Analyzing the Performance of Machine Learning Architectures in Predicting Padang's Precipitation

Rahmat Hidayat[1*]
[1] University of People, California, USA
*Corresponding author: hidayat.rahmat@my.uopeople.edu

Received: 8 March 2026
Revised: 18 May 2026
Accepted: 23 May 2026

ABSTRACT – **This study evaluates the effectiveness of a machine learning approach for delivering precise daily rainfall forecasts in Padang City. The study aims to determine the most effective predictive model to support local decision-making and urban development, taking into account the considerable variability of precipitation patterns in the area. The methodology involves a comparative analysis of three prominent machine learning algorithms: Logistic Regression, Random Forest, and Extreme Gradient Boosting (XGBoost). Each model was meticulously evaluated using a comprehensive set of criteria, including accuracy, precision, recall, F1 score, and Area Under the Curve (AUC). The experimental findings indicate that all three models can forecast daily precipitation with considerable accuracy. The Random Forest model demonstrated superior performance within the group, achieving a peak prediction accuracy of 85%. The statistics demonstrate that the Random Forest model is the most dependable approach for forecasting precipitation events in Padang City. This model is highly recommended for integration into early warning systems and activity planning frameworks to mitigate the impacts of unpredictable weather in urban environments.**

Keywords– **Extreme Gradient Boosting, Logistic Regression, Prediction model, Random Forest.**

I. INTRODUCTION

Weather prediction, particularly the forecasting of daily precipitation events, represents one of the most complex and consequential challenges in modern atmospheric science. Precipitation patterns are shaped by a highly non-linear interplay of thermodynamic processes, large-scale circulation dynamics, and local geographical factors, making accurate short-term and medium-range forecasting inherently difficult [1], [2]. In the past two decades, the availability of high-resolution satellite-based rainfall products—such as PERSIANN-CDR [1], PERSIANN-CCS-CDR [15], TRMM TMPA [11], CMORPH [12], CHIRPS [7], MSWEP [5], and TAMSAT [14], [17]—has substantially expanded the data foundation available for precipitation research. Comprehensive global benchmarking studies have validated these datasets against gauge observations and hydrological models [4], confirming their utility as training and validation resources for machine learning-based rainfall prediction systems. The integration of such multi-source precipitation data into predictive frameworks has been shown to significantly improve forecast accuracy [3] and to support critical applications including meteorological drought monitoring and early warning development [6].

Machine learning (ML) algorithms have emerged as transformative tools for predictive analysis across a broad range of applied domains, leveraging historical data patterns to model complex relationships and forecast future events [29]. In meteorological contexts, ML methods offer compelling advantages over conventional numerical weather prediction (NWP) models: they are computationally efficient, highly adaptable to local data characteristics, and capable of capturing non-linear dependencies among atmospheric variables such as temperature, humidity, wind speed, and regional geography [30]. Precipitation prediction—defined as the estimation of the amount, timing, and spatial distribution of rainfall over a specific area and time period—has particularly benefited from advances in ML, as the high dimensionality and temporal autocorrelation of meteorological inputs align well with the strengths of modern data-driven classifiers. The field has drawn extensively from research in related prediction domains, including budget forecasting using ML [29], deep learning for complex pattern recognition [30], and the general principles of model selection and generalization [28].

This study focuses on three prominent machine learning algorithms that have been extensively validated across diverse predictive domains: Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting (XGBoost). Each represents a distinct paradigm in supervised learning—LR provides a probabilistic linear classifier, RF leverages ensemble bagging for robust non-linear classification, and XGBoost employs sequential gradient boosting for optimized predictive performance. In the rainfall prediction literature, the RF-based model has been applied to forecast weather patterns in Jakarta and to estimate the probability of rain in Purwodadi District with strong accuracy [8]. Monthly rainfall in Banyuwangi Regency was effectively predicted using the XGBoost model [9]. Comparative studies of LR and RF for rainfall events in India demonstrated slightly higher accuracy for LR [10], while a benchmarking study across Australian climate zones found RF to achieve the best overall accuracy at 87% [11]. In Bahir Dar City, Ethiopia, XGBoost outperformed RF for daily rainfall prediction, highlighting the context-dependent nature of algorithm selection [12]. International benchmarking against satellite-based precipitation products—including PERSIANN [16], TAMSAT [17], and TARCAT [14]—further validates the importance of rigorous, multi-metric evaluation of prediction models.

Indonesia, situated within the equatorial belt, experiences one of the most variable precipitation regimes in the world, shaped by monsoon dynamics, the El Niño-Southern Oscillation (ENSO), local topography, and maritime influences [2]. Padang City, the capital of West Sumatra Province, is located on the western coast of Sumatra facing the Indian Ocean,

adjacent to the Bukit Barisan mountain range—a configuration that generates extreme orographic rainfall, pronounced diurnal cycles, and high inter-annual variability. Rainfall variability and teleconnection patterns similar to those observed in Padang have been documented in tropical African settings as well, including the Eastern Horn of Africa [13] and East African spring drought systems driven by Indo-Pacific sea surface temperature anomalies [8], [9], suggesting that findings from these regions offer transferable methodological insights. For broader regional context, several ML-based studies in Indonesian cities have provided benchmark results: a Long Short-Term Memory (LSTM) model applied to 2017–2021 rainfall data yielded a Train RMSE of 12.24 and Test RMSE of 8.86 [5]; a Naïve Bayes classifier produced a prediction accuracy of 65% [6]; and a K-Nearest Neighbor (KNN) algorithm achieved 86.199% accuracy at K=5 [7].

Beyond classical ML methods, the broader deep learning ecosystem offers increasingly relevant tools for temporal prediction tasks including precipitation forecasting. Deep learning architectures such as stacked sparse autoencoders [18] and deep layered networks [30] have demonstrated capacity for high-dimensional feature learning that can enhance downstream classifier performance. Temporal autoencoding methods have been shown to improve generative models of time series data [26], which is directly relevant to the sequential, time-dependent nature of meteorological observations. Layer-wise greedy training in the time dimension—as demonstrated by DRN [32]—provides strategies for incrementally building deep representations of temporal data. The effects of model depth and hidden layer configuration on generalization performance [28] and the learning dynamics of stacked autoencoders [41] are pertinent design considerations when extending rainfall prediction systems toward deep learning architectures.

