LEVERAGING MACHINE LEARNING FOR RAINFALL PREDICTION IN NORTH-CENTRAL NIGERIA: COMPARATIVE ALGORITHM STUDY

G. B. Balogun (1)
D. T. Kudabo (1)
O. J. Peter (2)*
A. G. Akintola (1)

Received: 27/01/2025
Revised: 09/03/2025
Accepted: 10/03/2025

[1] Department of Computer Science, University of Ilorin, Nigeria
[2] Department of Mathematical and Computer Sciences, University of Medical Sciences, Ondo City Ondo State, Nigeria
*Corresponding Author's Email: peterjames4real@gmail.com

https://doi.org/10.20428/jst.v30i4.2771

Leveraging Machine Learning for Rainfall Prediction in North-Central Nigeria: Comparative Algorithm Study

G. B. Balogun
Department of Computer Science, University of Ilorin, Nigeria
balogun.gb@unilorin.edu.ng

D. T. Kudabo
Department of Computer Science, University of Ilorin, Nigeria

O. J. Peter
Department of Mathematical and Computer Sciences, University of Medical Sciences, Ondo City Ondo State, Nigeria

A. G. Akintola
Department of Computer Science, University of Ilorin, Nigeria

Abstract— Rainfall has been a major worry in recent times because of the weather patterns that are constantly changing. The last ten years, in particular, have seen significant changes in rainfall patterns due to global warming. The principal determinants of this rainfall pattern include precipitation, dew points, wind speed, pressure, temperature, and humidity. For the purposes of agricultural growth and precise rainfall forecasting, it is essential to comprehend the relationship between these variables and rainfall behaviour. This is particularly true for the north central region of Nigeria, which is known as the country's "food basket." This work aims to investigate how machine learning techniques might infer precipitation patterns in north-central Nigeria. This study looked at several algorithms and evaluated how well they performed in respect to each variable that was connected to the goal variable. We also examine the performance of several machine learning methods in predicting rainfall. The outcome demonstrates the potential of the Random Forest Regression Algorithm as a flexible method for comprehending and controlling rainfall patterns. Thus, it is advised that the Nigerian Meteorological Agency (NIMET) use the outcome in conjunction with the traditional NWP (Numerical Weather Prediction) method to further improve rainfall prediction. The result will help farmers optimise their planting and harvesting schedules, assist water resource managers in planning for different water usages, allow disaster management authorities to issue timely warnings, support the conservation of natural resources, and ultimately promote economic development through infrastructure planning.

Keywords— Computer architecture; Neural networks; Predictive models; Biological system modeling; Neurons; Data models; ConvNet; Deep Learning; LSTM; Precipitation; Rainfall Prediction.

I. Introduction

Accurate rainfall prediction is vital for managing water resources, supporting sustainable agriculture, and mitigating socioeconomic impacts in regions dependent on rainfall. This is particularly relevant in North Central Nigeria, which features diverse geography, including the Jos Plateau, the Benue River Valley, and the lower Niger River Basin, with annual rainfall ranging from 1,000 to 1,400 millimetres during the rainy season from April to October. It rains in very complicated patterns that are affected by the Inter-Tropical Convergence Zone (ITCZ) and things like the El Niño Southern Oscillation (ENSO), the North Atlantic Oscillation (NAO), and the temperature of the Atlantic Ocean. The region's vulnerability to flooding and other climate-induced stressors underscores the importance of precise rainfall forecasting [1].

Advances in remote sensing, such as radar and satellite data, have significantly improved the accuracy of rainfall prediction. Tools like artificial neural networks (ANNs) and other machine learning models have been widely studied for their effectiveness in capturing complex, non-linear relationships in weather data. For instance, [2] showed that feed-forward neural networks can accurately predict rain in Manaus. Studies in West Africa have also shown that accurate rainfall forecasts can help with planning farms [3]. Fig. 1 shows the yearly average rainfall (mm) in North Central Nigeria, while Fig. 2 shows the average monthly temperature in North Central Nigeria. The orange line represents the daytime, while the blue line represents the nighttime (Courtesy WorldData.info).

Machine learning techniques, including Random Forest, Support Vector Machines (SVM), and ensemble models, have shown promise in rainfall prediction. These models use historical data on variables such as humidity, air pressure, and windspeed to deliver accurate forecasts. Recent research has emphasised combining machine learning algorithms with numerical weather prediction (NWP) methods to improve accuracy and address local challenges in north-central Nigeria [4]. For instance, the integration of ConvNet and LSTM networks has been effective in capturing precipitation patterns [5].

Flood forecasting is another critical application of rainfall prediction, particularly in high-risk areas where accurate forecasts can save lives and resources [6]. Studies in Bangkok, Thailand, and elsewhere have demonstrated that neural networks can improve short-term rainfall forecasts, aiding disaster preparedness and agricultural planning [7, 8].

Despite the progress, challenges, such as data quality and modelling uncertainties, persist. Addressing these issues through improved algorithms and expanded datasets is essential for enhancing the reliability of rainfall predictions. Combining machine learning with traditional NWP approaches offers a novel pathway for addressing the region's environmental and socioeconomic challenges.

This study is unique because it compares different machine learning algorithms for predicting rain in North Central Nigeria. Specifically, it looks at how the Random Forest Regression Algorithm and the NWP models work together. This approach enhances rainfall forecast accuracy and provides actionable insights for agricultural planning, water resource management, and disaster mitigation, offering a unique contribution to addressing the region's pressing environmental and economic challenges.

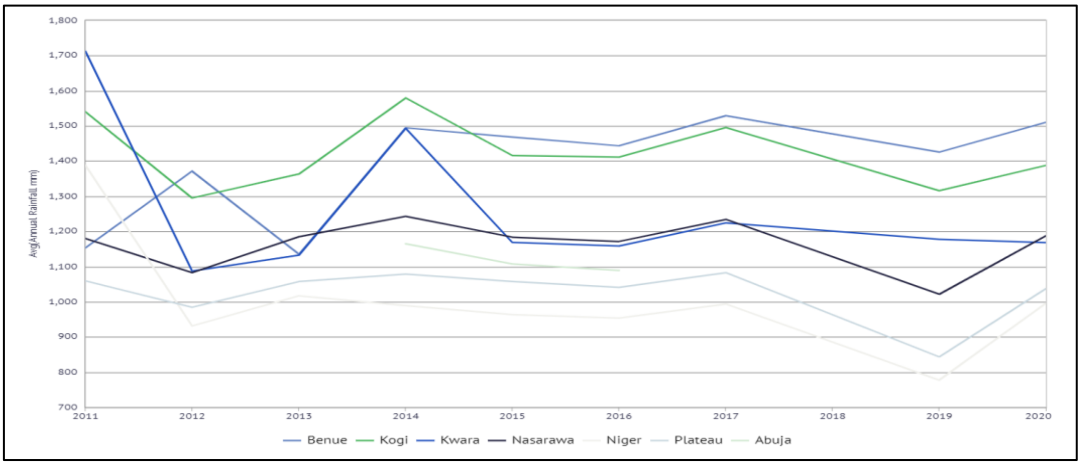


Fig. 1. Yearly Average rainfall (mm) in North Central Nigeria. (Courtesy NIMET)

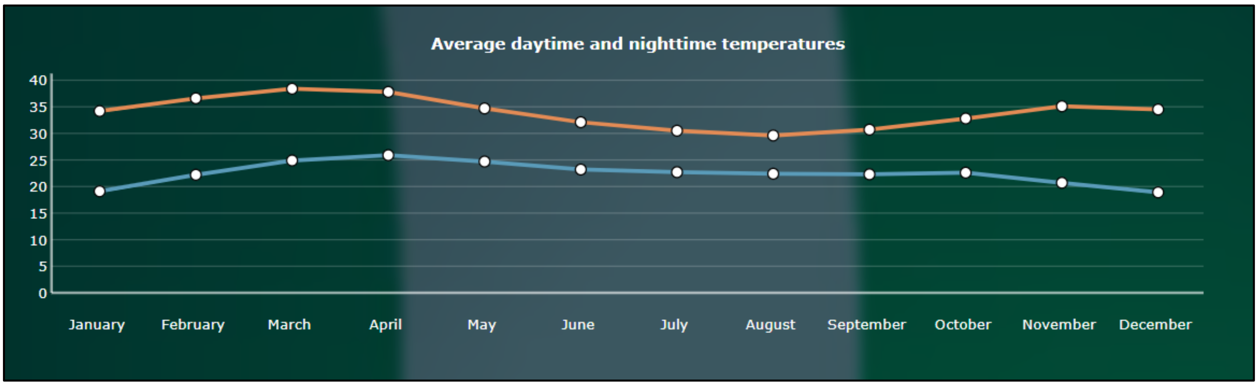


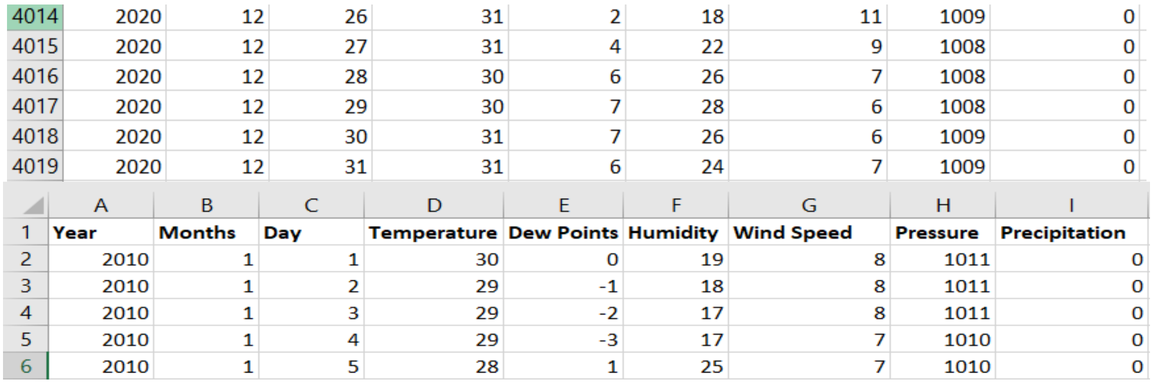
Fig. 2. Average Monthly Temperature in North Central Nigeria. (Orange line (Day), Blue (Nighttime) (Courtesy WorldData.info).

II. Methodology

In the fascinating and quickly expanding field of computer technology known as "machine learning," researchers explore the algorithms that allow computers and other devices to automatically learn new skills and increase their efficiency through testing and training with different factors. Machine learning aims to teach computers to carry out activities like data processing, recognition, and prediction that would otherwise need human interaction. Machine learning may improve the precision and effectiveness of data processing in many domains, including artificial intelligence applications in computer engineering and medicine, by employing powerful algorithms and tools. Among its numerous advantages is the reduced necessity for labour-intensive, error-prone manual tasks. Creating rules for input data is a crucial part of machine learning since it enables computers to handle similar scenarios more quickly and effectively. Prediction and comprehending the transformation of variables into vectors are two further areas of emphasis for machine learning. All things considered, machine learning is a crucial part of contemporary computer technology and has many real-world uses across a range of sectors. As long as we continue to develop and advance, the future of computing and technology holds enormous promise.

