DOI: 10.31577/ahs-2023-0024.02.0024

ACTA HYDROLOGICA SLOVACA

Volume 24, No. 2, 2023, 214 – 220

A CNN Bidirectional LSTM framework for predicting monsoon rainfall in India

Rajaprasad SVS*, Rambabu MUKKAMALA

Rainfall prediction has evolved as a paramount research significance in recent times due to its complexities and ongoing demand such as water resource planning and management. Agriculture is a major source of employment in India, as well as a substantial contributor to gross domestic product, and crop output is dependent on the monsoon season. Rainfall prediction is useful to authorities for water storage and timely release to increase crop productivity. The current study proposes a Deep Neural Network (DNN) based hybrid model using a combination of convolutional neural network bi-directional long short-term memory (CNN BiLSTM) to predict monthly rain fall during monsoon seasons. The DNN models were used to analyze the average monthly rainfall data collected across the country from 1871 to 2019 during the monsoon seasons. Furthermore, the hybrid model's results were compared to the Bidirectional LSTM (BiLSTM) architecture. In predicting rainfall in India, the proposed hybrid model framework has been found to be more accurate than the BiLSTM. The findings of the study suggest that a DNN frame work can be successfully adopted for time series analysis in water resource management and related domains to reduce the associated risks.

KEY WORDS: rainfall, prediction, DNN, CNN-BiLSTM, monsoon

Introduction

Rainfall prediction is crucial in Indian civilization, and it plays a significant part in human life. Rainfall water management boosts productivity in agriculture in developing countries, especially India. Rainfall prediction is critical because fluctuations in rainfall can have severe consequences, such as crop and property damage; hence, a better prediction tool is useful for early warning, which can reduce risks. Rainfall prediction is a complex situation due to unreliable climate parameters. Accurate rainfall prediction is vital for countries like India, whose economy is primarily based on agriculture. The parameters that have a direct impact on the amount of rainfall include temperature, relative humidity, pressure, and evaporation (Liyew and Melese, 2021). Rainfall prediction has recently received the greatest scholarly significance due to its complex and continual application in flood prediction. As a way to minimize damage, studies have explored and suggested rainfall prediction approaches in order to prepare for any type of situation (Barrera-Animas et al., 2022). As a result, several techniques for predicting rainfall have evolved in order to ensure efficient and precise results.

To predict rainfall, statistical approaches such as basic regression analysis, exponential smoothing, and autoregressive integrated moving average (ARIMA) are often used. Because environmental information is non-linear; multiple research studies indicate that these methods are still ineffective for predicting rainfall, but a few studies using these methods have proven to be effective (Shrivastava et al., 2012; Farajzadeh et al., 2014). In the past studies have been carried out employing empirical approaches to predict rainfall; however, the complexity of rainfall, such as its non-linearity, makes it difficult to predict (Wu and Chau, 2013). However, the statistical models used involve substantial computational capacity and can be time-consuming with few outcomes. As a result, the researchers' emphasis is focused on applying artificial neural networks for accurately predicting rainfall. The spatial and temporal heterogeneity of rainfalls influences prediction results, and recurrent Neural Networks (RNNs) are the best tool for addressing these issues (Hossain et al., 2020). Conversely, a drawback of the RNN framework is its limited capacity to understand in predicting the long-term time series data. To address this issue, a long short-term memory (LSTM) network came into existence, consisting of memory cells that govern the transfer of information between the different cells (Greff et al., 2016). When supervised machine learning (ML) approaches are integrated with fuzzy logic, the results outperform conventional approaches in rainfall prediction (Rahman et al., 2022). LSTM framework implementation is a relatively technically advanced process, it is a proven for accurate prediction of rainfall (Billah et al., 2022). Although empirical models can predict rainfall, ANN models outperformed

due to their computing capability and precision.
Based on the significance of LSTM architecture in rainfall prediction, the paper proposes two LSTM-based architectures; Bidirectional LSTM and CNN Bidirectional LSTM, which have been used to predict monthly rainfall during the monsoon seasons across India. To assess the performance of the DNN algorithms, the evaluation indicators mean absolute error (MAE), mean absolute percentage error (MAPE), root-mean-squared error (RMSE), and R square (R^2) have been adopted (Zhang et al., 2015). The study used all India monthly rainfall time series data relating to pre, post and monsoon periods from 1871 to 2019, and the model parameters are evaluated and compared.

