Rainfall analysis and prediction using ensemble learning for Karnataka State

Govardhana Meti[1], Ravi Kumar Guralamata Krishnegowda[2], Gudihalli Savitha Swamy[3]

[1]Department of Computer Science and Engineering, BGS Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[2]BGS College of Engineering and Technology, Bangalore, Visvesvaraya Technological University, Belagavi, India
[3]Lecturer, Government Polytechnic for Women, Bengaluru, India

Article Info

Article history:

Received Jun 6, 2023
Revised Jul 2, 2023
Accepted Aug 18, 2023

Keywords:

Ensemble
Rainfall forecasting
Rainfall prediction
Stacking
XGBoost regressor

ABSTRACT

Accurate prediction of rainfall can save farmers from crop damage as erratic rains in India have caused agricultural losses. Due to continuous climate change, rainfall and weather prediction has become essential. Rain causes damage to people and property. So, rainfall forecasting is crucial for crop as our country's economy is still heavily dependent on agriculture. Using machine learning (ML) and deep learning (DL) models, we can train these models for the rainfall dataset and make predictions. In this paper, we estimate rainfall over the Karnataka region using a stacking ensemble model on a rainfall dataset collected between 1901 and 2015.

Corresponding Author:

Govardhana Meti
Department of Computer Science and Engineering, BGS Institute of Technology
Visvesvaraya Technological University
Nagarabhavi, Belagavi, India
Email: govardhan.meti@gmail.com

1. INTRODUCTION

India's agricultural production is affected by heavy rainfall. Detailed rainfall information can help farmers sustain their crops and thereby boost the country's economy. Rainfall forecasting helps in reducing the damage caused by floods, which pose a serious threat to people's lives and property. Determining rainfall is a challenging task for scientists, as the actual duration and amount of rainfall may vary [1]. Due to fluctuating meteorological conditions, it is hard to predict the rainfall with accuracy. For both the summer and the rainy seasons, forecasting the amount of rainfall is difficult. Researchers have proposed many rainfall prediction techniques, most of which use random numbers and resemble climatic data [2]. For predicting rainfall, researchers typically adopt classification or regression techniques [3]. Fuzzy logic, decision tree (DT) algorithm, regression, artificial neural network (ANN) along with data manipulation techniques are some computing methods used for weather forecasting [1]. ANN is often used to forecast rainfall [2], [3]. Long short-term memory (LSTM), a recurrent neural network (RNN) version, is applied for both short-and long-term forecasting. For weather forecasting, a temporal convolutional network (TCN) layer model is also used [4]. Machine learning (ML) algorithms evaluate the data that might be linear or non-linear to provide the predicted value with high accuracy [5]. The occurrence of rain and its intensity are affected by various climatic conditions [6]. In comparison to various ML approaches, the deep neural network (DNN) performs better [7]. ANN-MLP is useful in rainfall trend analysis of annual and seasonal rainfall [8]. A DNN model's capacity to solve classification or regression problems improves when more layers are used [9]. Since deep learning approaches such as LSTM and RNN can be used to learn non-linear relationships, it is particularly useful for predicting rainfall [10], [11]. Although the XGBoost model and stacked ensemble DTs are used as the final step, the results show that it is a better approach to take advantage of what is provided by the ensemble ML approach rather than relying on a single regression algorithm [12]. Therefore, we use XGBoost in our proposed method in ensemble stacking with other models for prediction.

Further, section 2 of this paper discusses research works related to this study. The section 3 presents the proposed methodology and the ML approach proposed by this research. The section 4 presents the results and discussions. In section 5 concludes the paper.

2. LITERATURE REVIEW

In this section, we will review some machine learning approaches and deep learning methods used for rainfall prediction. In research on applying deep learning algorithms to determine rainfall, Basha *et al.* [1] explored various problems that may arise while employing the prediction approaches. Two deep learning methods have been applied, the multilayer perceptron (MLP) and the autoencoder. Their study found that due to the non-linear connections in rainfall datasets and ANN model's ability to learn from the past, they provide a superior solution than all existing technologies. Ganesh *et al.* [3] created a composite ensemble regression model by combining bagging regression (BAR), extra tree regression (ETR), random forest regression (RFR), gradient boosting regression (GBR), and extreme gradient boosting regression (XGBoostR). Consequently, these ensemble regression models, in which two or more models are implemented in various combinations, are used to predict precipitation instead of features.

Grace and Suganya [2] proposed a rainfall prediction model for an Indian dataset using multiple linear regression (MLR). The information being provided contains of multiple climatic parameters that can result in more precise precipitation forecasts. Using mean square error (MSE), precision, and correlation, the performance of the proposed model is evaluated. Hewage *et al.* [4] proposed a weather forecasting model based on LSTM and TCN. The performance was then compared to current classical ML approaches, statistical forecasting approaches, a dynamic ensemble algorithm, and the established weather research and forecasting (WRF) numerical weather prediction model.

Kaushik *et al.* [5] examined rainfall data for the state of Punjab using k-nearest neighbor (KNN), extreme learning machine (ELM), and support vector machine (SVM). Parameters tested include humidity, wind speed, maximum temperature, and minimum temperature for the data from 1973 to 2019. For the training dataset, SVM accurately predicted results 95% of the time, compared to 92% for the testing dataset. Liyew *et al.* [6] conducted a study on the prediction of Ethiopian rainfall using three ML techniques, namely MLR, random forest (RF), and XGBoost. Using mean absolute error (MAE) and root mean absolute error (RMSE) metrics, the efficiency of models is analyzed. According to their findings, the XGBoost ML method performed better than others.

Different ML approaches, including SVM, RF and DTs, were investigated by Naik *et al*. [7] to predict rainfall. The Adam optimizer is used to improve the model's average weights and bias. Their proposed system based on DNN is outperformed over other ML approaches to predict rainfall. Praveen *et al*. [8] employed the Pettitt test to compute the abrupt change over a period. In addition, they performed the mann-kendall (MK) test and sen's trend analysis method was applied to investigate the trend of rainfall. ANN-MLP is employed to predict the amount of India's rainfall for the next 15 years. From their results, it is found that the annual rainfall pattern was prominently negatively declining (-8.5).