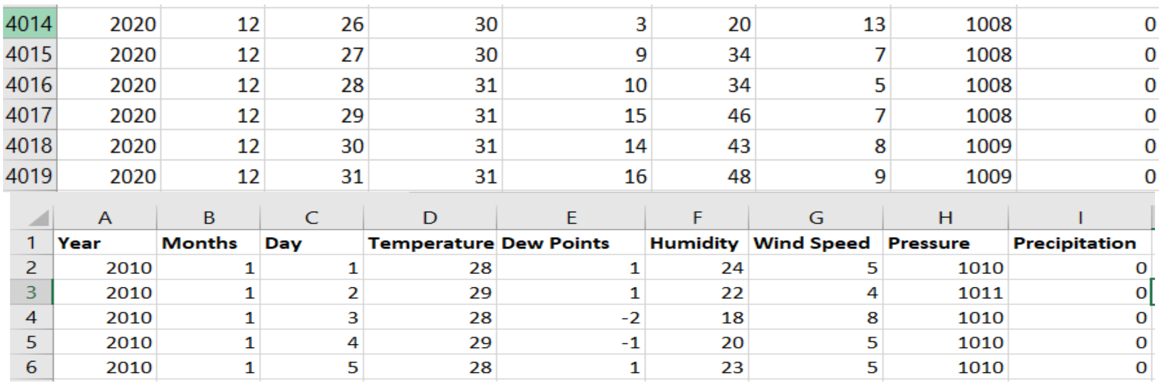
III. Data Collection

The Global Campaign for Climate Action (GCCA) collected the rainfall data over a 10-year period (2010–2020), spatially distributed across North-Central Nigeria.



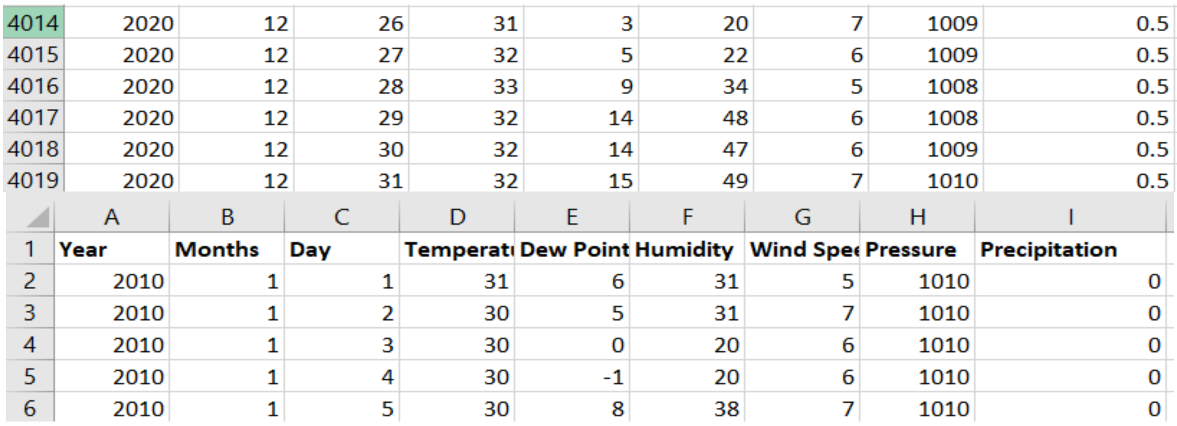
4014	2020	12	26	31	2	18	11	1009	0
4015	2020	12	27	31	4	22	9	1008	0
4016	2020	12	28	30	6	26	7	1008	0
4017	2020	12	29	30	7	28	6	1008	0
4018	2020	12	30	31	7	26	6	1009	0
4019	2020	12	31	31	6	24	7	1009	0
	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
2	2010	1	1	30	0	19	8	1011	0
3	2010	1	2	29	-1	18	8	1011	0
4	2010	1	3	29	-2	17	8	1011	0
5	2010	1	4	29	-3	17	7	1010	0
6	2010	1	5	28	1	25	7	1010	0

Fig. 3. Abuja Dataset including the temperature, dew points wind speed, pressure, precipitation.



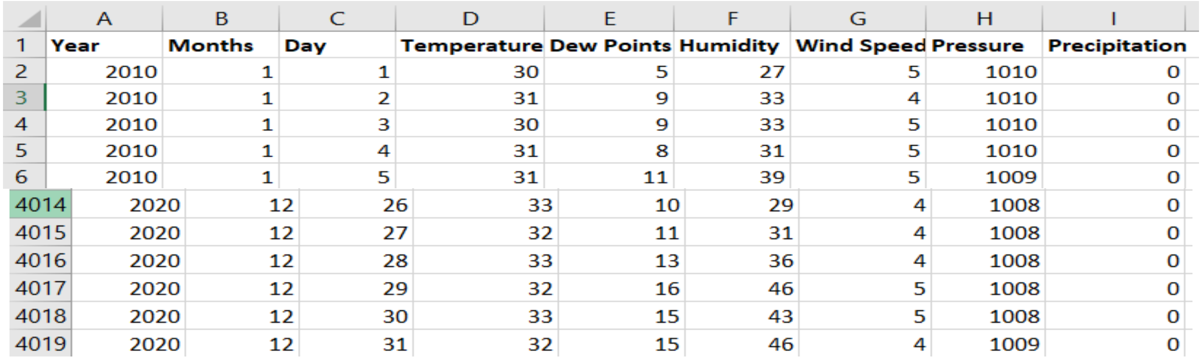
4014	2020	12	26	30	3	20	13	1008	0
4015	2020	12	27	30	9	34	7	1008	0
4016	2020	12	28	31	10	34	5	1008	0
4017	2020	12	29	31	15	46	7	1008	0
4018	2020	12	30	31	14	43	8	1009	0
4019	2020	12	31	31	16	48	9	1009	0
	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
2	2010	1	1	28	1	24	5	1010	0
3	2010	1	2	29	1	22	4	1011	0
4	2010	1	3	28	-2	18	8	1010	0
5	2010	1	4	29	-1	20	5	1010	0
6	2010	1	5	28	1	23	5	1010	0

Fig. 4. Benue Dataset including the temperature, dew points wind speed, pressure, precipitation.



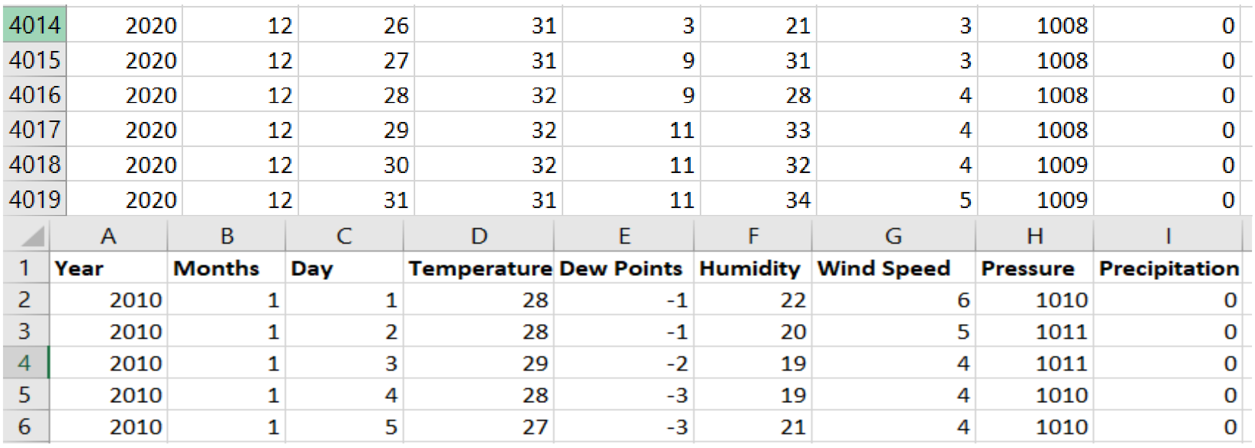
4014	2020	12	26	31	3	20	7	1009	0.5
4015	2020	12	27	32	5	22	6	1009	0.5
4016	2020	12	28	33	9	34	5	1008	0.5
4017	2020	12	29	32	14	48	6	1008	0.5
4018	2020	12	30	32	14	47	6	1009	0.5
4019	2020	12	31	32	15	49	7	1010	0.5
	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperatı	Dew Point	Humidity	Wind Spee	Pressure	Precipitation
2	2010	1	1	31	6	31	5	1010	0
3	2010	1	2	30	5	31	7	1010	0
4	2010	1	3	30	0	20	6	1010	0
5	2010	1	4	30	-1	20	6	1010	0
6	2010	1	5	30	8	38	7	1010	0

Fig. 4. Kwara Dataset including the temperature, dew points wind speed, pressure, precipitation.



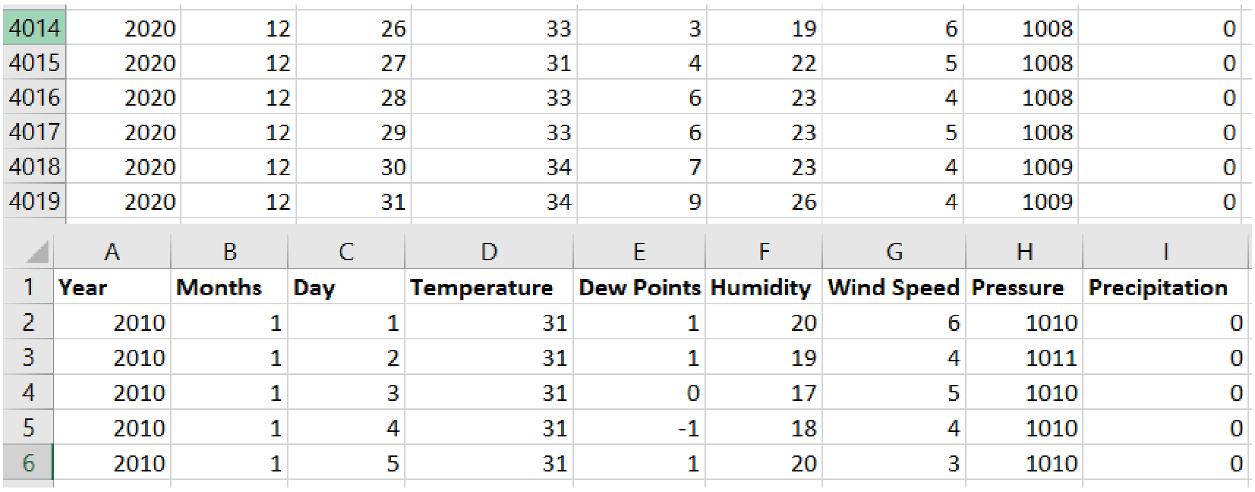
	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
2	2010	1	1	30	5	27	5	1010	0
3	2010	1	2	31	9	33	4	1010	0
4	2010	1	3	30	9	33	5	1010	0
5	2010	1	4	31	8	31	5	1010	0
6	2010	1	5	31	11	39	5	1009	0
4014	2020	12	26	33	10	29	4	1008	0
4015	2020	12	27	32	11	31	4	1008	0
4016	2020	12	28	33	13	36	4	1008	0
4017	2020	12	29	32	16	46	5	1008	0
4018	2020	12	30	33	15	43	5	1008	0
4019	2020	12	31	32	15	46	4	1009	0

Fig. 5. Kogi Dataset including the temperature, dew points wind speed, pressure, precipitation.



