Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2025, 2025:108

https://doi.org/10.28919/cmbn/9385

ISSN: 2052-2541

RAINFALL FORECASTING USING EXTREME GRADIENT BOOSTING

IVANA SAKURA INDAH MARGARETH, YUYUN HIDAYAT, SRI WINARNI*

Department of Statistics, Universitas Padjadjaran, Jl. Raya Bandung Sumedang km 21 Jatinangor, Sumedang 45363, Indonesia

Abstract: Rainfall is an important factor affecting the agricultural sector, especially in areas such as Grobogan Regency, Central Java, which is one of the national rice barns. Rainfall uncertainty often leads to crop failure due to drought or flooding, which is caused by inaccurate rainfall predictions. This research aims to build a dasarian rainfall forecasting model using the Extreme Gradient Boosting (XGBoost) method. The data used is ten-day (decadal) rainfall data from January 1991 to February 2025. The research steps start from data preprocessing which includes lag and seasonal feature creation, hyperparameter determination and hyperparameter tuning, and model evaluation. After conducting hyperparameter tuning to find the best model, the best model was found with *n_estimator* 448, *learning rate* 0.14, *max_depth* 3, *min_child_weight* 1, *subsample* 0.71, *colsample_bytree* 0.97, *gamma* 5.1, *alpha* 3.8, and *lambda* 8.89. The model formed is then evaluated using SMAPE (Symmetrical Mean Absolute Error) for test data of 36.48%. From the model formed, forecasting is carried out and the forecasting results can be utilized as a basis for decision making in agricultural planning.

Keywords: rainfall; forecasting; XGBoost; Grobogan; rice.

2020 AMS Subject Classification: 62M20, 68T07.

*Corresponding author

E-mail address: sri.winarni@unpad.ac.id

Received May 27, 2025

1. INTRODUCTION

Rainfall is one of the important factors in various aspects of human life. The economic sector, agriculture, plantations, and infrastructure are some of the sectors that are greatly influenced by rainfall. In the agricultural sector, rainfall not only determines water availability but also directly impacts the success of agricultural activities, such as planting schedule planning, irrigation management, and pest and disease control.

Central Java Province, as one of the national rice granaries, plays a strategic role in supporting national food security. Based on data from the Badan Pusat Statistik (BPS), Central Java is recorded as the third-largest rice producer in Indonesia in 2023, with a total rice production of 9,084,107.53 tons and an average productivity of 55.30 quintals/ha [1].Based on BPS data from 2023, the largest rice-producing area is in Cilacap Regency with a total production of 772,488.96 tons, followed by Grobogan Regency with a total production of 678,350.41 tons, and the third place is Sragen Regency with a total production of 641,988.28 tons [2].

In the years 2019-2024, the inaccuracy of rainfall predictions became a serious challenge in the agricultural sector, especially in tropical regions like Central Java. Inaccurate predictions, whether overestimating or underestimating rainfall intensity, often disrupt planting, irrigation, and harvesting schedules. This results in losses for the farmers. One of the main causes of this inaccuracy is the high data fluctuations and the limitations of conventional prediction methods in capturing dynamic and unpredictable weather patterns. BMKG has used several statistical models and machine learning to predict rainfall, such as ARIMA, SARIMA, and ANFIS. However, these models have weaknesses such as sensitivity to missing data, difficulty in handling non-linear patterns, and limitations in recognizing seasonal and fluctuating characteristics. Therefore, a more flexible approach that can accommodate the complexity of climate data is needed.

Extreme Gradient Boosting (XGBoost) has emerged as one of the ensemble learning methods that demonstrate high performance in various weather forecasting studies. XGBoost can forecast monthly rainfall in the city of Bandung better than the SARIMA method, Exponential Smoothing, and Artificial Neural Network [3]. Meanwhile, there is also a study comparing machine learning models for predicting rainfall in New Delhi, India, and it found that ensemble methods like CatBoost, LGBM, and XGBoost outperform linear regression [4]. However, the CatBoost method is used for categorical data, whereas the data used is numerical, making it less suitable, and LGBM cannot handle outliers.

