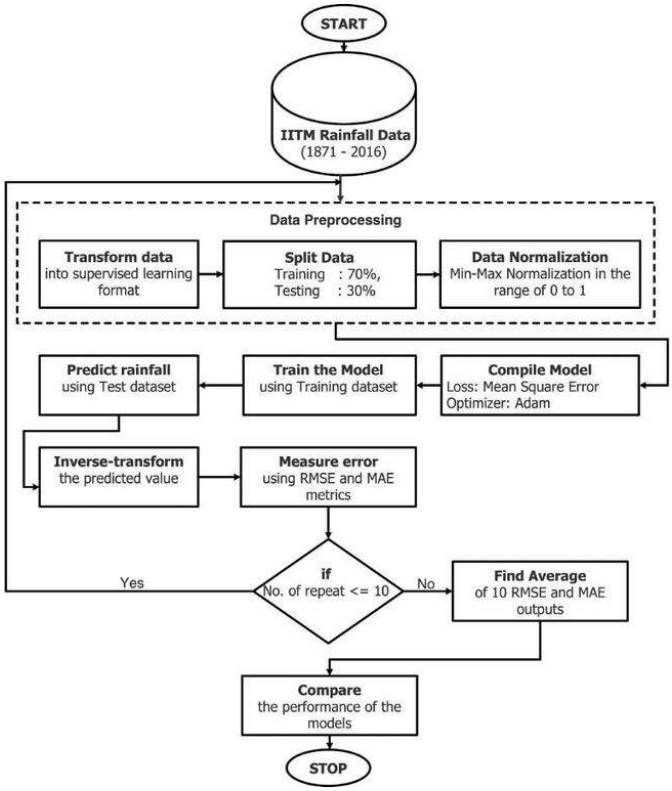

Fig 3. Flowchart of the proposed approach for monthly rainfall prediction

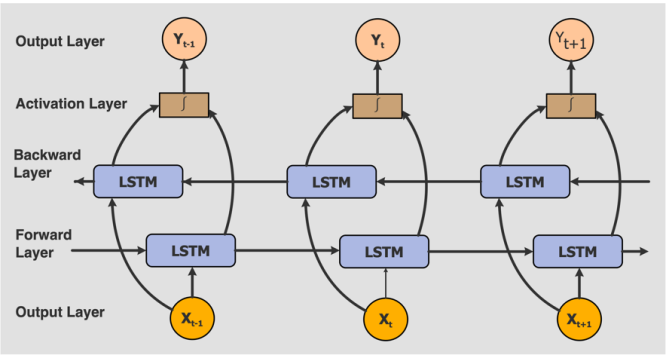

Fig 4. Architecture of the proposed Bidirectional LSTM

In this study, Mean Squared Error (MSE) was used as the loss function, and Adam optimizer was implemented to minimise the loss function. Random values are chosen as the initial weights of the models in each execution. Each LSTM model was trained and tested ten times using the same training and testing dataset to solve the stochastic nature of the algorithm, and the mean of the ten observed results was taken as the final performance of the model.

The benchmark RNN and LSTM models proposed by[17] consist of a single cell in the hidden layer, and they have implemented the models using 20 input time steps and 500 epochs. The Vanilla LSTM[18] was implemented using one hidden layer of 50 cells with 12 input time steps and trained for 400 epochs. In comparison, the Stacked LSTM[18] has two hidden LSTM layers with 12 cells each, followed by one hidden ANN layer with 12 neurons. The Stacked LSTM also consists of 28 input time steps to predict the next month's rainfall, and the model is trained for 500 epochs.

The architecture and parameters of the proposed Bidirectional LSTM models were determined by utilizing a grid search method. The final architecture consists of one hidden Bidirectional LSTM layer with 80 cells and an output ANN layer. The input time step is 24 months of rainfall to predict the next month's rainfall, and the model converges at 150 epochs. Table 1 summarizes the proposed model architecture and the compared existing models.

Table 1. Summary of Models architecture under study

LSTM Model	Input Timesteps	Model Architecture	Epochs
RNN[17]	20	RNN (1) – Dense (1)	500
LSTM[17]	12	LSTM (1) – Dense (1)	500
Vanilla LSTM[18]	12	LSTM (50) – Dense (1)	400
Stacked LSTM[18]	28	LSTM (12) – LSTM (12) – Dense (12) – Dense (1)	500
Bidirectional LSTM	24	Bi-LSTM (80) – Dense (1)	150

2.7 Performance Metrics

To find the performance of the LSTM models, two statistical measures, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), were employed[28]. MAE measures the mean of the differences between the actual and predicted values. RMSE gives the standard deviation of the errors among the predictions. A lower MAE and RMSE scores suggest a better performance. The final value of the MAE and RMSE of a model were obtained by computing the average of ten MAE and ten RMSE outputs. The MAE and RMSE can be represented using the following equations- Equations (2) and (3):

$$MAE = \left(\frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y}_i|\right) \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y})^2} \tag{3}$$

Where y_i indicates the ith actual rainfall value, $\widehat{y}_i$ the ith predicted rainfall amount of the model, and n represents the total observations.

