International Journal of Statistical Sciences *ISSN 1683-5603*
Vol. 25(1), March, 2025, pp 39-54
DOI: https://doi.org/10.3329/ijss.v25i1.81044

Performance of Deep Learning Algorithms to Predict the Monthly Rainfall Data of Rajshahi District, Bangladesh

Md. Asaduzzaman Khondoker[1], Md. Mostafizur Rahman[1*], Sojaul Islam[1], Md. Abdur Rahman[1], Md. Abdul Khalek[1] and M. Sayedur Rahman[1]

[1]Data Mining and Environment Research Group, Department of Statistics,
University of Rajshahi, Rajshahi-6205, Bangladesh
*Correspondence should be addressed to Md. Mostafizur Rahman
(Email: mostafiz_bd21@yahoo.com)

[Received November 05, 2024; Accepted March 1, 2025]

Abstract

The economic development of Bangladesh highly depends on agriculture production and rainfall is one of the most influential factors. A number of variables, including temperature, relative humidity, wind direction, wind speed, and cloud cover, influence the likelihood of rainfall. There is currently a deficiency in the ability to accurately and precisely predict rainfall, which would be beneficial in a variety of industries, including flood prediction, water conservation, and agriculture. Recently, machine learning algorithms showed better performance for predicting climatic variables than tradition models. Using deep learning algorithms to forecast rainfall is an innovative method that makes use of sophisticated computer methods to examine complex patterns in meteorological data. So, in this paper we compare the forecasting performance of deep learning algorithms and machine learning algorithms in case of Rajshahi district in Bangladesh. The historical data from January 1964 to December 2017 is considered for study. The empirical results suggest that, for the subsequent timeframes, the deep learning algorithms MLP is the most suitable algorithm for forecasting the monthly rainfall data of this study area.

Keywords: Rainfall, Climatic variables, Machine Learning algorithms, deep learning algorithms, Rajshahi.

AMS Classification: 68T05, 62P12.

1. Introduction

The main driver of Bangladesh's economy is its agricultural output. Rainfall is one of the most important factors in this manufacturing. Rainfall is a type of precipitation where water falls on the surface, in the oceans, and in the atmosphere of the Earth. The condensed water smoke rises into the atmosphere and causes rain to fall. A number of variables, including temperature, relative humidity, wind direction, wind speed, and cloud cover, influence the likelihood of rainfall. Deep Learning is a technique for working with data to manipulate and retrieve implicit, known, and potentially relevant information. It includes many supervised and unsupervised learning classifiers that are employed to forecast and pinpoint the exact forecasting model for the provided data set. Using deep learning algorithms to predict rainfall is an innovative method that makes use of sophisticated computer methods to examine complex patterns in meteorological data. By improving rainfall predicting accuracy, this novel approach has the potential to transform our knowledge of precipitation dynamics and provide important new information for climate resilience and efficient catastrophe management. One major development in meteorological research is the use of machine learning algorithms to predict rainfall. This work intends to improve rainfall forecast accuracy by utilizing data-driven models. This would provide insightful information about environmental conditions and support proactive decision-making for climate resilience and catastrophe preparedness.

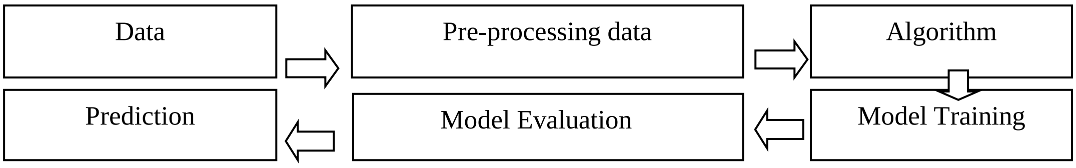
Zainudin et al. (2016) examined the performance of Naive Bayes, Support Vector Machine, Decision Tree, Neural Network and Random Forest in case of Malaysian rainfall data and found that the Random Forest model gives better result. Solanki and G.P.B. (2018) proposed Hybrid Intelligent System by integrating Artificial Neural Network and Genetic Algorithm and found the better performance of their hybrid model than other existing model for prediction. Mishra et al. (2018) created one-month and two-month forecasting models for rainfall prediction using monthly rainfall data spanning 141 years from various meteorological stations in the north area of India. Bagirov et al. (2017) compared the performance of multiple linear regression, artificial neural networks and support vector machines with their proposed Cluster wise Linear Regression (CLR) method and found that their proposed model performed better. Basha et al. (2016) used Deep learning algorithms such as MLP and Auto-Encoder NN for predicting rainfall data. In this study, the CNN algorithm was used for taking input from past data. Performance of these techniques evaluated using MSE and RMSE. To predict the maximum temperature, rainfall, evaporation and wind speed for Nigeria Olaiya and Adeyemo (2012) compared the performance of ANN model and decision tree algorithms and found that ANN model give better result. Haidar et al. (2018) developed a monthly rainfall prediction model. A deep convolution neural network (CNN) was used for prediction found the preference of their proposed algorithm for predicting rainfall data. Along with these authors, additional authors Talib et. al. (2017), Tharun et al. (2018), and others examined the effectiveness of several deep learning and machine learning algorithms for predicting rainfall for certain cities or areas. Besides these several authors examined the performance of different deep learning and machine learning algorithms for predicting rainfall data for example, Aftab et al (2018), Sivapragasam et al (2001), Monira et al (2010), Kannan et al. (2010), Sethi and Garg (2014), Gupta and Ghose (2015) etc.