Based on these advantages, this research aims to apply XGBoost in forecasting daily rainfall in Grobogan Regency, Central Java, Indonesia as an alternative prediction. The forecasting results using the developed model can be used to make decisions in planning rice planting periods or other policy-making that impacts the agricultural sector in the region.

2. Material and Method

2.1 Data

The data used in this research is secondary data on decadal rainfall in Grobogan Regency, obtained and managed by the BMKG Central. The data to be studied is historical decadal data starting from January 1, 1991, to February 1, 2025 (each month has 3 decadal data points). Where the data has characteristics of autocorrelation, non-linearity, seasonality, outliers, and high variability as shown in Figure 1.

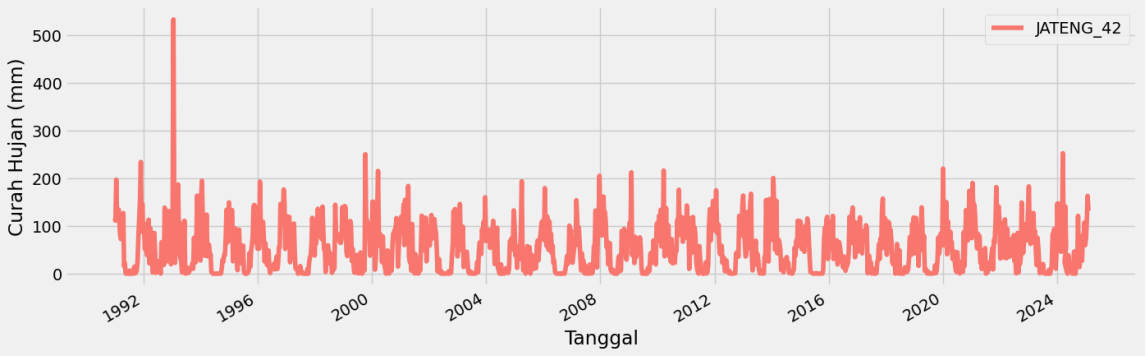

Figure 1. Rainfall Plot for Grobogan Regency, Central Java January 1991 – February 2025

2.2 Data Preprocessing

Data preprocessing is a series of techniques used to clean, transform, and prepare raw data to meet the needs of further analysis or modeling. The main objective of this stage is to improve data quality by eliminating defects, reducing noise, and ensuring the data is ready to be used in generating deeper insights through analysis or model building. At this stage, several processes will be carried out so that the rainfall data is ready for analysis using the XGBoost method.

2.2.1 Windowing

Windowing is a technique used to transform time series data into a cross-sectional machine learning input dataset. The available data is divided into several parts or windows based on a specific time range. Windowing can be done by adding several lag features to the data.

Lag features are values from previous time steps that are considered useful because based on the assumption that what happened in the past can influence or provide information about the future.

The use of previous time steps to predict the next time step is called the sliding window method.

2.2.2 Date Time Feature

Date time feature is a feature generated from time-related data, such as dates, months, and others, created from time units of each observation. This is done because it is necessary to extract additional information from the existing time so that the model can better understand seasonal patterns, trends, or cycles in the data. The transformation of the date-time feature is done by adding new columns such as hour, month, day of the month, day of the year, and others if necessary.

2.2.3 Rolling Window Statistics

Rolling Window Statistics is a technique for calculating statistical values such as mean, median, maximum value, quartiles, and others on time series data using a rolling window. This window includes the current data sample as well as several previous or subsequent samples. The size or period of this window is usually determined by the researcher based on the characteristics of the data and the purpose of the analysis. The calculated values are then added to the data by creating a new column. After the rolling window period is established, the statistical calculations are repeated for each window according to the specified period.