Material and methods

The study aimed to evaluate the performance DNN algorithms. The LSTM architecture was exhaustively discussed in order to better understand the architectures of the Bidirectional and CNN-Bidirectional LSTM models despite the fact that the LSTM model was not employed in data analysis. The main feature of the analysis is to develop a hybrid model based on CNN. This section illustrates in detail the structure of deep learning(DL) networks adopted in the analysis. DNN were used to investigate the effectiveness of the model parameters on the prediction of all India rainfall during the monsoon seasons.

Rainfall data collection

In India, the yearly rainfall contribution is 97.9% from March to December and 2.1% during the winter months. The monsoon seasons are divided into three types: pre-monsoon, southwest monsoon, and post-monsoon. The data corresponding to three monsoon seasons from 1871 to 2019 are included for analysis in this study. For the period mentioned, all-India average monthly rainfall values were calculated by weighing each of the 30 sub-divisional rainfalls with their corresponding area as a weight (Kothawale and Rajeevan, 2017). To eliminate outliers and its marginal annual rainfall contribution, the winter rainfall data was not considered. The Indian Meteorological Department's (IMD) data from 1871 to 2019 were utilized to train and validate BiLSTM and CNN BiLSTM architectures.

LSTM Model

LSTM networks are a type of recurrent neural network (RNN) that increases memory recall by remembering past data (Liu et al., 2022) and backpropagation is used to train the model, which is ideally suited to predicting time series with unpredictable time lags. The LSTM networks were explicitly developed to circumvent the long-term dependency issues and handle the vanishing gradient problem successfully, and the model is divided into three sections which perform a specific function as shown in Fig. 1.
The first section determines whether the data from the previous time step is relevant or can be ignored.

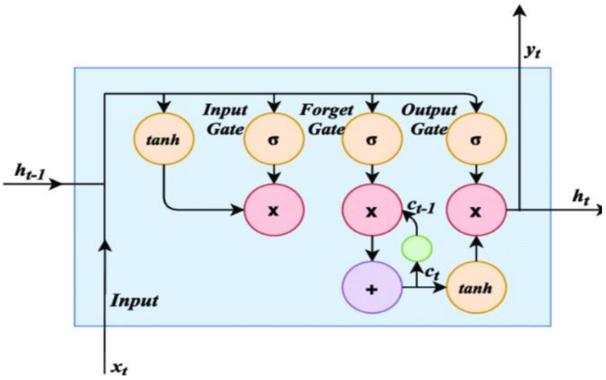

Fig. 1. LSTM block diagram.

In the second and final sections, the cell seeks to learn new information from the input and transfers the updated information from the current time step to the next time step respectively, and the LSTM cycle is viewed as a single-time step. These three sections of the LSTM unit are known as forget gate, input gate, and output gate. A memory cell in an LSTM network can be viewed as a layer of neurons in a typical feedforward neural network, with each neuron having a hidden layer and an ongoing state. These gates also solve the problem of vanishing gradient, which generally occurs in RNNs. As a result, it is widely used in a variety of applications in time series prediction (Huang, et al., 2022). The first section determines whether to retain or remove the information from the previous time step. Equation 1 depicts the forget gate. The activation value of forget gate f_t at time t is calculated using a sigmoid function. The f_t is then multiplied by the previous time step's cell condition.

$$f_t = \sigma\,(X_t * U_f + h_{t-1} * W_f) \quad (1)$$

where
X_t –input to the on-going time step,
U_f –weight related with the input,
h_{t-1} –hidden state of the preceding time step,
W_f –weight matrix related with the hidden state.