From the above discussion we found that researchers were consistently working to predict rainfall with maximum accuracy by optimizing and integrating different deep learning and machine learning techniques for different weather station and found different models showing better performance for different weather stations. Predicting rainfall by deep learning algorithm for Rajshahi district is rare. So, the aim of this paper is to compare the forecasting performance of deep learning algorithms with several machine learning algorithms and to find out the most accurate rainfall forecasting model in case of economically important districts Rajshahi, Bangladesh. This study will be helpful to give information to support crop, water and flood management which will save lives and properties of the people and contribute for the development of the economy.

2. Methods and Materials

2.1 Methodology

The objective of this investigation is to assess the effectiveness of deep learning algorithms in contrast to various machine learning algorithms in forecasting the rainfall data of Rajshahi, Bangladesh. Consequently, in this section, we elaborate on the traditional framework, elucidate the data collection process, delineate data pre-processing techniques, expound on the modeling phases, and elucidate the evaluation phase. The working framework is given below:



2.2 Predictive Models

In this study, we utilize commonly used deep learning algorithms like Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP), along with other machine learning algorithms such as the Gradient Boosting method, k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT) for predicting monthly rainfall data in case of Rajshahi district in Bangladesh. These models are described below.

2.2.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a particularly useful deep learning technique for processing sequential data and predictions. Long short-term memory (LSTM) is a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs in pattern recognition and extraction from sequences. LSTM networks outperform ordinary RNNs in tasks like time series forecasting and natural language processing due to the memory cells' long-term capacity to store and retrieve data.

An input gate, an output gate, a forget gate, and a cell are all included in an LSTM unit, which together control the information flow in the network. The gates decide what to remember and what to discard, controlling the flow of information and improving the model's capacity to identify dependencies in sequential data. The cell stores long-term memory. Because of its exceptional ability to learn complex patterns and maintain context, the LSTM architecture is especially well-suited for applications like weather forecasting, language translation, and stock price prediction.

The following is a summary of how the LSTM (Long Short-Term Memory) algorithm operates:

- The LSTM unit's memory cells and gates must be configured during initialization.
- The network computes the input, forget, and output gates for each input in the sequence, figuring out what data to store, discard, or output.
- The memory cell is updated in accordance with the computed gates, enabling the network to gather and store pertinent data over lengthy sequences.
- The last result is generated, which either indicates the outcome of the sequence or the prediction.

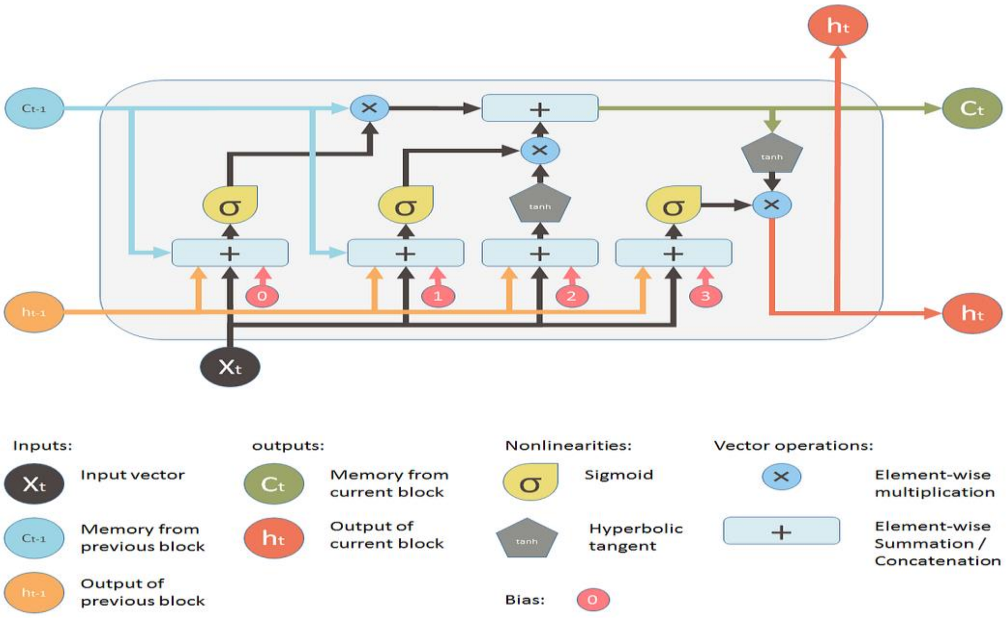


Fig. 1: Diagram for LSTM (Source: Shi, 2016)

2.2.2 Multi-Layer Perceptron (MLP)

A flexible artificial neural network architecture, the Multilayer Perceptron (MLP) has many benefits. Like LSTM, MLP is an effective tool across a range of areas because it performs well in challenging pattern recognition tasks. Capturing complex linkages and dependencies within data is one of its main advantages. To improve the stability of the training process, MLP is a good choice for problems such as vanishing and exploding gradients. It works well in a variety of domains and may be applied to both sequential and time-series data. Since MLP is so versatile and effective in a variety of contexts, it is widely used in applications including speech recognition, natural language processing, and weather forecasting.

Regression and classification are two of the many tasks that MLP excels at. During training, MLP adjusts its internal weights to reduce the discrepancy between expected and actual results through a procedure called back-propagation. MLP is capable of adjusting to the nuances of various datasets thanks to its iterative learning process. MLP can learn from data and generalize to new contexts, it is a valuable tool in the field of deep learning.

- The MLP unit's neurons and connections are configured at initialization.
- The network determines what data to pass through or discard for each input by computing the weighted total, applying the activation function, and producing the output.
- The neural network is updated based on the computed weights and biases, enabling it to capture and learn nuanced patterns over extended sequences.
- The final output, which represents the prediction or the subsequent step in the sequence, is generated.