2.3 Data Splitting

Data splitting is the separation of a dataset into several subsets used in the development and evaluation of machine learning models. This method is carried out with the aim of avoiding overfitting on the data and measuring the model's performance. Generally, in machine learning models, data is divided into two parts: training data and testing data. In this research, the data will be split into training and testing data in an 80:20 ratio. The training data will be used to train or build the XGBoost model, while the test data will be used to evaluate how well the model performs on new data.

2.4 XGBoost Modeling

Extreme Gradient Boosting (XGBoost) is an enhancement of the gradient boosting method that uses decision trees as weak learners. XGBoost improves model performance by adding regularization to control tree complexity [5]. XGBoost applies a parallel process during the training phase, particularly when building trees and calculating the best splits through gradient histograms, making the training faster and more efficient. For a dataset of n data points, the equation of the ensemble tree model is as follows:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i), f_t \in \mathcal{F} \quad (1)$$

In this equation, $\hat{y}_i$ representing the predicted value, and f_t is a regression tree of size t in the set of regression trees $\mathcal{F}$. Next, calculate the objective function.

$$\mathcal{L}(\phi) = \sum_{i=1} l(y_i, \hat{y}_i) + \sum_t \Omega(f_t) \quad (2)$$

In Equation 2, $l(y_i, \hat{y}_i)$ is a loss function that used to measure the error between predictions and actual values. $\Omega(f_t)$ is a regularization that used to used to control model complexity to prevent overfitting. Regularization is stated in the Equation (3).

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{t=i}^{T} {\omega_j}^2 \quad (3)$$

In this equation, ω_j represent weight of the j-th leaf, γ represent L1 regularization factor, λ represent L2 regularization factor, T represent number of iterations, and t represent iteration. The equation (3) cannot be optimized using traditional optimization, so an iterative process was carried out. Then, f_t was added to minimize the objective function, so it can be written as an equation (4).

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, {\hat{y}_i}^{(t-1)} + f_t(x_i)\right) + \sum_{i=1}^{t} \Omega(f_t) \quad (4)$$

Optimizing the loss function is usually done using a second-order Taylor approximation. It forms as in the equation as follows:

$$\tilde{\mathcal{L}}^{(t)} \simeq \sum_{i=1}^{n} l\left(y_i, {\hat{y}_i}^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)\right) + \Omega(f_t) \quad (5)$$

Where, $g_i = \partial_{\hat{y}_{(t-1)}} l(y_i, {\hat{y}_i}^{t-1})$ and $h_i = \partial^2{}_{\hat{y}_{(t-1)}} l(y_i, {\hat{y}_i}^{t-1})$ are the first and second orders of the statistical gradient on the loss function. By removing the constant in equation (5), a simpler form is obtained to achieve a simpler objective function that is written in equation (6).

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left(g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)\right) + \Omega(f_t) \quad (6)$$

i represent the i-th data, t represent the t-th iteration, n represent number of data, g_i represent gradien of the i-th data, h_i represent hessian of the i-th data, $f_t(x_i)$ represent prediction of the t-th iteration on the i-th data (improvement contribution), and $\Omega(f_t)$ represent regularization. By using $\Omega(f_t)$ in equation (3), equation (6) is further decomposed into a simpler form bellow:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left(g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)\right) + \gamma T + \frac{1}{2}\gamma \sum_{t=1}^{T} {\omega_j}^2$$

$$\tilde{\mathcal{L}}^{(t)} = \sum_{j=1}^{T} \left[\left(\sum_{i \epsilon I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i \epsilon I_j} h_i + \lambda\right)\left(\sum_{i \epsilon I_j} g_i\right){\omega_j}^2\right] + \gamma T \quad (7)$$

i represent the i-th data, t represent the t-th iteration, n represent number of data, j represent the j-th leaf, g_i represent gradien of the i-th data, h_i represent hessian of the i-th data, ω_j

represent weight of the j-th leaf, γ represent L1 regularization factor, λ represent L2 regularization factor, and T represent number of iterations. It can be seen that the value $\left[\left(\sum_{i\epsilon I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\epsilon I_j} h_i + \lambda\right)\left(\sum_{i\epsilon I_j} g_i\right)\omega_j{}^2\right]$ is a polynomial in terms of ω_j. Thus, the optimal value ω_j is obtain when

$$\omega^*{}_j = -\frac{\sum_{i\epsilon I_j} g_i}{\sum_{i\epsilon I_j} h_i + \lambda} \tag{8}$$

