A DEEP LEARNING APPROACH USING LONG SHORT-TERM MEMORY NETWORKS FOR ENHANCED PREDICTION OF RAINFALL IN THE NORTHEASTERN REGION OF BANGLADESH

Sujan Chandra Mondol*[1], Sajal Kumar Adhikary[2], Hrithik Nath[3], Shuvendu Pal Shuvo[4]

[1] *Department of Civil Engineering, Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh, e-mail: scm.mondol@gmail.com*
[2] *Department of Civil Engineering, Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh, e-mail: sajal@ce.kuet.ac.bd*
[3] *Department of Civil Engineering, University of Creative Technology Chittagong (UCTC), Chattogram, Bangladesh, e-mail: hrithiknath.ce@gmail.com*
[4] *Department of Civil Engineering, Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh, e-mail: shuvenduce@gmail.com*

***Corresponding Author**

ABSTRACT

Rainfall is a significant climatic parameter that is often used as a crucial input for hydrological analysis and modeling for the effective management of water resources. Accurate and reliable prediction of rainfall is essential for the design of hydraulic structures and the development of decision-making frameworks for different water resource projects, agricultural expansion, climate change studies, and disaster prevention schemes. Recently, deep learning and machine learning-based artificial intelligence (AI) techniques have gained significant popularity among researchers for the prediction of rainfall, as they are able to capture the underlying non-linear relationships between inputs and outputs. In the past, various AI techniques, including artificial neural networks (ANN), support vector machines (SVM), and genetic programming (GP), have been used for rainfall prediction across various regions all over the world. Although these methods have been improved over the years to achieve enhanced performance in rainfall prediction, their limitations continue to exist. Therefore, the current study explores a long short-term memory (LSTM) network-based deep learning approach for the prediction of rainfall, considering its inherent suitability for time-series data. The LSTM networks are a type of recurrent neural network (RNN) that is capable of learning order dependence in sequence prediction problems and offers a solution to the vanishing gradient problem present in traditional RNNs. In addition to the LSTM network-based deep learning model, two other AI models, namely ANN and SVM, are developed in order to prove the efficiency of the LSTM network-based deep learning approach for the enhanced prediction of rainfall. A large dataset of the monthly values of various hydrometeorological variables, including rainfall, temperature, humidity, windspeed, and cloud cover, for more than a 66-year period (1956–2021) is used as the input to predict the one-month ahead monthly rainfall for Sylhet and Srimangal stations in the northeast region of Bangladesh. The performance of each of the aforementioned models is assessed based on various model performance evaluation criteria, including root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R2). The results indicate that the LSTM network-based deep learning approach gives enhanced prediction of rainfall and accordingly outperforms the individual ANN and SVM techniques in terms of prediction accuracy based on different model performance criteria. Thus, the study concludes that the LSTM network-based deep learning approach could be a viable technique for the accurate prediction of rainfall in the northeast region of Bangladesh and could be applicable to other similar areas.

Keywords: *Rainfall prediction, Machine learning, RNN, ANN, SVM, Long Short Term Memory*

1. INTRODUCTION

Rainfall is the paramount hydrometeorological variable due to its copious scope in and around many domains. It is required to be incorporated as one of the key considerations in designing many systems ranging from irrigation to national development. Various water resources management schemes, agricultural strategies, hydrologic modeling and simulation projects, and disaster mitigation programs consider it quite imperative in their decision-making process. Hence, the prediction of rainfall can in no way be considered a mere part of weather forecasting. Rather, it remains a critical research problem in the face of the rising complexity of the meteorological and environmental systems imposed by the impact of climate change.

Bangladesh is one of the most vulnerable countries to long-term climate risk (Islam et al., 2022). The history of prominent and frequent visits by a number of natural hazards, mostly of hydroclimatic nature, and the consequent impact on the people and economy are very likely to account for that (Karim, 1995). Furthermore, the geographic location of Bangladesh is also responsible for its disaster-prone nature and remarkable sensitivity towards climate change (Brouwer et al., 2007). Bangladesh is located at the lowermost part of the Ganges-Brahmaputra-Meghna (GBM) basin with an extensive drainage area, and so it is responsible for draining a huge amount of water through its landscape to the Bay of Bengal. Therefore, it is very crucial to look for better ways to predict rainfall in order to better prepare for various rainfall-instigated extremities in Bangladesh.

Rainfall is essentially condensed water from the atmosphere and is affected by many meteorological and atmospheric variables. Due to the inherent uncertainty and its stochastic nature, it is really hard to model it with comparable performance through a physics-based model. Consequently, the prediction of rainfall in the literature is primarily based on data-driven models. At the primary stage, statistical models and various time series models were the key resort for the researchers to estimate rainfall. Many time series modeling techniques remain popular even in very recent times. Although these methods were developed over time, their limitations continue to exist. Recently, artificial neural networks (ANN) have received a considerable attention to be implemented for prediction tasks in the fields of hydrology and water resources. Past studies have demonstrated the wide application of ANN in rainfall prediction all over the world (Sahai et al., 2000; Iseri et al., 2005; Hossain et al., 2020). The convolutional neural network (CNN) is one of the most popular deep ML models with its well-known capability for image processing, and such a model manifested better results than MLP in predicting the monthly rainfall in Australia (Haidar & Verma, 2018).

Deep learning models are computationally intensive with better predictive results as they include more than one hidden layer with a potentially high number of neurons and have grown popular recently due to the availability of high-performance computation devices in the public sphere. The very recent trend dictates the implementation of a special type of RNN called LSTM for time series forecasting, and it has applications from financial to hydrologic forecasting. In a study seeking to judge the aptness of the LSTM model in comparison with other statistical and ML models, improved performance was observed for the LSTM model, with no significant variation in performance for the difference in the number of hidden units (Kang et al., 2020). Billah et al. (2022) used an LSTM model coupled with principal component analysis (PCA)-based feature selection to predict the daily rainfall category in Bangladesh. The model performed significantly better than six other classifier models by producing more than 97% accuracy. In another study comparing both the stacked LSTM and bidirectional LSTM with four other machine learning algorithms, it was evident that the LSTM models showed superior performance in the prediction of hourly rainfall (Barrera-Animas et al., 2022). Furthermore, a recent review study on rainfall prediction by various machine learning models indicates that the LSTM model is regarded as possessing the best performance for rainfall prediction (Latif et al., 2023). Some studies even tried various modified versions of the typical LSTM models and demonstrated their prospects for further improved performance (Poornima & Pushpalatha, 2019). On the whole, the aim of the current study is to explore a long short-term memory (LSTM) network-based deep learning approach for the enhanced prediction of rainfall, considering its inherent suitability for time-series data.