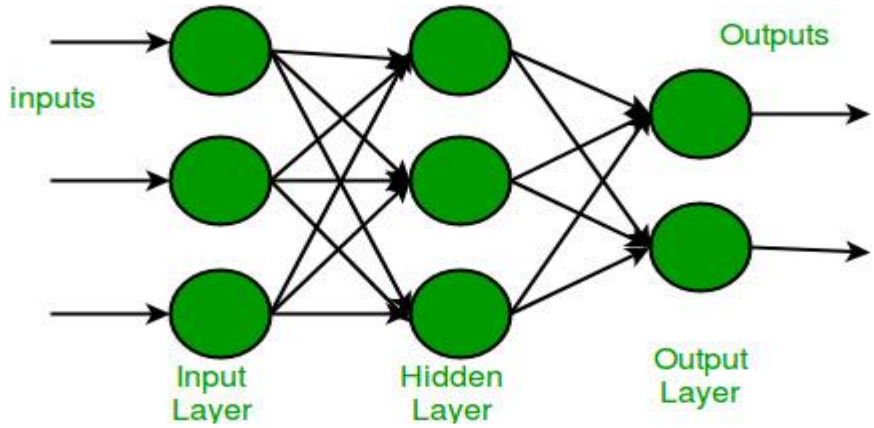


Fig. 2: Diagram for MLP (Multi-Layer Perceptron) (Source: GeeksforGeeks)

2.2.3 Gradient Boosting Method

The gradient boosting strategy is a powerful machine learning technique for both regression and classification applications. It works by combining the predictions of many weak learners—usually decision trees—to create a strong predictive model. The basic idea is to build trees one after the other, correcting the flaws of the ones that came before. Using the residuals from the previous model, a new tree is trained each time, and its predictions are contributed to the ensemble as a whole. It is extensively utilized in several fields, including as natural language processing, finance, and healthcare. The operational procedure of the Gradient Boosting method algorithm is outlined as follows:

- Initialization entails configuring the base learner within the Gradient Boosting framework.
- The algorithm computes the gradient of the loss function for the predictions made by the current model at each iteration of the boosting process.
- The negative gradient is fitted using a weak learner, typically a decision tree, to highlight the areas where the current model performs badly.
- The predictions of the new weak learner are scaled by a learning rate and included in the model ensemble.
- Every new model in the iterative process corrects the errors of the entire ensemble.
- The total predictions made by all models result in the final output, which is the enhanced and improved forecast for the given task.

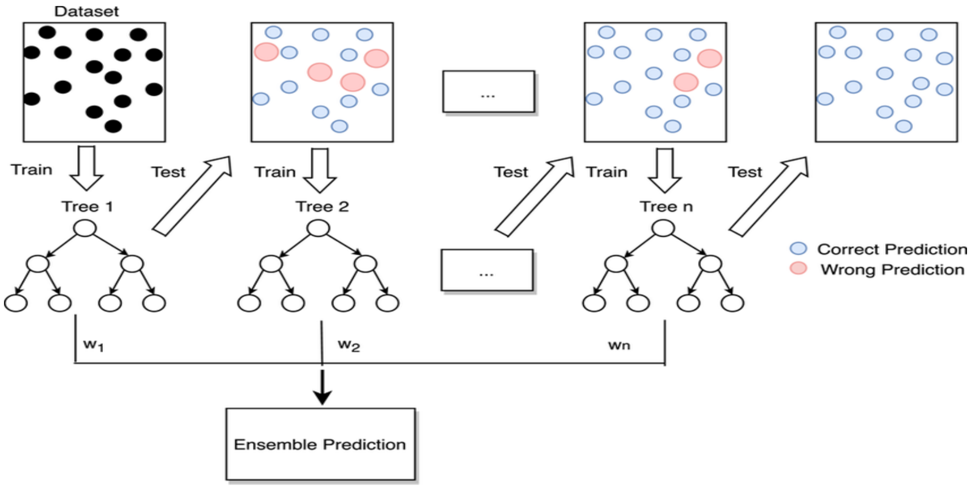


Fig. 3: Diagram for Gradient Boosting method (Source: Zhang, 2021)

2.2.4 Support Vector Machine (SVM)

A popular supervised learning method for regression analysis and classification is Support Vector Machine (SVM). It was first introduced by Vapnik and associates (Boser et al., 1992, Vapnik, 1995). The development of support vector machines in 1990 was a major step forward in the search for a trustworthy algorithm. The approach of support vector machines is based on finding an N-dimensional hyperplane in the space that properly categorizes the data points. A nice illustration of a support vector machine may be seen in Figure 3.4.

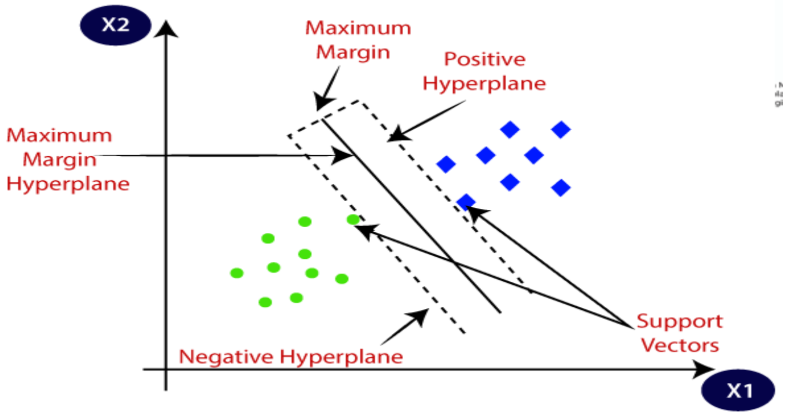


Fig. 4: Diagram for Support Vector Machine (Source: Gandhi, 2018)

It is possible to split the two classes of data points using many hyperplanes. The aim is to find the plane with the broadest margin, which represents the biggest difference between data points from both classes. It is necessary to optimize the margin distance in order to increase the confidence in the categorization of upcoming data points.