The second gate, the input gate, is used to evaluate the relevance of the new information carried by the input, and its equation is shown in Equation 2.

$$i_t = \sigma\,(X_t * U_i + h_{t-1} * W_i) \quad (2)$$

where
U_i –weight matrix of input,
W_i –weight matrix of input associated with hidden state.

The sigmoid function is then applied on top of this, resulting in the value of 'i' at time step t being between 0 and 1. Equation 3 is used to calculate the new information.

$$N_t = tanh\,((X_t * U_c + h_{t-1} * W_c) \quad (3)$$

The new information that has to be provided to the cell state is now a function of a hidden state at time step t-1 and input x at time step t. Tanh is the activation function, and the value of the new information runs from -1 to 1. If N_t is negative, information is subtracted from the cell state and added to the continuing state, and vice versa. The N_t, however, is not directly added to the cell state. Equation 4 depicts the modified cell state equation.

$$C_t = f_t * C_{t-1} + i_t * N_t \tag{4}$$

where
C_{t-1} – the cell state at the on-going time step.

The equation of the output gate is shown as Equation 5.

$$o_t = \sigma\ (X_t * U_o + h_{t-1} * W_o) \tag{5}$$

Equation 6 is used to compute the current hidden state using the output state 'o_t' and tanh of the updated cell state. As indicated in Equation 7, the hidden state is a function of long-term memory (C_t) and the current output, and the output of the current time step is determined by applying the SoftMax activation to hidden state ht. The predicted value is nothing but the maximum score in the output.

$$h_t = o_t * tanh\ (C_t) \tag{6}$$

$$Output = Softmax\ (h_t) \tag{7}$$

Finally, backpropagation is used to obtain the LSTM, which is then stored in memory blocks. Now, the LSTM can successfully familiarize the inputted time series data to produce a long-term memory function (Wang et al., 2021). In this model, the outputs of the cell are controlled by the gates. Fig. 2 depicts the architecture of a single LSTM cell (Metlek et al., 2021).

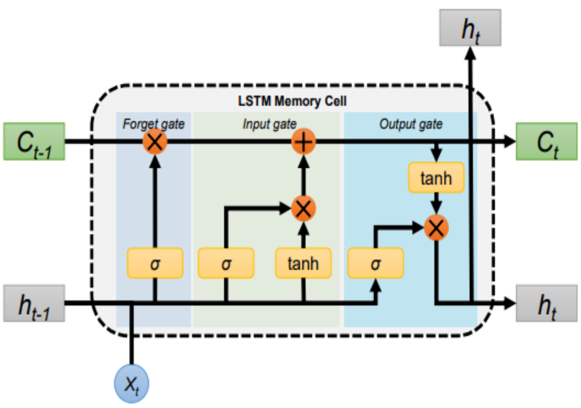

Fig. 2. Schematic diagram of the LSTM cell architecture.

BiLSTM Model

LSTM may receive long range information prior to the output time but cannot use reverse information. To greatly increase prediction accuracy, the forward and backward information of time series data should be completely considered in time series prediction. BiLSTM is made up of two LSTM's, one forward and one backward layer. In comparison to the regular LSTM's one-way-state transmission, BiLSTM evaluates data changes before and after data transmission and can make more full and detailed decisions based on past and future information (Zhuang and Cao, (2022). BiLSTM model performs forward and backward calculations as shown in the model structure in Fig. 3 and it depicts the two-way flow of time series information in the model, whereas data information flows vertically in only one direction from the input layer to the hidden layer to the output layer (Varghese et al., 2022). The purpose of using the LSTM twice makes the model to learn the model long-term dependencies and increase its accuracy (Metlek et al., 2021).

