IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

DESIGN AND IMPLEMENTATION OF A MACHINE LEARNING-BASED MODEL FOR PRECISE DAILY RAINFALL PREDICTION

[1]Moni Mandal, [2]Prof. Namrata Kumari
[1]M.Tech Student, [2]Assistant Professor
[1] Department of Computer Science and Engineering
RVS College of Engineering and Technology, Jamshedpur, India

Abstract: This research explores the design and implementation of a machine learning-based model to enhance the precision of daily rainfall predictions. By leveraging historical meteorological data and employing advanced machine learning algorithms such as Linear Regression, Random Forest and XGBoost, the model aims to capture intricate relationships between various atmospheric variables. The proposed model is trained, validated, and tested on real-world data, and its performance is evaluated using metrics like the R^2 score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The dataset, preprocesses to handle missing values and outliers, is divided into training and testing subsets. Key hyperparameters of the machine learning models, such as learning rate, maximum tree depth, and the number of boosting rounds, are optimized through cross-validation to prevent overfitting and ensure generalizability. The performance of the Linear Regression, Random Forest and XGBoost model is evaluated against traditional regression techniques and other ensemble methods using metrics like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Our results demonstrate that Linear Regression significantly outperforms on this dataset, providing more accurate and reliable rainfall predictions. The result demonstrate that machine learning techniques can significantly improve the accuracy of daily rainfall predictions, offering a promising tool for better decision-making in weather-dependent industries.

Keywords: *Linear Regression, XGBoost, Random Forest, overfitting, Root Mean Square Error, Mean Absolute Error*

1. *INTRODUCTION*

Rainfall prediction is a fundamental component of weather forecasting, with far-reaching implications for agriculture, water resource management, disaster preparedness, and urban infrastructure planning. Accurate daily rainfall predictions are essential for mitigating the risks associated with extreme weather events, optimizing agricultural practices, and managing water resources effectively. However, predicting rainfall is inherently challenging due to the complex, dynamic, and non-linear nature of atmospheric processes. Traditional forecasting methods, which often rely on statistical models and numerical weather prediction techniques, face limitations in capturing these complexities, leading to less precise predictions.
In recent years, advancements in machine learning (ML) have opened new avenues for improving the accuracy of weather forecasts. Machine learning models, with their ability to learn from large datasets and uncover hidden patterns, offer a powerful alternative to traditional methods. By analysing historical weather data, including variables such as temperature, humidity, pressure, and wind speed, ML models can generate more accurate predictions of daily rainfall.

The dataset Sub Divisional Monthly Rainfall from 1901 to 2017 is downloaded from [16] is Open Government data (OGD) platform India. Open Government Data (OGD) Platform India – data.gov.in – is a platform of Government of India to support Open Data initiative. The portal is proposed to be used by the Government of India Ministries/ Departments their organizations to publish different things like documents, services, tools, datasets and applications collected by them for public use. It anticipates to increase transparency in the functioning of Government and open avenues for many more innovative uses of Government Data to give different perspective.

Traditional statistical methods, such as linear regression have been extensively used for rainfall prediction. However, these methods often struggle to capture the intricate interactions among various atmospheric variables. Recent advancements in machine learning offer

mising alternatives that can model complex patterns and improve predictive accuracy. One such advanced technique is XGBoost (extreme Gradient Boosting), a scalable and efficient implementation of gradient boosted decision trees.

```
Index(['SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL', 'JF', 'MAM', 'JJAS',
       'OND'],
      dtype='object')
```

Fig 1.1: list of variables presents in the dataset(source:"mydataset")

This research focuses on the design and implementation of a machine learning-based model tailored for precise daily rainfall prediction. The study explores various ML algorithms, including ensemble methods like Random Forest and XGBoost which are particularly suited for time series forecasting. The primary objective is to develop a model that can reliably predict daily rainfall, thereby providing a valuable tool for stakeholders in weather-dependent sectors. The model's performance is rigorously evaluated using real-world meteorological data, with the results highlighting the potential of machine learning to enhance the precision of rainfall predictions.

The subsequent sections of this paper will detail the methodology used for data collection and preprocessing, the selection and training of machine learning models, the evaluation metrics employed, and the results obtained. Finally, the paper will discuss the implications of these findings for practical applications and future research directions in the field of weather forecasting.