2.2.5 K-Nearest Neighbors (K-NN)

Fix and Joseph (1951) developed the K-Nearest Neighbors (K-NN) algorithm, which was further refined by Thomas and Peter (1967). This supervised learning system uses the training dataset directly for prediction. To generate predictions for a new instance (x), the algorithm searches the whole training set for the K examples (neighbors) that are the most similar, then summaries the output variable for those K instances. This might be the mean output variable in regression and the modal (most common) class value in classification.

To determine which of the K instances in the training dataset is closest to a new input, a distance metric is used, most often the Euclidean distance for input variables with real values. K's value is usually set to be odd, however it can be found by adjusting the procedure. K-NN's computational complexity increases as the training dataset's size does. To make K-NN stochastic for very large training sets, one can take a sample from the training dataset and use it to find the K-most similar occurrences. The functioning of K-NN is described by the following algorithm:

- Step 1: Find out the neighbors' K-numbers.
- Step 2: In step two, get the Euclidean distance between K neighbors.
- Step 3: Using the computed Euclidean distance, select the K closest neighbors.
- Step 4: Find out how many data points in each category each of these k neighbors has.
- Step 5: Assign the recently gathered data points to the group with the greatest number of neighbors.
- Step 6: Our model is complete.

First, we will choose k=5 as the number of neighbors.

Next, we'll calculate the Euclidean distance between each data point. The Euclidean distance is the distance between two points, which we have already looked at in geometry.

The Euclidean distance was computed to find the nearest neighbors. In category A, there were three closest neighbors, while in category B, there were two closest neighbors. Consider the following picture:

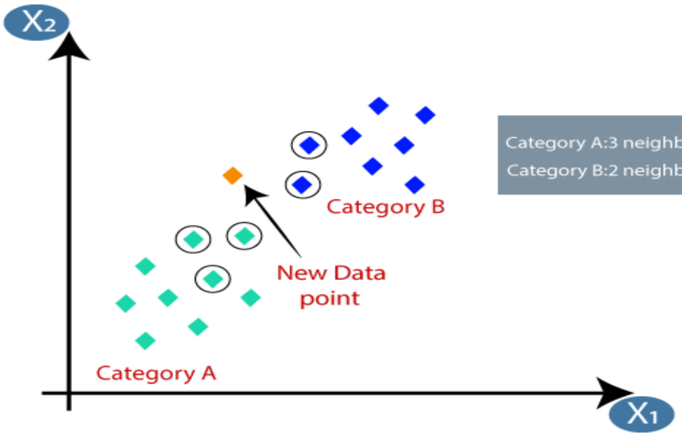


Fig. 5: Selection of new data point from A and B

As we can see, the three nearest neighbors of this new data point are likewise from group A, hence it must also belong to that category.

2.2.6 Decision Tree

The Decision Tree algorithm is a well-liked supervised learning method for applications involving regression and classification. The model is arranged in a tree-like fashion, wherein every internal node symbolizes a test conducted on a certain characteristic, every branch signifying the test's outcome, and every leaf node holding the ultimate decision or outcome. This method operates by iteratively splitting the data into subgroups based on the most significant characteristic at each step. The algorithm, which starts at the root node, determines which characteristic best splits the data, using metrics like mean squared error for regression and Gini impurity for classification. Until specific stopping conditions are satisfied, such as a predefined tree depth or a minimum number of samples in a leaf node, the recursive partitioning process continues. The resulting Decision Tree offers a simple and easily interpreted model that can handle categorical and numerical data; nevertheless, techniques such as pruning are used to avoid overfitting, especially in deeper trees. Decision Tree is illustrated in the picture below:

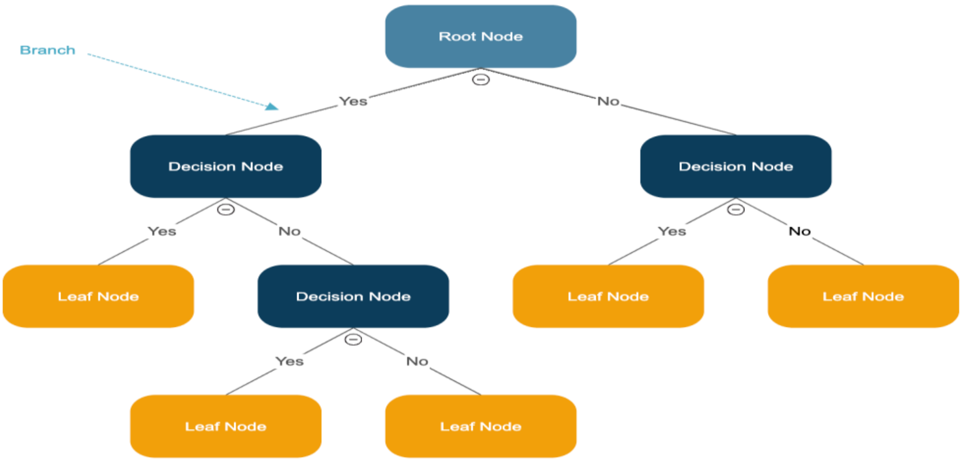


Fig. 6 Decision Tree model

2.3 Model Evaluation

Evaluating a model's performance is crucial to machine learning and deep learning algorithms. The degree of future prediction accuracy of a model may be assessed using a variety of metrics. For different machine learning and deep learning techniques, different assessment criteria could be required. The four metrics that are usually used to evaluate predicting abilities are the Nash-Sutcliffe model efficiency (NSE), R-square, Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). The percentage of all projections that were accurate is represented by overall accuracy, which is also taken into account (Hyndman and Athanasopoulos, (2018)).