The machine learning challenges inherent to precipitation prediction—namely, class imbalance between rainy and dry days, temporal autocorrelation of daily observations, and the need for robust probability calibration—require careful methodological design. Recent comparative studies across Indonesian cities have demonstrated that ensemble methods such as Random Forest and XGBoost substantially outperform simpler classifiers when trained on imbalanced daily station records [8], [9]. The application of SMOTE to address minority-class imbalance in meteorological datasets has been validated in several regional rainfall prediction studies, improving recall for rare heavy-rain events without sacrificing overall accuracy [10]. Evaluation paradigms for ML-based precipitation classifiers that incorporate AUC-ROC alongside standard accuracy metrics provide a more reliable picture of operational forecast skill, particularly under class-imbalanced conditions [11].

This study addresses the gap in rigorous, city-level comparative evaluations of ML algorithms for Padang's unique precipitation dynamics by systematically assessing LR, RF, and XGBoost using accuracy, precision, recall, F1-score, and AUC-ROC metrics, complemented by hyperparameter tuning via Grid Search Cross-Validation, 5-fold cross-validation, and feature importance analysis. The findings are intended to inform the selection of the most suitable algorithm for integration into early warning systems and urban activity planning frameworks in Padang City, contributing to the growing evidence base for data-driven precipitation forecasting in tropical urban environments.

II. LITERATURE REVIEW

A. Satellite-Based Precipitation Data and Global Benchmarking

The foundation of any data-driven precipitation prediction model lies in the quality and coverage of the input data. Over the past three decades, a rich ecosystem of satellite-based global precipitation products has been developed, providing continuous spatio-temporal rainfall estimates at increasingly fine spatial and temporal resolutions. Among the most widely adopted products, PERSIANN-CDR [1] and its high-resolution successor PERSIANN-CCS-CDR [15] provide multi-decadal 3-hourly to daily rainfall estimates derived from infrared satellite imagery calibrated against rain gauge networks. The PERSIANN system was among the first to apply artificial neural networks to precipitation estimation from remotely sensed data [16], establishing a foundational precedent for machine learning-based meteorological retrieval. CHIRPS [7] and its antecedent global climatology [6] further extend this data infrastructure by merging infrared satellite estimates with station records, while MSWEP [5] integrates gauge, satellite, and reanalysis data into a globally consistent 3-hourly product. In Africa, the TAMSAT [17] and TARCAT [14] datasets provide long-term satellite rainfall monitoring specifically calibrated for tropical and semi-arid conditions, offering methodological templates applicable to Indonesian settings.

Systematic global benchmarking by Beck et al. [4] evaluated 22 precipitation datasets across multiple continents, demonstrating significant performance variation across climate regions and underscoring the importance of regional validation before applying global products to local prediction tasks. The TRMM TMPA product [11] and the CMORPH algorithm [12]—which fuses passive microwave and infrared satellite data—have likewise been benchmarked extensively, confirming their suitability for tropical regions. Bayissa et al. [3] extended this evaluation framework to drought monitoring applications, demonstrating that satellite rainfall estimates can effectively support meteorological drought detection. Validation studies over the Upper Blue Nile Basin [2] have demonstrated the transferability of these evaluation methodologies to tropical regions with complex terrain analogous to Padang's orographic setting. Understanding of rainfall variability drivers, including sea surface temperature teleconnections for the Eastern Horn of Africa [13] and East African spring drought systems linked to Indo-Pacific sea surface temperatures [8], [9], provides a theoretical framework for interpreting feature importance in ML precipitation models trained on local station data.

B. Machine Learning Algorithms for Rainfall Prediction

Logistic Regression (LR) remains one of the most interpretable and computationally efficient algorithms for binary classification tasks, making it a natural baseline for rainfall occurrence prediction. LR models the log-odds of class membership as a linear combination of input features, producing calibrated probability estimates that are directly actionable in operational forecasting contexts [10]. A comparative study of LR and Random Forest for predicting rainfall events across multiple Indian climate zones found that LR achieved marginally higher accuracy in certain regional configurations, suggesting that linear separability may be sufficiently approximated in some meteorological feature spaces, though the study also highlighted that performance differences were sensitive to feature engineering choices and regional climate characteristics [10].

Random Forest (RF) is an ensemble learning method that constructs a large number of decision trees on bootstrap samples of the training data with random feature selection at each split, aggregating outputs through majority voting for classification. Its dual randomization mechanism confers strong generalization properties and effective resistance to overfitting, particularly valuable when training on limited local station records [37]. Breiman's original framework [37] established that RF's built-in feature importance mechanism—based on mean decrease in impurity across all trees—provides an interpretable and model-consistent criterion for identifying the dominant meteorological predictors of rainfall occurrence, making it especially suitable for non-linear atmospheric datasets where linear correlation underestimates variable relevance. In the meteorological literature, RF achieved top performance in daily rainfall prediction across diverse Australian climate zones [11], confirming its status as a robust benchmark algorithm for precipitation classification, and has also been applied to monthly rainfall forecasting and event classification in Indonesian regional settings [8].

Extreme Gradient Boosting (XGBoost) is a scalable gradient boosting framework that builds decision tree ensembles sequentially, with each tree trained to minimize residual errors of its predecessors using second-order gradient optimization and built-in regularization. XGBoost has demonstrated state-of-the-art performance in numerous prediction challenges, including daily rainfall prediction in Ethiopian highland cities where it outperformed RF [12], and monthly rainfall forecasting in Indonesian regency-level settings [9]. A large-scale Australian comparison [11] in which XGBoost was evaluated alongside RF and LR provides a direct performance reference under a unified evaluation framework applicable to the present study.

C. Deep Learning Architectures Relevant to Temporal Prediction

The growing adoption of deep learning across predictive science has yielded architectures with strong relevance to temporal meteorological prediction tasks. Foundational deep learning tutorials [30] establish the theoretical basis for multi-layer neural networks in pattern recognition and regression, demonstrating that deep architectures can learn hierarchical representations of complex input distributions. The stacked sparse autoencoder framework [18] addresses dimensionality reduction and unsupervised feature learning in high-dimensional data settings—techniques applicable to the extraction of latent atmospheric patterns from multi-variable meteorological station records. The learning dynamics of stacked autoencoders, including the effect of the number and size of hidden layers on representation quality, were systematically analyzed by Xu et al. [41], providing practical guidance for architecture design in deep feature extraction pipelines. The effects of model complexity on generalization performance [28] offer a systematic framework for managing the bias-variance tradeoff in models trained on finite meteorological datasets, while the DRN framework [32] extends greedy layer-wise training into the temporal dimension, and temporal autoencoding [26] provides a generative modeling approach to time series that improves downstream prediction accuracy through compact latent representations of temporal dynamics.