To enhance the performance of traditional ANNs, Putra *et al*. [9] presented a new DNN model as deep auto encoder semi convolutional neural network (DAESCNN). Tests were conducted using rainfall datasets collected annually between 2006 and 2016 from meteorological stations in the city of Samarinda, Indonesia. The results showed that DAESCNN outperformed better and provided 99% accuracy. A timely assessment of rainfall will improve the yield and lower agricultural expenses. Salehin *et al*. [10], designed an architecture based on LSTM and RNN model to estimate the extent of rainfall in Dhaka city from 2000 to 2014. The factors, including temperature, humidity, dew point, wind direction, wind pressure and wind speeds, are considered while making predictions. After detailed analysis, 76% accuracy was obtained. Hussain and Zoremsanga [11] observed that DL methods can be successfully applied for rainfall forecasting and found to be outperform traditional ML methods and shallow neural network models.

Barrera-Animas *et al*. [12] proposed a paradigm using LSTM, bidirectional-LSTM networks, XGBoost, Stacked-LSTM, and also includes GBR, linear SVR, and ETR. The models were then compared in the prediction of hourly precipitation using time-series data. The tests were conducted on climate datasets from five cities in the UK for the period 2000 to 2020. Applied a cascading and stacking approach to combine the ML approach with Ridge, GBR, LGBM, and XGBoost. They employed bayesian optimization technique to get better accuracy. The model's performance was evaluated using R^2, RMSE, MAE, and relative RMSE for cascading regression and stacking regression used for all features. The model has accuracy of 0.984 R^2 and 179.898 RMSE.

3. PROPOSED METHOD

The proposed method includes data collection, pre-processing, DEA-State wise, DEA-Karnataka State and district wise and customised model construction and then results are predicted. The process flow is as shown in Figure 1. It consists of collecting the data from open-source data available from Indian government web resources, pre-processing of data, analysis the data by visualisation to understand the insights of the data and creating a customised ensemble model to predict the rainfall.

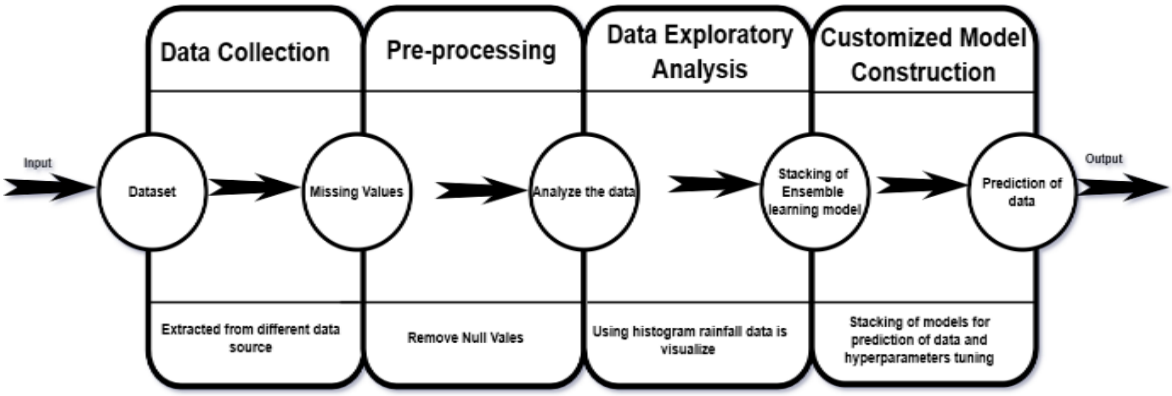

Figure 1. Proposed method flow

3.1. Dataset collection

Data of rainfall was gathered from the Indian Government's website (http://data.gov.in). The collected dataset consists of subdivision wise rainfall and its departure i.e., rainfall dataset between year 1901 to 2015. The dataset includes parameters such as mean, standard deviation, coefficient of variation, actual, percentage departure, and number of districts.

3.2. Pre-processing

Naik *et al.* [7] explained in their review that data pre-processing is an essential task; we should perform it before building a model. Pre-processing involves removing redundant information, filling in missing values, and extracting features so that it can be fed to a model in the training process. Missing values are filled during the data cleaning task by means of mean and median values [5], [6]. Depending on the extent of the dataset, the data set is divided into two parts: a training dataset and a testing dataset. The training dataset is used to train the model (using actual data), whereas the dataset for testing is used for assessing the model's predictions on new (unknown) data.

3.3. Data exploratory analysis-statewise

Data exploratory analysis (DSA) is used to analyze the data before pre-processing using different visualizations. From the dataset, we observed the distribution of rainfall from month of January to December using histogram in Figure 2. The statewise rainfall with respect to the season from January to February is graphically visualised in Figure 3. The statewise rainfall with respect to the season from March to May is graphically visualised in Figure 4. The statewise rainfall with respect to the season from June to September is graphically visualised in Figure 5. The statewise rainfall with respect to the season from October to December is graphically visualised in Figure 6. Rainfall occurs more in July in all states. Also, the extent of rain is more from June to September. The heatmap for annual distribution along with different months is shown in Figure 7.