Where i represent the i-th data, j represent the j-th leaf, g_i represent gradien of the i-th data, h_i represent hessian of the i-th data, ω_j represent weight of the j-th leaf, and λ represent L2 regularization factor. To obtain the optimal value of the loss function, equation (8) is substituted into equation (7).

$$\tilde{\mathcal{L}}^{(t)} = -\frac{1}{2}\sum_{i=1}^{n} \frac{\left(\sum_{i\epsilon Ij} g_i\right)^2}{\sum_{i\epsilon Ij} h_i + \lambda} + \gamma T \tag{9}$$

i represent the i-th data, n represent number of data, j represent the j-th leaf, g_i represent gradien of the i-th data, h_i represent hessian of the i-th data, γ represent L1 regularization factor, and T represent number of iterations. Next, equation (9) can be used as a scoring function to measure the quality of the tree. The measurement is conducted by evaluating the best splitting result of the node. Let I_L and $I_R$be the sets of examples from the left and right nodes after the split, then the reduction in loss after the split is given by:

$$Gain = \frac{1}{2}\left[\frac{G^2{}_L}{H_L+\lambda} + \frac{G^2{}_R}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{10}$$

Where H_L, H_R represent number of Hessians in the left and right branches, G_L, G_R represent number of gradients in the left and right branches, γ represent L1 regularization factor, and λ represent L2 regularization factor.

If Gain has a positive value or Gain > 0 and is significant, then a split or construction of a new tree is performed. Conversely, if Gain is negative or Gain < 0 or too small (less than γ), the split is not performed. The resulting value will be said to be optimal when the value of each iteration has a downward trend and is not too low (almost zero). A downward trend in value indicates that the new tree is learning from the previous tree. This means the model is learning well from previous errors.

To make it a fit model, the XGBoost method requires hyperparameter tuning. Hyperparameter tuning is done in order to optimize model building to improve the model created by changing the parameter values that affect the training process. The value of the hyperparameter has a subjective

element where the value needs to be initiated by the model builder before building the model and the value is independent of the data or model used. The parameter that used in this research include n_estimator, learning rate, max depth, min child weight, subsample, colsample_bytree, gamma, alpha, and lambda [6].

Hyperparameter tuning process is required to determine the parameter combinations that yield optimal performance to obtain the best XGBoost model. Several commonly used methods for tuning include Grid Search, Random Search, and Bayesian Optimization. In this study, the Bayesian Optimization method was used with the help of the Optuna library. This method works efficiently by using information from previous experiments to guide the search for the next parameter combinations based on probabilistic distribution. The process begins by determining the range of hyperparameter values, followed by the exploration of initial combinations randomly, and then iterating towards the best results. Evaluation is conducted using 5-fold cross-validation to maintain result consistency and prevent overfitting.

In addition to tuning, feature selection is also performed to filter the features that have the most influence on the prediction results. This process aims to reduce noise, speed up computation time, and improve model interpretability. Evaluation is conducted using the feature importance method, which shows the relative contribution of each feature to the model's performance. Features that are considered insignificant will be removed, and model training will be conducted again using the selected features. This approach helps produce a model that is more efficient and relevant to the rainfall patterns that need to be predicted.

2.5 Evaluation Model

Model evaluation needs to be calculated to assess how well the algorithm of the selected model performs. This research will use the evaluation method of SMAPE (Symmetric Mean Absolute Percentage Error) to compare the performance of the prediction results with the methods used in terms of prediction errors.

SMAPE takes into account the relative difference between the predicted value and the actual value presented as a percentage, considering the magnitude of the values equally. Unlike other error measures, SMAPE gives equal weight to overestimates and underestimates.