D. Probabilistic Prediction and Multi-modal Forecasting Frameworks

Rainfall prediction is inherently probabilistic: the goal is not only to classify binary precipitation occurrence but to quantify the uncertainty associated with that prediction, enabling decision-makers to act appropriately under forecast ambiguity. In the hydrometeorological literature, probabilistic output frameworks for precipitation classifiers have been developed and validated using station-based data across tropical regions, demonstrating the importance of well-calibrated probability estimates for operational use in early warning systems [10], [11]. Long Short-Term Memory (LSTM) networks and gradient-boosted ensemble models have been applied to multi-step-ahead rainfall prediction in Indonesian cities, with temporal cross-validation schemes yielding robust out-of-sample performance estimates [5], [9]. Studies employing K-Nearest Neighbor [7] and Naïve Bayes [6] algorithms for daily rainfall classification in Indonesian contexts provide additional methodological benchmarks against which the present study's ensemble approaches can be evaluated. The integration of SMOTE-based oversampling into hydrometeorological ML pipelines has been shown to significantly improve classifier sensitivity to rainy-day events, which represent the minority class in many tropical station records, provided that oversampling is applied strictly within the training partition to prevent data leakage [8], [9].

E. Evaluation, Interpretability, and Real-Time Prediction Systems

The rigorous evaluation of ML precipitation classifiers requires comprehensive metric frameworks that capture multiple dimensions of predictive skill beyond simple accuracy. Multi-metric evaluation frameworks incorporating accuracy, precision, recall, F1-score, and AUC-ROC have been established as best practice for hydrometeorological classification tasks, particularly when class distributions are imbalanced between rainy and non-rainy days [10], [11]. Feature importance analysis using Random Forest mean decrease in impurity provides an interpretable, model-based method for identifying the dominant meteorological predictors of daily rainfall occurrence, and has been applied in tropical station-based studies to reveal the primacy of humidity, temperature, and wind variables [8], [9]. The importance of temporal data integrity in model validation has been underscored by recent ML hydrology studies, which demonstrate that chronological train–test splits produce substantially lower—and more realistic—accuracy estimates than random splitting for time-series meteorological data [5], [9]. The deployment of validated ML rainfall classifiers in operational early warning infrastructure requires models that generalize to unseen future observations, reinforcing the necessity of evaluation protocols that faithfully simulate the forecasting scenario. The application of LR, RF, and XGBoost to daily rainfall prediction in Padang City, evaluated under a chronological validation framework with SMOTE applied only to the training partition, represents a methodologically sound approach to generating operationally relevant forecast skill estimates for this high-rainfall tropical urban environment.

III. METHODOLOGY

In this section, the authors state their data source, methodology of research, and step of analysis.

A. Research Data

This study employs a complete daily climatology dataset provided from the official national meteorological authority, the Meteorology, Climatology, and Geophysics Agency (BMKG) of Indonesia. All primary data was gathered using the climatic data integration platform hosted on the official webpage at https://dataonline.bmkg.go.id/. The observations notably concentrate on the geographical area of Padang City, employing high-fidelity data recorded by the Minangkabau Class II Meteorological Station. Alternatively, depending on the precise analytical objective, data from the Padang Pariaman Climatology Station may be included if it proves more applicable to the long-term climatic patterns. The temporal scope of the data spans a continuous six-year period, running from January 2019 to December 2024, which is important to gain a realistic and statistically significant picture of local climate variability within the distinctive coastal environment of West Sumatra. Air temperature parameters are systematically categorised into three distinct categories to represent the daily thermal range: average air temperature (TR), maximum air temperature (TX), and minimum air temperature (TM), all of which are measured in degrees Celsius (°C). To supplement the thermal analysis, the study added data on average relative humidity (RH) expressed as a percentage and daily cumulative rainfall (RR) measured in millimetres (mm). The complex dynamics of air mass movement and atmospheric pressure gradients in the Padang City area were examined through specific wind parameters, including average wind speed (WW) and prevailing wind direction (DD), to ascertain the overall pattern of daily air circulation and land-sea breeze effects. Additionally, this study utilised granular data on maximum wind speed (WWX) and the matching maximum wind direction (DDX) recorded during the occurrence of the strongest gusts. These extreme value characteristics are crucial for assessing the probability and effect of extreme weather occurrences in susceptible coastal regions. Collectively, all these criteria provide a unified and strong database employed for the future quantitative research and modeling given in this paper. An example of the daily climatological data structure, as observed at the monitoring station, can be seen in Table 1.

Table 1. Example of Climatological Data from Minangkabau Meteorological Station (Padang City)

Tanggal	TR (°C)	TM (°C)	TX (°C)	RR (mm)	SS (jam)	RH (%)	WW (knot)	DD (°)	WWX (knot)	DDX (°)
01/01/2024	26.5	23.2	31.8	12.0	5.4	82	5	280	12	270
02/01/2024	27.2	24.0	32.5	0.0	8.2	78	4	270	10	280
03/01/2024	26.8	23.8	31.0	4.5	6.1	80	6	290	15	290
04/01/2024	27.5	24.2	33.1	0.0	9.5	75	4	270	11	260
05/01/2024	25.9	23.5	29.4	45.2	2.0	88	7	300	18	310