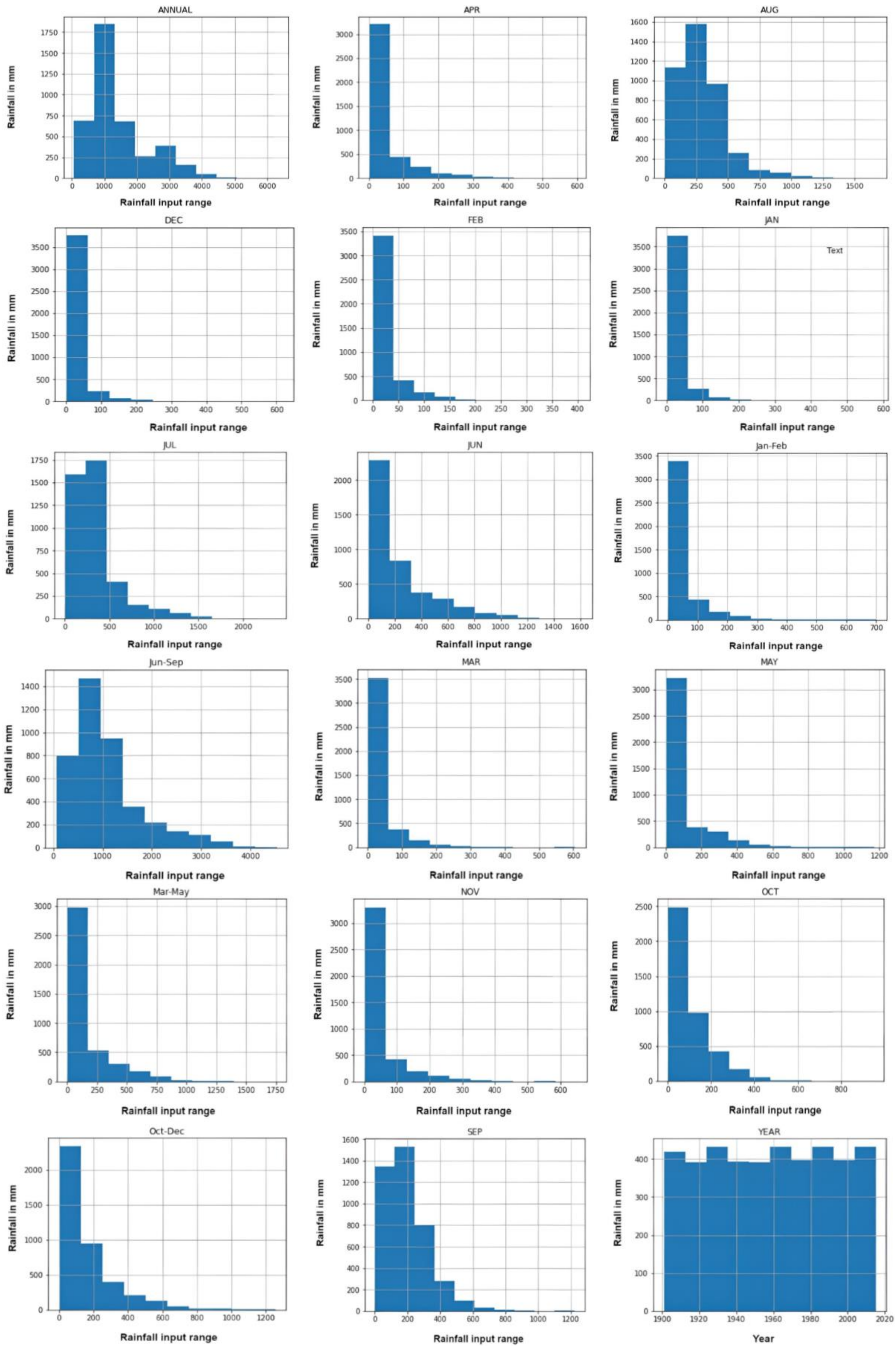

Figure 2. Histogram of rainfall for each month

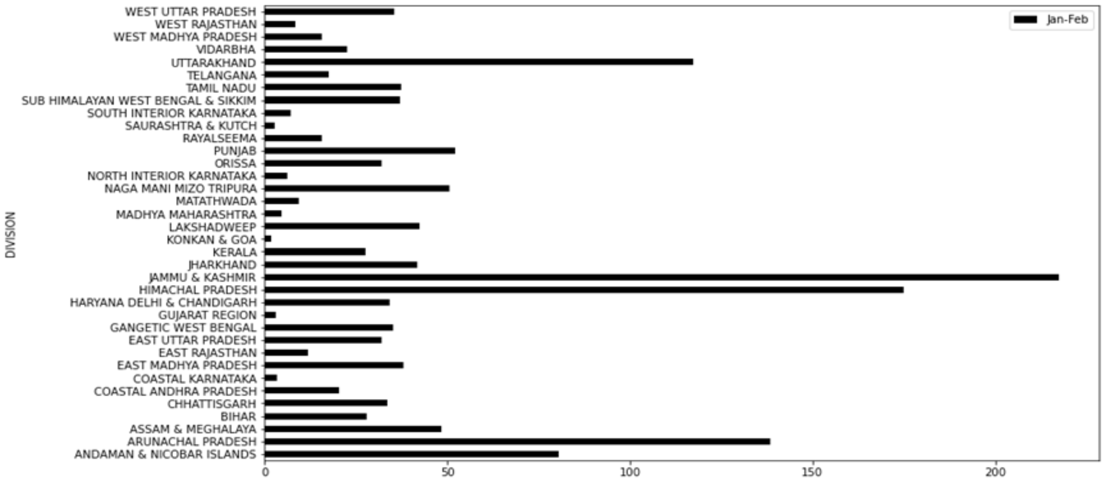

Figure 3. State wise rainfall for each session [Jan-Feb]

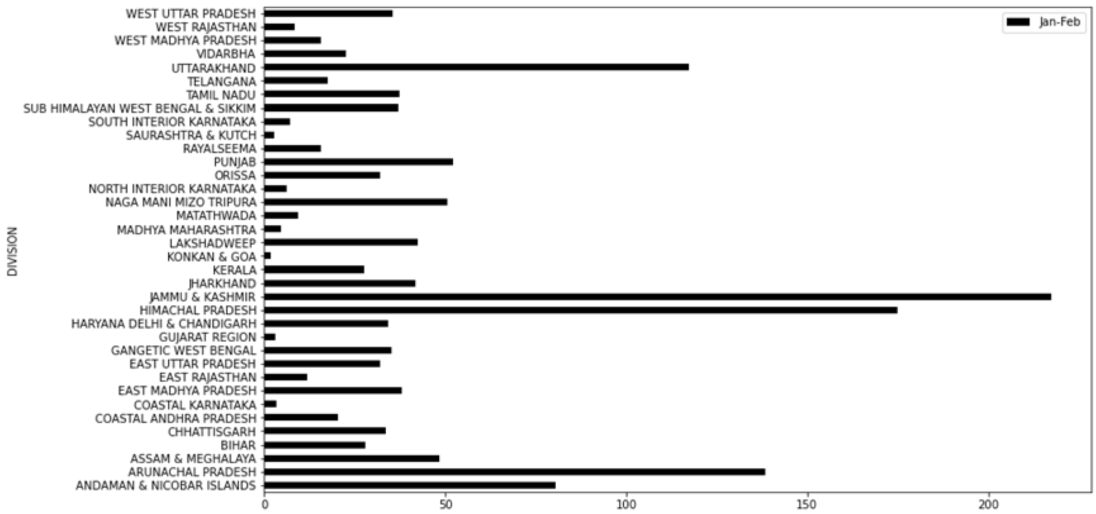

Figure 4. State wise rainfall for each session [March-May]