2. METHODOLOGY

2.1 Study Area and Datasets

The current study focuses on the rainfall prediction of two stations, namely Srimangal and Sylhet, located in Moulvibazar and Sylhet districts of Bangladesh, respectively, as shown in Figure 1. These two stations are located at the northeast end of Bangladesh. The study region is located at the lower end of the Meghna basin and is usually prone to heavy rainfall during the monsoon and pre-monsoon. The rainfall in this area is particularly of significant interest due to a few reasons. This area is the major host of the cultivation of tea, and tea is one of the noteworthy cash crops, markedly contributing to the national economy. Rainfall affects the production of tea both from an agricultural and socioeconomic perspective (Farukh et al., 2020; Rahman, 2022). Furthermore, Hasan et al. (2012) found a link between flood occurrences in other areas of the country and the exceedance of the normal yearly rainfall in this area. In addition, this area itself is prone to flash floods. Again, this is the region in Bangladesh that accommodates many wetland ecosystems, locally named 'haors'. The huge biodiversity supported by these ecosystems is not only important for local environmental stability but also too deeply rooted for the overall ecological balance of Bangladesh (Jakariya & Islam, 2017).

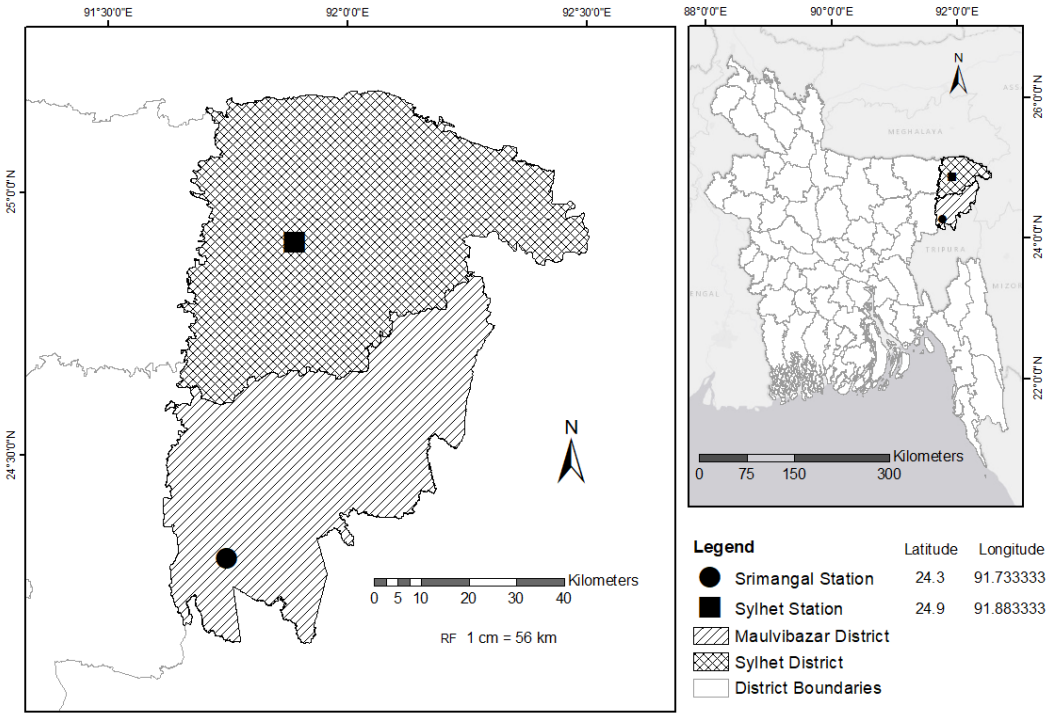


Figure 1: Selected rainfall stations in northwestern Bangladesh on the map

The hydrometeorological dataset with variables including rainfall, dew point temperature, temperature (maximum, average, and minimum), pressure (mean sea level and station level), cloud amount, wind speed and direction, bright sunshine hour, relative humidity, present weather, past weather, and shapefiles was collected from the Bangladesh Meteorological Department (BMD). The provided data have data ranges of variable periods based on availability. The dataset for the teleconnection climate indices, namely Nino 3.4, Dipole Mode Index (DMI), Pacific Decadal Oscillation (PDO), and Southern Oscillation Index (SOI), was obtained from the website of the Physical Sciences Laboratory (PSL) of the National Oceanic and Atmospheric Administration (NOAA).

2.2 Data Preprocessing

There were thirteen hydrometeorological variables in the originally collected dataset from BMD, among which twelve variables contained only numeric values. The data for a particular variable was provided inside a separate file in which the data for different stations were arranged in tabular form.

The format was similar for the four climate teleconnection indices. Each of the data tables had rows and columns pointing out the corresponding year and month, respectively, for a particular data point. First of all, the matrix format datasets were converted to timeseries data for each of the variables separately for the two stations. The prevailing wind direction (WD) was the only categorical variable, and it encoded the sixteen-point wind directions with corresponding angles in degrees. This variable, along with the prevailing wind speed, was replaced with two numeric variables, namely the north (WS_{north} ¿ and east component (WS_{east} ¿ of the prevailing wind speed, using the formula shown in Eqs. (1)- (2).

$$WS_{north} = (WS)(\cos WD) \quad (1)$$

$$WS_{east} = (WS)(\sin WD) \quad (2)$$

(a)