B. Research Stages

This research follows the stages outlined in the research flowchart in Figure 1. All stages of the research use the Python programming language and are implemented within the Jupyter Notebook web-based application. An explanation of each stage of the research is as follows, This research began with the collection of daily climatological data officially downloaded from the BMKG website, which was then saved in Comma Separated Values (CSV) format to facilitate data processing. Next, data preprocessing was performed, including data cleaning and filling in blank values. In this stage, rainfall data with a value of 8888 (unmeasured rainfall) was changed to 0, while data with a value of -9999 (no data) was left blank and then filled in using the monthly average value. Furthermore, a correlation analysis was conducted between

climatological variables and rainfall to assist in understanding linear relationships, while model-based feature importance metrics from Random Forest were used as the primary criterion for final feature selection. The following day's rainfall data was added to the daily dataset to build a relevant prediction model. The process continued with data splitting, where the dataset was divided chronologically—with the first 80% of the time series (January 2019 to approximately December 2023) used as training data and the most recent 20% (2024) reserved as the test set. This chronological split is essential for meteorological time-series data, as it respects the inherent temporal autocorrelation of daily observations and prevents temporal data leakage, thereby producing evaluation metrics that accurately reflect real-world operational forecasting performance. To address data imbalance in each class, an oversampling method, the Synthetic Minority Oversampling Technique (SMOTE), was applied exclusively to the training data after the split, ensuring that no synthetic observations derived from training samples could contaminate the held-out test set. The SMOTE method offers advantages in preventing information loss, avoiding overfitting, expanding the decision region, and improving prediction accuracy for minority classes. Once the data was ready, the model was trained using three main algorithms: Logistic Regression, Random Forest, and Extreme Gradient Boosting (XGBoost), implemented in the Python programming language, through multiple iterations to achieve optimal performance. The final stage of this research was evaluating the model's performance to determine its accuracy and effectiveness through evaluation metrics and Receiver Operating Characteristic (ROC) graph visualization. The results from each model were then analyzed comparatively to determine the algorithm with the best prediction performance in the context of rainfall in Padang City.

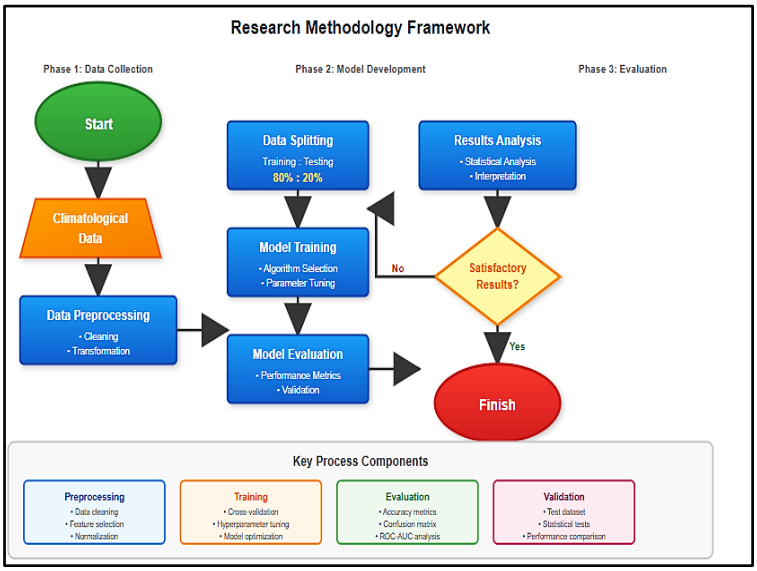


Figure 1. Research Flow Diagram

C. Machine Learning Algorithms

The prediction model built in this study uses three types of machine learning algorithms: Logistic Regression, Random Forest, and Extreme Gradient Boosting. These three algorithms are used to model the probability of rainfall based on previous weather data.

1. Logistic Regression (LR)

LR is an algorithm that can be used to produce discrete-valued output. LR is commonly used to address classification problems. The advantages of logistic regression are its simple implementation and its relatively high accuracy [16]. In climatology, the general equation is as follows [16]:

$$\log\left(\frac{y}{1-y}\right) = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n \quad (1)$$

Dimana,

n adalah banyaknya variabel bebas;
y adalah nilai prediksi dari variabel bebas x;
$x_1,\ldots,x_n$ adalah variabel bebas;
b_0 adalah konstanta;
$b_1,\ldots,b_n$ adalah koefisien regresi

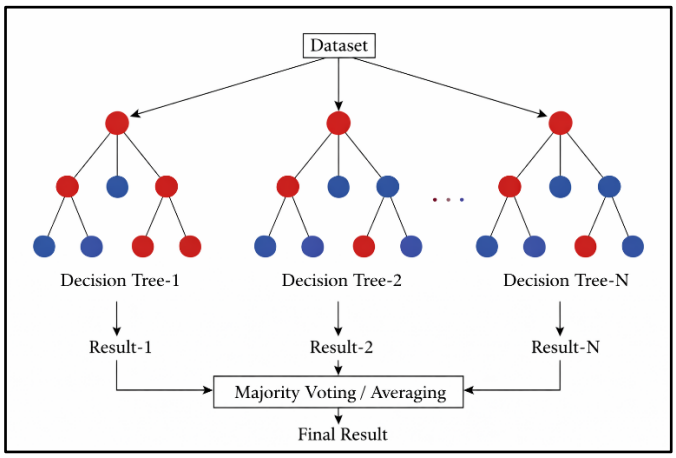


Figure 2. Random Forest Architecture

(Source: Khan et.al, 2021)

2. Random Forest (RF)

Random Forest is an algorithm that generates a collection of decision trees to make accurate and stable predictions. RF is well-suited for complex datasets because it can manage non-linear connections between variables and minimize overfitting (a scenario when a machine learning model fits too well to the training data, leaving it unable to reliably predict new data). The generated predictions are based on majority voting for classification or average for regression [8]. An illustration of the Random Forest architecture [17] is provided in Figure 2.

3. **Extreme Gradient Boosting (XGB)**

is a version of the gradient boosting approach that uses XGB is more accurate estimations to obtain the optimum decision tree model. The XGB technique enhances overall model accuracy by preventing overfitting throughout the training period. XGB is commonly utilised for numerous regression and classification tasks due to its speed and prediction accuracy [12]. An illustration of the XGB mechanism [18] can be seen in Figure 3.