2.3.1 Root Mean Squared Error (RMSE)

Regression analysis commonly uses the Root Mean Squared Error (RMSE) metric to calculate the average magnitude of the differences between actual and projected values. The square root of the average of the squared discrepancies between the expected and actual numbers is used to calculate it. The formula for RMSE is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$$

Where, n is the number of data points, y_i is the actual value, and $\hat{y_i}$ is the predicted value.

2.3.2 Mean Absolute Percentage Error (MAPE)

A statistic called Mean Absolute Percentage Error (MAPE), which is especially useful in time series analysis, is used to evaluate how accurate a forecasting or prediction model is. The percentage difference between the expected and actual numbers is measured.

The formula for MAPE is given by:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{(y_i - \hat{y_i})}{y_i} \right| * 100$$

Where, n is the number of data points, y_i is the actual value, and $\hat{y_i}$ is the predicted value.

2.3.3 Nash–Sutcliffe model efficiency (NSE)

A statistical metric called the Nash-Sutcliffe Efficiency (NSE) is frequently used to assess how well environmental or hydrological models perform. It offers an evaluation of the model's capacity to reproduce the data's observed variability. The NSE is especially well-liked in the hydrological community. The formula for NSE is given by:

$$\text{NSE} = 1 - \frac{\sum_{t=1}^{T}(O_t - M_t)^2}{\sum_{t=1}^{T}(O_t - \bar{O})^2}$$

Where, O_t is the observed value at time t, M_t is the modeled (predicted) value at time t, $\bar{O}$ is the mean of the observed values.

The NSE is between $-\infty$ and 1. A perfect fit between the modeled and observed values is indicated by a value of 1, whilst a poorer fit is suggested by lower values. Negative values suggest that the observed values' mean performs better as a predictor than the model.

2.3.4 R^2 (R-square)

A regression model's independent variables (features) can account for a percentage of the variation in the dependent variable (goal), which is measured statistically by a metric called the coefficient of determination, or R^2(R-squared). It is often used in regression analysis to evaluate the model's fit to the observed data. The formula for R^2 is given by:

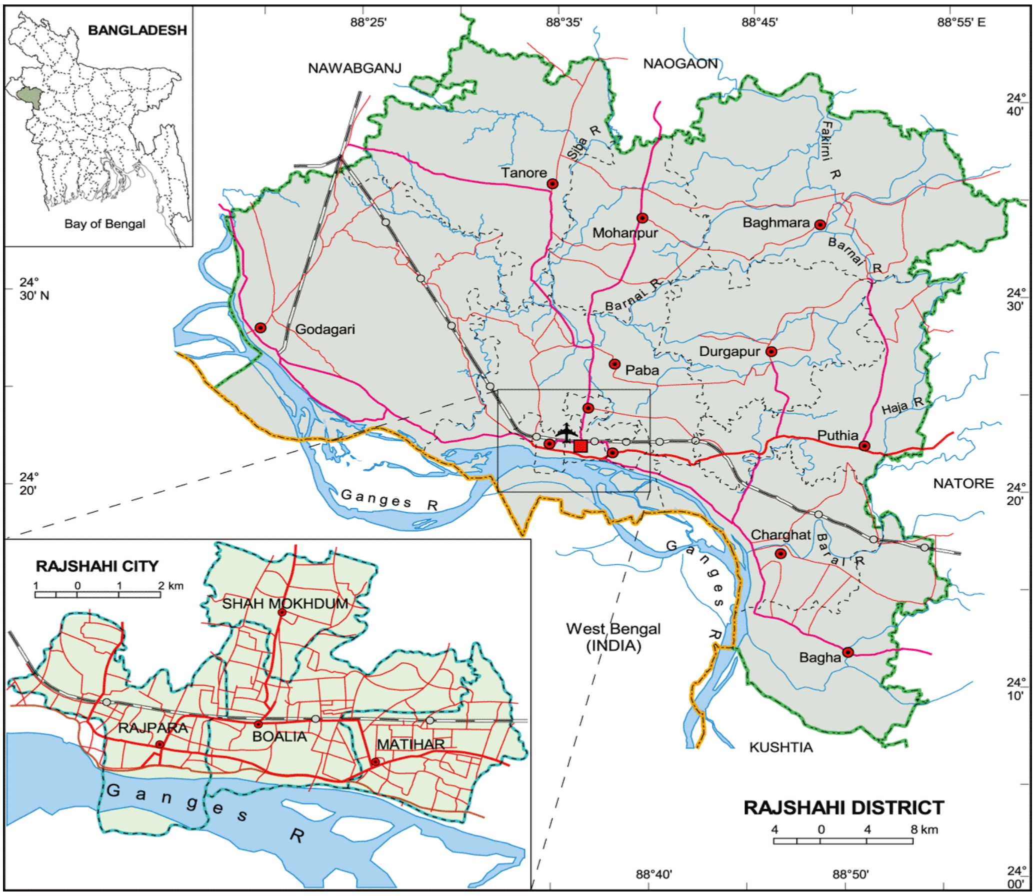
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where, n is the number of data points, y_i is the observed value of the dependent variable for data point i, f_i is the predicted (modeled) value of the dependent variable for data point i, $\bar{y}$ is the mean of the observed values.

The value of R^2 is between 0 and 1. When the model's value is 1, it means that the dependent variable is perfectly predicted; when its value is 0, it means that the dependent variable's variability is not explained by the model at all.