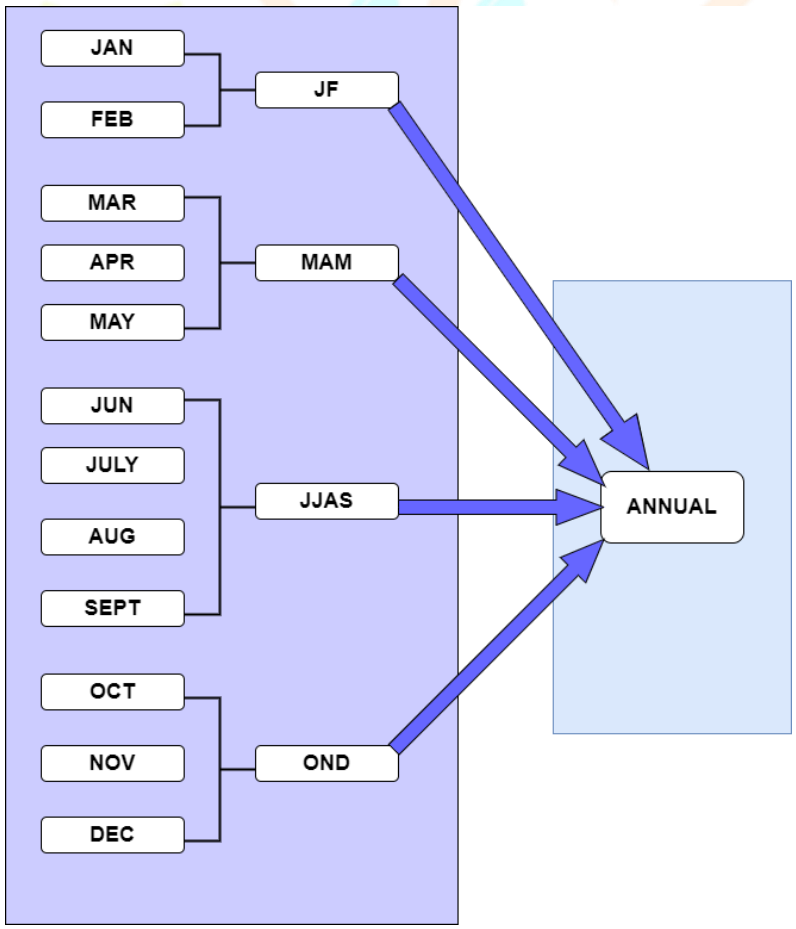

Fig 1.2: variable and its types working on the models (source:"mydataset")

This study investigates the comparison of Linear Regression, XGBoost and Random Forest for daily rainfall prediction of the dataset of each subdivision of India from 1901 to 2015. The dataset contains record of monthly, JF(Jan-Feb), MAM(March-May), JJAS(June-September), OND(October-December), we aim to develop a model that delivers accurate and reliable rainfall forecasts.

2. METHODOLOGY

The dataset is uploaded to the google drive and all the algorithms are trained and tested in google colab. The dataset then passed through the data cleaning process in which all the missing values are replaced with the mean value with the help of '*fillna*' function of python. Then the new dataset is used for training and testing the model. In the training process 20 percent of the whole data is used for testing and rest 80 percent of data is used to train the three models. Different packages are used for different models to train the dataset.

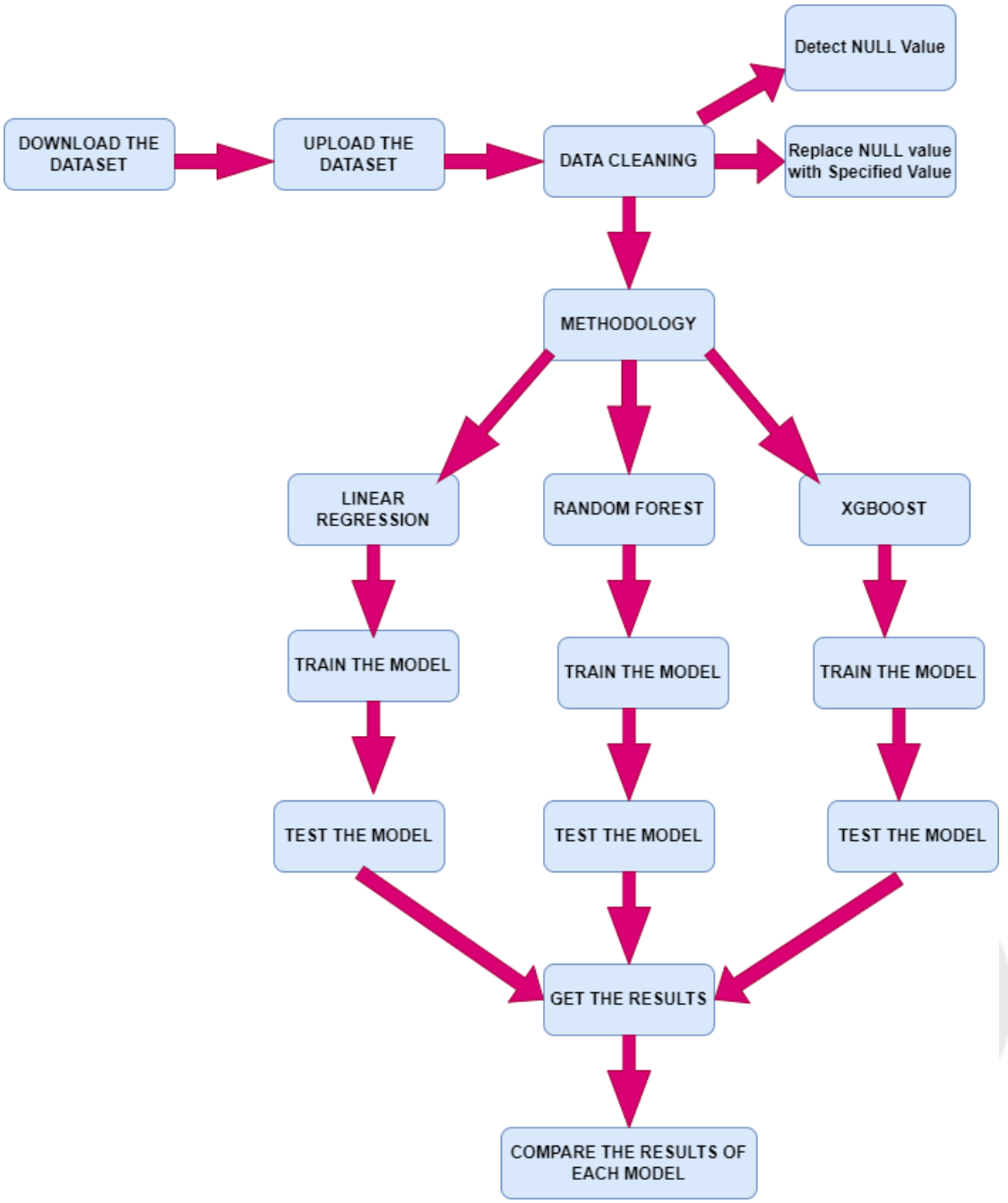

Fig 2.1: work flow diagram of daily rainfall prediction (source:"mydataset")

When you apply machine learning models like Linear Regression, Random Forest, and Boost to a given dataset, each method calculates the result differently.

a) Linear Regression:

Linear regression is a statistical algorithm which works on regression and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X) and the dependent variable (Y). **Simple linear regression** is that where only one input variable (X) is present and if there is more than one input variable, then such linear regression is called **multiple linear regression**.