2.8 System Setup

Experiments are conducted using Python 3.10, Jupyter Notebook 6.5.4, Keras and TensorFlow 2.10.0 on an Apple Mac Studio with M1 Max with 10 core CPU, 32 core GPU and 32 GB of RAM.

3 Results and Discussion

In this study, the prediction performance of the Bidirectional LSTM was studied by searching for the optimum hyperparameters such as the number of previous time steps, the number of epochs, and the number of cells. The study used all-India average monthly rainfall data from 1871 to 2016 prepared by[19] measured in 10th of mm. The minimum rainfall observed was 3, and the maximum value observed was 3460, with a mean value of 904.94 and a standard deviation of 951.71. The dataset was divided into training and testing at a ratio of 70:30, and due to better learning outcomes generated by the normalized data, the dataset was normalized using Min-Max normalization in the range of 0 to 1. However, the predicted values were inverse transformed

to their original values to get the final predicted value in the original scale. This study used the grid search method to find the optimal hyperparameters of the proposed model.

The optimal input time step is searched in the range of 1 to 30 and the prediction results for the 8, 12 and 24 time steps for the proposed Bidirectional LSTM model are shown in Figure 4. From this figure, it is observed that increasing the input time steps can lead to better prediction. However, as the results do not improve beyond 24 time steps, the input time step for the proposed model is set to 24 time steps which is 24 months of previous rainfalls.

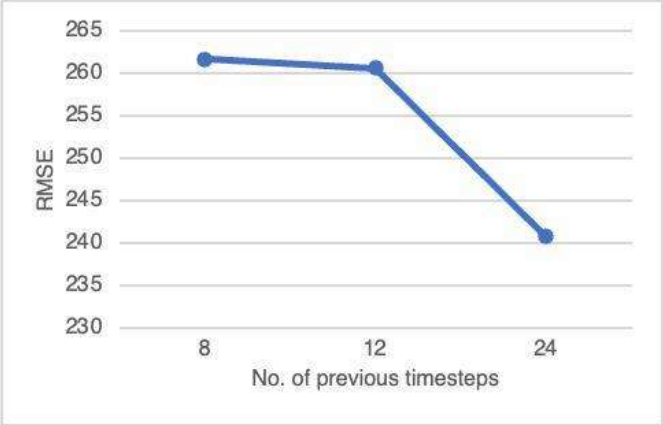

Fig 5. Input Time steps vs RMSE

The optimal value of the number of epochs to train the proposed model is searched in the range of 50 to 500, and the performance of the Bidirectional LSTM model is presented in Figure 5. The performance of the model improved by incrementing the number of epochs up to 150. However, the RMSE has started increasing beyond 150, which could be due to the overfitting of the data. Thus, an epoch of 150 was used to train the final model. The number of epochs for the RNN and LSTM models of[17] is 500, whereas the Vanilla LSTM and Stacked LSTM models of[18] converge at 400 and 500 epochs, respectively. In contrast, the proposed model requires only 150 epochs to find the minimum RMSE for the model. This shows that the proposed model requires much less time compared to the existing models.

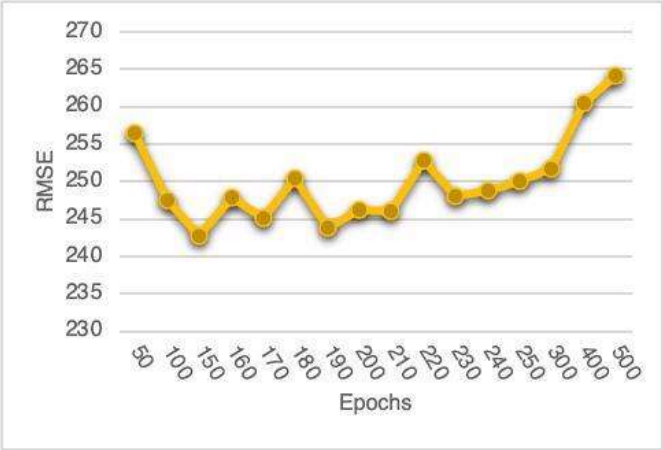

Fig 6. No. of epoch vs RMSE

The number of cells in the hidden layer of the Bidirectional LSTM model was incremented from 10 to 100 and the performance of the models is given in Figure 6. The RMSE of the model decreased by increasing the number of cells up to 80. However, it can be seen that the RMSE increased when the number of cells was above 80. Thus, the number of cells in the Bidirectional LSTM model is configured at 80. This result also shows that the number of cells in the hidden layer needs fine-tuning and cannot be fixed randomly.