	Total Rainfall	Avg. Cloud Amount	Avg. Dewpoint Temperature	Avg. Humidity	Avg. MSL Pressure	Avg. SL Pressure	Avg. Temperature	Maximum Temperature	Minimum Temperature	Wind Speed East Comp.	Wind Speed North Comp.	DMI	Nino 3.4	PDO	SOI	Rainfall (1 month ahead)
Total Rainfall	1.00	0.86	0.77	0.71	-0.83	-0.79	0.66	0.55	0.75	-0.08	-0.26	-0.05	0.25	0.12	-0.03	0.64
Avg. Cloud Amount	0.86	1.00	0.90	0.75	-0.92	-0.87	0.80	0.69	0.88	-0.10	-0.29	-0.06	0.27	0.07	-0.05	0.66
Avg. Dewpoint Temperature	0.77	0.90	1.00	0.81	-0.88	-0.84	0.91	0.76	0.96	-0.12	-0.23	-0.06	0.20	0.04	-0.02	0.55
Avg. Humidity	0.71	0.75	0.81	1.00	-0.65	-0.60	0.51	0.33	0.73	-0.07	-0.15	-0.11	0.10	0.07	-0.05	0.34
Avg. MSL Pressure	-0.83	-0.92	-0.88	-0.65	1.00	0.95	-0.85	-0.75	-0.88	0.13	0.32	0.10	-0.21	-0.01	-0.02	-0.72
Avg. SL Pressure	-0.79	-0.87	-0.84	-0.60	0.95	1.00	-0.82	-0.72	-0.84	0.14	0.29	0.09	-0.19	-0.01	-0.04	-0.68
Avg. Temperature	0.66	0.80	0.91	0.51	-0.85	-0.82	1.00	0.88	0.91	-0.12	-0.25	-0.03	0.21	0.01	0.00	0.57
Maximum Temperature	0.55	0.69	0.76	0.33	-0.75	-0.72	0.88	1.00	0.77	-0.12	-0.19	0.03	0.27	0.07	-0.01	0.56
Minimum Temperature	0.75	0.88	0.96	0.73	-0.88	-0.84	0.91	0.77	1.00	-0.13	-0.24	-0.05	0.16	0.01	-0.01	0.53
Wind Speed East Comp.	-0.08	-0.10	-0.12	-0.07	0.13	0.14	-0.12	-0.12	-0.13	1.00	0.36	0.03	0.04	-0.01	-0.04	-0.04
Wind Speed North Comp.	-0.26	-0.29	-0.23	-0.15	0.32	0.29	-0.25	-0.19	-0.24	0.36	1.00	0.04	-0.09	0.05	0.00	-0.27
DMI	-0.05	-0.06	-0.06	-0.11	0.10	0.09	-0.03	0.03	-0.05	0.03	0.04	1.00	0.23	-0.04	-0.20	-0.03
Nino 3.4	0.25	0.27	0.20	0.10	-0.21	-0.19	0.21	0.27	0.16	0.04	-0.09	0.23	1.00	0.45	-0.64	0.38
PDO	0.12	0.07	0.04	0.07	-0.01	-0.01	0.01	0.07	0.01	-0.01	0.05	-0.04	0.45	1.00	-0.41	0.13
SOI	-0.03	-0.05	-0.02	-0.05	-0.02	-0.04	0.00	-0.01	-0.01	-0.04	0.00	-0.20	-0.64	-0.41	1.00	-0.04
Rainfall (1 month ahead)	0.64	0.66	0.55	0.34	-0.72	-0.68	0.57	0.56	0.53	-0.04	-0.27	-0.03	0.38	0.13	-0.04	1.00

(b)

	Total Rainfall	Avg. Cloud Amount	Avg. Dewpoint Temperature	Avg. Humidity	Avg. MSL Pressure	Avg. SL Pressure	Avg. Temperature	Maximum Temperature	Minimum Temperature	Wind Speed East Comp.	Wind Speed North Comp.	DMI	Nino 3.4	PDO	SOI	Rainfall (1 month ahead)
Total Rainfall	1.00	0.69	0.70	0.29	-0.73	-0.66	0.64	0.51	0.71	0.38	-0.43	-0.07	0.28	0.11	-0.07	0.57
Avg. Cloud Amount	0.69	1.00	0.77	0.27	-0.79	-0.71	0.73	0.56	0.79	0.37	-0.46	-0.06	0.20	0.00	0.02	0.49
Avg. Dewpoint Temperature	0.70	0.77	1.00	0.37	-0.85	-0.77	0.93	0.74	0.96	0.48	-0.49	-0.10	0.21	0.05	-0.03	0.50
Avg. Humidity	0.29	0.27	0.37	1.00	-0.22	-0.21	0.01	-0.20	0.33	0.29	0.02	-0.09	-0.18	-0.17	0.06	-0.13
Avg. MSL Pressure	-0.73	-0.79	-0.85	-0.22	1.00	0.85	-0.83	-0.70	-0.86	-0.42	0.56	0.13	-0.19	-0.02	-0.03	-0.58
Avg. SL Pressure	-0.66	-0.71	-0.77	-0.21	0.85	1.00	-0.74	-0.62	-0.79	-0.39	0.49	0.12	-0.17	-0.03	-0.02	-0.52
Avg. Temperature	0.64	0.73	0.93	0.01	-0.83	-0.74	1.00	0.88	0.91	0.41	-0.53	-0.07	0.30	0.11	-0.05	0.58
Maximum Temperature	0.51	0.56	0.74	-0.20	-0.70	-0.62	0.88	1.00	0.71	0.28	-0.54	-0.04	0.33	0.08	-0.02	0.57
Minimum Temperature	0.71	0.79	0.96	0.33	-0.86	-0.79	0.91	0.71	1.00	0.45	-0.48	-0.08	0.20	0.06	-0.02	0.51
Wind Speed East Comp.	0.38	0.37	0.48	0.29	-0.42	-0.39	0.41	0.28	0.45	1.00	-0.19	-0.08	0.08	-0.05	0.00	0.25
Wind Speed North Comp.	-0.43	-0.46	-0.49	0.02	0.56	0.49	-0.53	-0.54	-0.48	-0.19	1.00	0.05	-0.22	0.01	0.01	-0.48
DMI	-0.07	-0.06	-0.10	-0.09	0.13	0.12	-0.07	-0.04	-0.08	-0.08	0.05	1.00	0.25	-0.01	-0.22	-0.04
Nino 3.4	0.28	0.20	0.21	-0.18	-0.19	-0.17	0.30	0.33	0.20	0.08	-0.22	0.25	1.00	0.46	-0.64	0.37
PDO	0.11	0.00	0.05	-0.17	-0.02	-0.03	0.11	0.08	0.06	-0.05	0.01	-0.01	0.46	1.00	-0.42	0.12
SOI	-0.07	0.02	-0.03	0.06	-0.03	-0.02	-0.05	-0.02	-0.02	0.00	0.01	-0.22	-0.64	-0.42	1.00	-0.04
Rainfall (1 month ahead)	0.57	0.49	0.50	-0.13	-0.58	-0.52	0.58	0.57	0.51	0.25	-0.48	-0.04	0.37	0.12	-0.04	1.00

Figure 2: Correlation matrices of different variables for (a) Sylhet and (b) Srimangal stations

Then, the timeseries for different variables for a particular station were combined in a table in which each column represents a variable and the rows represent months. In this process, it was found that the

data range for all the variables doesn't match. Hence, data points only within the selected range (for Srimangal station: 1950-01 to 2022-01 and for Sylhet station: 1956-01 to 2022-01) were extracted. Furthermore, since two timeseries variables, namely average sunshine hours and average visibility, had a significantly low number of records, they were removed from the data table. Since there were many missing values in the dataset for the wind speed component variables at Srimangal station and below 10% for all other variables, the missing values were imputed using the month-wise mean method. In this approach, the mean value was computed for a particular month of a specific variable, and whenever any missing value corresponding to that month appeared, it was filled in with that mean value.