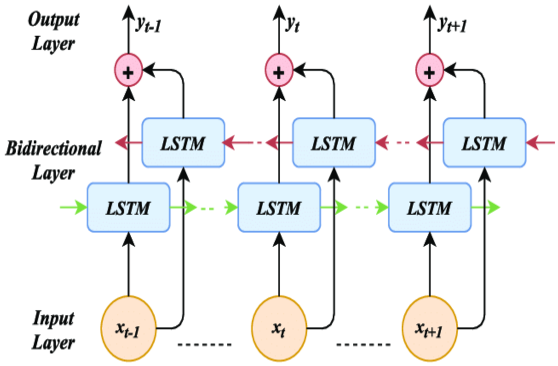

Fig. 3. Schematic diagram of BiLSTM structure.

CNN BiLSTM Model

The model combining CNN, BiLSTM and the connection layer proposed to predict the average monthly rainfall during the monsoon seasons across India is referred to as the CNN BiLSTM or hybrid model. The input of the hybrid model first enters the CNN layer, and after convolution calculation and max-pooling; a new feature matrix is generated. The feature matrix obtained from the CNN is used as the BiLSTM's input, and the BiLSTM's hidden output is obtained. The hidden output is routed through the connection layer, which is made up of a linear layer. The connection layer then returns the final results (Ullah et al., 2019; Metlek, 2023). Fig. 4 depicts the hybrid model architecture (Varghese et al., 2022).
The hybrid model describes how input and output interact. The recursive multi-step forecasting approach is used to create the univariate time series forecasting model. The univariate time series must be modified to suit the CNN input and output BiLSTMs because the hybrid model employs supervised learning. Assuming a univariate time series sample $y^{(1)}$, $y^{(2)}$.,. $Y^{(n)}$ with lag, the predicted value of y $^{(\tau+1)}$) can be obtained using the preceding steps. The one-dimensional vector is then rebuilt into a (τ + 1) dimensional matrix. Equation 8 shows how to generate the reconstructed sample matrix, Θ.

$$\Theta = [Y^{(1)}, Y^{(2)}, \ldots, Y^{(\tau)}, Y^{(\tau+1)}] \quad (8)$$

where
$Y^{(1)} = [y^{(1)}, y^{(2)}, \ldots, y^{(\tau)}, y^{(\tau+1)}]$ - a column vector with lag τ.

Then, the input to the hybrid model is a matrix X consisting of the previous τ column vectors $[Y^{(1)}, Y^{(2)}, \ldots, Y^{(\tau)}]$, and the output is the $(\tau + 1)$ as shown in Equation (8). When the forecasting reaches step $\tau + 1$, the input vector already contains all of the predicted values, implying that the extrapolation is complete (Chen and Fu, 2023). The hybrid model training process to predict the average monthly rainfall during the monsoon seasons is as follows:
Step 1: Clean up the data by removing unnecessary items, serializing time data, and dividing the training and testing sets.
Step 2: Input the pre-processed time series data into the hybrid model for training.
Step 3: Feed the training data into the trained model to predict.
Step 4: Using the formulas, restore the predicted data.
Step 5: Plot a graph that compares the actual and expected rainfall values, and use the actual and predicted values to evaluate the model's prediction power.

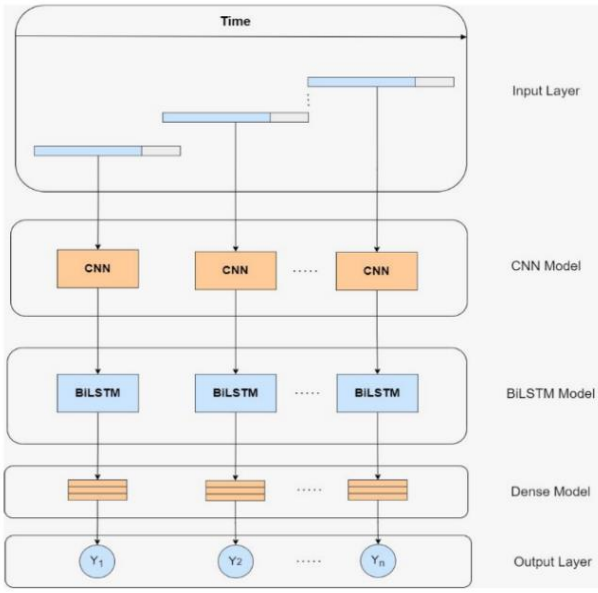

Fig. 4. Hybrid model architecture.