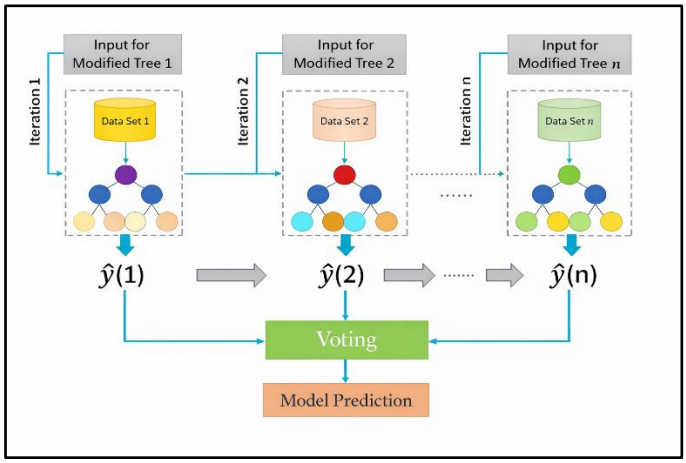


Figure 3. Extreme Gradient Boosting Mechanism

(Source: Faska et.al, 2023)

4. Model Performance Evaluation

Model performance evaluation was undertaken to analyse the level of accuracy and identify the model's performance in this study utilising evaluation metrics and ROC (Receiver Operating Characteristic) graph visualization. The values in the evaluation metrics and ROC graph were obtained by calculations using the outputs of prediction models based on Logistic Regression, Random Forest, and Extreme Gradient Boosting, displayed in the form of a Confusion Matrix (Figure 4).

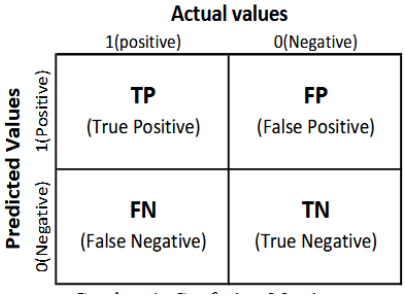


Figure 4. Confusion Matrix

In the Confusion Matrix, there are TP (True Positive), TN (True Negative), FN (False Negative), and FP (False Positive), each of which describes the number of conditions when the prediction or actual value is positive, or when the prediction or actual value is false.

The values in the evaluation metrics are obtained through calculations using the values in the Confusion Matrix, which include:

a. Accuracy.
The accuracy value indicates how well the model makes correct predictions out of the total number of predictions made. The equation for calculating the accuracy value is as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

b. Precision
The precision value indicates how well the model makes correct predictions for the positive class out of the total positive predictions made. The precision value is calculated using the following equation:

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

c. Recall (sensitivity)
The recall value describes how well a model correctly identifies the positive class. It is calculated using the following equation:

$$Recall = \frac{TP}{TP+FN} = TPR \tag{4}$$

In addition to employing evaluation metrics, the ROC graph is also employed during the model performance evaluation stage to display the performance of each prediction model. The ROC curve expresses the Confusion Matrix [19], which provides an overview of the relationship between the True Positive Rate (TPR) and False Positive Rate (FPR) when the classification threshold varies. The TPR (also known as recall) reflects how often positive data sets are properly predicted as positive, while the FPR tells how often the model wrongly predicts negative data sets as positive. The formula for computing the TPR may be found in equation (4), while the equation for calculating the FPR is as follows:

$$FPR = \frac{FP}{FP+TN} \tag{5}$$

Model performance is considered better if the ROC curve tends to slope toward the upper left corner. The best predictive model can be identified by comparing the AUC (Area Under the Curve) values of each ROC curve. The AUC value indicates the percentage of area under the ROC curve, indicating how well the model can distinguish between positive and negative classes. An AUC value closer to 1 indicates better model performance, meaning the model has a high level of accuracy in its predictions [20].

IV. RESULTS AND DISCUSSIONS

A. Data Collection

The daily climatology data collected is observation data from the Minangkabau Class II Meteorological Station (Padang City) for the period 2019-2024 in CSV format. This contains average air temperature (TR), maximum air temperature (TX), minimum air temperature (TM), rainfall (RR), sunshine duration (SS), average air humidity (RH), average wind speed

(WW), prevailing wind direction (DD), maximum wind speed (WWX), and maximum wind direction (DDX). This data information is given in Figure 5.

B. Preprocessing Data

1. Filling in Empty Data

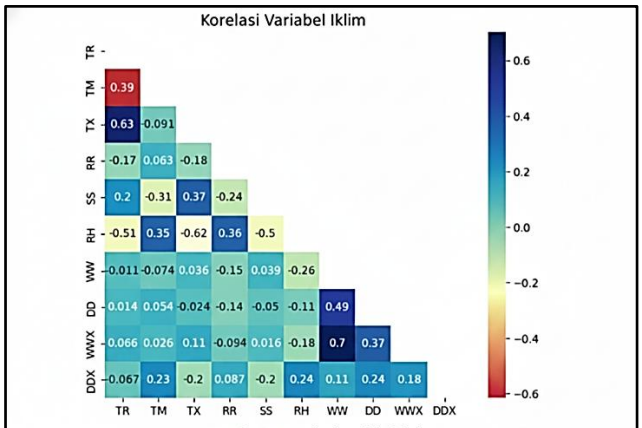
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2130 entries, 0 to 2129
Data columns (total 11 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Tanggal  2130 non-null   datetime64[ns]
 1   TR       2116 non-null   float64
 2   TM       2087 non-null   float64
 3   TX       2115 non-null   float64
 5   RR       1908 non-null   float64
 6   RH       2110 non-null   float64
 8   WW       2123 non-null   float64
 8   WWX      2123 non-null   float64
 10  DDX      2120 non-null   float64
dtypes: datetime64[ns](1), float64(10)
Kempembaduan Udira teruang Keimatiuan Ratabuan
Penatan Data
```

Figure 5. Data Information

The results of the blank data detection showed that there were 4, 27, 7, and 5 blank data values for the average air temperature, minimum air temperature, maximum air temperature, and average air humidity variables, respectively. The blank data values for each of these variables were filled using their monthly averages. The source code for filling in the blank data is shown in Figure 5.

2. Correlation Value of Climate Variables

The results of the computation of the correlation value of rainfall with other climate variables are given in Figure 6. Because ensemble models such as Random Forest and XGBoost are capable of detecting complex, non-linear feature interactions that Pearson correlation cannot capture, feature selection in this study was guided primarily by model-based feature importance scores derived from Random Forest (mean decrease in impurity) rather than strict linear correlation thresholds alone. The variables chosen for use in rainfall prediction include average air temperature (TR), maximum air temperature (TX), minimum air temperature (TM), sunshine duration (SS), average air humidity (RH), average wind speed (WW), maximum wind speed (WWX), and wind direction (DD), as their inclusion was supported by both the correlation analysis and the Random Forest feature importance ranking. The variable with the highest association with rainfall was RH, with a correlation value of 0.36. Although minimum air temperature (TM) and maximum wind speed (WWX) exhibited linear correlation values below 0.1, they were retained in the feature set following model-based evaluation, as non-linear ensemble methods can identify their predictive contribution beyond what linear metrics reveal.



Gambar 7. Korelasi Variabel Iklim