In this paper multiple linear regression is used where multiple independent features are used to predict the result/output i.e. dependent feature.

The general equation of Linear Regression is:

$$Y_{\beta}(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots \epsilon$$

Where:

i. β0 is the intercept.
ii. β1, β2,…,βn are the coefficients for each feature.
iii. ϵ\epsilon is the error term.

Also

Y=dependent feature

X= independent feature

β = regression coefficient

Linear Regression attempts to find the best-fit line that predicts the target variable YYY as a linear combination of the independent variables X1, X2,… , Xn.

Assume you want to predict the ANNUAL rainfall using monthly data. The features (X) would be monthly rainfall (JAN, FEB, ..., DEC). The target (Y) would be the ANNUAL rainfall.

The model calculates the weights (coefficients) β0, β1,… , βn such that:

$Y=\beta0+\beta1X1+\beta2X2+\cdots+ \beta n.$

It minimizes the sum of squared differences between the actual and predicted values (Mean Squared Error, MSE). The result is a linear equation that can be used to predict new values. The prediction for ANNUAL rainfall is calculated using the linear combination of the monthly rainfall data with the learned coefficients.

The model is trained by minimizing the cost function (often Mean Squared Error, MSE), which measures the difference between the predicted values and actual values in the training data.

After training, the model uses the learned coefficients to predict the target values for the test data. For each instance in the test set, the predicted value is calculated as:

$\hat{Y}=\beta0+\beta1X1+\beta2X2+\cdots+\beta nXn$

- The predicted values are plotted against the actual values, which is shown in the scatter plot.

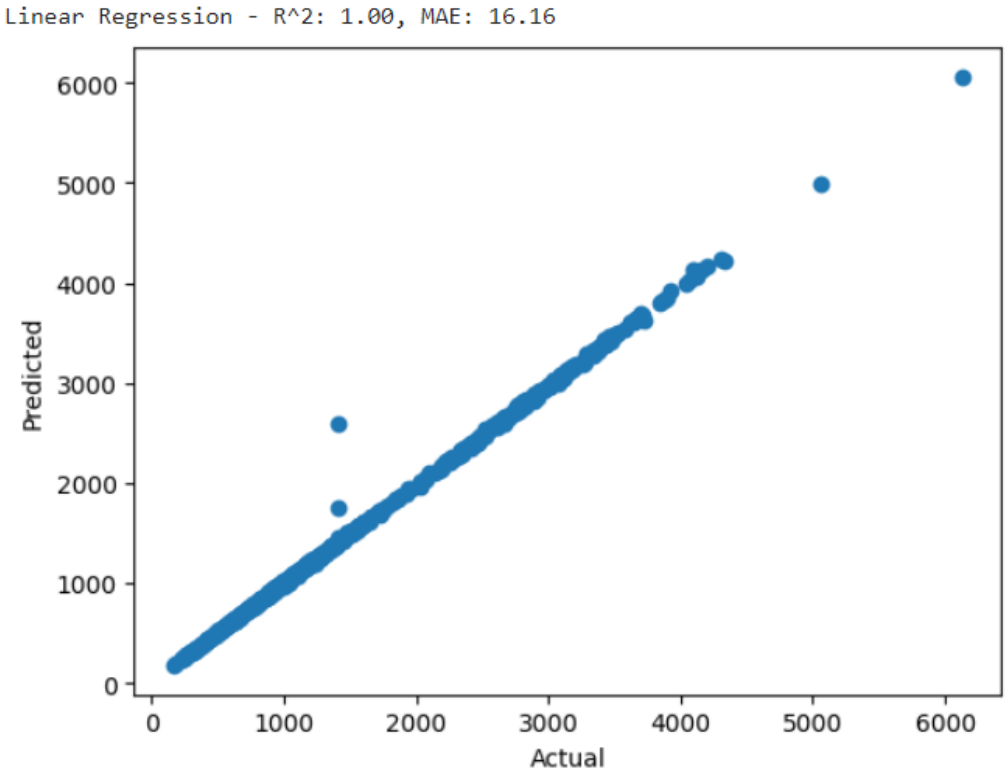

Fig 2.2: Predicted result of Linear Regression

b) Random Forest:

Random Forest is a supervised learning algorithm which means that the data on which it operates contains labels or outcomes. It works by creating many decision trees, each built on arbitrarily chosen subsets of the data. The individual decision tree models are built using a technique called bagging random forest. It comprises randomly selecting subsets of the training data and building smaller decision trees from them. Then we combine the smaller models to form the random forest model, which outputs a single prediction value. The procedure helps reduce variance and improve accuracy by combining the predictions from several decision trees. The model then combines the outputs of all of these decision trees to make an overall prediction for unseen data points. The RF algorithm works on the following steps[1]

i. Take at random p data points from the training set
ii. Build a decision tree associated with these p data points
iii. Take the number N of trees to build and repeat a and b steps
iv. For a new data point, make each one of the N tree trees predict the value of y for the data point and assign the new data point to the average of all of the predicted values of y.

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of individual trees.

Features (X): we were typically using monthly or seasonal rainfall data to predict a target variable, such as ANNUAL rainfall and Target (Y): The target could be the ANNUAL rainfall.

To train the model Bootstrap Sampling methods is used. Random Forest selects random subsets of the data (both samples and features) to train each tree. This reduces overfitting and ensures that the trees are diverse. Each tree is grown by recursively splitting the dataset at different nodes. The best split at each node is determined based on a criterion like Mean Squared Error (MSE), which measures the variance in the target variable. The tree continues to split until it reaches a stopping criterion (e.g., a maximum depth, a minimum number of samples per leaf, or no further reduction in error). Each decision tree makes a prediction for the target variable (e.g., annual rainfall) based on the splits learned during training. For regression tasks, the predictions from all trees are averaged to give the final prediction. The final predicted ANNUAL rainfall would be the average of all the individual tree predictions. This averaging reduces the variance and typically results in better generalization to unseen data compared to a single decision tree. This ensemble approach ensures that the model is less sensitive to noise and outliers, leading to more accurate and stable predictions.