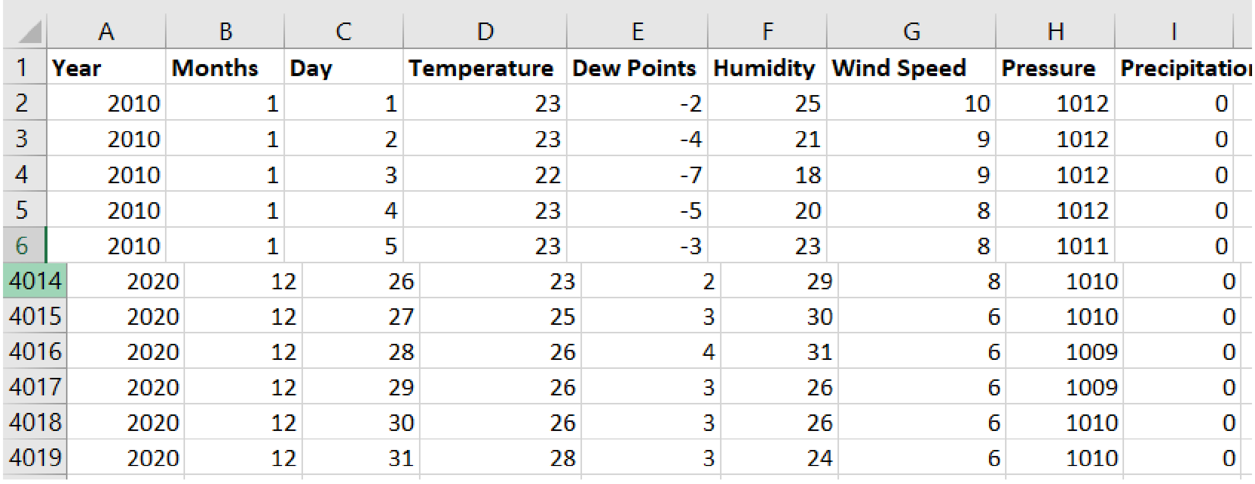
4014	2020	12	26	31	3	21	3	1008	0
4015	2020	12	27	31	9	31	3	1008	0
4016	2020	12	28	32	9	28	4	1008	0
4017	2020	12	29	32	11	33	4	1008	0
4018	2020	12	30	32	11	32	4	1009	0
4019	2020	12	31	31	11	34	5	1009	0
	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
2	2010	1	1	28	-1	22	6	1010	0
3	2010	1	2	28	-1	20	5	1011	0
4	2010	1	3	29	-2	19	4	1011	0
5	2010	1	4	28	-3	19	4	1010	0
6	2010	1	5	27	-3	21	4	1010	0

Fig. 6. Nasarawa Dataset including the temperature, dew points wind speed, pressure, precipitation.



4014	2020	12	26	33	3	19	6	1008	0
4015	2020	12	27	31	4	22	5	1008	0
4016	2020	12	28	33	6	23	4	1008	0
4017	2020	12	29	33	6	23	5	1008	0
4018	2020	12	30	34	7	23	4	1009	0
4019	2020	12	31	34	9	26	4	1009	0
	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
2	2010	1	1	31	1	20	6	1010	0
3	2010	1	2	31	1	19	4	1011	0
4	2010	1	3	31	0	17	5	1010	0
5	2010	1	4	31	-1	18	4	1010	0
6	2010	1	5	31	1	20	3	1010	0

Fig. 7. Niger Dataset including the temperature, dew points wind speed, pressure, precipitation



	A	B	C	D	E	F	G	H	I
1	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitatio
2	2010	1	1	23	-2	25	10	1012	0
3	2010	1	2	23	-4	21	9	1012	0
4	2010	1	3	22	-7	18	9	1012	0
5	2010	1	4	23	-5	20	8	1012	0
6	2010	1	5	23	-3	23	8	1011	0
4014	2020	12	26	23	2	29	8	1010	0
4015	2020	12	27	25	3	30	6	1010	0
4016	2020	12	28	26	4	31	6	1009	0
4017	2020	12	29	26	3	26	6	1009	0
4018	2020	12	30	26	3	26	6	1010	0
4019	2020	12	31	28	3	24	6	1010	0

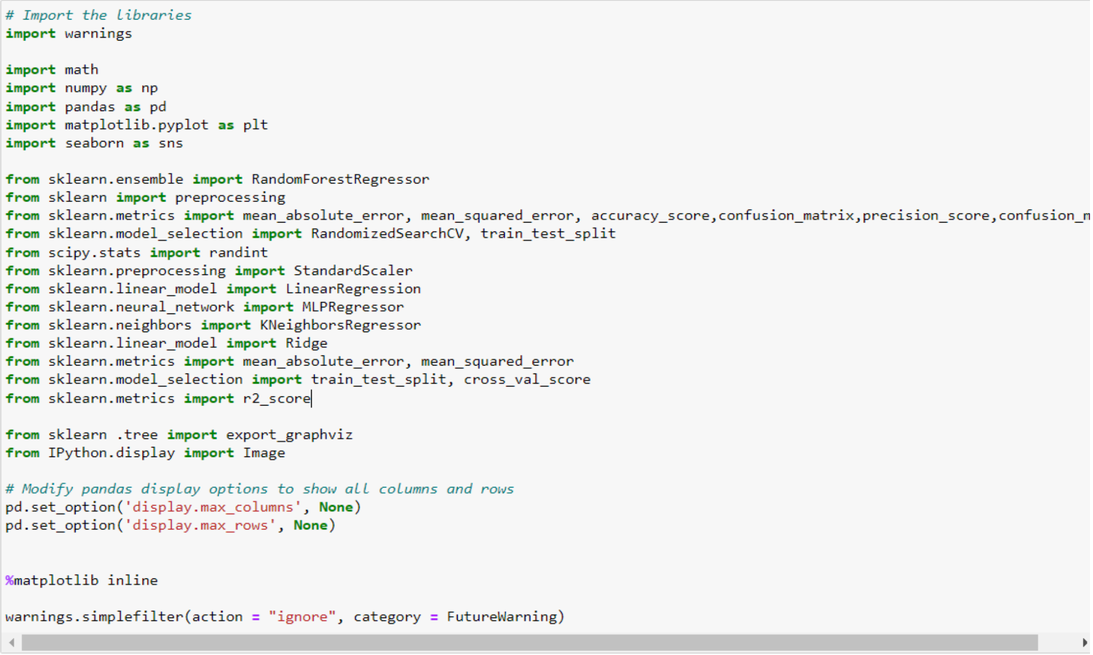
Fig. 8. Plateau Dataset including the temperature, dew points wind speed, pressure, precipitation.

IV. IMPLEMENTATION

Ten sections make up the structure of the project implementation. First, the necessary libraries will be brought in and thoroughly examined. The dataset will next be prepared by choosing the required attributes and carrying out the required data transformations. To learn more about the dataset, data analysis, including correlation analysis, will be done. To make it easier to evaluate the model and extract features, the dataset will next be divided into train and test sets. Using the training data, machine learning models will be created and trained. For assessment, the test data will be utilized to generate predictions using the trained models.

https://doi.org/10.20428/jst.v30i4.2771

Model development will entail improving hyper parameters and fine-tuning the models to get better performance. The model that performs the best will be chosen for deployment. To maintain the accuracy and dependability of the model over time, monitoring and maintenance will be carried out. The complete machine learning process, including data pretreatment, model selection, assessment outcomes, and any model interpretability analysis, will be documented.



```
# Import the libraries
import warnings

import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.ensemble import RandomForestRegressor
from sklearn import preprocessing
from sklearn.metrics import mean_absolute_error, mean_squared_error, accuracy_score,confusion_matrix,precision_score,confusion_m
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from scipy.stats import randint
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import r2_score

from sklearn .tree import export_graphviz
from IPython.display import Image

# Modify pandas display options to show all columns and rows
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)


%matplotlib inline

warnings.simplefilter(action = "ignore", category = FutureWarning)
```

Fig. 9. Importing Libraries

A. STEP 1—Importing Libraries

Warnings: The code uses the warnings library to control the display of warning messages.

Math: The math library provides mathematical functions and operations.

NumPy: NumPy is a fundamental library for numerical computing in Python.

Pandas: Pandas is a powerful data manipulation and analysis library.

Matplotlib: Matplotlib is a widely used plotting library in Python. Seaborn is a data visualisation library built on top of Matplotlib.

Scikit-Learn (sklearn): Scikit-Learn is a comprehensive machine learning library for Python.

Standard Scaler: Standard Scaler is a preprocessing method in Scikit-Learn for standardising features by removing the mean and scaling it to unit variance.

Linear Regression: Linear regression is a basic regression algorithm for modelling the relationship between a dependent variable and one or more independent variables.

MLP Regressor (Multi-layer Perceptron Regressor): MLPRegressor is a neural network-based regression algorithm in Scikit-Learn.

KNeighborsRegressor: The KNeighborsRegressor is a regression algorithm based on K-nearest neighbours.

Ridge Regression: Ridge Regression is a linear regression technique that includes L2 regularisation to prevent overfitting. Scikit-Learn's Ridge class provides an implementation for Ridge Regression.

Train-Test Split and Cross-Validation: Techniques for evaluating machine learning models' performance.

R2 Score: R2 Score (Coefficient of Determination) is a metric that quantifies the proportion of the variance in the dependent variable that is predictable from the independent variables.

RandomForestRegressor: Random Forest Regressor is a machine learning model from the scikit-learn library.

Preprocessing: The preprocessing module from scikit-learn provides functions for data preprocessing tasks such as scaling, encoding, and imputing missing values.

Mean Absolute Error (MAE), Mean Squared Error (MSE), Accuracy Score, Confusion Matrix, and Precision Score: These are evaluation metrics from the scikit-learn library used to measure the performance of machine learning models.

Grid Search CV: Grid Search CV is a hyperparameter tuning technique in machine learning. It searches through a specified hyperparameter grid to find the best combination for the highest model performance.

Randint: The randint function from the scipy.stats module generates random integers from a specified discrete uniform distribution.

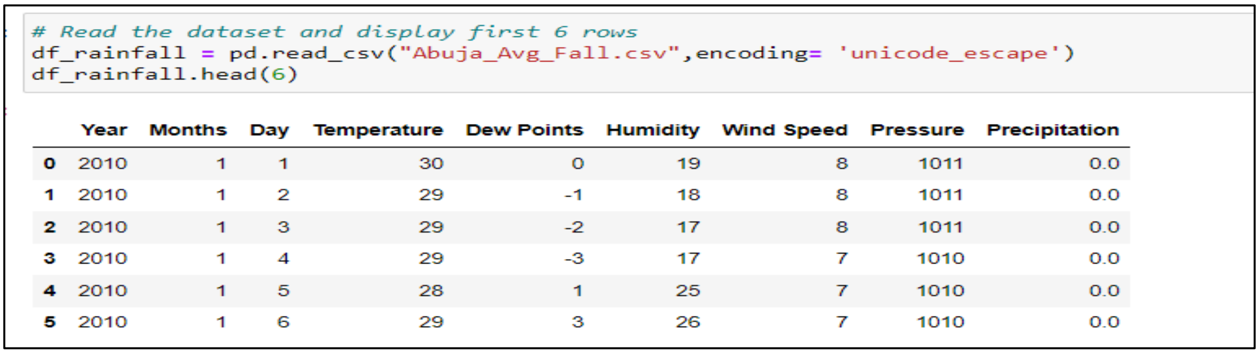
Export_graphviz: The export_graphviz function from the sklearn.tree module is used to export decision tree models in a Graphviz dot format.

Image: The Image module from the Ipython.display library is used to display images within the Jupyter Notebook environment.

These libraries provide various functionalities and tools that are essential for data preprocessing, visualisation, machine learning modelling, and model evaluation. After splitting the data into a training set and a test set, the training process begins.

B. STEP 2— Preparing Dataset

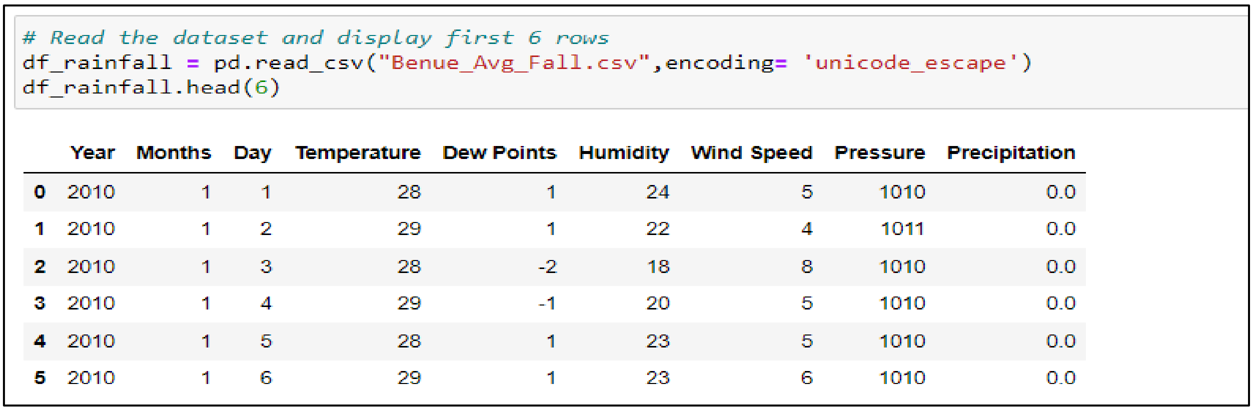
The dataset for our case study has been carefully prepared by selecting and extracting the necessary attributes from various datasets. This process ensures that we have the relevant information to address our research objectives effectively. Before proceeding with further analysis, it is crucial to develop a comprehensive understanding of our dataset. By exploring its characteristics and gaining insights from the data, we can establish a solid foundation for our subsequent analysis and model development.