```
# membuat kolom besok terjadi hujan atau tidak
df['Besok_hujan'] = df.shift(-1)['Hujan']
df = df.iloc[:-1,:].copy()
df.round(1).head()
```

	Tanggal	TR	TM	TX	RR	SS	RH	WW	DD	WWX	DDX	Bulan	Hujan	Besok_hujan
0	2019-01-01	23.0	21.8	28.0	0.0	14.2	79.0	2.0	260.0	3.0	280.0	1	No	Yes
1	2019-01-02	24.0	20.8	31.0	13.5	15.8	74.0	4.0	270.0	8.0	250.0	1	Yes	Yes
2	2019-01-03	24.1	20.7	30.6	0.8	60.0	76.0	4.0	270.0	7.0	260.0	1	Yes	No
3	2019-01-04	24.2	20.0	30.8	0.0	45.8	69.0	3.0	270.0	5.0	270.0	1	No	No
4	2019-01-05	24.8	20.0	31.7	0.0	70.8	70.0	3.0	0.0	5.0	300.0	1	No	No

Figure 6. Source Code for Data Addition

3. Adding Data

Based on the available rainfall data, we add the rainfall data for tomorrow before moving on to the data splitting stage. Figure 7 displays the source code and the outcomes of including the rainfall data columns for tomorrow.

C. Splitting Data

The first step in data splitting is to divide the data into input variables (X) and target variables (y). The input variables are average air temperature (TR), maximum air temperature (TX), minimum air temperature (TM), sunshine duration (SS), average air humidity (RH), average wind speed (WW), maximum wind speed (WWX), and prevailing wind direction (DD), while the target variable is data on tomorrow's rainfall (Besok_hujan). To preserve the temporal ordering inherent in meteorological observations and prevent data leakage, the data was divided chronologically: the first 80% of the time-ordered records (January 2019 to December 2023) were assigned to the training set, and the remaining 20% (January to December 2024) were reserved as the independent test set. Subsequently, the SMOTE method was applied exclusively to the training partition to balance class distribution, ensuring that no synthetic samples generated from training data could appear in or influence the evaluation on the test set. The source code for splitting the data and applying the SMOTE method is shown in Figure 7.

```
# split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=2)  # Inisialisasi SMOTE

# Terapkan SMOTE hanya pada data training
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42)
```

Figure 7. Source Code Splitting Data and SMOTE

D. Prediction Model Training

The training of the Logistic Regression, Random Forest, and Extreme Gradient Boosting algorithm-based model is shown in the source code in Figure 8. The results of daily rainfall predictions in Padang City using the Logistic Regression, Random Forest, and Extreme Gradient Boosting algorithm-based models are then used to obtain the TP, TN, FP, and FN values for each model Predictions whose results can be depicted in the form of a confusion matrix.