Afterwards, the Pearson's correlation coefficient (PCC) statistic was used to develop the correlation matrix for the dataset of each of the stations. PCC was chosen for feature selection as it indicates the intensity of the association between two variables. This quantity can be obtained by dividing the covariance of two variables by the product of their standard deviations. From the correlation matrix for Srimangal station as shown in Figure 2(b), it can be seen that the two teleconnection indices, namely DMI and SOI, had absolute PCC values less than 0.1 with respect to the target variable (one-month ahead rainfall), and since such a value is considered too low to have any kind of association, these two variables were removed from the dataset. Under this criterion, the east component of prevailing wind speed was removed from the dataset along with DMI and SOI for Sylhet station (as shown in Figure 2(a)). In addition, some other variables have such high absolute PCC values (greater than 0.9) among themselves that it indicates that one of the variables in the pair is very likely to be explained by another and is not going to contribute significantly as an independent predictor, and so one of them was dropped. In this consideration, two more variables, namely average cloud amount and average dewpoint temperature, were eliminated, leaving 10 final predictor variables for Sylhet station, while further elimination of two variables, namely average dewpoint temperature and minimum temperature, left 11 predictor variables for Srimangal station.

From this dataset composed of only the predictor variables and the target variable, a small portion (20%) was held out as test data, and the remaining 80% was separated for model development. The numbers of data in the training and testing set were 691 (From 1950-01 to 2007-07) and 173 (From 2007-08 to 2021-12) respectively for Srimangal station and 633 (From 1956-01 to 2008-09) and 159 (From 2008-10 to 2021-12) respectively for Sylhet station. Finally, the data variables were scaled based on the training dataset using Eq. (3), and the same scaling parameters were applied to the test set.

$$x_{i(scaled)} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (3)$$

Here, for a variable x, the scaled value of i_{th} element is $x_{i(scaled)}$ and it is obtained by dividing the difference of i_{th} element (x_i¿ and the minimum value element (x_{min}¿ to the difference between the maximum value element (x_{max}) and the minimum value element (x_{min}¿. Since the LSTM model demands the input dataset to be in a particular shape with the inclusion of a lookback sequence, the dataset was prepared in that three-dimensional format from the existing 2D condition by zipping 12 lookback samples (accounting for the preceding one-year) for each of the inputs.

2.3 LSTM Model

LSTM is a further developed version of the traditional RNN with its inherent nature to utilize the sequential structure of time series data for improved performance (Hochreiter & Schmidhuber, 1997). The feature that makes it distinct from an ordinary RNN is its encompassed structure of cell state that can account for the long-term dependency in the data sequence.

The basic unit of an LSTM layer is the LSTM cell, which updates a total of six parameters within its structure and is composed of three gates namely the forget gate, the input gate, and the output gate. The forget gate determines what portion of the memory information is supposed to be retained from the linear combination of the current cell input (X_t) and the hidden state output from the previous cell (h_{t-1}) and a corresponding bias value with its parameter f_t by applying a sigmoid activation as expressed in equation (4). In this equation and the subsequent ones W represents the corresponding weight vectors and b represents the corresponding biases with their representative subscripted notations. The value for f_t is within the range from 0 to 1 and 0 represents complete elimination while 1 represents complete retention.

$$f_t = \sigma(W_f \cdot [X_t, h_{t-1}] + b_f) \quad (4)$$

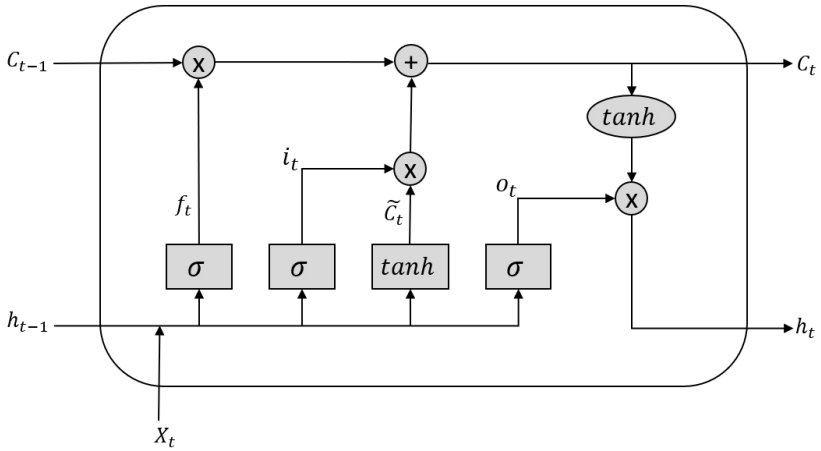


Figure 3: Schematic architecture of an LSTM cell

Then the input gate with its parameter i_t and candidate cell ($\widetilde{C}_t$) both dictate the amassment of new information in the cell state. The computation of i_t involves the sigmoid activation of the linear combination of X_t and h_{t-1} as shown in Eq. (5) while the computation of $\widetilde{C}_t$ involves the hyperbolic tangent activation of the linear combination of X_t and h_{t-1} as in Eq. (6). From these two parameters, the new cell state parameter (C_t) gets updated by summing the product of the previous cell state and the forget parameter and the product of the input parameter and the candidate cell as shown in Eq. (7).