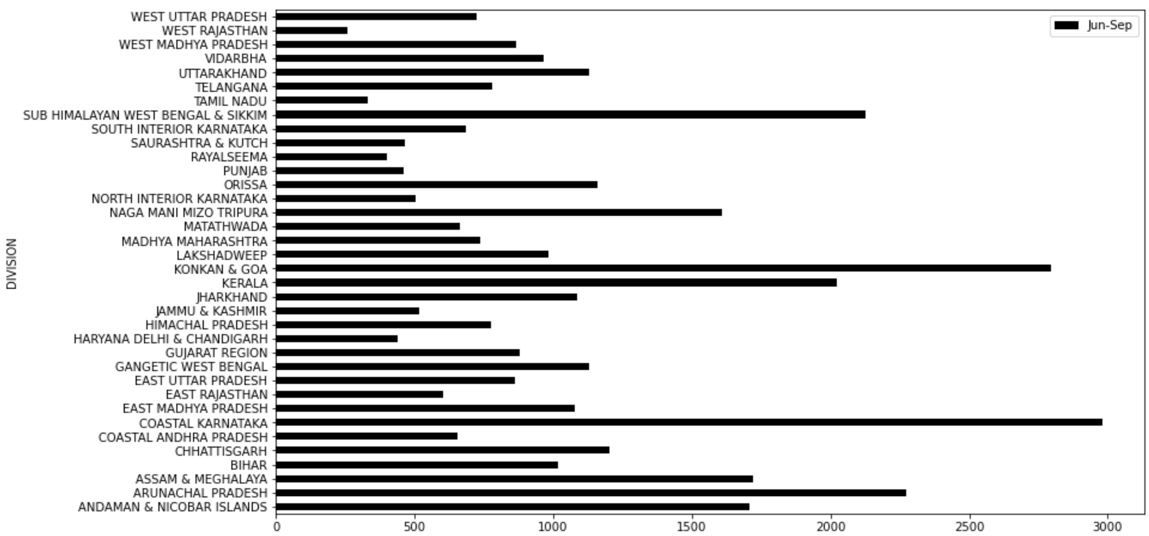

Figure 5. State wise rainfall for each session [Jun-Sept]

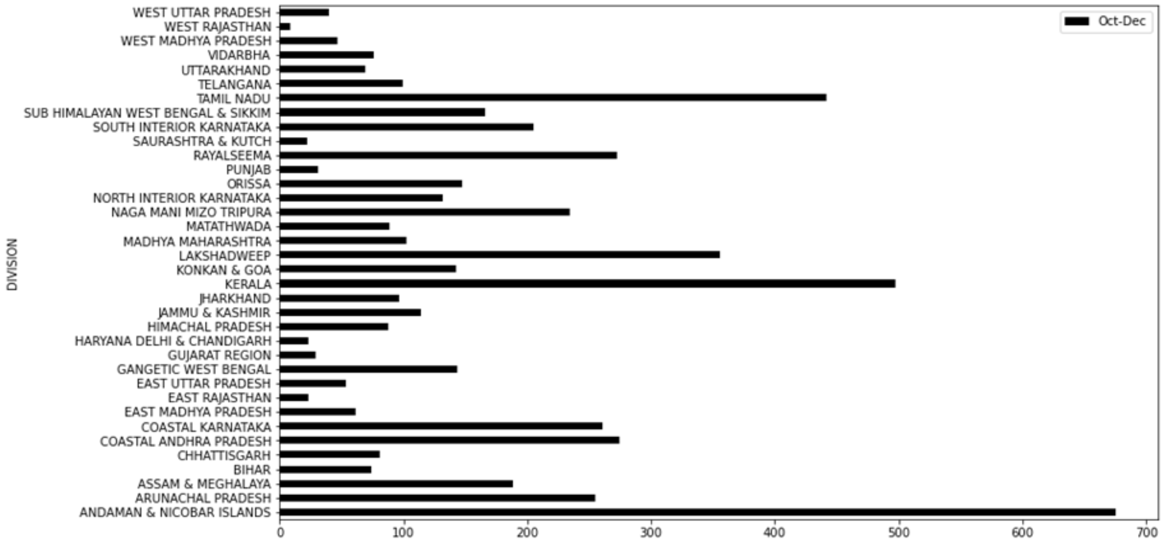

Figure 6. State wise rainfall for each session [Oct-Dec]

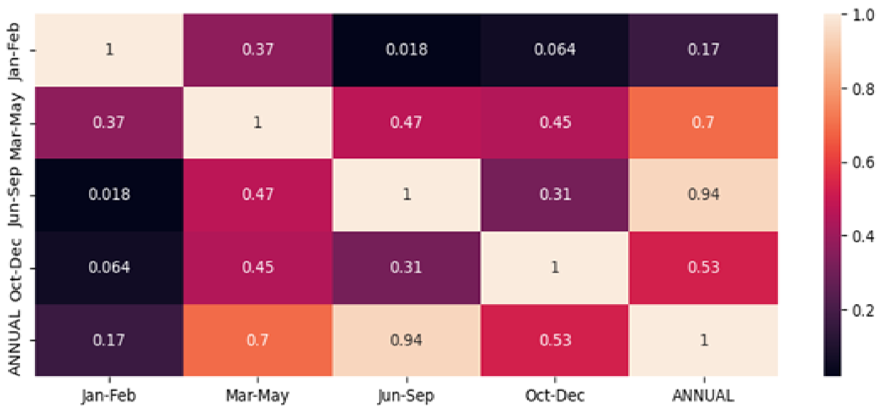

Figure 7. Heatmap of rainfall of data

3.4. Data exploratory analysis-Karnataka

In DEA for Karnataka region, Figure 8 displays year wise rainfall in this region for the past 115 years. This past data is used to predict the rainfall. The distribution of rainfall in the state of Karnataka including all the districts, can be seen in Figure 9. Bar plot for south interior karnataka region wise rainfall for each month is represented in Figure 10, for north interior karnataka region wise rainfall for each month it is represented in Figure 11 and for costal karnataka region wise rainfall for each month is represented in Figure 12, which shows the amount of rainfall regionwise. The average rainfall for some locations is shown in Table 1.