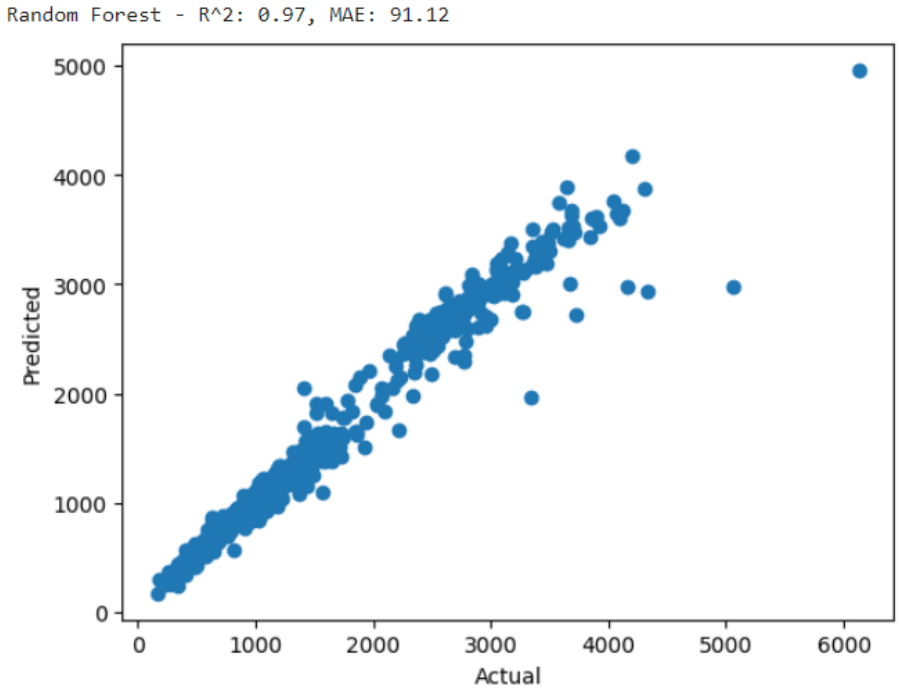

Fig 2.3: Predicted result of Random Forest Classification

c) X G Boost (Extreme Gradient Boosting):

XGBoost a well-known and robust machine learning algorithm often used for supervised learning tasks such as classification, regression, and ranking. Extreme Gradient Boosting is an advance implementation of gradient boosting algorithm, specially designed for speed and performance. It is known for its scalability, efficiency and accuracy, making it one of the most popular machine learning models for tabular data.
XGBoost is based on the gradient boosting framework where multiple weak learners (decision tree) are sequentially trained to correct the errors of the preceding model. Each subsequent tree is trained on the residuals (the difference between the actual and predicted values) of the ensemble. Ensemble learning method combine the prediction of multiple individual model (base learners) to improve overall predictive performance. The iterative process results in a strong predictive model with high accuracy and generalization ability. XGBoost incorporates various regularization techniques to prevent overfitting, such as shrinkage (also known as learning rate) which scales the contribution of each tree and the tree depth regularization which limits the complexity of individual trees.

XGBoost is a boosting algorithm that builds an ensemble of weak learners (usually decision trees) sequentially. Each tree tries to correct the mistake made by the earlier ones. The model optimizes a loss function (e.g., Mean Squared Error) and adds regularization to control complexity.

Features (X): monthly rainfall data (e.g., JAN, FEB, etc.) as features and Target (Y): The target could be the ANNUAL rainfall.

XGBoost starts with a simple initial model, usually a prediction of the mean of the target variable for regression tasks. The residuals (errors) between the predicted and actual values are calculated. These residuals represent what the model failed to predict correctly. A decision tree is built to predict these residuals. The goal of this tree is to correct the errors made by the initial prediction. The tree is constructed by splitting the data to minimize a loss function (e.g., Mean Squared Error for regression). XGBoost uses additional techniques like regularization to prevent overfitting and make the model more generalizable. The predictions are updated by adding the predictions from the new tree to the existing predictions. This is done with a learning rate (shrinkage parameter) that controls how much of the new tree's predictions are added. Each iteration builds a new tree to correct the residuals of the current model. The model continues to improve by focusing on areas where the previous trees made large errors. After a specified number of iterations or when the residual errors stop improving significantly, the model outputs the final prediction. The result is a highly accurate prediction, as XGBoost is designed to optimize performance with techniques like shrinkage, column sampling, and regularization. This ensemble approach ensures that the model is less sensitive to noise and outliers, leading to more accurate and stable predictions.

Hyperparameters like learning rate, depth of the trees, number of trees, and regularization were optimized to reduce overfitting and improve generalization.

i. After training, the model was applied to the **test data** (data that the model hadn't seen during training) to generate predictions.
ii. These predictions were then compared to the actual values, leading to the R^2 and MAE metrics shown in the scatter plot.