3. Study Area

Rajshahi, situated in northwestern Bangladesh between latitudes 24.23°N and 25.11°N, falls within a subtropical zone with distinct seasons. The climate is generally warm and humid, experiencing an average annual rainfall of around 1971 mm, primarily during the monsoon. Rainfall varies across locations and years, exemplified by 1,738 mm in 1981 and 798 mm in 1992. The region, which is classified as drought-prone, experiences average temperatures of 35° to 25°C during the warmest season and 9° to 15°C during the coldest. In Rajshahi, summertime highs can reach 45°C, while wintertime lows plummet as low as 5°C. The importance of rainfall analysis is highlighted by the unique climatic extremes observed in this older alluvium region compared to the rest of the nation. The map of Rajshahi district is given below:



Map 1: Map of Rajshahi District (Source: bdmaps.blogspot.com)

3.1 Data Sources

The data utilized in this study was obtained from the Bangladesh Meteorological Department (BMD). Rainfall occurrences are influenced by various natural factors including dry bulb temperature, humidity, wind speed, sunshine, minimum temperature, and maximum temperature. The dataset spans from January 1964 to December 2017, providing a total of 648 observations for these attributes in the Rajshahi district, Bangladesh. Table 1 presents the names, types, and measurement units of the attributes, including dry-bulb, humidity, sunshine, wind speed, minimum temperature, and maximum temperature. Data analysis performed by statistical package SPSS, Python, R program and other related software.

Table 1: Data and measuring unit

Variables	Type	Measurement
Rainfall	Continuous	mm
Dry-bulb	Continuous	Degrees Celsius
Humidity	Continuous	%
Wind Speed	Continuous	Meters per second
Sunshine	Continuous	Hour
Maximum Temp	Continuous	Degrees Celsius
Minimum Temp	Continuous	Degrees Celsius

3.2 Data Transformation

In our investigation, we incorporate monthly rainfall data, dry-bulb, humidity, wind speed, sunshine hours, maximum and minimum temperature. These variables exhibit distinct measurement units. To eliminate unit variations, we employ the Min-Max Normalization method in this study. This technique standardizes all numerical variables to a range of 0 to 1.

3.3 Data Smoothing

Missing data is a regular issue in environmental study. Even though there are missing values in the data collected for each station, the World Meteorological Organization (WMO) states that it is possible to successfully estimate missing values because less than 10% of the data must be estimated using the Statistical Package for Social Sciences (SPSS) program. The missing values were dispersed at random, and over variable periods of time, some years had constant missing data. We used SPSS to interpolate the closest values to fill in the missing data. We readied the data for analysis by approximating the missing data using the smoothing technique in SPSS software.

4. Result and Discussion

4.1 Characteristics of Data

To find out the initial pattern and trend from the data it is necessary to calculate summary statistics which present mean, minimum, maximum, first quartile, third quartile. The estimated summary statistics of these attributes dry bulb, temperature, humidity, wind speed, sunshine hour, maximum and minimum temperature is given in Table 2.

Table 2: The summary statistics of these variables

	Drybulb	Max.Tem	Min.Tem	Humidity	Sunshine	Wind speeds	Rainfall
Minimum	14.3	21.28	8.387	44	2.58	0	0
Maximum	33.1	39.69	27.477	90	10.78	34	763
Mean	25.37	31.1	20.33	76.18	6.72	2.87	120.96
1st Quartile	21.75	28.39	15.03	70	5.49	2	6
Median	27.5	32.05	22.345	78	7	2.3	70
3rd Quartile	28.8	33.505	25.633	84	7.99	3.25	198
Skewness	-0.74	-0.36	-0.46	-0.85	-0.37	3.46	1.26
Kurtosis	0.84	0.59	1.29	0.02	0.69	9.23	1.2

Table 2 showed that mean value of these attributes are 25.37, 31.1, 20.33, 76.18, 6.72, 2.87 and 120.96 for dry bulb, max.tem, min.tem, humidity, sunshine hour, wind speeds and rainfall respectively. For all of these attributes, the minimum and maximum values are same which is equal to 0 and 1 respectively. We see that, 1st quartile of these attributes are 21.75, 28.39, 15.03, 70, 5.49, 2 and 6 respectively and 3rd quartile of these attributes are 28.8, 33.51, 25.63, 84, 7.99, 3.25 and 198 respectively. The median of all of these attributes are 27.5, 32.05, 22.35, 78, 7, 2.3 and 70 respectively. Therefore, the skewness of all of these attributes are -0.74, -0.36, -0.46, -0.85, -0.37, 3.46 and 1.26 respectively and the kurtosis of all of these attributes are 0.84, 0.59, 1.29, 0.02, 0.69, 9.23 and 1.2 respectively. So, finally we found that skewness values of dry bulb, max.tem, min.tem, humidity and sunshine hour indicate negative skewness while wind speeds and rainfall indicate positive skewness. The kurtosis values confirmed that all of these attributes showed platykurtic where wind speeds showed leptokurtic shape.

4.2 Estimation Result of Predictive models

Performance of any machine learning algorithm is assessed by comparing the result with pre-classified data. One part is used for training the models and the other part is used for testing and validation. Usually 70% data is used for training and rest of 30% is used for test and validation. In this study we split the total data set as 80%, 70%, 60%, 50% as training data where rest of data 20%, 30%, 40% and 50% as test data respectively. For reducing the paper size we only present the result for 70% training and 30% test data.

- **For 70% train data and 30% test data**

Following the pre-processing stage of the procedure, the effectiveness of the deep learning algorithms Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP) was compared with other machine learning algorithms like Gradient Boosting method, k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT). These procedures are all considered to be monitored. The efficacy of any supervised deep learning and machine learning method can be assessed by comparing the outcomes with pre-classified data, or known classes. 30% is used as the test set and 70% is used as the training set in our study. Using a variety of metrics, including Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Nash-Sutcliffe model efficiency (NSE) and R-square experiments were carried out to determine the predicted models.