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
3542	1901	1.8	0.6	10.7	52.4	81.6	960.9	991.2	606.4	108.0	120.5	104.9	17.8
3543	1902	3.2	0.3	4.9	10.2	54.6	698.4	1401.6	454.2	708.4	180.4	50.8	132.2
3544	1903	0.7	0.0	0.0	4.1	202.8	536.5	1405.5	593.8	304.4	185.0	79.3	5.3
3545	1904	2.4	0.0	4.8	23.7	93.2	1108.2	1070.0	465.6	245.3	127.2	0.7	0.0
3546	1905	0.0	0.2	0.0	6.4	83.1	767.3	777.3	586.9	172.9	222.2	36.1	0.0

Figure 8. Year wise rainfall in Karnataka

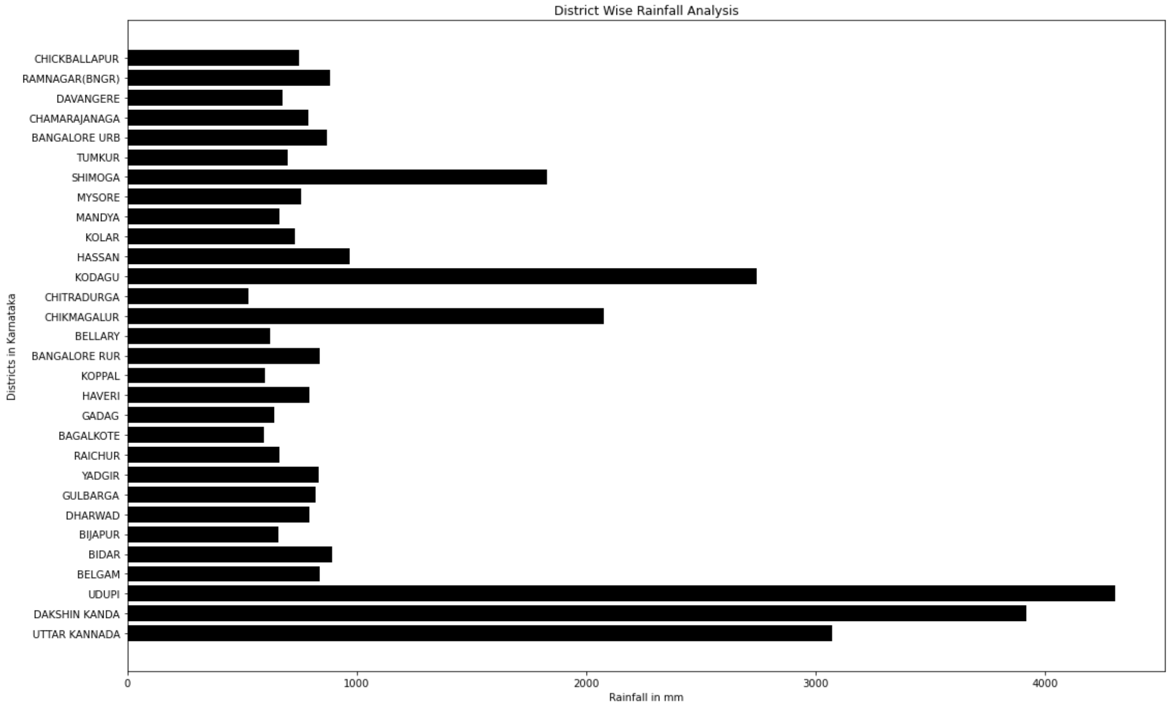

Figure 9. Karnataka district wise rainfall

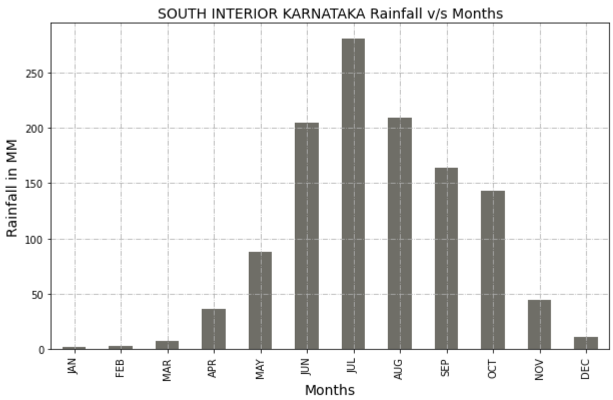

Figure 10. Bar plot for south interior karnataka region wise rainfall for each month

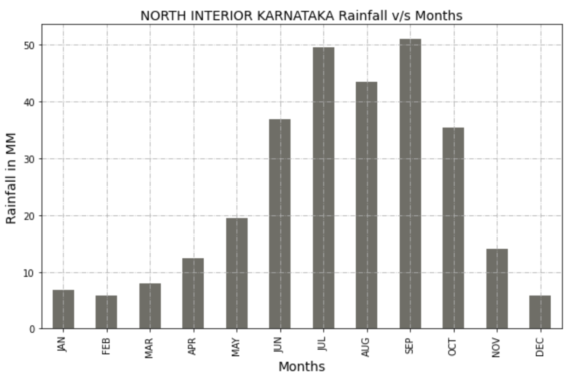

Figure 11. Bar plot for north interior karnataka region wise rainfall for each month

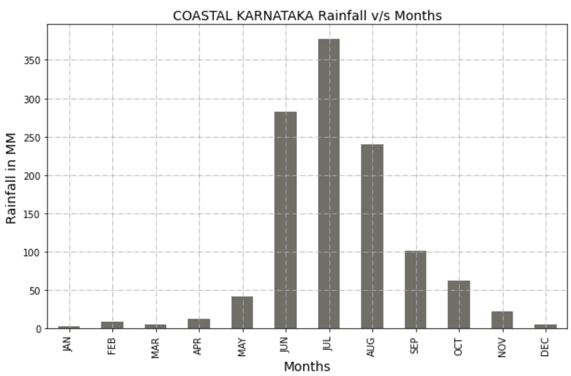

Figure 12. Bar plot for costal karnataka region wise rainfall for each month

Table 1. Average rainfall

Location index	Avg. rainfall
0	1.8
115	0.6
230	10.7
345	52.4
460	81.6

3.5. Customized model construction

In this section, the stacking of ensemble learning models created to predict the rainfall of a particular city in the Karnataka. Stacking, also known as stacked generalization, is an advanced ensemble learning technique used to improve the predictive performance of machine learning models. It involves combining the predictions of multiple base models by training a higher-level model (meta-model) on their outputs. The idea behind stacking is to leverage the strengths of different models to create a more accurate and robust predictive model.

3.5.1. Staking of model

Stacking typically generates improved performance than any single trained model [13]. It has been used on tasks such as regression [14], classification [15], [16]. It has also been used to study the error rate of bagging [17]. In our study, we trained ridge, lasso, support vector regression (SVR), RF, light gradient boosting machine (LGBM), XGboost regression models individually with the same number of CV on our rainfall training dataset. Our proposed approach is represented as Figure 13.