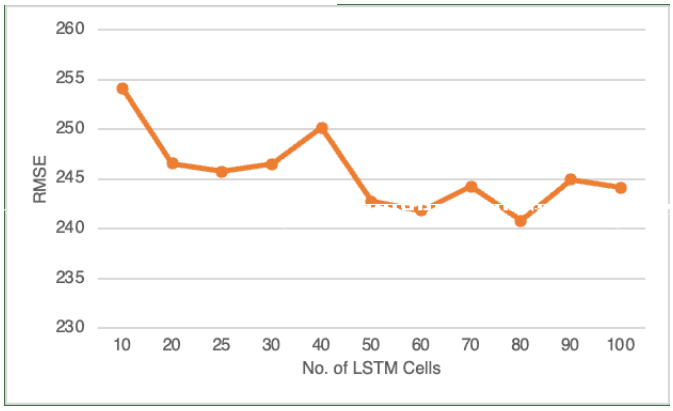

Fig 7. No. of LSTM cell vs RMSE

The final hyperparameters of the Bidirectional LSTM model are thus constructed using 24 input time steps, 80 cells in the hidden layer trained for 150 epochs and produce an RMSE of 240.79. This result shows an improvement of 8% compared to the single-cell RNN model of[17], 4% better compared to the single-cell LSTM of[17] and the Vanilla LSTM of[18]. The result also shows an improvement of 2% in comparison to the stacked LSTM of[18]. Figure 7 shows the performance of the proposed models in contrast to the existing models in terms of RMSE and MAE, and Table 2 shows the final hyperparameters of the models.

These results show that the Bidirectional LSTM that considered the past and future time steps is more capable in capturing the hidden patterns within the input data compared to the single-cell RNN, single-cell LSTM, Vanilla LSTM, and stacked LSTM. It could be considered for predicting time-series data. This results also shows that increasing the input time steps, number of cells in the hidden layer and number of epochs can improve the prediction performance of the model.

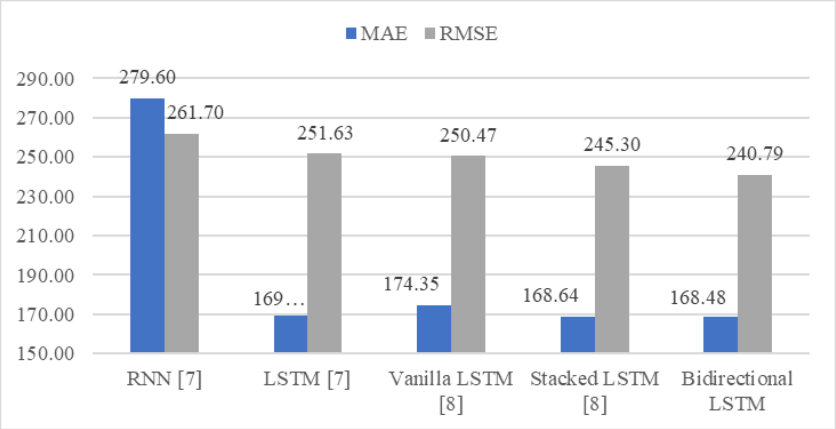

Fig 8. Performance of compared Model

Table 2. Final configuration of the models

LSTM Model	Input Timesteps	No. of Epochs	Loss function	Optimizer
RNN[17]	20	500	MSE	SGD
LSTM[17]	20	500	MSE	SGD
Vanilla LSTM[18]	12	400	MSE	Adam
Stacked LSTM[18]	28	500	MSE	Adam
Bidirectional LSTM	24	150	MSE	Adam