$$i_t = \sigma(W_i \cdot [X_t, h_{t-1}] + b_i) \quad (5)$$

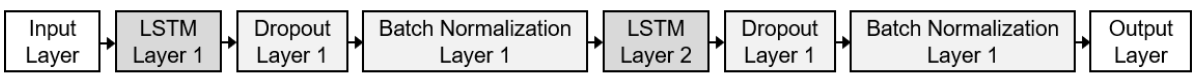
$$\widetilde{C}_t = \tan h(W_C \cdot [X_t, h_{t-1}] + b_C) \quad (6)$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \quad (7)$$

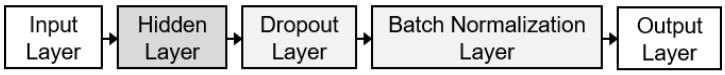
Lastly, the output gate determines the portion of information exits as the output with its parameter o_t. The final hidden state output (h_t) results from the product of o_t and the hyperbolic tangent-activated cell state as in equation (9).

$$o_t = \sigma(W_o \cdot [X_t, h_{t-1}] + b_o) \quad (8)$$

$$h_t = o_t \times \tanh(C_t) \quad (9)$$



(a)



(b)

Figure 4: Implemented model architectures: (a) LSTM (b) MLP

An LSTM layer typically consists of a series of such LSTM cells in which each of the cells takes the cell state and hidden state from the previous cell, if available, along with the actual input. In the LSTM model used in this study, two such LSTM layers, followed by the input layer, were used. The first layer of these two returns the last output in the sequence, and the second one returns the total sequence. Each of these LSTM layers is then followed by a batch normalization layer and a dropout layer, respectively. The inclusion of batch normalization is due to its imparted advantages, such as the improvement of test accuracy resulting from the effect of regularization and the steadier propagation of gradients. On the other hand, the concept of the dropout layer involves the dropping of some random units and their corresponding connections in an attempt to reduce the overfitting phenomenon (Srivastava et al., 2014). The dropout fraction of the model was set to 0.50 for the LSTM-to-LSTM layer connection and 0.20 for the LSTM to output layer connection. Altogether, the overall model structure can be represented by Figure 4(a). The implementation of this model will be performed using the Keras API in the TensorFlow environment of Python.

2.4 ANN Model

The imitation of the mechanism of transmitting and processing information through a complex network in a typical human brain led to the development of an artificial neural network (ANN) and such naming. The building block of such a computational tool is a neuron that simply sums the weighted products of the inputs with an additional bias value. To introduce nonlinearity in an otherwise linear structure, the neurons are followed by activation functions in a network. In a basic multilayer perceptron (MLP) model, a hidden layer comprises such neurons and is put between an input and an output layer. The neurons of a layer are interconnected with those of the preceding and following layers with weighted connections. This study uses an MLP model with one hidden layer (Figure 4(b)) and the hyperparameters obtained through the random search algorithm are the number of hidden neurons, activation function, batch size, and learning rate. This model also uses the same environment as that of the LSTM model and includes dropout regularization and batch normalization.

2.5 Activation Functions

The nonlinear activation function serves the useful purpose of imparting the capability of capturing the nonlinearity of the modeled function in neural networks. The sigmoid or logistic function (σ) with its well-known S-curved shape maps the input values in the range from 0 to 1 by using Eq. (10). Contrarily, although the hyperbolic tangent activation function ($\tanh$) as depicted in Eq. (11) also has an S-shape, it maps the inputs in the range from -1 to 1 as the output. Apart from them, the rectified linear unit ($relu$) as shown in Eq. (12) maps all the negative inputs to 0 but keeps the positive values as it is.

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{10}$$

$$\tanh(x)=\frac{(e^{x}-e^{-x})}{(e^{x}+e^{-x})} \tag{11}$$

$$relu(x)=max(0,x) \tag{12}$$

2.6 Adam Optimization

The adaptive moment estimation (Adam) optimization algorithm presented by Kingma & Ba (2015) is a modern alternative to the stochastic gradient descent algorithm. This algorithm combines the benefits of both the moment-based optimization and root mean square propagation (RMSProp) algorithm. The momentum update imparts speed to the optimization by increasing the learning rate if

the gradient matches the direction of the previous ones, while updating the learning rate based on steepness provides adaptability. The optimization parameters such as β_1, β_2, and ϵ have been used as the defaults given in the corresponding Keras class, and the only parameter tuned is the learning rate. This optimization was used for training both the LSTM and ANN models in this study. The epoch and early stopping were set at 2000 and 50, respectively.

2.7 SVR Model

With the necessary adjustments in the typical algorithm of a support vector machine, it can be successfully equipped with the ability to perform regressions, and such a model is called a support vector regression model. The three kernels explored in the grid search are the radial basis function (RBF), the polynomial kernel of third degree, and the sigmoid kernel. The parameter C controls the error margin in such a way that larger C values ensure a smaller error margin and vice versa. The gamma parameter is specifically needed for the RBF kernel as it defines the extent of curvature of the boundary. The 'scikit-learn' package in Python was used to employ this model.

2.8 Model Evaluation Metrics

A wide variety of evaluation metrics has been used for comparing the performances of the models in various papers dealing with hydrologic prediction problems (Latif et al., 2023). This paper uses such five metrics given in Eqs. (13)-(17), namely mean squared error (MSE), root mean squared error ($RMSE$), mean absolute error (MAE), Nash-Sutcliffe efficiency (NSE), and coefficient of determination (R^2) due to their wide acceptance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{ai} - y_{pi}\right)^2 \quad (13)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{ai} - y_{pi}\right)^2} \quad (14)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{ai} - y_{pi}\right| \quad (15)$$

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(y_{ai} - y_{pi}\right)^2}{\sum_{i=1}^{n}\left(y_{ai} - \overline{y_a}\right)^2} \quad (16)$$

$$R^2 = ¿¿ \quad (17)$$

Where, the actual and predicted data points are represented with y_{ai} and y_{pi} respectively, $\overline{y_a}$ and $\overline{y_p}$ are the mean of the actual and predicted data series respectively, and there are n number of elements in a data series.

3. RESULTS AND DISCUSSION

From the 300 tuning runs for the LSTM and MLP models and the 3000 runs for the SVR model, the random grid search operation with 4-fold cross-validation selected the best models based on the minimum MSE values, and the hyperparameters are listed in Table 1. The optimal hyperparameters are completely different for the same model at differing stations. For example, the optimum batch size chosen for the LSTM model in Srimangal is 16, while it is 64 for the Sylhet station. On the other side, the chosen number of LSTM units in each of the two LSTM layers is 8 for the LSTM model in Srimangal, and this number doubles to 16 for the Sylhet station. All these optimum hyperparameters were used along with the preselected set of model parameters, as mentioned before, to build each of the final models.