Results and discussion

Data preprocessing

In the present study, data refer to the year 1871 to 2019 for pre-monsoon, south west monsoon and post monsoon seasons was considered for analysis. The rainfall data relating to the winter months was excluded. The dataset consists of 1472 monthly average of all India rainfall data of three monsoon seasons. To maintain high data quality, the authors removed the outliers and standardized the data format and there are no null data points present. Data points were normalized between 0 and 1 using the MinMax scaler before splitting the data set for training and testing. Out of 1472 data points, 1030 (70%) and 442 (30%) were used to train and test the dataset to assess the models' predictive capability. Finally, five evaluation metrics namely; MAE, MAPE, RMSE and R^2 were used to check the model's predictive capability.

Parameters and the Experimental Models

In this study, the results of the hybrid model were compared to the evaluation metrics of BiLSTM model. The model and training parameters of two models are kept the same for comparison purposes. The Adam optimizer is used to calculate the adaptive parameter learning rate based on the mean of the first and second moments of the gradient. Several experiments were carried out to determine the best possible architectural configuration by varying model parameters (viz. number of hidden layers, number of filters, kernel size, pooling size, and dropout percentage) and optimizing several hyperparameters (viz. learning rate, batch size, number of epochs, and loss functions). The learning rate is 0.0001, and MAE is used as the loss function. The forecasting model was developed using a value of 12 preceding monthly rainfall instances (Deepak et al., 2019). It only measures the mean modulus length of the predicted value error, disregarding direction, and is more robust to outliers. The batch size and the number of epochs is 32 and 250 respectively with early stopping. Table 1 displays the hybrid model's parameter settings.

Table 1. Parameter settings of hybrid model

Parameters	Value
Conv1D(filters)	16
Conv1D kernel_size	2
Conv1D activation function	LeakyReLU
MaxPooling1D pool_size	1
BiLSTM units	32
BiSLSTM activation function	Tanh
Dense units	1

The average monthly rainfall data during the monsoon seasons after preprocessing are put into the BiLSTM, and hybrid models for training. The test dataset is used for prediction after completing the training. The plot of the hybrid model relating to the actual and predicted values in the last 1000 months are shown in Fig. 5. The learning curve of the hybrid model in prediction of average monthly rainfall is shown in Fig. 6. The learning curves are plotted to check whether the train or validation datasets are appropriately representative of the domain area. The learning curve shows that the model fit is good, with a training and validation loss reducing to a stability

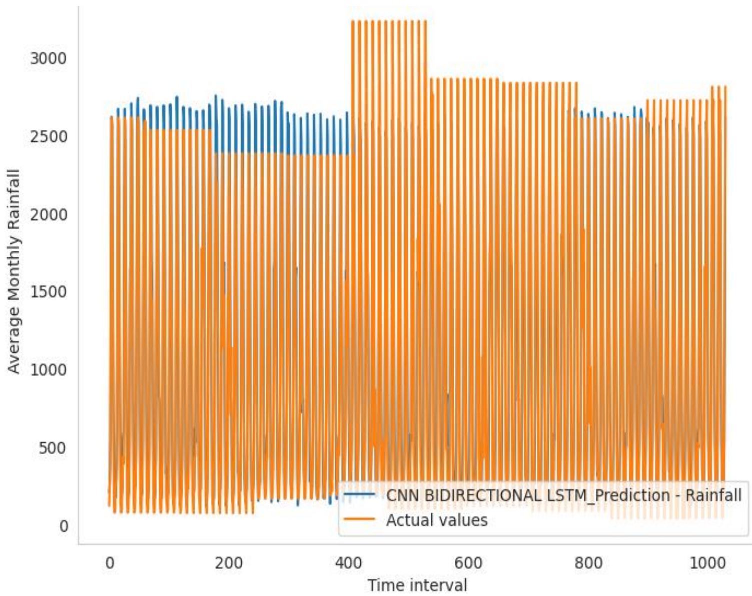

Fig. 5. Hybrid model – Actual vs predicted value of GHI.