SMAPE is used because the data used contains zero or near-zero values. The smaller the SMAPE value produced, the better the model can make predictions. Suppose there are many samples n with forecast results $\hat{y}_i$ (the i-th forecasted data) and y_i (the i-th actual data). The formula for SMAPE is formulated in Equation (11)

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{\left(\frac{|y_i| + |\hat{y}_i|}{2}\right)} \; X \; 100\% \quad (11)$$

3. MAIN RESULTS

3.1 Data Preprocessing

Rainfall data is a time series data that certainly has temporal dependencies among its data points. The algorithm used in XGBoost cannot directly learn the patterns of time series data. However, the algorithm is capable of learning the temporal aspects of time series data in the form of features. Therefore, in order for the rainfall data to be processed, the data is handled through feature engineering processes such as windowing, date time feature, and rolling window statistics.

Windowing is done by adding lag features to the data. With the presence of lag features, the model can learn patterns from previous observations and recognize them as features. In this research, 6 lag features, or 6 previous data points, are used to learn the patterns.

Date time feature is performed by extracting time elements found in the rainfall data. Some date time features used include the date, decadel, month, and year. Thus, the data does not lose the time information contained within it because, in time series data, time information is very important.

Rolling window statistics are performed by calculating several statistical values such as mean, median, mode, standard deviation, or other simple statistical analyses. In this research, the mean and standard deviation that continuously move along the data for 3, 6, 9, and 18 decades will be used. Thus, the model is given a better understanding of the trends, patterns, and variability of the data.

3.2 Data Splitting

After preprocessing the data, the next step is to split the training data and testing data. Training data is useful for recognizing patterns in the data and building the model, while testing data is useful for assessing the accuracy of the model that generates predictions. In this research, the total data used amounts to 1227 data points. The data distribution to be used is 80% for training data, which amounts to 981 data points, and 20% for testing data, which amounts to 246 data points.

3.3 XGBoost Modeling

3.3.1 Determination of Hyperparameter and Hyperparameter Tuning

After the training and testing data have been divided, modeling will be carried out using XGBoost. To determine the parameters that will be used in the model, hyperparameter tuning is first conducted, with the parameter values used visible in Table 1.

Table 1. Hyperparameter XGBoost

Parameter	Value
Learning rate	[0.01-0.3]
N_*estimators*	[50-1000]
Gamma	[0-10]
Subsample	[0.5-1]
Colsample_by tree	[0.5-1]
Max_depth	[3-10]
Min_child_weight	[1-5]
Alpha	[0-10]
Lambda	[0-10]

After determining the parameters used, the hyperparameter tuning process is performed. The best model is selected based on the smallest SMAPE (Symmetric Mean Absolute Percentage Error) value in the cross-validation results. The cross-validation process uses 5-fold, meaning that the data is divided into five parts, where each part in turn is used as validation data while the rest is used as training data. This approach helps to produce a more stable and accurate model. The average prediction error of all parameter combinations was calculated, and the model with the lowest error was selected as the best model.

3.3.2 Evaluation Model

With the help of computers, several of the best models that enable forecasting have been obtained. The best values of each parameter in those models can be seen in Table 2.

Table 2. Value of Each Model Parameter

Parameter	Model 1	Model 2	Model 3
n_estimators	448	450	900
Learning rate	0,14	0,07	0,05
Max_depth	3	4	3
Min_child_weight	1	2	1
Subsample	0,71	0.77	0,87
Colsample_bytree	0,97	0,89	0,97
Gamma	5,1	9,73	1,75
Alpha	3,8	7,79	5,92
Lambda	8,89	3,65	4,84

Next, that value is chosen as the XGBoost model. After selection, the model is tested against the test data. This is done to test whether the selected model can indeed capture the patterns from the data. After the model was trained on the test data, to assess the model's performance in generating ten-day rainfall forecasts in Grobogan Regency, the evaluation value was calculated using SMAPE. The SMAPE comparison values for each model can be seen in Table 3.