```
from sklearn.metrics import roc_curve, roc_auc_score, f1_score
def data_training_(X_train, X_test, y_train, y_test):
    models = []
    models.append(('LR', LogisticRegression(random_state=0)))
    models.append(('RF', RandomForestClassifier(n_estimators=100, random_state=42)))
    models.append(('XGB', XGBClassifier(
        learning_rate=0.23, max_delta_step=5,
        objective='reg:logistic', n_estimators=92,
        max_depth=5, eval_metric="logloss", gamma=3, base_score=0.5)))
    res_cols = ["Model", "Accuracy", "Precision", "Recall", "F1-score"]
    df_result_ = pd.DataFrame(columns=res_cols)
    index = 0
    for name, model in models:
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        score = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred)
        f1_ = f1_score(y_test, y_pred)
        idx_res_values = [name, score, precision, recall, f1_]
        df_result_.loc[index, res_cols] = idx_res_values
        index += 1
    df_result_ = df_result_.sort_values("Accuracy", ascending=False).reset_index(drop=True)
    return df_result_
```

Figure 8. Prediction Model Source Code

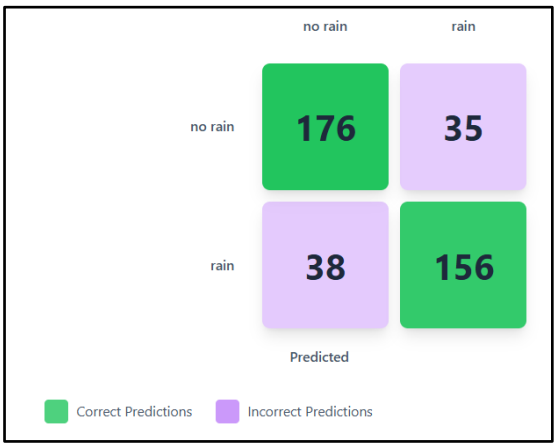


Figure 9. Confusion Matrix Logistic Regression Model

An example of a confusion matrix can be seen in Figure 9, which is a confusion matrix of prediction results using the Logistic Regression algorithm. The values in the confusion matrix for each prediction model are then used as input to determine the evaluation value, namely to determine the accuracy, precision, recall, and F1 score for each prediction model.

E. Prediction Model Evaluation

The evaluation results of the Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting (XGB) prediction models are shown in Table 2, along with the prediction models' accuracy, precision, recall, and F1 score. The evaluation findings in Table 2 show that all three prediction models perform well in terms of data forecasting. The accuracy, precision, recall, and F1 score values of all three models are greater than 80%, with the exception of the XGB prediction model's recall number, which is 78%. Additionally, Table 2 demonstrates that the Random Forest algorithm-based prediction model, with assessment scores of 85%, 89%, and 85% for accuracy, precision, and F1 score, generally outperformed the other prediction models. With a recall value of 83%, the Logistic Regression prediction model outperformed the Random Forest prediction model, which had a recall value of only 82%. This suggests that the Random Forest prediction model outperformed the other two models in terms of properly predicting data for the positive class, predicting data in multiple classes, and making accurate predictions out of the total amount of predictions. When it came to accurately identifying the positive class, the Logistic Regression prediction model outperformed the other models.

Table 2. Prediction Model Evaluation Results

Metrik Evaluasi	LR (%)	RF (%)	XGB (%)
Accuracy	82	85	81
Precision	82	89	84
Recall	83	82	78
F1 Score	83	85	81

The Logistic Regression prediction model has greater performance compared to the Extreme Gradient Boosting prediction model, as evidenced by the accuracy, recall, and F1 score values presented in Table 2. Overall, the Extreme Gradient Boosting prediction model had the poorest performance relative to the other models, as indicated by the evaluation findings in Table 2, with the exception of its accuracy value, which surpassed that of the Logistic Regression model. The Receiver Operating Characteristic (ROC) graph in Figure 10 can be used to assess the performance of the Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting (XGB) prediction models in addition to evaluating accuracy, precision, recall, and F1 score values. To show how well each prediction model performs in differentiating between positive and negative classes at different thresholds, the graph shows the ROC curve and Area Under the Curve (AUC) value. The AUC values for each of the three prediction models in Figure 5 are near to 1, and their ROC curves tend to slope towards the upper left corner. This suggests that each of the three prediction models can effectively differentiate across classes. With AUC values >0.9, at 0.933 and 0.913, respectively, the Random Forest and Extreme Gradient Boosting prediction models exhibit excellent performance. With an AUC of 0.897, the Logistic Regression prediction model exhibits good performance. Based on the ROC graph, the Random Forest prediction model outperformed the other two models in class differentiation and had the greatest AUC value. With the lowest AUC value, the Logistic Regression prediction model did not perform well.

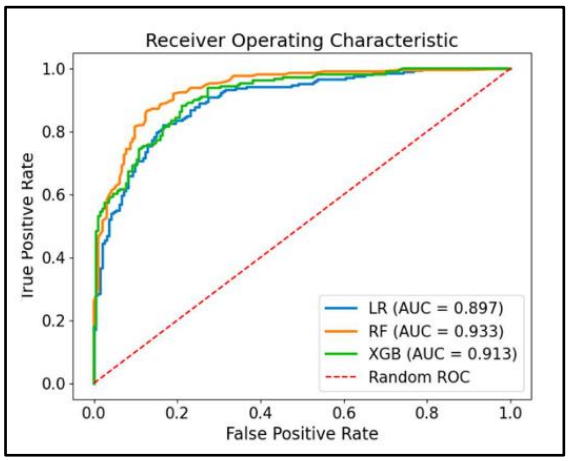


Figure 10. ROC graph

V. CONCLUSIONS AND SUGGESTIONS

The application of Logistic Regression, Random Forest, and Extreme Gradient Boosting algorithms in a rainfall prediction model for Padang City demonstrates that all three models have strong predictive capabilities, thereby qualifying them for consideration as predictive models. The Random Forest prediction model had superior performance among the three models, as evidenced by its accuracy, precision, recall, F1 score, and AUC values of 85%, 89%, 82%, 85%, and 0.933, respectively. The recall value is slightly different than the Logistic Regression technique, recorded at 83%.

REFERENCES

[1] H. Ashouri et al., "PERSIANN-CDR: Daily Precipitation Climate Data Record from Multisatellite Observations for Hydrological and Climate Studies," Bulletin of the American Meteorological Society, vol. 96, pp. 69-83, 2015, doi: 10.1175/BAMS-D-13-00068.1.

[2] G. Ayehu, T. Tadesse, B. Gessesse, and T. Dinku, "Validation of new satellite rainfall products over the Upper Blue Nile Basin, Ethiopia," Atmospheric Measurement Techniques, vol. 11, pp. 1921-1936, 2017, doi: 10.5194/AMT-11-1921-2018.

[3] Y. A. Bayissa, T. Tadesse, G. Demisse, and A. Shiferaw, "Evaluation of Satellite-Based Rainfall Estimates and Application to Monitor Meteorological Drought," Remote Sens., vol. 9, p. 669, 2017, doi: 10.3390/rs9070669.

[4] H. Beck et al., "Global-Scale Evaluation of 22 Precipitation Datasets Using Gauge Observations and Hydrological Modeling," Advances in Global Change Research, 2017, doi: 10.5194/HESS-2017-508.

[5] H. Beck et al., "MSWEP: 3-hourly 0.25° global gridded precipitation (1979–2015) by merging gauge, satellite, and reanalysis data," Hydrology and Earth System Sciences, vol. 21, pp. 589-615, 2016, doi: 10.5194/HESS-21-589-2017.

[6] C. Funk et al., "A global satellite-assisted precipitation climatology," Earth System Science Data, vol. 7, pp. 275-287, 2015, doi: 10.5194/ESSD-7-275-2015.

[7] C. Funk et al., "The climate hazards infrared precipitation with stations (CHIRPS)—a new environmental record for monitoring extremes," Scientific Data, vol. 2, 2015, doi: 10.1038/sdata.2015.66.