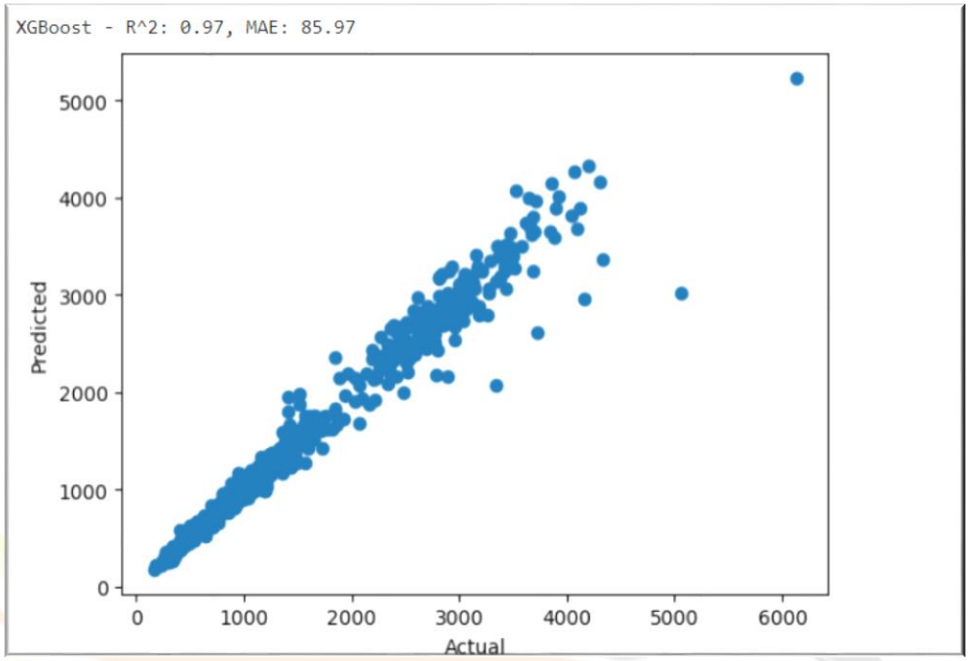

Fig 2.4: Predicted result of XGBoost model

3. *RESULT AND DISCUSSION*

Table 3.1: ***Comparison chart of Linear Regression, Random Forest and XGBoost***

Models / Properties	R^2	MAE (Mean Absolute Error)
Linear Regression	1.00	16.16
Random Forest	0.97	91.12
XGBoost	0.97	85.97

This table compares the performance of three different machine learning models — Linear Regression, Random Forest, and XGBoost — using two key evaluation metrics: R^2 (Coefficient of Determination) and MAE (Mean Absolute Error).

Key Terms:

i. R^2 (Coefficient of Determination): This metric measures how well the model's predictions match the actual data. It ranges from 0 to 1, where:
   a. 1 indicates perfect predictions (i.e., the model explains 100% of the variance in the data).
   b. 0 means the model does not explain any variance in the data.
ii. MAE (Mean Absolute Error): MAE measures the average absolute difference between the expected values and actual values. It gives an indication of how wrong the model's predictions are on average.
iii. The lower the MAE, the better the model is at making accurate predictions.

(a). Linear Regression

i. The plot shows the predicted values versus the actual values. A perfect model would have all points lying exactly on the diagonal line (where predicted = actual).
ii. Outliers: The few points that are away from the diagonal line are outliers, where the model's prediction deviates from the actual value.
iii. High R^2 (1.00): Indicates a perfect fit of the model.
iv. MAE of 16.16: Suggests the model's average prediction error is 16.16 units, which is relatively low, indicating good predictive performance.
v. Title Information:
    a. R^2 (R-squared): The value is 1.00, which indicates that the linear regression model explains 100% of the variance in the dependent variable. This suggests a perfect fit of the model to the data.
    b. MAE (Mean Absolute Error): The MAE is 16.16, which represents the average absolute difference between the predicted and actual values. A lower MAE indicates a more accurate model, and in this case, the value is relatively low.
vi. Scatter Points:
    a. The points are plotted close to the diagonal line, indicating that the predicted values are very close to the actual values. This further confirms the high accuracy and good fit of the model.
    b. There are a few outliers, where the predicted values are slightly different from the actual values. These points are off the diagonal line but still within a reasonable range, given the

(b). Random Forest:

The scatter plot shows the performance of a Random Forest model in predicting values, with an impressive
R2 value of 0.97 and a Mean Absolute Error (MAE) of 91.12. Here's a breakdown of what these results indicate:

i. **R2 Value (0.97)**: This value, also known as the coefficient of determination, indicates that 97% of the variance in the actual values can be explained by your model. This high value suggests that your model fits the data very well.
ii. **Mean Absolute Error (MAE) (91.12)**: This metric measures the average magnitude of errors in your predictions, without considering their trend. An MAE of 91.12 means that, on average, your model's predictions are off by about 91.12 units from the actual values. Given the range of your data (0 to 6000), this is relatively low, indicating good predictive accuracy.
iii. **Scatter Plot**: The dense cluster of blue dots following a linear trend indicates a strong positive correlation between the actual and predicted values. This visual representation aligns with the high
iv. R2 value, showing that your model's predictions are closely aligned with the actual values.

(c). XGBoost

i. The plot shows a **high correlation** between predicted and actual values, with most points falling close to the line y=xy = xy=x, indicating that the predictions closely match the actual values.
ii. **R-squared (R^2)**: 0.97 Similar to random forest, XGBoost explains 97% of the variance in rainfall.
iii. **High R^2** shows that the model explains a lot of the variability in the data.
iv. **Moderate MAE (Mean Absolute Error)**: 85.97 indicates that while the predictions are very close to actual values, there is still an average error of about 86 units (which might be acceptable depending on the application).

How These Results Came About

a. **Data Quality**: High-quality, well pre-processed data likely contributed to the model's performance. Ensuring that your data was clean, free of outliers, and properly scaled would have helped the model learn more effectively.
b. **Model Parameters**: The Random Forest algorithm's parameters, such as the number of trees, depth of trees, and the criteria for splitting nodes, were likely well-tuned. Proper hyperparameter tuning can significantly enhance model performance.
c. **Feature Engineering**: Effective feature selection and engineering would have played a crucial role. By selecting the most relevant features and possibly creating new ones, you provided the model with the most informative inputs.
d. **Cross-Validation**: Using techniques like cross-validation ensures that the model's performance is consistent across different subsets of the data, preventing overfitting and ensuring generalizability.