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Abuja_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
0	2010	1	1	30	0	19	8	1011	0.0
1	2010	1	2	29	-1	18	8	1011	0.0
2	2010	1	3	29	-2	17	8	1011	0.0
3	2010	1	4	29	-3	17	7	1010	0.0
4	2010	1	5	28	1	25	7	1010	0.0
5	2010	1	6	29	3	26	7	1010	0.0

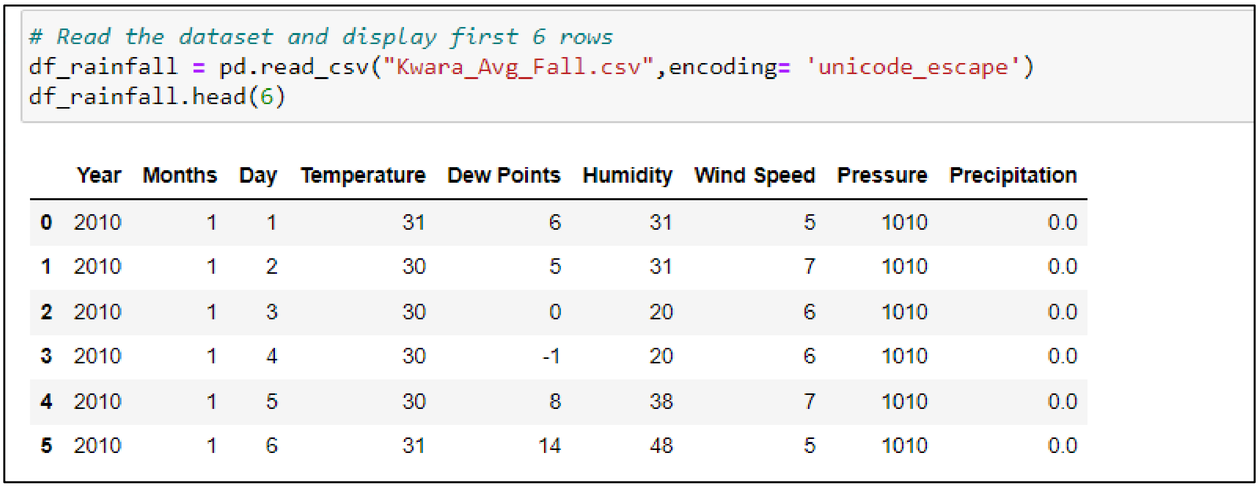
Fig. 10. Loading Abuja Dataset



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Benue_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
0	2010	1	1	28	1	24	5	1010	0.0
1	2010	1	2	29	1	22	4	1011	0.0
2	2010	1	3	28	-2	18	8	1010	0.0
3	2010	1	4	29	-1	20	5	1010	0.0
4	2010	1	5	28	1	23	5	1010	0.0
5	2010	1	6	29	1	23	6	1010	0.0

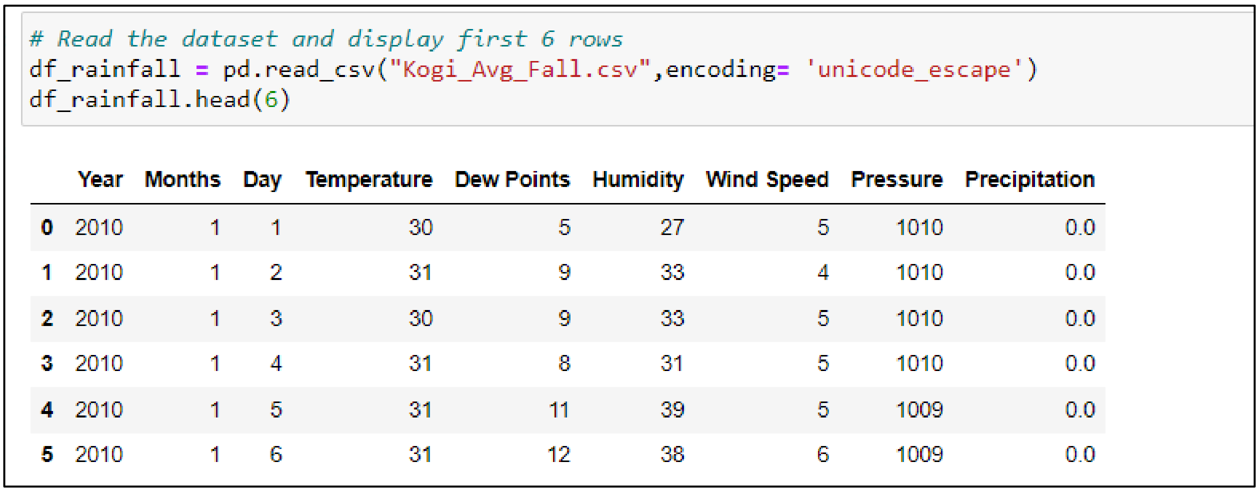
Fig. 11. Loading Benue Dataset



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Kwara_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
0	2010	1	1	31	6	31	5	1010	0.0
1	2010	1	2	30	5	31	7	1010	0.0
2	2010	1	3	30	0	20	6	1010	0.0
3	2010	1	4	30	-1	20	6	1010	0.0
4	2010	1	5	30	8	38	7	1010	0.0
5	2010	1	6	31	14	48	5	1010	0.0

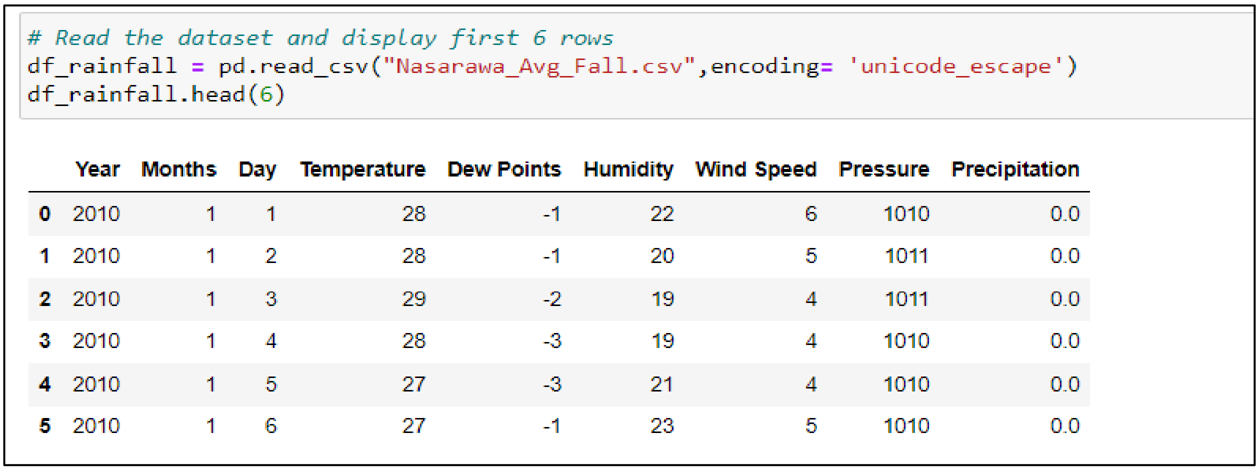
Fig. 12. Loading Kwara Dataset



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Kogi_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
0	2010	1	1	30	5	27	5	1010	0.0
1	2010	1	2	31	9	33	4	1010	0.0
2	2010	1	3	30	9	33	5	1010	0.0
3	2010	1	4	31	8	31	5	1010	0.0
4	2010	1	5	31	11	39	5	1009	0.0
5	2010	1	6	31	12	38	6	1009	0.0

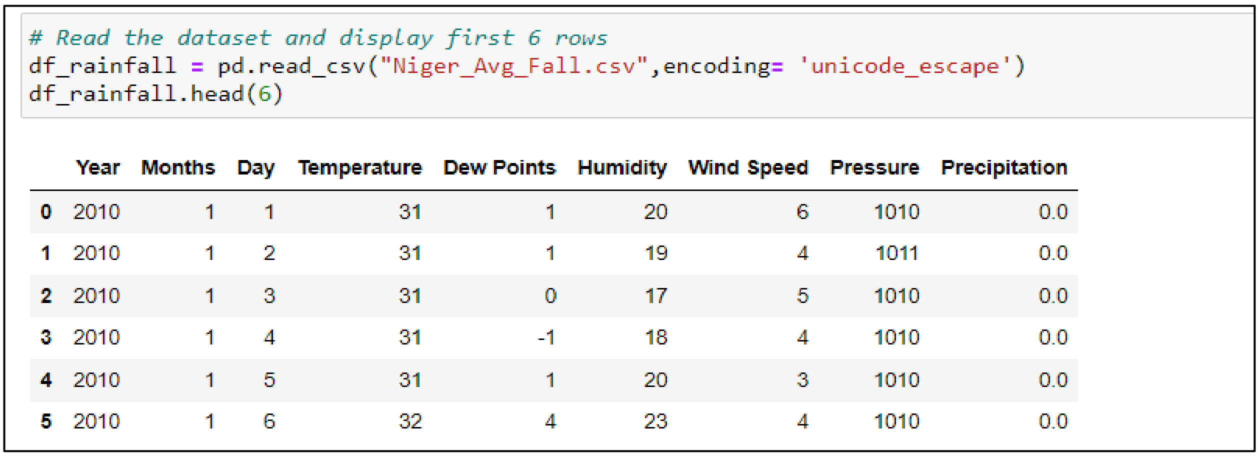
Fig. 13. Loading Kogi Dataset



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Nasarawa_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
0	2010	1	1	28	-1	22	6	1010	0.0
1	2010	1	2	28	-1	20	5	1011	0.0
2	2010	1	3	29	-2	19	4	1011	0.0
3	2010	1	4	28	-3	19	4	1010	0.0
4	2010	1	5	27	-3	21	4	1010	0.0
5	2010	1	6	27	-1	23	5	1010	0.0

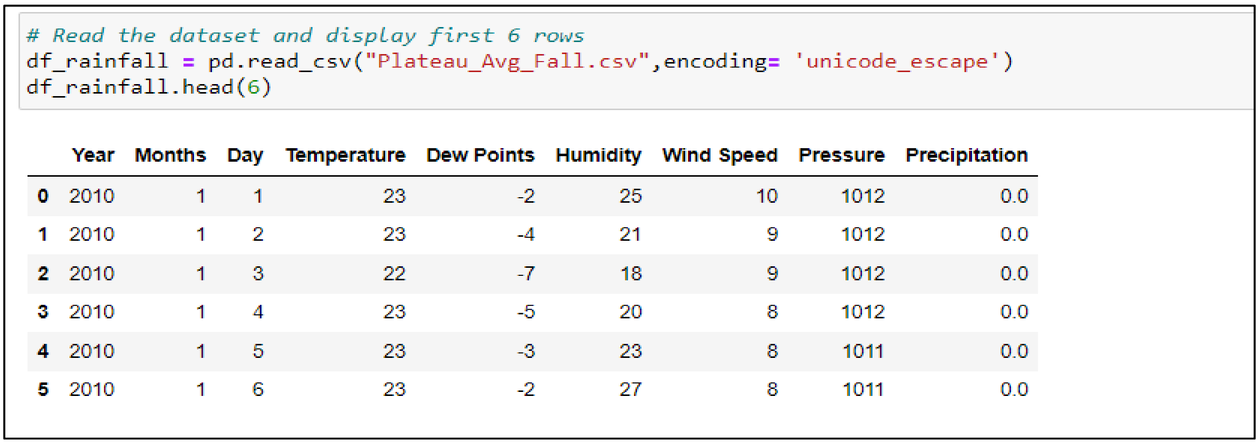
Fig. 14. Loading Nasarawa Dataset



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Niger_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
0	2010	1	1	31	1	20	6	1010	0.0
1	2010	1	2	31	1	19	4	1011	0.0
2	2010	1	3	31	0	17	5	1010	0.0
3	2010	1	4	31	-1	18	4	1010	0.0
4	2010	1	5	31	1	20	3	1010	0.0
5	2010	1	6	32	4	23	4	1010	0.0