4 Conclusion

Rainfall prediction is a critical component in weather forecasting, agriculture, and other sectors. It is crucial to have a dependable and accurate method to predict rainfall so that informed decisions can be made. This study proposed a Bidirectional LSTM to forecast the average monthly rainfall in India using the all-India average monthly rainfall data from 1871 to 2016. The hyperparameters of the Bidirectional LSTM model are searched using the grid search method, and the final model is constructed using 24 input time steps, 150 epochs and 80 cells. The proposed Bidirectional LSTM achieved an RMSE value of 240.79, which shows an improvement of 8% compared to the benchmark single-cell RNN, 4% compared to both the benchmark single-cell LSTM and the Vanilla LSTM and 2% in comparison to the stacked LSTM of the existing studies. This study discovered that considering more input time steps, increasing the number of cells in the hidden layer and increasing the epoch can improve the prediction accuracy to some extent. However, overfitting occurs if the model is over trained. Thus, it is necessary to balance the size of the input data with the number of cells and epochs. Even though the Bidirectional LSTM model is able to capture long-term dependencies within the data, the limitation is that they are computationally more expensive compared to the Vanilla LSTM and Stacked LSTM. Finally, this study produced a new benchmark model that outperforms the existing benchmark model for monthly rainfall prediction in India. This study could be beneficial for farmers who depend heavily on rainfall, helping decision-makers to assign water resources more efficiently, preparation for disasters due to heavy rain and researchers interested in the area of rainfall predictions. This study finds the hyperparameter of the models using grid search method which does not guarantee the findings of optimal hyperparameters. Thus, future works will include the prediction of average monthly rainfall in India by tuning the LSTM and Bidirectional LSTM model's hyperparameters using swarm optimization methods such as Particle Swarm Optimization.

Acknowledgements

The authors are grateful for the infrastructural and other research facilities provided by the Department of Mathematics and Computer Science, Mizoram University. The authors thank the anonymous reviewers for their constructive comments on the manuscript. No funds, grants, or other support were received during the preparation of this manuscript.

References

1) Sahoo M, Yadav RK. Teleconnection of Atlantic Nino with summer monsoon rainfall over northeast India. *Global and Planetary Change*. 2021;203. Available from: https://dx.doi.org/10.1016/j.gloplacha.2021.103550.
2) Dube A, Karunasagar S, Ashrit R, Mitra AK. Spatial verification of ensemble rainfall forecasts over India. *Atmospheric Research*. 2022;273. Available from: https://dx.doi.org/10.1016/j.atmosres.2022.106169.
3) Kim HJ, Kim MK. A novel deep learning-based forecasting model optimized by heuristic algorithm for energy management of microgrid. *Applied Energy*. 2023;332. Available from: https://doi.org/10.1016/j.apenergy.2022.120525.
4) Abbot J, Marohasy J. Application of artificial neural networks to rainfall forecasting in Queensland, Australia. *Advances in Atmospheric Sciences*. 2012;29(4):717–730. Available from: https://dx.doi.org/10.1007/s00376-012-1259-9.
5) Abbot J, Marohasy J. Input selection and optimisation for monthly rainfall forecasting in Queensland, Australia, using artificial neural networks. *Atmospheric Research*. 2014;138:166–178. Available from: https://dx.doi.org/10.1016/j.atmosres.2013.11.002.
6) Abbot J, Marohasy J. Skilful rainfall forecasts from artificial neural networks with long duration series and single-month optimization. *Atmospheric Research*. 2017;197:289–299. Available from: https://dx.doi.org/10.1016/j.atmosres.2017.07.015.
7) Chattopadhyay S, Chattopadhyay G. Univariate modelling of summer-monsoon rainfall time series: Comparison between ARIMA and ARNN. *Comptes Rendus Géoscience*. 2010;342(2):100–107. Available from: https://dx.doi.org/10.1016/j.crte.2009.10.016.
8) Dash Y, Mishra SK, Panigrahi BK. Rainfall prediction for the Kerala state of India using artificial intelligence approaches. *Computers & Electrical Engineering*. 2018;70:66–73. Available from: https://dx.doi.org/10.1016/j.compeleceng.2018.06.004.
9) Mekanik F, Imteaz MA, Gato-Trinidad S, Elmahdi A. Multiple regression and Artificial Neural Network for long-term rainfall forecasting using large scale climate modes. *Journal of Hydrology*. 2013;503:11–21. Available from: https://dx.doi.org/10.1016/j.jhydrol.2013.08.035.
10) Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997;9(8):1735–1780. Available from: https://dx.doi.org/10.1162/neco.1997.9.8.1735.
11) Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*. 2005;18(5-6):602–610. Available from: https://dx.doi.org/10.1016/j.neunet.2005.06.042.
12) George B, Kutty G. Multivariate ensemble sensitivity analysis applied for an extreme rainfall over Indian subcontinent. *Atmospheric Research*. 2022;277. Available from: https://dx.doi.org/10.1016/j.atmosres.2022.106324.
13) Saha M, Santara A, Mitra P, Chakraborty A, Nanjundiah RS. Prediction of the Indian summer monsoon using a stacked autoencoder and ensemble regression model. *International Journal of Forecasting*. 2021;37(1):58–71. Available from: https://dx.doi.org/10.1016/j.ijforecast.2020.03.001.
14) Poornima S, Pushpalatha M. Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units. *Atmosphere*. 2019;10(11):1–18. Available from: https://dx.doi.org/10.3390/atmos10110668.
15) Viswanath S, Saha M, Mitra P, Nanjundiah RS. Deep Learning Based LSTM and SeqToSeq Models to Detect Monsoon Spells of India. In: International Conference on Computational Science, ICCS 2019;vol. 11537 of Lecture Notes in Computer Science. Springer, Cham. 2019;p. 204–218. Available from: https://doi.org/10.1007/978-3-030-22741-8_15.