Table 3: Model evaluation criteria for different models

	Model	**R-square**	**NSE**	**RMSE**	**MAPE**
Deep learning	**LSTM**	0.8682	0.8682	0.1551	2.3234
	MLP	0.8433	0.8433	0.1533	1.6184
Machine learning	**Gradient Boosting**	0.7966	0.7966	0.1604	4.2925
	KNN	0.7946	0.7946	0.1621	4.3256
	SVM	0.5963	0.5963	0.1909	5.1225
	DT	0.4956	0.4956	0.2161	6.5784

The estimated result from Table 3 indicated that Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Nash-Sutcliffe model efficiency (NSE) and R-square confirmed the following results.

The R-square value showed that MLP and LSTM deep learning algorithms present the highest value of R-square where machine learning algorithms like gradient boosting, K-NN, SVM and DT algorithms showed lower value. This results indicated that deep learning perform better than machine learning algorithms. The similar result was found in case of NSE statistics. The RMSE value indicated that the both deep learning algorithms present the lowest value and all of the machines learning algorithms present the higher RMSE value. The MAPE statistics also indicated deep learning algorithms give lowest value of MAPE compare to machine learning algorithms. Between these two deep learning algorithms MLP give better results based on all of the evaluation criteria which have been used here. In case of machine learning algorithms the most performing model for predicting the monthly rainfall of Rajshahi is Gradient boosting where Decision tree is the worst performing predictive model. Finally, all the models showed that the predictive performance in case of MLP model is high and Decision Tree model is low. So, from the Table 3, we concluded that MLP model is the most successful model to predict the monthly rainfall data of Rajshahi district.

Here, we present a graphical line plot for comparing the value of actual rainfall data and predicted rainfall value of LSTM, MLP, Gradient Boosting, KNN, SVM and Decision Tree model on 30% test data:

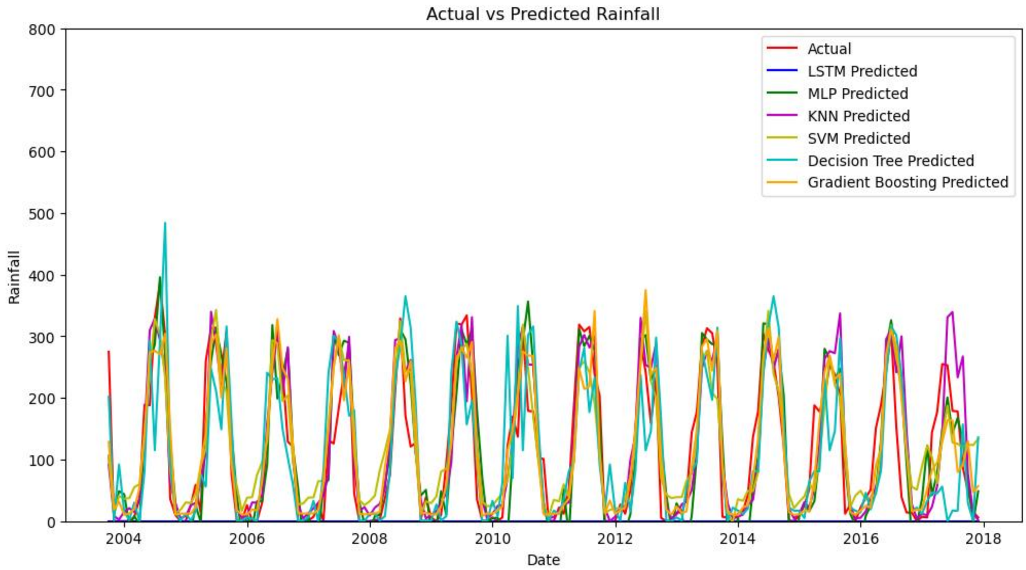


Fig. 7: Actual vs Predicted rainfall for different model based on test data

From Figure 7, it can be seen that, the predicted rainfall trend line of MLP model is flowing very close to the actual rainfall trend line. So we can say that, MLP is the best and Decision tree is the worst model from all of these deep learning and machine learning model. The similar result is obtained for other set of data such as 80% training 20% test, 60% training and 40% test and 50% training and 50% test data. Now we will predict the monthly rainfall data with most successful predictive algorithms MLP in case of Rajshahi district, Bangladesh. The predictive value is given in Table 4.

Table 4: Predicted rainfall data for next 12 months in case Rajshahi

Months	**Forecasted Rainfall**	**Months**	**Forecasted Rainfall**
January	36.79	July	251.94
February	4.53	August	172.71
March	114.14	September	166.96
April	93.71	October	10.54
May	182.34	November	5.65
June	224.92	December	0

The estimated result from Table 4 indicated that, In January, The forecasted rainfall with MLP model is 36.79 mm. Then, the forecasted rainfall in February is 4.53 mm, in March is 114.14 mm, in April is 93.71 mm, in May is 182.34 mm, in June is 224.92 mm, in July is 251.94 mm, in August is 172.71 mm, in September is 166.96 mm, in October is 10.54 mm, in November is 5.65 mm and in December is 0 mm. Therefore, we can say that, the highest amount of rainfall will occur in the month of July and lowest amount of rainfall will occur in December which confirm the historical pattern in Rajshahi dirtrict, Bangladesh.

5. Conclusions

Rainfall prediction is a useful but challenging undertaking. Rainfall forecasting can be achieved by employing different developed techniques to extract hidden knowledge from previous meteorological data. Rainfall has a major impact on agriculture and the economy everywhere in the world. Deep learning systems accurately predict rainfall data by extracting and utilizing hidden information from historical meteorological data. There is currently no single model or technique that can accurately anticipate climate data across all geographic locations, despite the fact that there are numerous models and methods for forecasting climate data. In order to predict monthly rainfall data from Rajshahi district in Bangladesh from January 1964 to December 2017, we regularly used deep learning algorithms like Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP), along with other machine learning algorithms like the Gradient Boosting method, k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT). As a result, this paper compared these algorithms.