Fig. 15. Loading Niger Dataset



```
# Read the dataset and display first 6 rows
df_rainfall = pd.read_csv("Plateau_Avg_Fall.csv",encoding= 'unicode_escape')
df_rainfall.head(6)
```

```
   Year  Months  Day  Temperature  Dew Points  Humidity  Wind Speed  Pressure  Precipitation
0  2010       1    1           23          -2        25          10      1012            0.0
1  2010       1    2           23          -4        21           9      1012            0.0
2  2010       1    3           22          -7        18           9      1012            0.0
3  2010       1    4           23          -5        20           8      1012            0.0
4  2010       1    5           23          -3        23           8      1011            0.0
5  2010       1    6           23          -2        27           8      1011            0.0
```

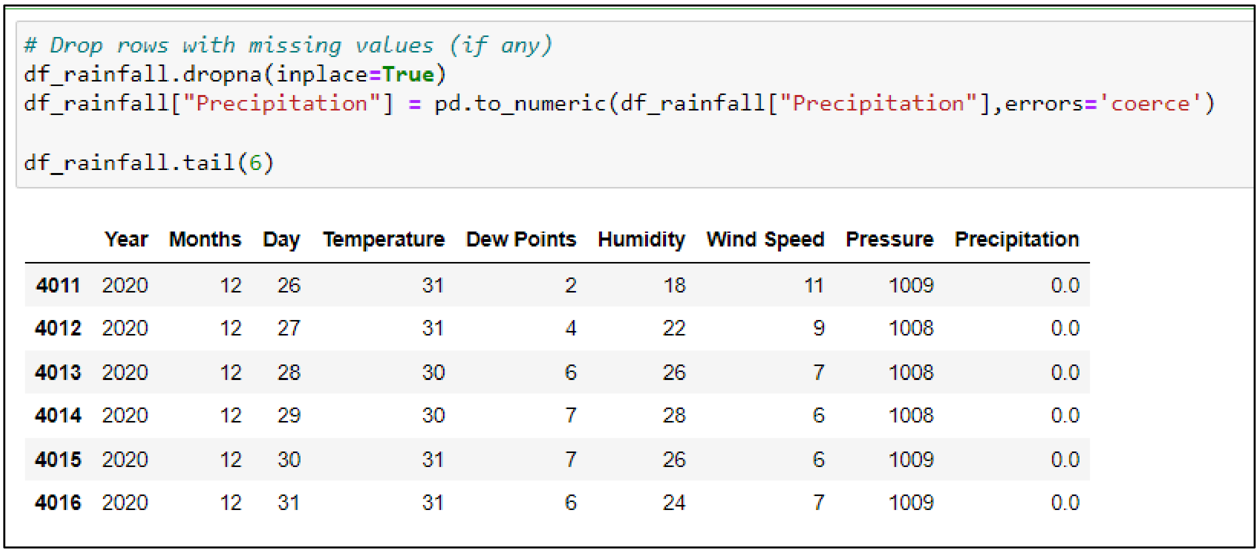
Fig. 16. Loading Plateau Dataset

C. STEP 3— Data Preprocessing

This work focuses on the exact and accurate prediction of rainfall by machine learning methods. The main goal is to efficiently anticipate rainfall by utilising machine learning capabilities. The study uses a thorough strategy that includes closely examining source data during the pre-processing stage in order to accomplish this goal. Next, this data is used in the processing stage to produce precise and effective rainfall forecasts.

Statistical data from the Global Campaign for Climate Action (GCCA) has been gathered in order to conduct this analysis. The dataset includes temperature, dew point, wind speed, air pressure, precipitation, humidity, and other critical rainfall-related characteristics. For the corresponding locations of Benue, Kogi, Kwara, Nasarawa, Niger, Plateau, and Federal Capital Territory (Abuja), each of these criteria is examined separately. The study's statistical data is a useful source of information for the processing stage, which is where rainfall prediction is ultimately achieved. The project attempts to generate accurate and dependable rainfall forecasts by utilising suitable machine learning algorithms and approaches. This research project has the potential to advance our knowledge of rainfall patterns and lead to future improvements in forecasting techniques. An essential step in getting our dataset ready for model training is data preprocessing. Given the inherent difficulties with real-world data, like noise, incompleteness, inconsistencies, and undesirable data, this phase is essential.

Preprocessing approaches are used to try to mitigate these problems and make sure our models are reliable. This entails a number of actions, such as managing missing values, implementing suitable encoding methods, and carrying out data scaling. Managing missing values is the most crucial phase in this process. The distribution of our data can be greatly impacted by the existence of null values, which may result in forecasts that are not accurate. As a result, eliminating null values is crucial to preserving data integrity and raising the standard of our analysis. It is noteworthy that the dataset has a comparatively low number of null values, which emphasises how important it is to deal with them successfully. We may improve our dataset’s consistency, quality, and dependability by carefully applying data preprocessing techniques. This will pave the way for a strong model during training and precise predictions.



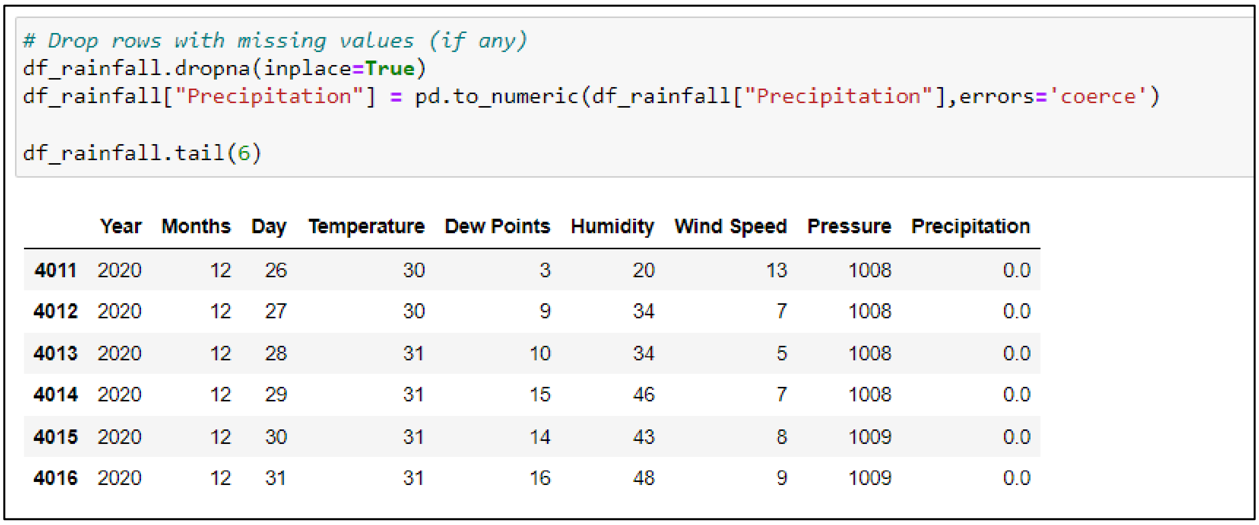
```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

```
      Year  Months  Day  Temperature  Dew Points  Humidity  Wind Speed  Pressure  Precipitation
4011  2020      12   26           31           2        18          11      1009            0.0
4012  2020      12   27           31           4        22           9      1008            0.0
4013  2020      12   28           30           6        26           7      1008            0.0
4014  2020      12   29           30           7        28           6      1008            0.0
4015  2020      12   30           31           7        26           6      1009            0.0
4016  2020      12   31           31           6        24           7      1009            0.0
```

Fig. 17. Preprocessing the Abuja Dataset

https://doi.org/10.20428/jst.v30i4.2771

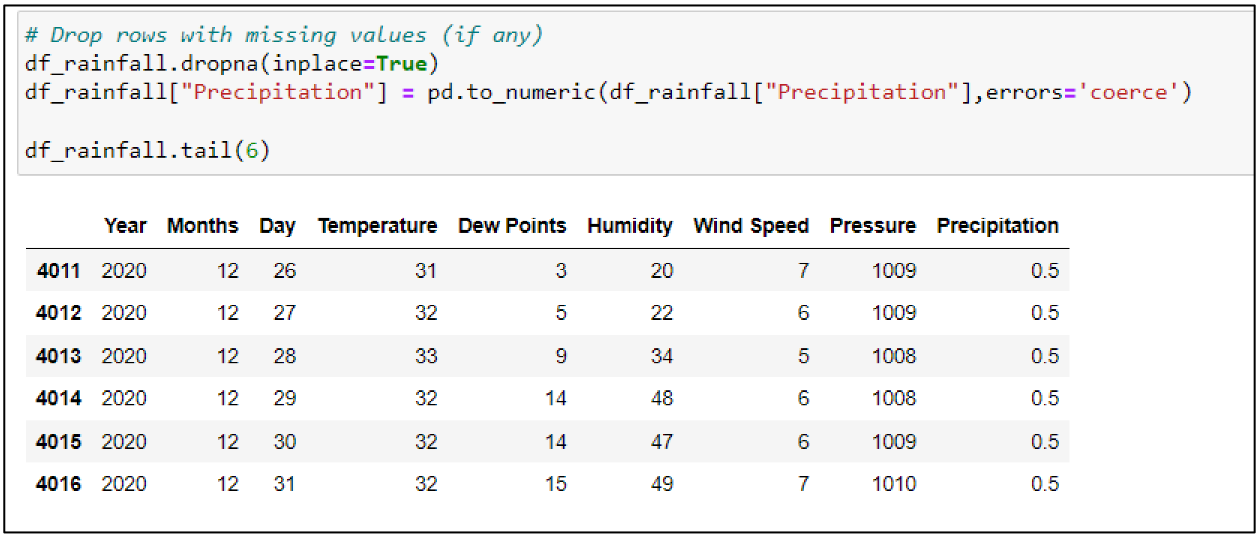


```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
4011	2020	12	26	30	3	20	13	1008	0.0
4012	2020	12	27	30	9	34	7	1008	0.0
4013	2020	12	28	31	10	34	5	1008	0.0
4014	2020	12	29	31	15	46	7	1008	0.0
4015	2020	12	30	31	14	43	8	1009	0.0
4016	2020	12	31	31	16	48	9	1009	0.0