Table 3. Comparison of SMAPE Evaluation Results

Data	Model 1	Model 2	Model 3
Training	26,65%	26,32%	26,75%
Testing	36,48%	37,78%	36,92%

Based on Table 3, it can be seen that the model with the best SMAPE value with the smallest SMAPE value is in model 1. The resulting SMAPE model is good enough to be able to do forecasting. To make sure, look at the comparison graph of actual data and prediction data using training data and test data.

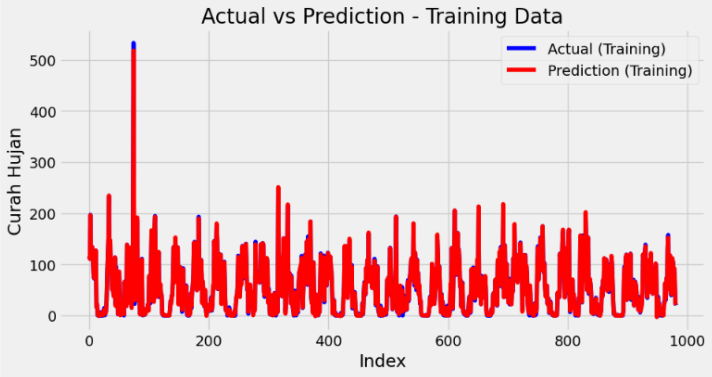

Figure 2. Comparison Between Actual and Prediction (Training)

From Figure 2, an evaluation of the training data was produced with an SMAPE value of 26.65%. It can be seen in the image that the model's prediction results can follow the pattern of the actual data, although not all of them are the same. The prediction line follows the pattern of the actual data line quite accurately, especially during fluctuations that indicate high rainfall, although not exactly the same. For the evaluation of the test data, it can be seen below.

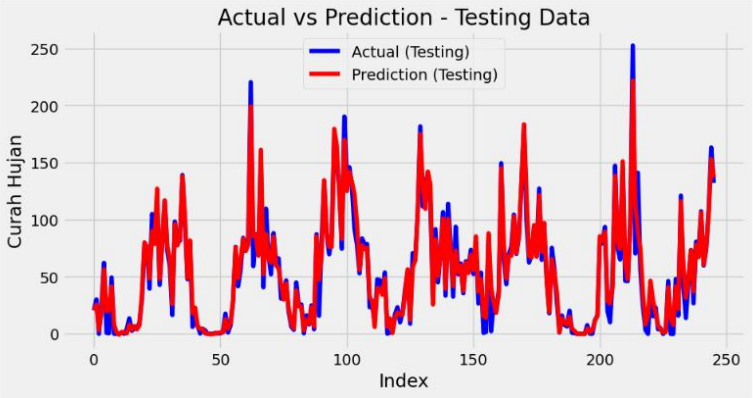

Figure 3. Comparison Between Actual and Prediction (Testing)

Based on Figure 3, the evaluation results on the test data show an SMAPE value of 36.48%. Although this value is quite high, the model generally manages to follow the actual data pattern, especially in areas with high rainfall fluctuations. Although not exactly the same, the prediction line is quite in line with the existing trends and seasonal patterns, indicating that the model has successfully captured the main characteristics of the data. Thus, the model can be considered quite good. Next, a feature importance analysis is conducted to determine the features that have the most influence in forming the decision tree. The results are displayed in Figure 4.