To fit our dataset, an SVR model with a linear kernel and hyper-parameter tuning was used. The lasso works through a continuous coefficient shrinking process, resulting in a significant decrease in the variance of the predicted values, which reduces regression coefficients to minimize the chance of overfitting [18]. The ridge will manage the trade-off bias and variance using regularization approach a and sum-of-squares error function [19], [20]. High and low misestimates are equally penalized in SVR training due to its symmetrical loss function. The fundamental advantage of SVR is that its computational difficulty is independent of the dimensions of the input space [21]. Random forests are a form of ensemble learning that builds several DTs during training [22]. LGBM are the most commonly used machine learning algorithm for prediction [23].

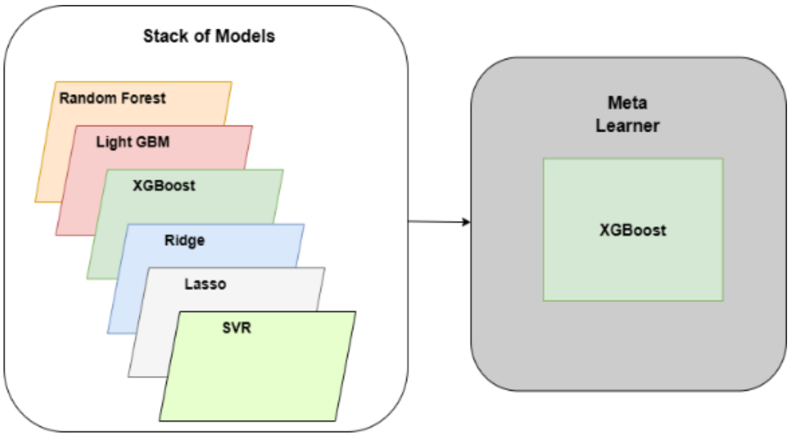

Figure 13. Model stacking with ensemble layers

GBR builds a weak ensemble prediction model similar to other boosting techniques [3]. Chen *et al*. [24] proposed XGBoost based on the gradient boosting paradigm. It is a tree-based ensemble algorithm applied to solve the supervised ML problems which has data with multiple features of *xi* to calculate a target variable *yi* and used in predictions [24]. Therefore, XGBoost is used to automatically identify features with CV. Due to the speed and prediction accuracy of the XGBoost, it is used for classification and regression problems [6]. It is an efficient linear model solver and tree learning algorithm [25].

XGBoost is effective, scalable, and provides higher accuracy [26], [27]. The LGBM, GBR, XGBoost, and ridge algorithms are used in combination with stacking and cascading approaches in [28]. We used 5-K fold for the initial training that means the dataset is iteratively trained 5 times. Then these models are combined to give a model stack. Thus, the features obtained from each model are stacked to provide improved accuracy. A meta-regressor is used in the ensemble learning approach of stacking to combine several regression models.

3.5.2. Grid search cross validation for hyper-parameter tuning

In ensemble regression models, the parameters such as maximal depth, number of estimators, and base estimator are optimized for optimal performance [4]. Using a grid search for hyperparameters [12], the optimal hyperparameter values for each model's training were determined. Utilizing a set of predetermined values for parameters required for n-fold cross-validation to obtain maximum accuracy, a grid search is developed [25], [28]. Optimizers tune hyper-parameters during the training phase in order to attain the maximum average values after multiple tests. Stratified cross-validation is utilized to address the issue of overfitting in the standard grid search, in which samples are randomly divided into K-folds. The grid search cross validation (GridSearchCV) model from scikit learn [29] is utilized to determine the optimal parameters. Using XGB regressor, LGBM regressor, and RF regressor, we instantiate some grid function input parameters with 5 K-fold CV. The methods used in the study might include multiple data visualization and data production for improved accuracy of the prediction. The dataset utilized here to train the proposed models is a dataset created using collected data from the Meteorological Department of India. Figure 1 depicts a block diagram of the methodology for EDA and pre-processing for rainfall data.

4. RESULTS AND ANALYSIS

In this section, we have evaluated the performance of our methodology along with ML models using the coefficient of determination metrics and we have determined the best ML model using MAE, MSE, root MSE and accuracy metrics. The models are developed in google colab pro platform using python language. The experiments are conducted by utilizing the Karnataka State rainfall dataset. Here to evaluate the accuracy the metrics such as MSE, MAE and root MSE are utilized.

4.1. Evaluation of best model

The coefficient of determination is also called R-square or R^2 is used to predict and explain the future outcomes of a model. R^2will help in deciding a model's best fit. Based on the proportion of the total range of outcomes explained by the model, it offers a measure of how well observed results are replicated by the model [30]. The (1) for R^2 is:

$$\mathrm{R}^2 = \frac{\sum_{i=1}^{n}|y_i - x_i|^2}{\sum_{i=1}^{n}|y_i - \bar{y}_i|^2} \tag{1}$$

where, $\bar{y_i}$ is mean value of y_i.

The best possible regression score is equal to 1. As we can see, XGB regressor achieved 0.8493 R^2score. Stacking cross validation regressor (StackingCVRegressor) is an ensemble-learning meta regressor implementation for stacking regression. It uses multiple regressors to predict. In our research, we use ridge, lasso, SVR, RF, LGBM, XGBoost ML to combine these models as a stack. Then we trained these stacked models on our rainfall dataset. We obtained R^2 score of 0.84266 using stackingCVRegressor and 12 K-folds CV with XGboost as meta-regressor. We can improve the results by using the staking approach as shown in Table 2.

Table 2. Score for models

Model	R^2 score
XGB regressor	0.8493
LGBM regressor	0.8561
Random forest regressor	0.8669

4.2. Performance evaluation and prediction

The most frequently utilized metrics for evaluating the accuracy of continuous variables were RMSE and MAE [6]. Mean absolute error (MAE) is calculated as the error between paired observations representing the same event. According to Willmott and Matsuura [31], the MAE is given by the sum of the absolute errors divided by the sample size. The (2) is given as:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \tag{2}$$

where y_i is the prediction and $x_i$the true value [31].

The MSE is calculated as the difference in the predicted values and actual values derived from square of the mean difference over the dataset.

$$MSE = \frac{\sum_{i=1}^{n}|y_i - x_i|^2}{n} \quad (3)$$

the RMSE is evaluated as the square root of the variance also known as standard deviation, [32], [33]. For a prediction array y_i and actual array x_i made up of n scalar observations, the RMSE is expressed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}|y_i - x_i|^2}{n}} \quad (4)$$

the values of MAE, MSE, RMSE and accuracy for training data and testing data can be analyzed in Tables 3-4.