Fig. 18. Preprocessing the Benue Dataset

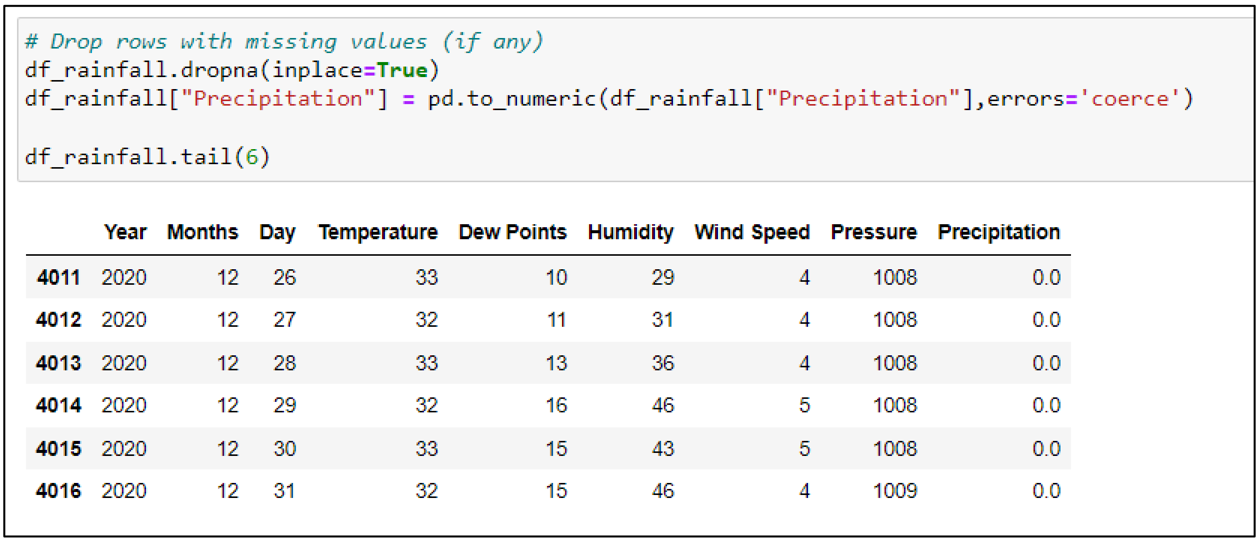


```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
4011	2020	12	26	31	3	20	7	1009	0.5
4012	2020	12	27	32	5	22	6	1009	0.5
4013	2020	12	28	33	9	34	5	1008	0.5
4014	2020	12	29	32	14	48	6	1008	0.5
4015	2020	12	30	32	14	47	6	1009	0.5
4016	2020	12	31	32	15	49	7	1010	0.5

Fig. 19. Preprocessing the Kwara Dataset

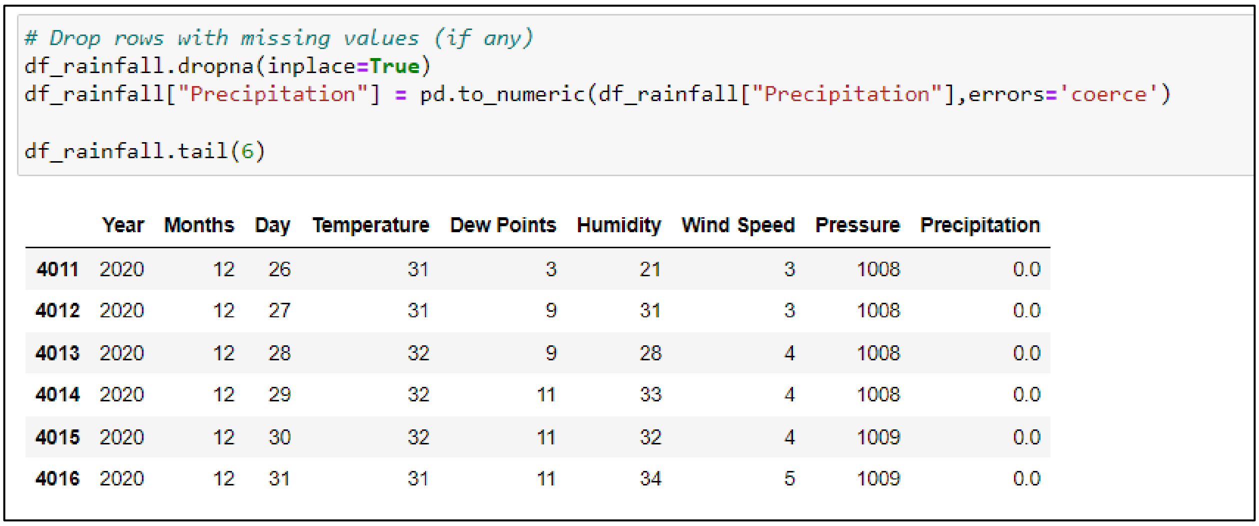


```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

	Year	Months	Day	Temperature	Dew Points	Humidity	Wind Speed	Pressure	Precipitation
4011	2020	12	26	33	10	29	4	1008	0.0
4012	2020	12	27	32	11	31	4	1008	0.0
4013	2020	12	28	33	13	36	4	1008	0.0
4014	2020	12	29	32	16	46	5	1008	0.0
4015	2020	12	30	33	15	43	5	1008	0.0
4016	2020	12	31	32	15	46	4	1009	0.0

Fig. 20. Preprocessing the Kogi Dataset

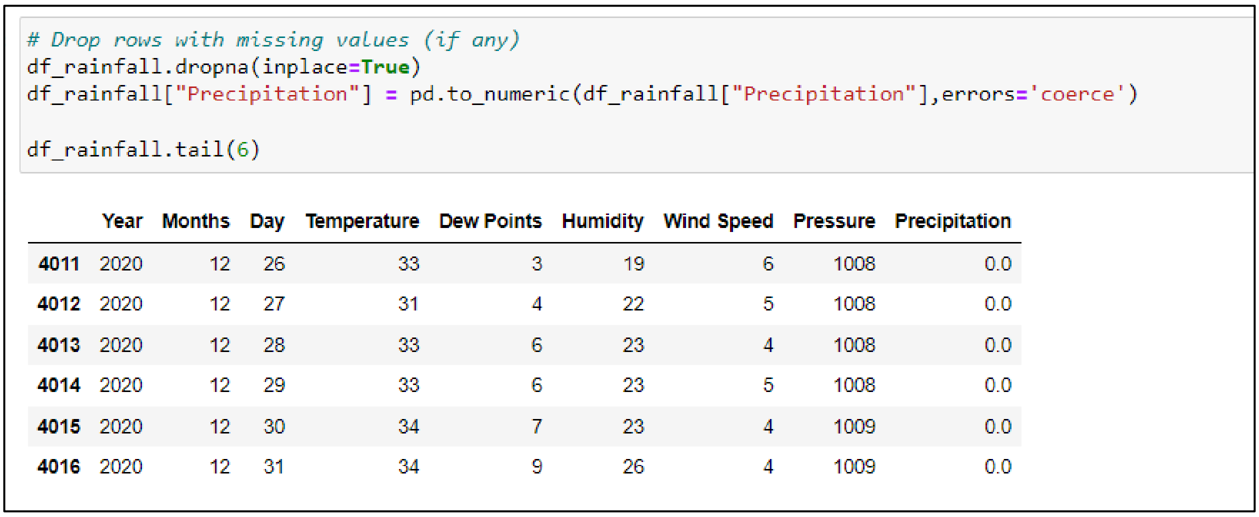


```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

```
      Year  Months  Day  Temperature  Dew Points  Humidity  Wind Speed  Pressure  Precipitation
4011  2020      12   26           31           3        21           3      1008            0.0
4012  2020      12   27           31           9        31           3      1008            0.0
4013  2020      12   28           32           9        28           4      1008            0.0
4014  2020      12   29           32          11        33           4      1008            0.0
4015  2020      12   30           32          11        32           4      1009            0.0
4016  2020      12   31           31          11        34           5      1009            0.0
```

Fig. 21. Preprocessing the Nasarawa Dataset

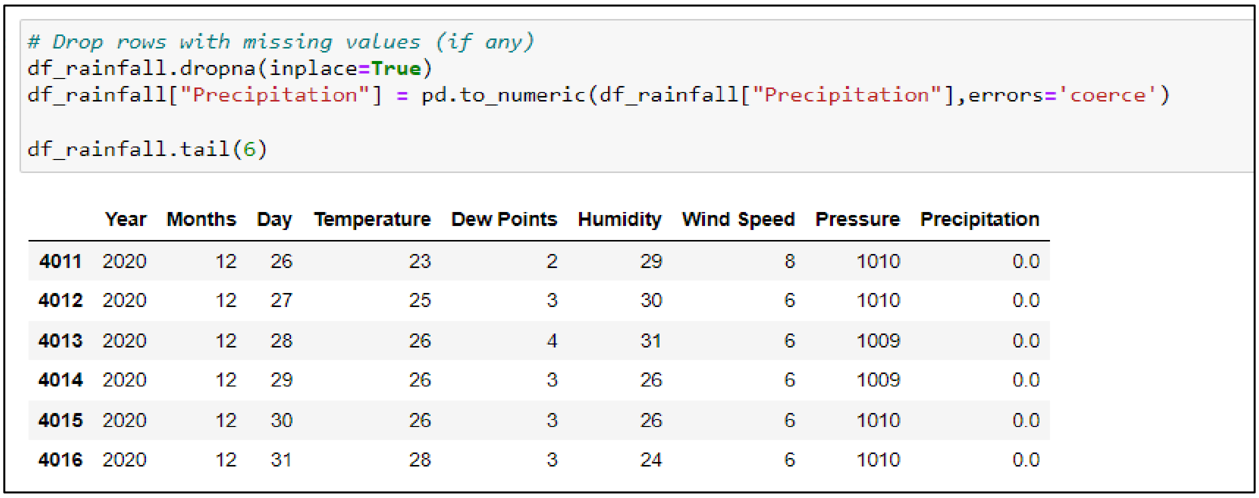


```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

```
      Year  Months  Day  Temperature  Dew Points  Humidity  Wind Speed  Pressure  Precipitation
4011  2020      12   26           33           3        19           6      1008            0.0
4012  2020      12   27           31           4        22           5      1008            0.0
4013  2020      12   28           33           6        23           4      1008            0.0
4014  2020      12   29           33           6        23           5      1008            0.0
4015  2020      12   30           34           7        23           4      1009            0.0
4016  2020      12   31           34           9        26           4      1009            0.0
```

Fig. 22. Preprocessing the Niger Dataset



```
# Drop rows with missing values (if any)
df_rainfall.dropna(inplace=True)
df_rainfall["Precipitation"] = pd.to_numeric(df_rainfall["Precipitation"],errors='coerce')

df_rainfall.tail(6)
```

```
      Year  Months  Day  Temperature  Dew Points  Humidity  Wind Speed  Pressure  Precipitation
4011  2020      12   26           23           2        29           8      1010            0.0
4012  2020      12   27           25           3        30           6      1010            0.0
4013  2020      12   28           26           4        31           6      1009            0.0
4014  2020      12   29           26           3        26           6      1009            0.0
4015  2020      12   30           26           3        26           6      1010            0.0
4016  2020      12   31           28           3        24           6      1010            0.0
```

Fig. 23. Preprocessing the Plateau Dataset

Missing Values: To address the issue, we use imputation techniques to substitute missing values. There are several imputation techniques, including random sampling imputation, mean, median, and mode imputation. The type of data at hand determines which imputation technique is used. We have employed median imputation in our investigation to address missing values, guaranteeing a fair strategy that is consistent with the features of our dataset.

https://doi.org/10.20428/jst.v30i4.2771

Label Encoding: One kind of encoding method used to convert categorical variables into numerical variables is label encoding. Since most machine learning algorithms can only handle numerical data, this conversion is crucial. By giving each category a distinct number identifier, we make it possible for our models to understand and process these variables efficiently. This encoding technique is very useful when working with categorical data because it makes it easier to extract important insights and patterns from the dataset.

We improve the quality and usefulness of our dataset by encoding categorical variables and using appropriate procedures for handling missing values. This process makes our research's analysis more accurate and reliable.

D. STEP 4— Visualisation using the Technique of Correlation

A thorough understanding of our data is necessary for any data analysis procedure. One effective method for helping us understand the patterns and trends that the data shows is data visualisation. Through the application of diverse visualisation techniques, we are able to efficiently investigate and analyse the properties of our dataset. Analysis of correlation is one such technique.

We can investigate the links between variables by using correlation analysis. It offers insightful information on the type and strength of the relationship between two or more variables. A substantial reliance between two variables is indicated when there is a strong correlation between them. Stated differently, alterations in one variable are correlated with comparable adjustments in the other.Strong correlations between variables and the target variable have a bigger effect on the target variable itself.

We are able to determine the primary factors influencing our target variable by utilising correlation analysis and showing the correlations between variables. With this understanding, we may concentrate on and prioritise the factors that have the biggest impact on the goal variable. In the end, having a solid grasp of the relationships between our data allows us to build reliable models for precise forecasts and insights and to make well-informed judgements.