Table 1: Selected hyperparameters for different models

Station	Model	Selected Hyperparameters
Srimangal	LSTM	Batch size: 16, Learning rate: 0.009099489329724531, Number of LSTM units: 8
	MLP	Activation function: tanh, Batch size: 32, Learning rate: 0.020612697, Hidden units: 4
	SVR	C: 0.3988186931209492, gamma: 0.7358779785637147, Kernel: RBF
Sylhet	LSTM	Batch size: 64, Learning rate: 0.025913875649481007, Number of LSTM units: 16
	MLP	Activation function: relu, Batch size: 64, Learning rate: 0.002057143, Hidden units: 32
	SVR	C: 2.1546248509014165, gamma: 0.8781346547028155, Kernel: RBF

In order to get a visual overview of how the model prediction goes with the actual timeseries, the timeseries plots of three models are plotted alongside the observed timeseries for each of the two stations in Figure 5. The deviations of the predicted time series from the actual one create a visually perceptible notion of the predictive performance. The variety of paths chosen by different models reflects their inherent differences. Although various models predict with variable accuracy, none of them can be stated as successful in predicting extreme peaks, as the prediction line is spotted significantly below that of the actual one as visible from the figure.

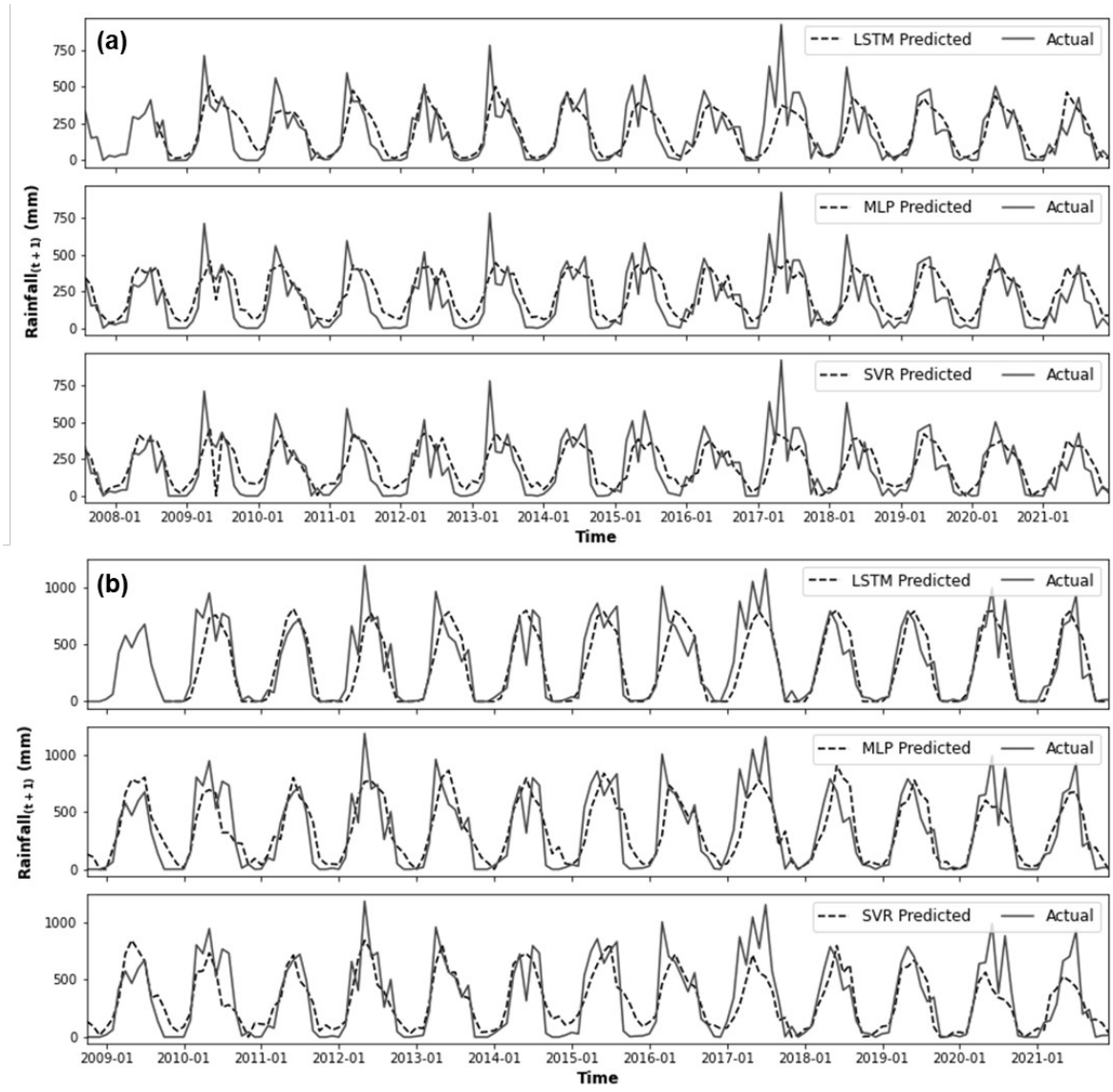


Figure 5: Actual and predicted rainfall for (a) Srimangal station and (b) Sylhet station using LSTM, MLP and SVR models in the testing phase

Furthermore, the scatterplots of predicted and observed one-month ahead monthly rainfall as shown in Figures 6 and 7 not only provide a way of comparison for training and testing but also a way to learn about differences from model to model. All the scatterplots establish that the prediction trendline is always slanted with a slope less than the perfect prediction condition, be it for training or testing. And part of this phenomenon can be attributed to the extreme data points that are underpredicted by quite a margin from the actual values, as confirmed by the time series plots. Again, the patterns of the data points from the training to testing do not disagree that largely indicate any overfitting or underfitting. Moving from the visual intuition to the numeric, Table 2 lists all five-evaluation metrics computed on the training and testing datasets for two stations. For the two error measures, MAE and RMSE, the values range from 73.85 mm to 139.93 mm and from 109.92 mm to 194.79 mm, respectively. The values of the training and testing metrics for the same model are within reach since MAE differs by a maximum of 12.7 mm and RMSE by 18.02 mm. Similarly, for the fitness measures NSE and R^2, the values range from 0.48 to 0.75 and from 0.53 to 0.76, respectively.