Table 3. MAE, MSE, RMSE and accuracy for training data

Algorithm	MAE	MSE	RMSE	Training accuracy in %
Linear regression	301.30	147533.8	384.10	3.1
XGB regressor	71.36	17869.9	133.67	88.3
Gradient boosting regressor	71.06	15244.0	123.46	90.0
Random forest regressor	67.8000	13974.5	118.21	90.8
Meta regressor-XGBoost regressor	25.55	1670.2	40.86	98.9

Table 4. MAE, MSE, RMSE and accuracy for test data

Algorithm	MAE	MSE	RMSE	Testing accuracy in %
Linear regression	333.59	177915.9	421.80	0.4
XGBoost regressor	110.26	31704.3	178.05	82.3
Gradient boosting regressor	94.41	24822.6	157.55	86.1
Random forest regressor	98.79	25950.5	161.09	85.5
Meta regressor-XGBoost regressor	94.53	25050.9	158.27	86.3

The value of RMSE will always be greater than or equal to MAE; the larger the difference between them, the greater the variability of individual errors in the sample [6]. In the training phase, meta regressor-XGBoost regressor has the lowest RMSE value of 133.67 among the other models. In the testing phase, meta regressor-XGBoost regressor has the lowest RMSE value of 158.27. Thus, the above results indicate that it can predict more accurately than all other models. We obtained an accuracy of 0.8493 R^2 and 158.27 RMSE using the stacking ML model with meta regressor-XGBoost regressor.

Accuracy is calculated as correct predictions of data divided by total data points. In our study, meta regressor-XGB regressor achieved training accuracy of 98.9% and testing accuracy of 86.3% see Figure 14. We get training accuracy of 98.39% and testing accuracy 86.3% for meta regressor-XGBoost regressor, respectively. Then, we used a linear regression model in further training with alpha=0.5. We obtained 2,418 value of MAE using meta XGBoost regressor for "Mysore" city as shown in Table 5. The prediction model proposed out performed better compare to state of art models.

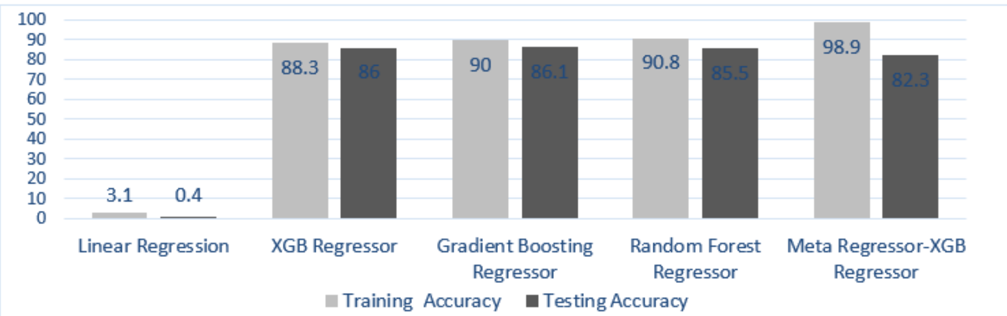

Figure 14. Training and testing accuracy

Table 5. Different model performance for "Mysore"

Model	MAE	Mean	Std. deviation
Linear	14.712	119.15882	173.401
SVR	10.182	123.68824	174.37629
XGBoost	5.359	128.51176	176.71068
Random forest regressor	2.829	131.04118	183.27556
Meta XGBoost	2.418	131.45294	183.89036

5. CONCLUSION

In this study, the stacking of ensemble models helps combine various ML algorithms that are used to evaluate rainfall prediction when tested on rainfall datasets in the Karnataka region. In this stacking architecture, an attempt is made to use the features of stack ML algorithms and apply the performance of meta-learners to improve the prediction ability. Our proposed approach achieved an accuracy of 0.8493 R^2 and 157.27 RMSE. The performance of the prediction is affected by the dataset's partitioning into training and test data and the ML model's architecture. In the future, our focus is to investigate the variability in the rainfall dataset to identify key components, and will analyze other ML models, such as RNN, LSTM, to improve predictive performance. Climatic, meteorological conditions, and variability data can be used to effectively extend a data-driven model using a stacking ensemble model onto various climatic regions.

REFERENCES

[1] C. Z. Basha, N. Bhavana, P. Bhavya, and V. Sowmya, "Rainfall prediction using machine learning deep learning techniques," in *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020*, Jul. 2020, pp. 92–97, doi: 10.1109/ICESC48915.2020.9155896.

[2] R. K. Grace and B. Suganya, "Machine learning based rainfall prediction," in *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, Mar. 2020, pp. 227–229, doi: 10.1109/ICACCS48705.2020.9074233.

[3] P. Ganesh, H. V. Vasu, and D. Vinod, "Estimation of rainfall quantity using hybrid ensemble regression," in *Proceedings of the 2019 9th International Conference on Advances in Computing and Communication, ICACC 2019*, Nov. 2019, pp. 300–309, doi: 10.1109/ICACC48162.2019.8986217.

[4] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Analysis and Applications*, vol. 24, no. 1, pp. 343–366, Feb. 2021, doi: 10.1007/s10044-020-00898-1.

[5] S. Kaushik, A. Bhardwaj, and L. Sapra, "Predicting annual rainfall for the indian state of punjab using machine learning techniques," in *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, Dec. 2020, pp. 151–156, doi: 10.1109/ICACCCN51052.2020.9362742.

[6] C. M. Liyew and H. A. Melese, "Machine learning techniques to predict daily rainfall amount," *Journal of Big Data*, vol. 8, no. 1, p. 153, Dec. 2021, doi: 10.1186/s40537-021-00545-4.

[7] A. R. Naik, A. V. Deorankar, and P. B. Ambhore, "Rainfall prediction based on deep neural network: a review," in *2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2020 - Conference Proceedings*, Mar. 2020, pp. 98–101, doi: 10.1109/ICIMIA48430.2020.9074892.

[8] B. Praveen *et al.*, "Analyzing trend and forecasting of rainfall changes in India using non-parametrical and machine learning approaches," *Scientific Reports*, vol. 10, no. 1, p. 10342, Jun. 2020, doi: 10.1038/s41598-020-67228-7.