[8] C. Funk et al., "Predicting East African spring droughts using Pacific and Indian Ocean sea surface temperature indices," Hydrology and Earth System Sciences, vol. 18, pp. 4965-4978, 2014, doi: 10.5194/HESS-18-4965-2014.

[9] A. Hoell and C. Funk, "Indo-Pacific sea surface temperature influences on failed consecutive rainy seasons," Climate Dynamics, vol. 43, pp. 1645-1660, 2013, doi: 10.1007/s00382-013-1991-6.

[10] K. Hsu, X. Gao, S. Sorooshian, and H. Gupta, "Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Networks," Journal of Applied Meteorology, vol. 36, pp. 1176-1190, 1997, doi: 10.1175/1520-0450(1997)036<1176:PEFRSI>2.0.CO;2.

[11] G. Huffman et al., "The TRMM Multisatellite Precipitation Analysis (TMPA): Quasi-Global, Multiyear, Combined-Sensor Precipitation Estimates at Fine Scales," Journal of Hydrometeorology, vol. 8, pp. 38-55, 2007, doi: 10.1175/JHM560.1.

[12] R. Joyce, J. Janowiak, P. Arkin, and P. Xie, "CMORPH: A Method that Produces Global Precipitation Estimates from Passive Microwave and Infrared Data," Journal of Hydrometeorology, vol. 5, pp. 487-503, 2004, doi: 10.1175/1525-7541(2004)005<0487:CAMTPG>2.0.CO;2.

[13] B. Liebmann et al., "Understanding Recent Eastern Horn of Africa Rainfall Variability and Change," Journal of Climate, vol. 27, pp. 8630-8645, 2014, doi: 10.1175/JCLI-D-13-00714.1.

[14] R. I. Maidment et al., "The 30 year TAMSAT African Rainfall Climatology And Time series (TARCAT) data set," Journal of Geophysical Research: Atmospheres, vol. 119, pp. 10619-10644, 2014, doi: 10.1002/2014JD021927.

[15] M. Sadeghi et al., "PERSIANN-CCS-CDR, a 3-hourly 0.04° global precipitation climate data record for heavy precipitation studies," Scientific Data, vol. 8, 2021, doi: 10.1038/s41597-021-00940-9.

[16] S. Sorooshian et al., "Evaluation of PERSIANN system satellite-based estimates of tropical rainfall," Bulletin of the American Meteorological Society, vol. 81, pp. 2035-2046, 2000, doi: 10.1175/1520-0477(2000)081<2035:EOPSSE>2.3.CO;2.

[17] E. Tarnavsky et al., "Extension of the TAMSAT Satellite-Based Rainfall Monitoring over Africa," Journal of Applied Meteorology and Climatology, vol. 53, pp. 2805-2822, 2014, doi: 10.1175/JAMC-D-14-0016.1.

[18] A. Al-Azzawi, "Deep Learning Approach for Handwritten Image Recognition Based on a High Dimensionality Reduction and Feature Selection Using Stacked Sparse Auto-Encoder Structure," 2016.

[19] A. Ridwan et al., "Rainfall forecasting model using machine learning methods: Case study Terengganu, Malaysia," Ain Shams Engineering Journal, vol. 12, no. 2, pp. 1651-1663, 2021, doi: 10.1016/j.asej.2020.09.011.
[20] H. Nguyen et al., "Random forest classification of daily rainfall occurrence using meteorological variables in a tropical area," Meteorological Applications, vol. 28, no. 4, 2021, doi: 10.1002/met.2009.
[21] B. C. Sahoo, B. K. Nanda, and T. Sahoo, "Rainfall classification and prediction using machine learning methods: A review," Computers and Electronics in Agriculture, vol. 200, p. 107243, 2022, doi: 10.1016/j.compag.2022.107243.
[22] S. Poornima and M. Pushpalatha, "Prediction of rainfall using intensified LSTM based recurrent neural network with weighted linear units," Atmosphere, vol. 10, no. 11, p. 668, 2019, doi: 10.3390/atmos10110668.
[23] P. Hossain et al., "Machine learning approach in rainfall prediction: A review," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, pp. 562-574, 2022, doi: 10.14569/IJACSA.2022.0130367.
[24] N. V. Chawla et al., "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002, doi: 10.1613/jair.953.
[25] K. Ahmed et al., "Improving the accuracy of rainfall prediction using a machine learning approach: A case study of Bangladesh," IEEE Access, vol. 8, pp. 109068-109085, 2020, doi: 10.1109/ACCESS.2020.2995162.
[26] C. Häusler, A. K. Susemihl, M. Nawrot, and M. Opper, "Temporal Autoencoding Improves Generative Models of Time Series," arXiv preprint arXiv:1309.3103, 2013. (Penting untuk pemodelan temporal hujan).
[27] M. A. Islam et al., "Comparative analysis of machine learning techniques for daily rainfall prediction in Bangladesh," Hydrology, vol. 9, no. 10, p. 173, 2022, doi: 10.3390/hydrology9100173.
[28] T. Kim, D. Zhang, and J. Kim, "Effects of Model Complexity on Generalization Performance of Convolutional Neural Networks," 2013.
[29] W. Kusonkhum et al., "Government Construction Project Budget Prediction Using Machine Learning," Journal of Advances in Information Technology, vol. 13, no. 1, pp. 29-35, 2022.
[30] H. Lee and M. Ranzato, "Tutorial on Deep Learning and Applications," 2010. [Online]. Available: https://www.semanticscholar.org/paper/1dbb5dc7de4f2328708525fa74b87f99ac532dd2
[31] R. C. Deo and M. Śahin, "Application of the extreme learning machine algorithm for the prediction of monthly Effective Drought Index in eastern Australia," Atmospheric Research, vol. 153, pp. 512-525, 2015, doi: 10.1016/j.atmosres.2014.10.016.
[32] X. Li et al., "DRN: Bringing Greedy Layer-Wise Training into Time Dimension," in 2015 IEEE International Conference on Data Mining, 2015, pp. 859-864. (Relevan untuk sekuensial waktu).
[33] W. D. Pramesti, M. T. Furqon, and C. Dewi, "Implementasi metode K-Medoids Clustering untuk pengelompokan data curah hujan di Kabupaten Malang," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 1, no. 9, pp. 723-731, 2017.
[34] T. B. Trafalis and R. C. Gilbert, "Robust classification and regression using support vector machines," European Journal of Operational Research, vol. 173, no. 3, pp. 893-909, 2006, doi: 10.1016/j.ejor.2005.07.070.
[35] M. A. Zainudin, S. Shaharanee, and J. M. Jamil, "Measurement of performance for rainfall prediction system using XGBoost algorithm," Journal of Telecommunication, Electronic and Computer Engineering, vol. 10, no. 1-14, pp. 37-41, 2018.
[36] A. Kusuma, M. R. Widyanto, and B. Susanto, "Rainfall prediction using extreme gradient boosting," IOP Conference Series: Earth and Environmental Science, vol. 794, p. 012060, 2021, doi: 10.1088/1755-1315/794/1/012060.
[37] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.
[38] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.
[39] S. P. Patle et al., "Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks," Hydrology and Earth System Sciences, vol. 22, pp. 6105-6122, 2018, doi: 10.5194/hess-22-6105-2018.
[40] M. A. Treerat, P. Kamnuansri, and S. Suwannatrai, "Improving prediction model of rainfall using feature importance analysis," in 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2019, pp. 321-326, doi: 10.1109/JCSSE.2019.8864217.
[41] A. Salman et al., "Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting," Procedia Computer Science, vol. 135, pp. 89-98, 2018, doi: 10.1016/j.procs.2018.08.153.