A heatmap is an effective tool for visualising correlation, in addition to its numerical representation. A heatmap is a visual aid that improves our comprehension of the relationships between the data. It is a member of the family of visualization methods that also includes box plots and histograms, which help with the understanding and study of complicated datasets.

We may display the correlation matrix in a way that is visually clear by using a heatmap. We can spot patterns and trends more easily when colours and intensity gradients are used. This visual representation enhances our cognitive capacities because it can be difficult for humans to understand and draw conclusions from raw numerical data alone. We can better understand and analyse the data because the correlation information is presented in a graphic fashion, which takes advantage of our innate ability to process visual information. One useful tool for capturing the subtleties and relationships in the dataset is the heatmap. It makes it easier to identify significant relationships by allowing us to determine the direction and degree of correlations between variables. By utilising visualisation, we can effectively convey the complex interdependencies in our data and gain insightful knowledge that helps us make well-informed decisions. Figure 24 shows the correlation heatmap. We can easily distinguish the correlation between the variables based on the colour.

Light color—less correlated.
Medium-light colour-correlated
Medium colourless, strongly correlated
Dark color—strongly correlated.

E. STEP 5— Splitting Dataset

One of the most important steps in machine learning is to precisely divide the dataset into two sets. The training model's performance and accuracy may be affected by how the dataset is divided. Creating a train set and a test set from the dataset is a popular method. Various ratios can be used to accomplish this, including 80% for the train set and 20% for the test set or 70% for the train set and 30% for the test set, among others. The precise ratio selected may vary based on variables like the dataset's size and the demands of the current issue. Making sure that the train set and the test set each contain a representative sample of the dataset is crucial when doing the split. This implies that the train set's feature distributions and attributes should be shared by the test set. Preventing biases and disparities that can affect the trained model's generalisability is crucial.

Data can be split into the required train and test sets using a slicing operation to create a meaningful division of the dataset. This ensures the integrity of the dataset and assigns every observation to the correct collection. We can precisely assess the model's performance and have faith in its ability to generalise to new data by meticulously dividing the dataset and making sure the training and test sets are statistically significant.

Considering the independent variables in 'input_ds' and the target variable in 'output_ds',

input_ds = df_rainfall.drop (columns = [“Precipitation,Year,Months,Day”])
output_ds = df_rainfall[“Precipitation”]

To ensure proper model training and evaluation, the dataset has been divided into two distinct sets: the training dataset and the testing dataset. In the figure provided, the division is as follows:

- 80% of the data is allocated to the training dataset.
- The remaining 20% is assigned to the testing dataset.

Before starting the model training process, it is imperative to split the dataset to get accurate and objective findings. We may evaluate the model's performance on untested data and determine how well it generalises to new instances by utilising distinct datasets for training and testing. Using both the training and testing datasets at the same time while developing a model can sometimes improve accuracy. By using this method, the model can be trained on a wider range of samples and be able to represent the underlying relationships and patterns in the data more broadly. The study made sure that the model was evaluated under realistic conditions and that it could successfully generalise its predictions to unknown data using our train-test split methodology. Using this method makes it easier to create strong, dependable models that can forecast outcomes accurately for fresh, untested data sets.

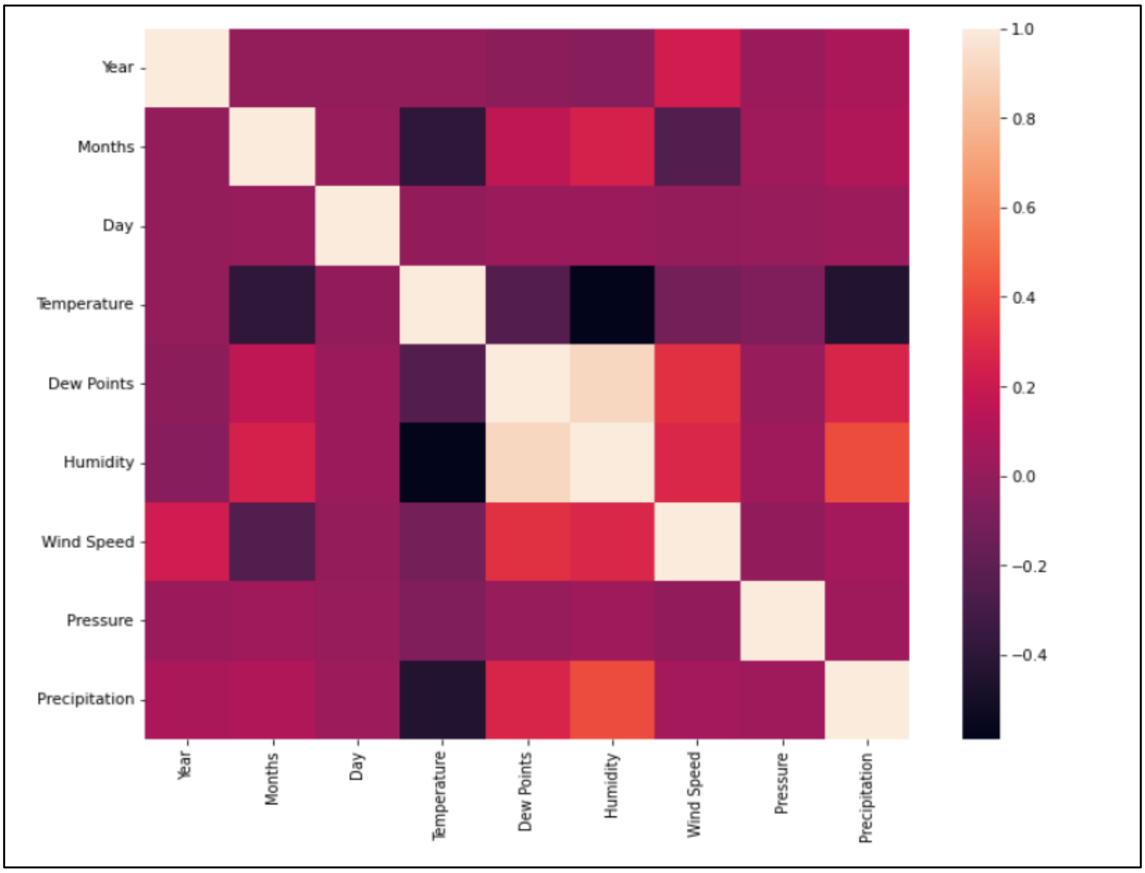
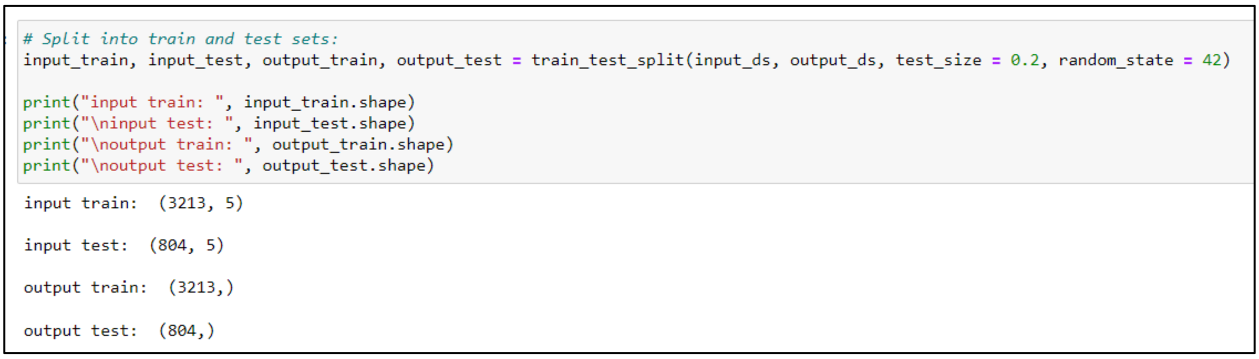


Fig. 24. Correlation Heatmap



```
# Split into train and test sets:
input_train, input_test, output_train, output_test = train_test_split(input_ds, output_ds, test_size = 0.2, random_state = 42)

print("input train: ", input_train.shape)
print("\ninput test: ", input_test.shape)
print("\noutput train: ", output_train.shape)
print("\noutput test: ", output_test.shape)
```

```
input train:  (3213, 5)

input test:  (804, 5)

output train:  (3213,)

output test:  (804,)
```

Fig. 25. Splitting the Dataset

F. STEP 6—Model Training

In the field of machine learning, there are many different types of algorithms. But we have purposefully chosen six particular algorithms to train our model using. Specifically:

1. *Multivariate Linear Regression (MLR):*
   - A straightforward approach that uses a linear equation to model the relationship between independent variables and the target variable.
   - The line that minimises the sum of squared discrepancies between the predicted and actual values is found to be the best fit.
   - Gives coefficients for every feature that show how each one affects the target variable. It is utilised to forecast numerical quantities that are continuous.
2. *Multi-layer Perceptron Regressor (Neural Network):*
   - A type of artificial neural network designed for regression tasks.
   - Consists of multiple layers of interconnected nodes (neurones) that process input data.
   - To produce an output, each neurone applies a weighted sum of inputs and processes it through an activation function.

- Used to capture complicated patterns in data and is useful for complex non-linear interactions.

3. *Kneighbors Regressor (KNN):*
   - This is an instance-based approach that
   - Uses the training data's k-nearest neighbours to predict the target variable.
   - Determines the weighted average or average of the goal values for each of these neighbours.
   - It is ideal for applications involving both regression and classification.
   - The value of k and the selected distance metric have a significant impact on performance.
4. *Ridge Regression:*
   - A regularisation technique for linear regression that prevents overfitting by adding a penalty term to the regression equation.
   - It helps mitigate multicollinearity by discouraging large coefficients for multiple features.
   - The trade-off between fitting the data and maintaining tiny coefficients is managed by the regularization strength (alpha).
5. *Random Forest Regressor:*
   - An ensemble technique that reduces overfitting by generating numerous decision trees and averaging their predictions.
   - A random subset of the characteristics and data is used to train each tree.
   - I am able to manage interactions and non-linear correlations in data.
6. *Support Vector Machines (SVM):*
   - It is used for both classification and regression tasks.
   - Maps data into a higher-dimensional space and finds a hyperplane that best separates data points while maximising the margin.
   - Kernel functions let you deal with non-linear interactions by transforming data into a space with more dimensions.
   - Used while choosing the right kernel, and it can be effective for complicated data distributions.

These algorithms have been applied to the supplied dataset in order to estimate and forecast the amounts of precipitation depending on different meteorological parameters. Every algorithm has advantages and disadvantages, and how well it works will depend on the particular situation at hand as well as the characteristics of the data.

We use well-known assessment metrics, such as the R² score, mean squared error (MSE), and root mean squared error (RMSE), to evaluate the correctness of our regression model. These metrics provide us important information about how the model is working and let us assess how predictive it is. We make use of the Sklearn software to put these algorithms into practice and assess their efficacy. This package offers a wide range of machine learning tools and functionalities. Through the process of model training, we can effectively employ the chosen algorithms by importing the required parts from Scikit-learn. We use strong assessment measures and carefully select the best algorithms for our particular assignment to create a high-performing model that predicts the target variable with accuracy. Making the most of the Scikit-learn package's features helps us train and assess the selected algorithms in our project more successfully. The Sklearn packages must be installed and imported to continue with our investigation. For machine learning tasks, this package offers a wide range of tools and functionalities. After installing and importing it, we can use its features for our project. After the training and testing stages, we compute a number of critical evaluation metrics to determine our model's precision and efficiency.

The Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R²-score are examples of these measurements. As you can see, the MSE and RMSE values show how big the differences are between what was expected and what actually happened. The R² score tells you how much of the target variable's volatility the model can predict. While lower RMSE and MSE values suggest better performance, higher R² scores indicate greater model efficiency. During the testing phase, we evaluate the model's effectiveness in processing real-time data using the test dataset. We may confirm the model's generalisability and usefulness outside of the training dataset by assessing its performance on untested data. These evaluation measures are essential for assessing the model's efficacy and accuracy in regression tasks. A model with a higher R² score is considered more efficient, and one with lower RMSE and MSE values is more accurate. To guarantee that our model predicts the target variable as accurately and ideally as possible, we must minimise error values.