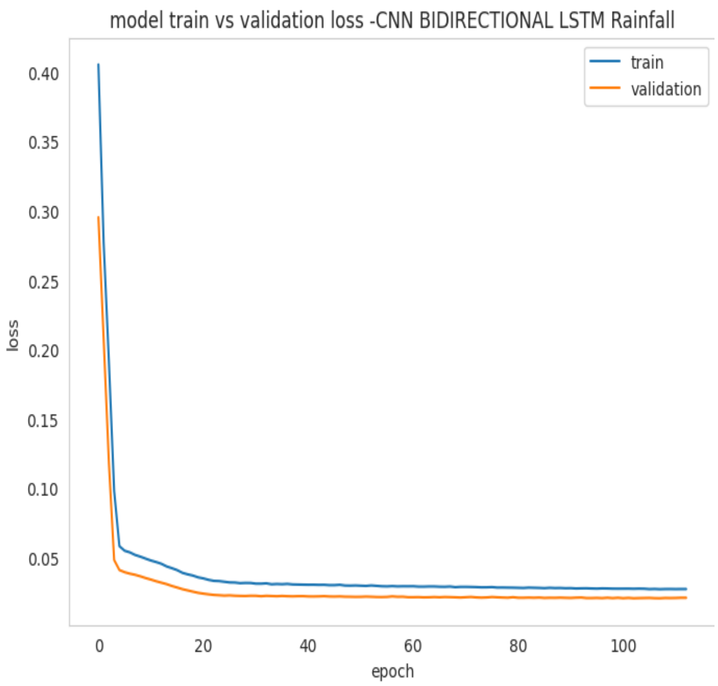

Fig. 6. Hybrid model learning curve (Model training vs validation loss of Rainfall).

point with a marginal difference between the two final loss values.

The most important evaluation parameters MAE, MAPE, RMSE, and R^2 are utilized in the analysis to compare the model results. These four parameters are used to assess the difference between predicted and actual rainfall, and their values are shown in Table 2. When compared to the Bidirectional LSTM model, the hybrid model has a higher R^2 score. The hybrid model's MAE, MAPE, and RMSE scores are significantly lower, showing that the model's predictive power is accurate.

To construct a hybrid model, the LSTM and BiLSTM prediction data are integrated with CNN for feature extraction, and more significant extended features are learned. The R^2 score of the hybrid model is 0.9485,which is greater than the R^2 scores of the BiLSTM model and it shows that the model

Table 2. Model evaluation parameters of rainfall

Model	MAE	MAPE	RMSE	R^2
BiLSTM	178.908	41.225	249.255	0.9243
Hybrid model	189.568	47.257	256.586	0.9485

outperformed the empirical and DL models in terms of predicting time-series data (Varghese et al., 2022). The LSTM model outperformed the RNN in forecasting average monthly yearly rainfall in India from 1871 to 2016 (Deepak et al., 2019) with a correlation coefficient of 0.960, whereas the hybrid model in the current study obtained a correlation coefficient of 0.974. A combination of a one-dimensional Convolutional Neural Network and a Multi-Layer Perceptron neural network proved to be a better model for predicting daily rainfall by considering nine variables that are closely associated with daily rainfall variation with a coefficient of correlation ranging from 0.41 to 0.76 (Mislan et al., 2015). The hybrid model has the best degree of fit based on the evaluation parameters between the predicted and actual values. The Extreme Gradient Boost gradient descent machine learning approach predicts rainfall in Bahir Dar, Ethiopia more accurately than the random forest algorithm and Multi-Layer Perceptron neural network with low MAE and RMSE scores (Liyew and Melese, 2021). Backpropagation Neural Network demonstrated better results for predicting rainfall in Tenggarong, East Kalimantan, Indonesia, with the lowest mean squared error when the hyperparameters were varied (Mislan et al., 2015). According to previous studies, the hybrid model was not applied for rainfall analysis, and the model findings in the present research are substantially improved. Recent approaches in predicting the rainfall using hybrid models of LSTM-Networks will be investigated. Recently developed architectures for time series data prediction will be studied using hybrid LSTM models (Altan et al., 2021), and the present study effectively analyzed the rainfall data using a CNN-BiLSTM hybrid model.