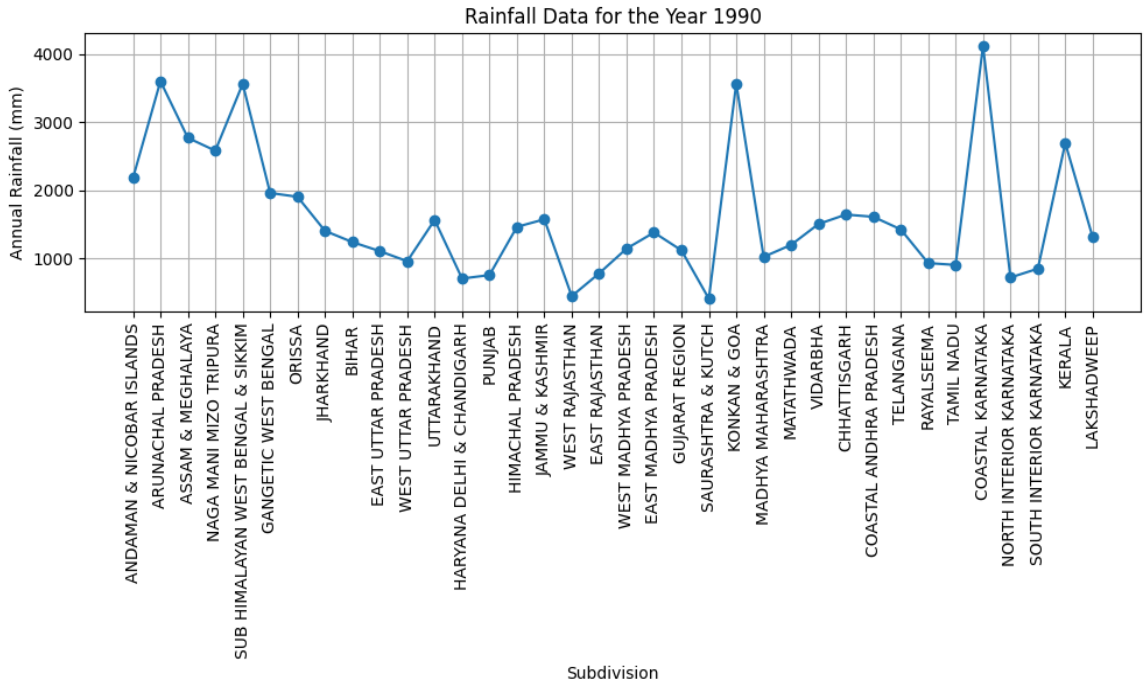

Fig 3.2: Annual rainfall in (mm) in the year 1990 at all the subdivision

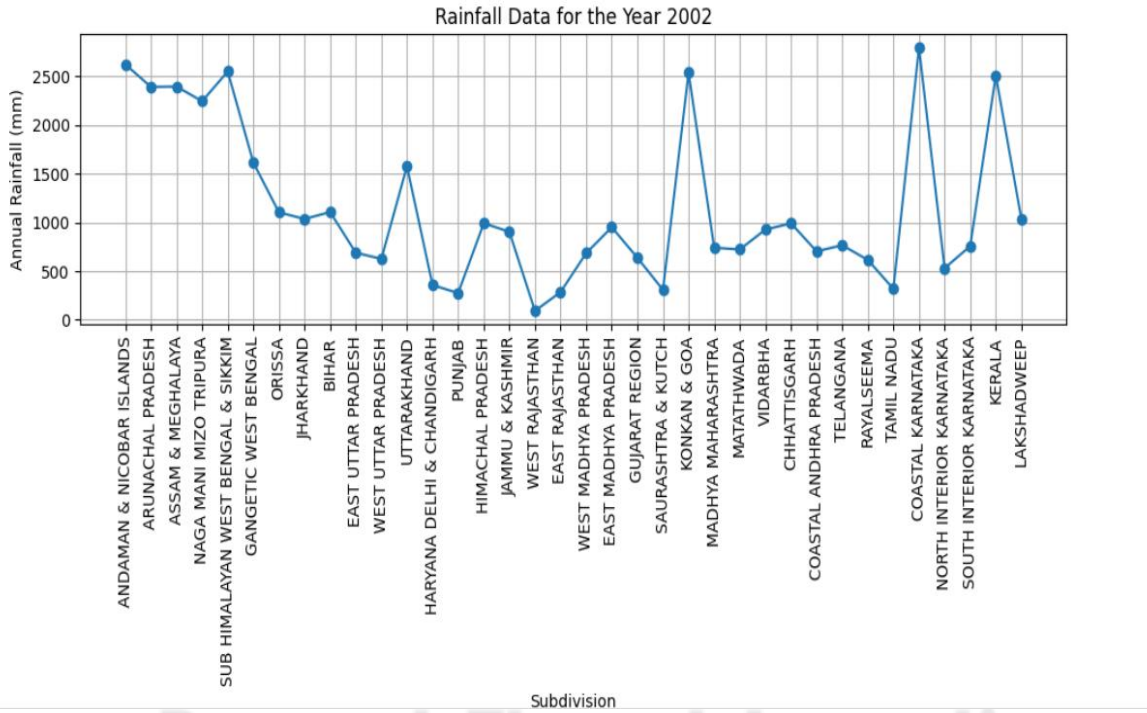

Fig 3.3: Annual rainfall in (mm) in the year 2002 at all the subdivision

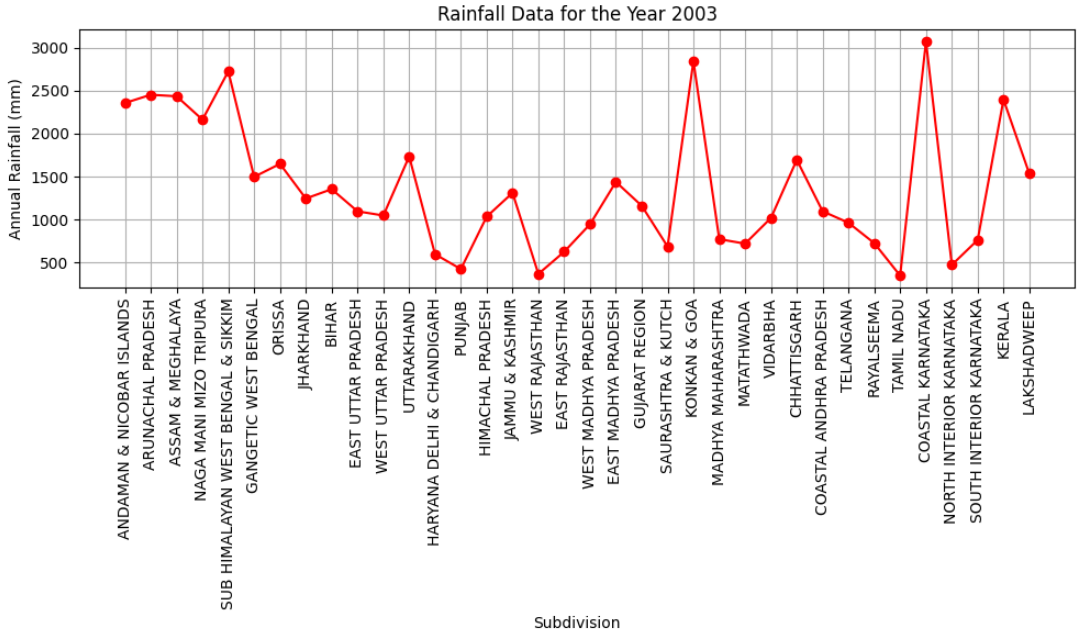

Fig 3.4: Annual rainfall in (mm) in the year 2003 at all the subdivision

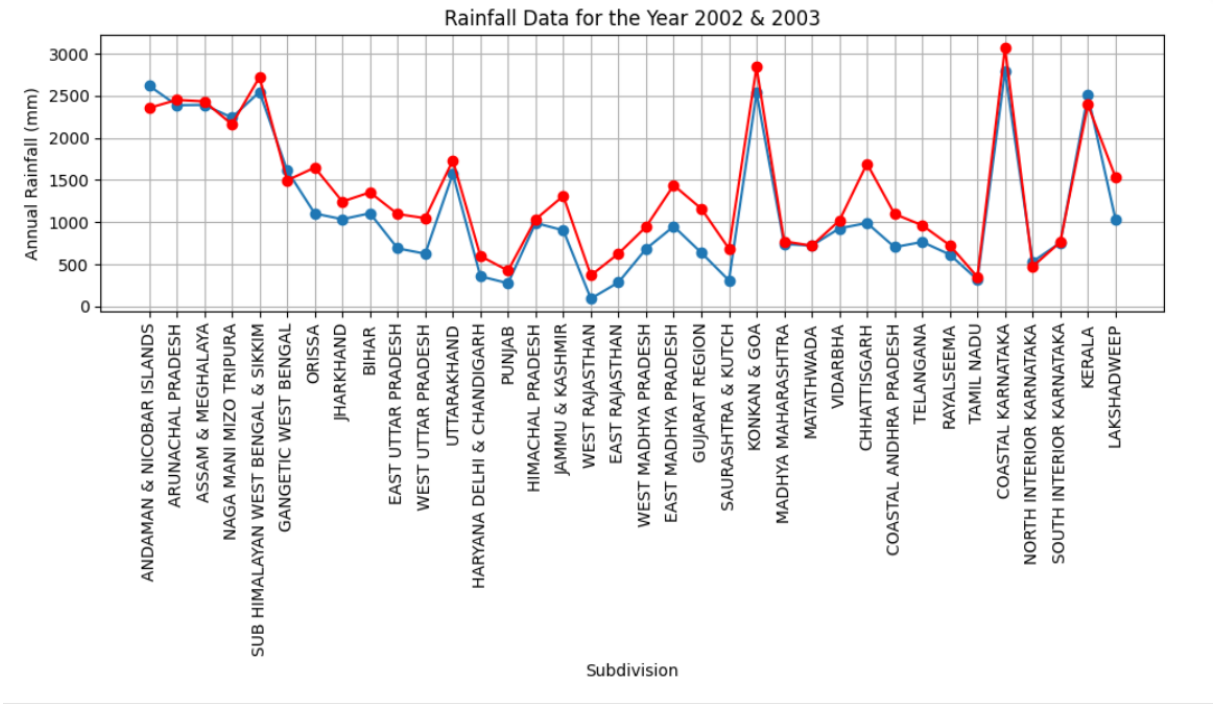

Fig 3.5: Comparison of annual rainfall between the year 2002 & 2003(Fig 3.3 & Fig 3.4)

a. 2003 appears to have been a wetter year overall, as indicated by the red line being higher than the blue line in many regions. This suggests that many parts of India received more rainfall in 2003 compared to 2002.
b. Some regions like Kerala and the northeastern states consistently receive high amounts of rainfall, regardless of the year.
c. In contrast, western and arid regions like Rajasthan and Gujarat receive lower rainfall, and the difference between the two years is more noticeable here.
d. The coastal regions, particularly around the Western Ghats (like Konkan & Goa), show a clear spike in 2003.

The graph highlights annual rainfall variations across different geographical regions in India, showing that certain regions, such as the northeastern states and the coastal regions, consistently receive more rainfall, while the arid regions like Rajasthan consistently receive less. Additionally, 2003 had generally higher rainfall compared to 2002 in many parts of the country, particularly in regions like Kerala, Konkan & Goa, and some parts of the northeast.