Table 2: Evaluation metrics for different rainfall prediction models

	Metric	Srimangal			Sylhet		
		LSTM	**MLP**	**SVR**	**LSTM**	**MLP**	**SVR**
Training	MAE (mm)	73.85	94.85	89.65	110.25	130.41	128.50
	MSE (mm^2)	12081.48	16884.54	15759.53	27893.64	34478.13	31270.02
	RMSE (mm)	109.92	129.94	125.54	167.01	185.68	176.83
	NSE	0.68731	0.55877	0.58817	0.75445	0.69781	0.72593
	R^2	0.68914	0.57908	0.60436	0.76018	0.69912	0.73096
Testing	MAE (mm)	86.55	104.42	97.45	115.06	139.93	139.48
	MSE (mm^2)	16369.61	18712.14	17101.73	32217.54	37942.87	35955.51
	RMSE (mm)	127.94	136.79	130.77	179.49	194.79	189.62
	NSE	0.56166	0.48243	0.52697	0.70963	0.64775	0.6662
	R^2	0.56229	0.52621	0.53531	0.72106	0.64865	0.67567

As can also be seen from Figure 5, LSTM predictions for both stations very closely follow the actual path for very low values at troughs and medium values with few exceptions. In opposition, the MLP and SVR models overpredict the troughs and are spread around the actual path by a visible left or rightward shift for moderate values. In addition, since the data points for the Srimangal station got somewhat stuck in the lower half of the scatterplots and a comparatively balanced pattern was obtained for the Sylhet station (Figure 6), the latter is supposed to yield better metric values. This shortcoming of the models for the Srimangal station can be attributed to the dataset itself, as the data points were mostly concentrated at lower values and there were a small number of training data points for learning extreme values.

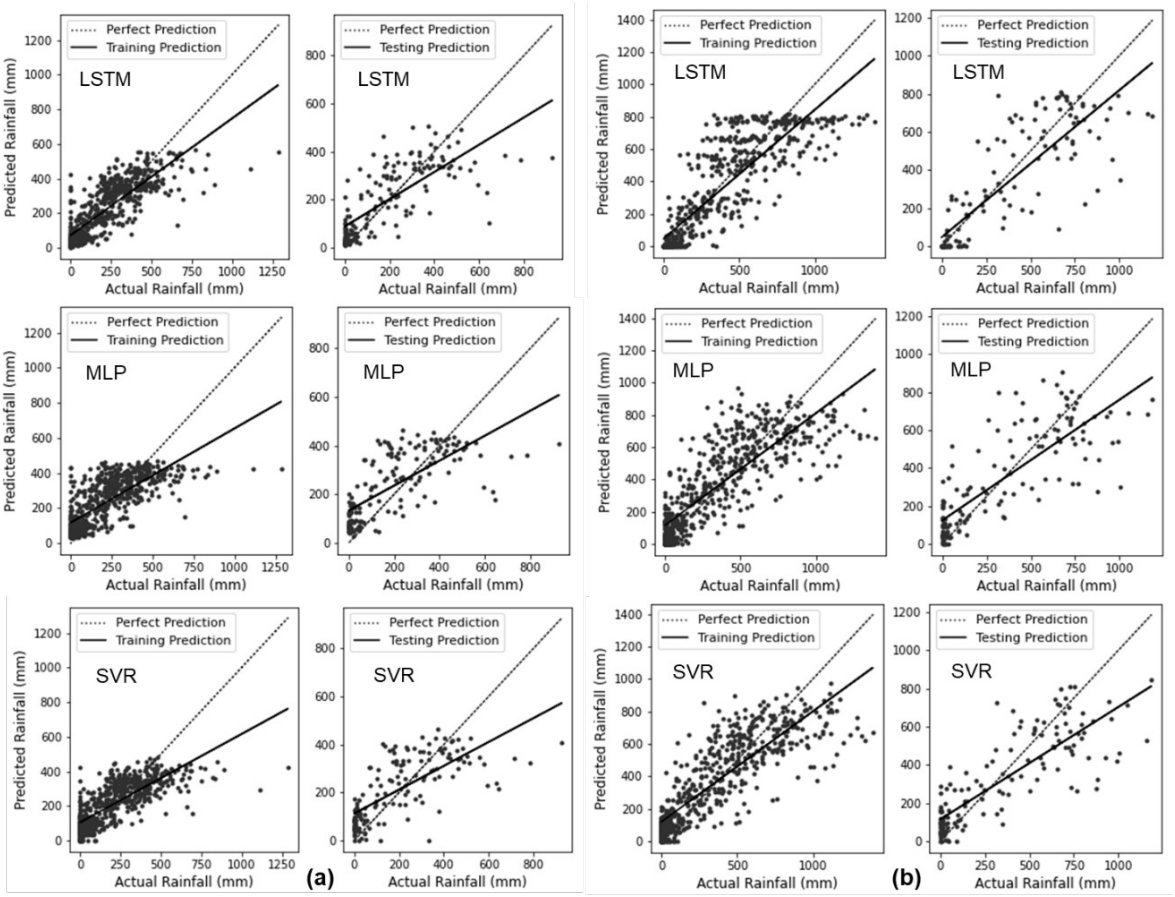


Figure 6: Actual and predicted rainfall scatterplots for (a) Srimangal station and (b) Sylhet station using LSTM, MLP and SVR models in the training and testing phases

As can be seen from Table 2, the least value for each of the three error measures in both stations was obtained for the LSTM models both on training and testing data. The LSTM model reduced the testing MAE by 17.11% and 11.19% for the Srimangal station and by 17.77% and 17.51% for the Sylhet station, respectively, compared to the MLP and SVR models. The improvement in RMSE values by the LSTM model follows a similar trend, with decreases of 5% and 0.63% for the Srimangal station and 7.85% and 5.34% for the Sylhet station, respectively. Likewise, the LSTM model retains its superiority in performance by showing the maximum values of the two fitness measures, namely NSE and R^2. In Srimangal station, the NSE and R^2 values are boosted by 16.42% and 6.86% with LSTM than that of MLP and by 6.58% and 5.04% with SVR. Similarly, in Sylhet station, the LSTM model increases the NSE value by 9.55% compared to that of MLP and by 6.52% compared to that of SVR. In addition, the R^2 values rise by 11.16% and 6.72% for LSTM compared to MLP and SVR, respectively, in this station.