The learning curve of the hybrid model shows that the model fit is good with no overfitting or underfitting. The accuracy of the models used was validated against the actual rainfall, resulting in an average absolute percent error value of less than 1%. Compared to the BiLSTM model, the results of the hybrid model show that the improvement in the predictive capability and the predicted values are closely following the trend of the original dataset. The proposed model can be trained and used with the other parameters of the meteorological data that have an impact on the rainfall. The hybrid model can be applied to regression and time series data via a one-dimensional filter as in the proposed model. As a result, the proposed hybrid model can be applied in many fields such as forecasting of meteorological, health care, and environmental datasets in the long term.

Conclusion

The study aims to model India's average monthly rainfall during the monsoon seasons. The BiLSTM model was used for this purpose and in addition to this model, a hybrid model is developed and proposed. The hybrid model R^2 scores provided better results for predicting rainfall. The results reveal that the hybrid model outperforms the BiLSTM deep learning neural networks in terms of prediction. The evaluation parameters of the models used were verified using the actual rainfall and the MAPE of the proposed model was less than 1%. The coefficient of correlation is closer to one which confirms the robustness of the model in predicting rainfall. The proposed model fit is good as it falls between an overfit and an underfit model. The proposed hybrid model can be applied to model the meteorological parameters that influence rainfall. The limitation of the study is parameters which are influencing the monthly rain were not considered. The study can be extended in the future by developing a multivariate CNN-BiLSTM architecture by considering the variables that have a direct impact on rainfall. Furthermore, a comprehensive evaluation of the inclusion of the variables influencing rainfall should be investigated to increase the effectiveness of prediction models.

Acknowledgement

We acknowledge the data utilized from Research Report No. RR-138 ESSO/IITM/STCVP/SR/02(2017)/189, Monthly, Seasonal and Annual Rainfall Time Series for All-India, Homogeneous Regions and Meteorological Subdivisions: 1871-2016, Indian Institute of Tropical Meteorology (IITM), India and also Open Government Data(OGD) platform, India (https://data.gov.in/catalog /rainfall-india).

References

Altan, A., Karasu, S., Zio, E. (2021): A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. Applied Soft Computing, vol.100, 106996.

Barrera-Animas, A. Y., Lukumon, O. O., Muhammad, B., Taofeek, D. A., Juan, M. D., Lukman, A. A. (2022): Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. Machine Learning with Applications, vol. 7, no. 100204, 1–20.

Billah, M., Adnan, Md. N., Akhond, M. R., Ema, R. R., Hossain, Md. A., Md. G. S. (2022): Rainfall prediction system for Bangladesh using long short-term memory. Open Computer Science, vol. 12, no. 1, 323–331.

Chen, Y., Fu, Z. (2023): Multi-Step Ahead Forecasting of the Energy Consumed by the Residential and Commercial Sectors in the United States Based on a Hybrid CNN-BiLSTM Model. Sustainability, vol. 15, no.3, 1–21.

Deepak, K., Anshuman, S., Pijush, S., Rishi Kumar, J. (2019): Forecasting monthly precipitation using sequential modelling. Hydrological Sciences Journal, vol.64, no. 6, 690–700.

Farajzadeh, J., Fard, A. F., Lotfi, S. (2014): Modeling of monthly rainfall and runoff of Urmia lake basin using feed-forward neural network and time series analysis model. Water Resources and Industry, vol. 7–8, 38–48.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., Schmidhuber, J. (2016): LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, vol.28, no.10, 2222–2232.