[9] A. B. W. Putra, R. Malani, B. Suprapty, and A. F. O. Gaffar, "A deep auto encoder semi convolution neural network for yearly rainfall prediction," in *Proceedings - 2020 International Seminar on Intelligent Technology and Its Application: Humanification of Reliable Intelligent Systems, ISITIA 2020*, Jul. 2020, pp. 205–210, doi: 10.1109/ISITIA49792.2020.9163775.

[10] I. Salehin, I. M. Talha, M. M. Hasan, S. T. Dip, M. Saifuzzaman, and N. N. Moon, "An artificial intelligence based rainfall prediction using LSTM and neural network," in *Proceedings of 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering, WIECON-ECE 2020*, Dec. 2020, pp. 5–8, doi: 10.1109/WIECON-ECE52138.2020.9398022.

[11] J. Hussain and C. Zoremsanga, "A survey of rainfall prediction using deep learning," in *ICECIE 2021 - 2021 International Conference on Electrical, Control and Instrumentation Engineering, Conference Proceedings*, Nov. 2021, pp. 1–10, doi: 10.1109/ICECIE52348.2021.9664730.

[12] A. Y. Barrera-Animas, L. O. Oyedele, M. Bilal, T. D. Akinosho, J. M. D. Delgado, and L. A. Akanbi, "Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting," *Machine Learning with Applications*, vol. 7, p. 100204, Mar. 2022, doi: 10.1016/j.mlwa.2021.100204.

[13] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.

[14] L. Breiman, "Stacked regressions," *Machine Learning*, vol. 24, no. 1, pp. 49–64, Jul. 1996, doi: 10.1007/bf00117832.

[15] C. F. Tsai, "Training support vector machines based on stacked generalization for image classification," *Neurocomputing*, vol. 64, no. 1-4 SPEC. ISS., pp. 497–503, Mar. 2005, doi: 10.1016/j.neucom.2004.08.005.

[16] J. Chen, C. Wang, and R. Wang, "Using stacked generalization to combine SVMs in magnitude and shape feature spaces for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2193–2205, Jul. 2009, doi: 10.1109/TGRS.2008.2010491.

[17] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010, doi: 10.1007/s10462-009-9124-7.

[18] T. Hastie, R. Tibshirani, and M. Wainwright, "Statistical learning with sparsity: the lasso and generalizations," *Statistical Learning with Sparsity: The Lasso and Generalizations*, pp. 1–337, 2015, doi: 10.1201/b18401.

[19] A. E. Hoerl and R. W. Kennard, "Ridge regression: applications to nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 69–82, Feb. 1970, doi: 10.1080/00401706.1970.10488635.

[20] G. C. McDonald, "Ridge regression. wiley interdisciplinary reviews: computational statistics," *Communications in Statistics-Simulation and Computation*, vol. 1, no. 1, pp. 93–100, 2009.

[21] M. Awad and R. Khanna, *Efficient learning machines*. Berkeley, CA: Apress, 2015.
[22] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.
[23] R. M. Aziz, M. F. Baluch, S. Patel, and A. H. Ganie, "LGBM: a machine learning approach for ethereum fraud detection," *International Journal of Information Technology (Singapore)*, vol. 14, no. 7, pp. 3321–3331, Dec. 2022, doi: 10.1007/s41870-022-00864-6.
[24] T. Chen and T. He, "Xgboost: extreme gradient boosting," *R Lecture*, vol. 1, no. 2016, pp. 1–84, 2014.
[25] H. R. Seireg, Y. M. K. Omar, F. E. A. El-Samie, A. S. El-Fishawy, and A. Elmahalawy, "Ensemble machine learning techniques using computer simulation data for wild blueberry yield prediction," *IEEE Access*, vol. 10, pp. 64671–64687, 2022, doi: 10.1109/ACCESS.2022.3181970.
[26] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, no. 2, pp. 337–407, Apr. 2000, doi: 10.1214/aos/1016218223.
[27] J. H. Freidman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, p. 1189, 2001.
[28] B. H. Shekar and G. Dagnew, "Grid search-based hyperparameter tuning and classification of microarray cancer data," in *2019 2nd International Conference on Advanced Computational and Communication Paradigms, ICACCP 2019*, Feb. 2019, pp. 1–8, doi: 10.1109/ICACCP.2019.8882943.
[29] P. Fabian *et al*., "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, [Online]. Available: http://scikit-learn.sourceforge.net.
[30] D. J. Ozer, "Correlation and the coefficient of determination," *Psychological Bulletin*, vol. 97, no. 2, pp. 307–315, Mar. 1985, doi: 10.1037/0033-2909.97.2.307.
[31] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005, doi: 10.3354/cr030079.
[32] C. J. Willmott and K. Matsuura, "On the use of dimensioned measures of error to evaluate the performance of spatial interpolators," *International Journal of Geographical Information Science*, vol. 20, no. 1, pp. 89–102, Jan. 2006, doi: 10.1080/13658810500286976.
[33] V. A. Bell, D. S. Carrington, and R. J. Moore, "Rainfall forecasting using a simple advected cloud model with weather radar, satellite infra-red and surface weather observations: an initial appraisal under UK conditions," NERC Open Research Archive, p: 69, 1994.

BIOGRAPHIES OF AUTHORS

Govardhana Meti he is a research scholar, doing research in BGS Institute of Technology at Visvesvaraya Technological University, Karnataka, India, having around 17+ years of total IT experience (worked 3 years in USA) in implementation, maintenance, enhancement, development and production support of SAP BW/BI, HANA, and ERP packages and mainframe application software. Having 6+ years (October 2017-till date) of Teaching experience in CSE and ISE Department of B.G.S Institute of Technology, B.G Nagar and S.J.B Institute of Technology, Bangalore. He can be contacted at email: govardhan.meti@gmail.com.

Dr. Ravi Kumar Guralamata Krishnegowda he working as Principal at BGS College of Engineering and Technology-Bangalore, India. Overall 25 years of experience, in which 15 years in teaching at the level of Professor and HOD, and 10 Plus years of experience in IT industry as a program manager. He can be contacted at email: ravikumargk@yahoo.com.

Dr. Gudihalli Savitha Swamy she working as Lecturer at Government Polytechnic for Women, Bengaluru, India. Overall, 18 years of experience, 16 years in teaching as a civil engineering faculty and 2 years of industrial experience. She can be contacted at email: savithaswamygss@gmail.com.