Since LSTM models produce more worthy values for both error and fitness measures, it can be unequivocally dictated that the best model outperforms the other two competitor models. Moreover, since the improvements effected by LSTM are of lesser magnitude for SVR than MLP, SVR is found to be the closest competitor, being the second-best model. Comparing the performance metrics for the two stations, it is evident from Table 2 that the models generate comparatively lower error values for the Srimangal station than for the Sylhet station. But a surprisingly similar case is apparent for the fitness measures as well, as we get to see lower values of *NSE* and R^2 for Srimangal station. Thus, for any of the two stations, the comparatively desirable values of either error or fitness measures are obtained, and no station proves to show better performance in terms of all evaluation measures.

4. CONCLUSIONS

The current study implements a long short-term memory (LSTM) network-based deep learning approach for the prediction of rainfall in two rainfall stations in the northeastern region of Bangladesh and compares its performance with two other proven machine learning models, namely MLP and SVR. A multivariate approach using the combination of hydrometeorological variables and climate indices as predictors has been adopted here after a systematic feature selection process. The key conclusions found from this study are:

- The LSTM model captures the rainfall values at low and medium values more satisfactorily than the SVR and MLP models. However, none of the models can predict extreme rainfall values with reasonable accuracy.
- The LSTM models suggested for the two stations well surpass the performance exhibited by the other two models according to all five model evaluation criteria.
- The models perform better on Srimangal station data based on error measures, while they perform better on Sylhet station data in terms of fitness measures.
- Since the values of evaluation metrics follow the same trend for both stations, the results of this study may be extendable to other stations located in northeastern Bangladesh.

It is worth mentioning that there is still the opportunity to try various other combinations of predictor variables in the current study and the univariate approach to achieve further improved performance, especially to capture the extreme values and simulate similar studies in other regions of Bangladesh.

REFERENCES

Barrera-Animas, A.Y., Oyedele, L.O., Bilal, M., Akinosho, T.D., Delgado, J.M.D., & Akanbi, L.A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, 100204. https://doi.org/10.1016/j.mlwa.2021.100204

Billah, M., Adnan, M.N., Akhond, M.R., Ema, R. R., Hossain, M.A., & Md. Galib, S. (2022). Rainfall prediction system for Bangladesh using long short-term memory. *Open Computer Science*, *12*(1), 323–331. https://doi.org/10.1515/comp-2022-0254

Brouwer, R., Akter, S., Brander, L., & Haque, E. (2007). Socioeconomic Vulnerability and Adaptation to Environmental Risk: A Case Study of Climate Change and Flooding in Bangladesh. *Risk Analysis*, *27*(2), 313–326. https://doi.org/10.1111/j.1539-6924.2007.00884.x

Farukh, M.A., Rahman, M.A., Sarker, S., & Islam, M.A. (2020). Impact of Extreme Precipitation Intensity on Tea Production in the North-East of Bangladesh. *American Journal of Climate Change*, *09*(04), 441–453. https://doi.org/10.4236/ajcc.2020.94028

Haidar, A., & Verma, B. (2018). Monthly Rainfall Forecasting Using One-Dimensional Deep Convolutional Neural Network. *IEEE Access*, *6*, 69053–69063. https://doi.org/10.1109/ACCESS.2018.2880044

Hasan, G.J., Alam, R., Islam, Q.N., & Hossain, M. S. (2012). Frequency structure of major rainfall events in the north-eastern part of Bangladesh. *Journal of Engineering Science and Technology*, *7*(6), 690–700.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hossain, I., Rasel, H.M., 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*, *132*(1), 131–141. https://doi.org/10.1007/s00703-019-00679-4

Islam, M.S., Okubo, K., Islam, A.H.M.S., & Sato, M. (2022). Investigating the effect of climate change on food loss and food security in Bangladesh. *SN Business & Economics*, *2*(1), 4. https://doi.org/10.1007/s43546-021-00177-z

Iseri, Y., Dandy, G.C., Maier, H.R., Kawamura, A., Jinno, K., (2005). Medium term forecasting of rainfall using artificial neural networks. In Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation, Modelling and Simulation Society of

Australia and New Zealand, December 2005, pp. 1834-1840. ISBN: 0-9758400-2-9.

Jakariya, M., & Islam, M.N. (2017). Evaluation of climate change induced vulnerability and adaptation strategies at Haor areas in Bangladesh by integrating GIS and DIVA model. *Modeling Earth Systems and Environment*, *3*(4), 1303–1321. https://doi.org/10.1007/s40808-017-0378-9

Kang, J., Wang, H., Yuan, F., Wang, Z., Huang, J., & Qiu, T. (2020). Prediction of precipitation based on recurrent neural networks in jingdezhen, jiangxi province, China. *Atmosphere*, *11*(3), 1–17. https://doi.org/10.3390/atmos11030246

Karim, N. (1995). Disasters in Bangladesh. *Natural Hazards*, *11*(3), 247–258. https://doi.org/10.1007/BF00613409

Kingma, D.P., & Ba, J.L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15. https://doi.org/https://doi.org/10.48550/arXiv.1412.6980

Latif, S.D., Alyaa Binti Hazrin, N., Hoon Koo, C., Lin Ng, J., Chaplot, B., Feng Huang, Y., El-Shafie, A., & Ahmed, A.N. (2023). Assessing rainfall prediction models: Exploring the advantages of machine learning and remote sensing approaches. *Alexandria Engineering Journal*, *82*, 16–25. https://doi.org/10.1016/j.aej.2023.09.060

Poornima, S., & Pushpalatha, M. (2019). Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units. *Atmosphere*, *10*(11), 668. https://doi.org/10.3390/atmos10110668

Rahman, M.M. (2022). Effect of Rainfall Pattern on the Tea Production in Bangladesh: An Analysis of Socio-economic Perspectives. *Journal of Agroforestry and Environment*, *15*(1), 43–55. https://doi.org/10.55706/jae1507

Sahai, A.K., Soman, M.K., & Satyan, V. (2000). All India summer monsoon rainfall prediction using an artificial neural network. *Climate Dynamics*, *16*(4), 291–302. https://doi.org/10.1007/s003820050328

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.