16) Manoj SO, Ananth JP. MapReduce and Optimized Deep Network for Rainfall Prediction in Agriculture. *The Computer Journal*. 2020;63(6):900–912. Available from: https://dx.doi.org/10.1093/comjnl/bxz164.
17) Kumar D, Singh A, Samui P, Jha RK. Forecasting monthly precipitation using sequential modelling. *Hydrological Sciences Journal*. 2019;64(6):690–700. Available from: https://dx.doi.org/10.1080/02626667.2019.1595624.
18) Zoremsanga C, Hussain J. A Comparative Study of Long Short-Term Memory for Rainfall Prediction in India. In: Proceedings of the NIELIT's International Conference on Communication, Electronics and Digital Technology;vol. 676 of Lecture Notes in Networks and Systems. Singapore. Springer. 2023;p. 547–558. Available from: https://doi.org/10.1007/978-981-99-1699-3_38.
19) Kothawale DR, Rajeevan M. Monthly, Seasonal and Annual Rainfall Time Series for All-India, Homogeneous Regions and Meteorological Subdivisions: 1871-2016. 2017. Available from: https://www.tropmet.res.in/~lip/Publication/RR-pdf/RR-138.pdf.
20) Hunasigi P, Jedhe S, Mane M, Patil-Shinde V. Multilayer perceptron neural network based models for prediction of the rainfall and reference crop evapotranspiration for sub-humid climate of Dapoli, Ratnagiri District, India. *Acta Ecologica Sinica*. 2023;43(1):154–201. Available from: https://dx.doi.org/10.1016/j.chnaes.2022.09.004.
21) Feng S, Wang Y, Liu L, Wang D, Yu G. Attention based hierarchical LSTM network for context-aware microblog sentiment classification. *World Wide Web*. 2019;22(1):59–81. Available from: https://dx.doi.org/10.1007/s11280-018-0529-6.
22) Ying W, Zhang L, Deng H. Sichuan dialect speech recognition with deep LSTM network. *Frontiers of Computer Science*. 2020;14(2):378–387. Available from: https://dx.doi.org/10.1007/s11704-018-8030-z.
23) Song S, Huang H, Ruan T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*. 2019;78(1):857–875. Available from: https://dx.doi.org/10.1007/s11042-018-5749-3.
24) Abdel-Nasser M, Mahmoud K. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Computing and Applications*. 2019;31(7):2727–2740. Available from: https://doi.org/10.1007/s00521-017-3225-z.
25) Yu Y, Si X, Hu C, Zhang J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*. 2019;31(7):1235–1270. Available from: https://dx.doi.org/10.1162/neco_a_01199.
26) Schuster M, Paliwal KK. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*. 1997;45(11):2673–2681. Available from: https://dx.doi.org/10.1109/78.650093.
27) Ewees AA, Al-qaness MAA, Abualigah L, Elaziz MA. HBO-LSTM: Optimized long short term memory with heap-based optimizer for wind power forecasting. *Energy Conversion and Management*. 2022;268. Available from: https://dx.doi.org/10.1016/j.enconman.2022.116022.
28) Mahmoodzadeh A, Nejati HR, Mohammadi M, Ibrahim HH, Rashidi S, Rashid TA. Forecasting tunnel boring machine penetration rate using LSTM deep neural network optimized by grey wolf optimization algorithm. *Expert Systems with Applications*. 2022;209. Available from: https://dx.doi.org/10.1016/j.eswa.2022.118303.