5. *CONCLUSION*

Daily rainfall prediction plays an important role in various sectors, including agriculture, water resource management, urban planning, and disaster management. With advancements in machine learning algorithms, the accuracy and reliability of rainfall predictions have significantly improved. These models leverage large datasets and refined computational techniques to capture complex patterns in meteorological data. The application of these models in daily rainfall prediction has demonstrated promising results due to its ability to handle nonlinear relationships and interactions between variables. By incorporating diverse data sources, such as satellite imagery, IoT sensors, and historical weather data, these models can provide high-resolution and timely forecasts. The potential to integrate these predictions with climate change models further enhances their value, offering insights into future rainfall patterns and aiding in long-term planning and adaptation strategies.

6. LIMITATION

Despite the advancements and potential of daily rainfall prediction using different types of machine learning techniques, these have several limitations also

i. The accuracy of rainfall predictions heavily depends on the quality and availability of data. In many regions, especially in developing countries, there may be a lack of wide ranging and high-quality atmospheric data. While XGBoost models are powerful, they can be complex and difficult to interpret. Understanding the model's decision-making process and the contribution of individual features can be challenging too, which may limit their acceptance in some applications where interpretability is crucial.

REFERENCES

[1] Chalachew Muluken Liyew and Haileyesus Amsaya Melese,2021 Dec 07, "Machine learning techniques to predict daily rainfall amount." https://doi.org/10.1186/s40537-021-00545-4

[2] S. Prabakaran, P. Naveen Kumar and P. Sai Mani Tarun,2017 June, "RAINFALL PREDICTION USING MODIFIED LINEAR REGRESSION".

[3] Sonali Pattanayak, Willis Towers Watson and D Nagesh Kumar, 2020," Review of Recent Advances in Climate Change Detection and Attribution Studies: A Large-Scale Hydro climatological Perspective". https://doi.org/10.2166/wcc.2020.091

[4] S. Swain, P. Patel, 2017 April, "A multiple linear regression model for precipitation forecasting over Cuttack district, Odisha, India". doi: 10.1109/I2CT.2017.8226150.

[5] Binh Thai Pham, Lu Minh Tien Thinh, KienTrinh Thi, Vu Hai Bang, Indra Prakash, 2020, "Development of advanced artificial intelligence models for daily rainfall prediction". https://doi.org/10.1016/j.atmosres.2020.104845

[6] R Vijayan, V Mareeswari, P Mohankumar, G Gunasekaran, K Srikar, June 2020," Estimating Rainfall Prediction using Machine Learning Techniques on a Dataset".

[7] Mia Lucas,2024, "The Role of AI in Climate Change Mitigation and Environmental Monitoring". DOI: 10.13140/RG.2.2.21153.38247

[8] Chandrasegar Thirumalai, K Sri Harsha, M Lakshmi Deepak, K Chaitanya Krishna, 2017, "Heuristic Prediction of Rainfall Using Machine Learning Techniques". DOI: 10.1109/ICOEI.2017.8300884

[9] Prof. Gayatri Naik, Mr.Tushar Patil, Miss.Akanksha Yadav, Miss.Jyoti Yadav, 2021," Prediction of Rainfall Using Machine Learning Techniques".

[10] Tharun V. P, Ramya Prakash, S. Renuga Devi, 2018 April, " Prediction of Rainfall Using Data Mining Techniques"'. DOI: 10.1109/ICICCT.2018.8473177

[11] Mohammad Kazemi Garajeh, Fatemeh Haji, Mahsa Tohidfar, Amin Sadeqi, Reyhaneh Ahmadi & Narges Kariminejad, 2024, "Spatiotemporal monitoring of climate change impacts on water resources using an integrated approach of remote sensing and Google Earth Engine". DOI:10.1038/s41598-024-56160-9

[12] N. Gnanasankaran, E. Ramaraj. 2020. "A Multiple Linear Regression Model to Predict Rainfall Using Indian Meteorological Data". *International Journal of Advanced Science and Technology, Vol-29 no 8s*

[13] Ting Zhang, Soung Yue Liew, Xiao Yan Huang, How Chinh Lee, Dong Hong Qin, 2021, "Research Trend Analysis of Artificial Intelligence Rainfall Prediction Algorithms Based on Knowledge Networks". IOP Conf. Ser.: Earth Environ. Sci. 945 012073. doi:10.1088/1755-1315/945/1/012073.

[14] Aditya Sai Srinivas T., Ramasubbareddy Somula, Govinda K., Akriti Saxena, Pramod Reddy A, 2019 April," Estimating rainfall using machine learning strategies based on weather radar data**"** DOI: 10.1002/dac.3999

[15] https://www.data.gov.in/resource/sub-divisional-monthly-rainfall-1901-2017

[16] Namitha K, Jayapriya A, SanthoshKumar G." Rainfall prediction using artificial neural network on map-reduce framework. ACM." 2015. https://doi.org/10.1145/2791405.2791468

[17] Arnav G, Kanchipuram Tamil Nadu. "Rainfall prediction using machine learning. Int J Innovative Sci Res Technol". 2019. 56–58.

[18] Vijayan R, Mareeswari V, Mohankumar P, Gunasekaran G, Srikar K, JUNE 2020, "Estimating rainfall prediction using machine learning techniques on a dataset". Int J Sci Technol Res. 2020;9(06):440–5.

[19] Salim Akhter Ansari1, Prince Kumar Raj, Sonu Kumar, KM Shaijal, Priyanka Garg,2024, " RAINFALL PREDICTION SYSTEM USING ML". IJLREC Volume 11, Issue 1, Page No.100-106

[20] R Praveena, T R Ganesh Babu, M Birunda, G Sudha, P Sukumar and J Gnanasoundharam, 2022, "Prediction of Rainfall Analysis Using Logistic Regression and Support Vector Machine". doi:10.1088/1742-6596/2466/1/012032