Hossain, I., Rasel, H., Imteaz, M. A., Mekanik, F. (2020): Long-term seasonal rainfall forecasting using linear and non-linear modelling approaches: A case study for western Australia. Meteorology and Atmospheric Physics, vol.132, no.1, 131–141.

Huang, X., Li, Q., Tai, Y., Chen, Z., Liu, J., Shi, J., Liu, W. (2022): Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM. Energy, vol. 246, 123403.

Kothawale, D. R., Rajeevan, M. (2017): Research Report No. RR-138 ESSO/IITM/STCVP/SR/02(2017)/189, Monthly, Seasonal and Annual Rainfall Time Series for All-India, Homogeneous Regions and Meteorological Subdivisions: 1871-2016, Indian Institute of Tropical Meteorology (IITM), 1–169.

Liu, Y., Meenakshi, V., Karthikeyan, L., Maroušek, J., Krishnamoorthy, N. R., Sekar, M., Xia, C. (2022): Machine learning based predictive modelling of micro gas turbine engine fueled with microalgae blends on using LSTM networks: An experimental approach. Fuel, vol. 322, 1–8.

Liyew, C. M., Melese, H. A. (2021): Machine learning techniques to predict daily rainfall amount. J Big Data, vol. 8, no. 153, 1–11.

Metlek, S., Kayaalp, K., Basyigit, I. B., Genc, A., Dogan, H. (2021): The dielectric properties prediction of the vegetation depending on the moisture content using the deep neural network model. International Journal of RF and Microwave Computer-Aided Engineering, vol.31, no.1, 1–10.

Metlek, S. (2023): A new proposal for the prediction of an aircraft engine fuel consumption: a novel CNN-BiLSTM deep neural network model. Aircraft Engineering and Aerospace Technology, vol. 95, no.5, 838–848.

Mislan, H., Sigit, H., Sumaryono, M. A. (2015): Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan – Indonesia. Procedia Computer Science, vol. 59, 142–151.

Rahman, A. U., Abbas, S., Gollapalli, M., Ahmed, R., Aftab, S., Ahmad, M., Khan, M. A., Mosavi, A. (2022): Rainfall Prediction System Using Machine Learning Fusion for Smart Cities. Sensors, vol.22, no. 3504, 1–15.

Shrivastava, G., Karmakar, S., Kowar, M. K. (2012): Application of Artificial Neural Networks in Weather Forecasting: A Comprehensive Literature Review. International Journal of Computer Applications, vol. 51, 17–29.

Ullah, F. U. M., Ullah, A., Haq, I. U., Rho, S., Baik, S. W. (2019): Short-term prediction of residential power energy consumption via CNN and multi-layer bi-directional LSTM networks. IEEE Access, vol. 8, 123369-123380.

Varghese, J., Anshuman, M., Sowmaye, T., Shruti, M., Sandeepkumar, S., Sachi, N. M. (2022): A hybrid deep learning framework with CNN and Bi – directional LSTM for store item demand forecasting. Computers and Electrical Engineering, vol.103, 1–14.

Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., Wang, J. (2021): A stock closing price prediction model based on CNN BiSLSTM. Complexity, 1–12.

Wu, C., Chau, K.W. (2013): Prediction of rainfall time series using modular soft computing methods. Engineering Applications of Artificial Intelligence, vol.26, no.3, 997–1007.

Zhang, J., Florita, A., Hodge, B. M., Lu, S., Hamann, H. F., Banunarayanan, V., Brockway, A. M. (2015): A suite of metrics for assessing the performance of solar power forecasting. Solar Energy, vol.111, 157–175.

Zhuang, W., Cao, Y. (2022): Short-Term Traffic Flow Prediction Based on CNN-BILSTM with Multicomponent Information. Applied Sciences, vol. 12, no.17, 1–15.

Dr. Rajaprasad Svs, Ph.D (*corresponding author, e-mail: rajaprasad@nicmar.ac.in)
Rambabu Mukkamala, M.Tech
National Institute of Construction Management and Research (NICMAR)
Hyderabad
India