G. STEP 7—Feature Importance Analysis:

1. *Temperature:* The term "temperature" describes how hot or cold a material or its surroundings are. Numerous elements, including latitude, height, and closeness to the sea, as well as wind patterns, including ocean currents, affect it. The intensity of precipitation, whether in the form of light showers or severe downpours, is determined by the amount of water vapour in the atmosphere, so the relationship between temperature and rainfall is important. As a result, warmer weather is frequently accompanied by heavier rainfall, which produces more significant precipitation events.
2. *Humidity***:** At a specific temperature, humidity is defined as the amount of moisture or water vapour in the air. It is essential for the development of clouds and the saturation of moisture vapour. Higher humidity levels indicate greater saturation capacity, which in turn enhances cloud formation and precipitation creation. As a result, areas with more humidity have a tendency to get larger amounts of precipitation than areas with lower humidity. Humidity is a critical component in comprehending precipitation patterns since it directly affects the frequency and severity of rainfall.
3. *Air Pressure*: Air pressure is the force that the atmosphere applies to the surface of the Earth. It is essential to the production of rainfall and weather

patterns. Elevation causes the air pressure to decrease; hence, higher altitudes generally have lower air pressure than lower heights. This variance in air pressure impacts the movement of air masses, moisture retention, and atmospheric stability. This variance affects how and when rainfall occurs across different regions. It is essential to comprehend how air pressure and rainfall patterns interact to forecast and analyse precipitation amounts in various geographic locations.

4. *Wind Speed:* The pace at which air molecules travel horizontally in the atmosphere is referred to as wind speed. Wind speed significantly influences the formation and distribution of rainfall. Air moves from high-pressure zones to low-pressure zones, creating wind patterns. These patterns affect the weather, including precipitation. Since wind speed and rainfall have a strong correlation, wind speed must be taken into account as an input for forecasting rainfall. Higher wind speeds are associated with less intense rainfall because the strong airflow disperses moisture and inhibits the collection of raindrops. On the other hand, slower wind speeds during the rainy season are better for greater precipitation because they let moisture pool and make it easier for rain to form. Understanding how wind speed influences rainfall patterns can enhance the accuracy of precipitation level forecasting and interpretation.
5. *Dew Point*: The temperature at which water vapour in the air becomes saturated and forms dew, fog, or clouds is known as the dew point. Because it shows the amount of moisture in the air, it is crucial for forecasting rainfall. Precipitation is more likely when the dew point is near to the actual temperature since it indicates a high moisture content. This condition is due to the air having reached saturation, and any additional cooling could allow water vapour to condense into liquid droplets, which would then cause clouds to develop and eventually rainfall.

H. *STEP 8—Prediction:*

- Apply the refined and trained model to forecast fresh or unobserved data.
- When you give the model the required input features (such as temperature and humidity), it will forecast the amount of rainfall that will occur.
- To preserve accuracy and consistency, make sure the input data is preprocessed in the same manner as the training data.

I. *STEP 9—Model Deployment:*

- To enable consumers to receive rainfall predictions, deploy the learnt model in a real-world setting.
- There are several ways to accomplish this, including building an API, a web application, or incorporating the model into already-in-use software systems.
- Ensuring that the model is put into use is reliable, scalable, and integrated with the infrastructure required to make predictions in real time.

J. *STEP 10—Monitoring and Maintenance:*

- To guarantee the correctness and dependability of the deployed model, periodically check its performance.
- To keep the model current and preserve its predictive power, continually gather fresh data and update it on a regular basis.
- To capture any evolving trends or changes in the link between input features and rainfall, think about retraining the model with more current data.
- Keep abreast of new findings, methodologies, and information sources that could improve the model's functionality and yield more precise rainfall forecasts.

V. Results

The main goal of the project is to forecast rainfall using machine learning techniques. The study demonstrated that applying these algorithms can yield precise and reliable rainfall forecast values. The study's main goal was to make forecasts with the highest level of efficiency and accuracy possible.The study used data from the previous ten years in seven different Nigerian locales to train and test to achieve this. The application of machine learning algorithms and techniques was the study's primary focus. In addition to Nigeria's Federal Capital Territory (Abuja), the rainfall data from 2010 to 2020 was evaluated in the regions of Benue, Kogi, Kwara, Nasarawa, Niger, and Plateau.The models have undergone successful training. It is possible to draft all of the values into tabular form as follows:

Table 1. Models Result for Abuja

Models (Abuja)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	8.19	4.26	0.29(29%)
Multi-layer Perceptron Regressor (Neural Network)	8.22	2.9	0.54(54%)
Random Forest Regression	7.73	1.44	0.88(88%)
Support Vector Machines	8.95	3.43	0.14(14%)
Kneighbors Regressor (KNN)	8.86	2.0	0.73(73%)
Ridge Regression	8.18	4.25	0.29(29%)

https://doi.org/10.20428/jst.v30i4.2771

Table 2. Models Result for Benue

Models (Benue)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	10.68	4.81	0.18(18%)
Multi-layer Perceptron Regressor (Neural Network)	10.6	3.82	0.41(41%)
Random Forest Regression	10.97	1.76	0.84(84%)
Support Vector Machines	11.42	3.8	0.07(7%)
Kneighbors Regressor (KNN)	11.19	2.48	0.68(68%)
Ridge Regression	10.68	4.81	0.18(18%)

Table 3. Models Result for Kwara

Models (Kwara)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	9.64	3.45	0.23(23%)
Multi-layer Perceptron Regressor (Neural Network)	16.08	2.74	0.42(42%)
Random Forest Regression	6.67	1.33	0.84(84%)
Support Vector Machines	6.89	2.91	0.09(9%)
Kneighbors Regressor (KNN)	7.59	1.85	0.65(65%)
Ridge Regression	9.43	3.45	0.23(23%)

Table 4. Models Result for Kogi

Models (Kogi)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	7.32	4.17	0.28(28%)
Multi-layer Perceptron Regressor (Neural Network)	6.92	3.14	0.46(46%)
Random Forest Regression	7.46	1.57	0.86(86%)
Support Vector Machines	8.04	3.56	0.12(12%)
Kneighboes Regressor (KNN)	8.69	2.27	0.67(67%)
Ridge Regression	7.32	4.17	0.28(28%)

Table 5. Models Result for Nasawara

Models (Nasarawa)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	7.82	4.38	0.31(31%)
Multi-layer Perceptron Regressor (Neural Network)	7.18	3.3	0.48(48%)
Random Forest Regression	7.89	1.72	0.84(84%)

Support Vector Machines	8.6	3.73	0.16(16%)
Kneighboes Regressor (KNN)	8.48	2.26	0.72(72%)
Ridge Regression	7.84	4.38	0.31(31%)

Table 6. Models Result for Niger

Models (Niger)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	7.83	4.17	0.33(33%)
Multi-layer Perceptron Regressor (Neural Network)	7.48	2.78	0.6(60%)
Random Forest Regression	7.74	1.43	0.87(87%)
Support Vector Machines	8.68	3.43	0.17(17%)
Kneighboes Regressor(KNN)	8.71	2.01	0.72(72%)
Ridge Regression	7.83	4.16	0.33(33%)

Table 7. Models Result for Plateau

Models (Plateau)	**RMSE**	**MAE**	**R-Squared (%)**
Multivariate Linear Regression (MLR)	6.59	3.81	0.4(40%)
Multi-layer Perceptron Regressor (Neural Network)	6.42	2.91	0.56(56%)
Random Forest Regression	6.82	1.47	0.87(87%)
Support Vector Machines	7.12	3.38	0.3(30%)
Kneighboes Regressor (KNN)	7.37	2.13	0.73(73%)
Ridge Regression	6.61	3.83	0.4(40%)

The table provides a comparison of the computed R²-score value, RMSE value difference, and MSE value difference. We determine the RMSE value difference by subtracting the RMSE value of the test data from the RMSE value of the train data. Similarly, we calculate the MSE difference value. We compute these differences to evaluate the testing dataset's correctness. If the RMSE value of the testing dataset exceeds that of the training dataset, we will obtain a negative value. This implies a superior accuracy level, as the testing dataset has undergone more successful training than the training dataset. Similarly, a model appears to be underperforming compared to a model that merely forecasts the target variable's mean if its R-squared value is negative. R-squared quantifies the percentage of the target variable's variance that the model can account for. It has a value between 0 and 1, where 1 denotes a perfect fit and 0 means that the model cannot account for any of the variance. If the target variable's mean is used as the forecast, the model's predictions are not as good as they would be if the R-squared value was negative. This phenomenon usually happens when the target variable is extremely unpredictable or when the model is failing to identify the underlying patterns in the data. It can also mean that the model is not properly generalising to new data, possibly due to overfitting the training set.

Additionally, the R² score value gives us an idea of how accurate each model is. Among the algorithms chosen, the Random Forest Regression algorithm stands out for having the highest accuracy. In addition, the relatively minor disparities in error values show that the Random Forest algorithm has been trained exceptionally well when compared to the other methods.

VI. Conclusions

Rainfall is a critical natural phenomenon with significant implications for agriculture, water resource management, and disaster preparedness. This study identified the Random Forest Regression Algorithm as the most effective machine learning model for rainfall prediction in North-Central Nigeria, demonstrating superior accuracy across evaluation metrics. Leveraging ten years of climatic data, the research compared various algorithms, emphasizing preprocessing methods to ensure reliable predictions. The Random Forest model's robust performance underscores its potential for operational use by meteorological agencies to enhance agricultural planning, water management, and disaster

https://doi.org/10.20428/jst.v30i4.2771

mitigation. These findings contribute to sustainable development efforts and resource optimisation in the region.

References

[1] A. G. Omonijo, A. Matzarakis, O. Oguntoke, and C. O. Adeofun, "Influence of weather and climate on malaria occurrence based on human-biometeorological methods in Ondo State, Nigeria," *Journal of Environmental Science and Engineering*, vol. 5, pp. 1215–1228, 2011.

[2] E. B. Guedes, P. M. de Lima, and M. B. L. de Oliveira, "Neural networks for time series rainfall forecasting: A case study in Manaus, Amazonas," 2017.

[3] J. B. Omotosho, A. Balogun, and K. Ogunjobi, "Predicting monthly and seasonal rainfall, onset and cessation of the rainy season in West Africa using only surface data," *International Journal of Climatology*, vol. 20, no. 1, pp. 1-16, Jan. 2000. Available: https://doi.org/10.1002/(SICI)1097-0088(200001)20:1<1::AID-JOC444>3.0.CO;2-K

[4] S. Aswin, P. Geetha, and R. Vinayakumar, "Deep Learning Models for the Prediction of Rainfall," *2018 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2018, pp. 0657-0661, doi: 10.1109/ICCSP.2018.8523829.

[5] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi, "An artificial neural network model for rainfall forecasting in Bangkok, Thailand," *Hydrology and Earth System Sciences*, vol. 13, no. 8, pp. 1413-1425, Aug. 2009. Available: https://doi.org/10.5194/hess-13-1413-2009

[6] N. D. Hoai, K. Udo, and A. Mano, "Downscaling global weather forecast outputs using ANN for flood prediction," *Journal of Applied Mathematics*, vol. 2011, pp. 1-12, 2011. Available: https://doi.org/10.1155/2011/587408

[7] S. K. Biswas et al., "Rainfall forecasting by relevant attributes using artificial neural networks—a comparative study," *International Journal of Big Data Intelligence*, vol. 3, no. 2, pp. 79-89, 2016.

[8] P. Wichitarapongsakun, C. Sarin, P. Klomjek, and S. Chuenchooklin, "Rainfall prediction and meteorological drought analysis in the Sakae Krang River basin of Thailand," *Agriculture and Natural Resources*, vol. 50, no. 6, pp. 401-407, 2016. Available: https://doi.org/10.1016/j.anres.2016.12.006