The empirical results suggested that, for the subsequent timeframes, the MLP model is the most suitable algorithm for predicting the monthly rainfall data in case of Rajshahi district, Bangladesh. The outcome also demonstrated that the Rajshahi district's decision tree model performed the poorest for prediction. It was shown that deep learning algorithm approaches must be improved, optimized, and integrated in order to investigate and address these challenges. In order to make more forecasts, it is proposed that future research look at different classification algorithms and climatic aspects on different weather stations data. The results of this study will assist policymakers in taking the required actions to address the water issue in order to maintain sustainable agricultural output in these study areas. This will guarantee efficient agricultural output.

References

[1] Aftab, S., Ahmad, M., Hameed, N., Bashir, M. S., Ali, I., and Nawaz, Z. (2018). Rainfall Prediction in Lahore City using Data Mining Techniques, International Journal of Advanced Computer Science and Applications,Vol.9, No.4,254-260.

[2] Aftab, S., Ahmad, M., Hameed, N., Bashir, M. S., Ali, I. and Nawaz, Z. (2018). Rainfall Prediction using Data Mining Techniques: A Systematic Literature Review. International Journal of Advanced Computer Science and Applications, Vol. 9, No.5, 143-150.

[3] Bagirov, A. M., Mahmood, A. and Barton, A. (2017). Prediction of monthly rainfall in Victoria, Australia: Cluster wise linear regression approach, Atmos. Res., vol. 188, pp. 20–29.

[4] Basha, O. M., Weng, L., Men, Z., Xu, W. and Badie, I., Morsi (2016). NICE's Indirect Coal-to-Liquid Process for Producing Clean Transportation Fuels Using Fischer-Tropsch Synthesis. Front Engineering 3: 362–76.

[5] Boser, B. E., Guyon, I. M., Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory – COLT '92. p. 144.

[6] Fix, E. and Joseph, H. L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. USAF School of Aviation Medicine, Randolph Field, Texas.

[7] Gandhi, R. (2018). Support Vector Machine-Introduction to Machine Learning Algorithms. Towards Data Science. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[8] Gupta, D and Ghose, U. (2015). A Comparative Study of Classification Algorithms for Forecasting Rainfall. pp. 0–5, 2015.

[9] Hyndman, R.J., and Athanasopoulos, G. (2018). Forecasting: principles and practice, 2nd edition. OTexts:Melbourne, Australia. OTexts.com/fpp2.

[10] Kannan, M., Prabhakaran, S. and Ramachandran, P. (2010). Rainfall Forecasting Using Data Mining Technique. International Journal of Engnieering and Technology, Vol. 2 (6), 397-401.

[11] Mishra A. K. (2018). Development and Analysis of Artificial Neural Network Models for Rainfall Prediction by Using Time-Series Data. Int. J. Syst. Appl., vol. 10, no. 1, pp. 16–23.

[12] Monira, S. S., Faisal, Z. M. and Hirose, H. (2010). Comparison of artificially intelligent methods in short term rainfall forecast, Proc. 2010 13th Int. Conf. Comput. Inf. Technol. ICCIT 2010, no. Iccit, pp. 39–44, 2010.

[13] Olaiya, F. and Adeyemo, A.B. (2012). Application of Data Mining Techniques in Weather Prediction and Climate Change Studies. I.J. Information Engineering and Electronic Business, 2012, 1, 51-59.

[14] Sethi, N. and Garg, D.K (2014). Exploiting Data Mining Technique for Rainfall Prediction. International Journal of Computer Science and Information Technologies, Vol. 5 (3), 3982-3984.

[15] Sivapragasam, C., Liong, S. and Pasha, M. (2001). Rainfall and runoff forecasting with SSA-SVM approach, J. Hydroinformatics, no. April 2016, pp. 141–152.

[16] Shi, Y. (2017). Understanding LSTM and its diagrams. https://shiyan.medium.com/seeing-is-believing-in-this-world-and-the-next-3a7b5d861e61

[17] Solanki, N. and G. P. B, (2018). A Novel Machine Learning Based Approach for Rainfall Prediction. Inf. Commun. Technol. Intell. Syst. (ICTIS 2017) - Vol. 1, vol. 83, no. Ictis 2017.

[18] Talib, M. R., Ullah, T., Sarwar, M. U., Hanif, M. K. and Ayub, N. (2017). Application of Data Mining Techniques in Weather Data Analysis. International Journal of Computer Science and Network Security, Vol. 17, No. 6, 22-28.

[19] Tharun, V. P., Prakash, P. and Devi, S. R. (2018). Prediction of Rainfall Using Data Mining Techniques. Proceeding of the 2 nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Eplore Compliant-Part Number: CFP18BAC-ART: ISBN: 978-1-5386-1974-2.

[20] Thomas, M. and Peter, H. E. (1967). Nearest neighbor pattern classification (PDF). IEEE Transactions on Information Theory. 13 (1): 21–27.

[21] Vapnik, V. (1995). Support-vector networks (PDF). Machine Learning. 20 (3): 273–297.

[22] Zainudin, (2016). Comparative Analysis of Data Mining Techniques for Malaysian Rainfall Prediction, Int. J. Sci. Eng. Inf. Technol., vol. 6, no. 6, pp. 1148–1153.

[23] Zhang, Y.D., Liao, L., YU, Q., Ma, W.G., Li, K.H. (2021). Using the gradient boosting decision tree (GBDT) algorithm for a train delay prediction model considering the delay propagation feature. Advances in Production Engineering & Management, Vol. 16, No. 3, pp 285-296.