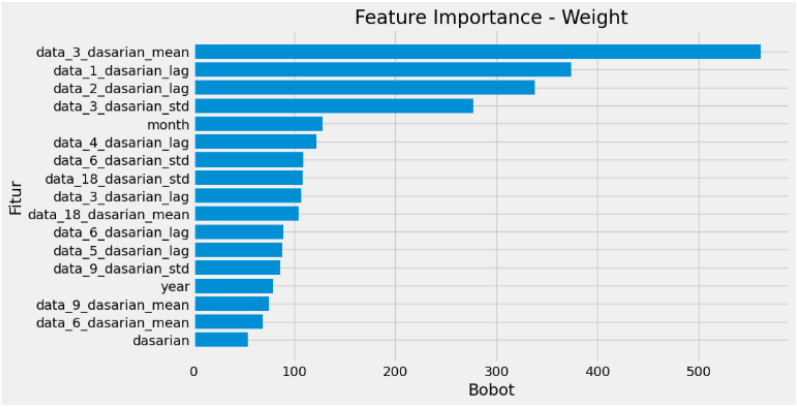

Figure 4. Feature Importance

Feature importance is calculated for a single decision tree based on how many times a feature causes a split that can improve the model's performance. Weight refers to the number of occurrences of a feature in the decision trees that have been built, where the displayed value is obtained from the average weight of all the decision trees in the model. Based on Figure 4, it can be seen that data_3_dasarian_mean has the highest level of importance that means data_3_dasarian_mean is the most frequently used feature in model development with a weight of more than 500. Next, data_1_dasarian_lag, data_2_dasarian_lag, and data_3_dasarian_std also

significantly contribute to the model development. After that, it is followed by the month in the year, data_4_dasarian_lag, data_6_dasarian_std, data_18_dasarian_std, and so on.

3.4 Forecasting

After obtaining the best model from XGBoost (Extreme Gradient Boosting) with the most optimal evaluation results, the next step is to forecast rainfall for the next 6 decadal.

Table 4. Forecast Results for the Next 6 Decadal

Decadal	Forecasting
11/02/2025	41,46
21/02/2025	37,92
01/03/2025	113,86
11/03/2025	201,8
21/03/2025	174,32
01/04/2025	30,14

Based on Table 4, the forecasting results show quite significant fluctuations in rainfall. The first two decadal forecast low rainfall (around 41.46 mm and 37.92 mm), followed by a sharp increase to 201.8 mm, then dropping back to 30.14 mm in the last decadal. Despite the low values, there are no definite indications of a seasonal change. Based on BMKG criteria, it does not yet meet the requirements to be considered the beginning of the dry season or the rainy season, but this period can be categorized as a transitional phase.

With this forecast, rainfall in Grobogan Regency is expected to increase significantly in March, particularly in the medium to high category. These conditions are favorable for rice planting, especially in early to mid-March. However, high rainfall has the potential to cause flooding, so it is important for farmers to ensure that the drainage system functions well and to prepare water drainage channels, including the possibility of using pumps. High humidity can also trigger pests and diseases, so monitoring of plants and fertilizer management need to be carried out. As April approaches, rainfall is expected to decrease sharply, so farmers need to maintain water availability to prevent drought during the planting season.

4. CONCLUSION

The XGBoost model that was formed is capable of capturing patterns in the data. The ten-day rainfall forecasting model in Grobogan Regency, built using the Extreme Gradient Boosting (XGBoost) method, was successfully optimized through the hyperparameter tuning process. The

best model was obtained with the *n_ estimator* 448, *learning rate* 0.14, *max_depth* 3, *min_child_weight* 1, *subsample* 0.71, *colsample_bytree* 0.97, *gamma* 5.1, *alpha* 3.8, and *lambda* 8.89. This model shows quite good performance with an SMAPE value of 36.48% on the test data, indicating its ability to capture seasonal patterns and the complexity of daily rainfall data in the region.

Based on the forecasting results, the researchers recommend that farmers are advised to plant rice from early to mid-March because high rainfall is predicted, which supports plant growth. The drainage system and water disposal channels need to be ensured to function well to prevent flooding, and routine monitoring of the plants should be conducted to anticipate pests and diseases due to high humidity. Additionally, farmers need to maintain water availability as rainfall is expected to decrease in early April.

ACKNOWLEDGEMENTS

The authors expresses gratitude to Directorate of Research and Community Engagement of Padjadjaran University for their valuable support during the writing of this paper and for providing financial support for publication in this journal. This assistance is highly valuable in facilitating the completion of this research and contributing to the author's academic work.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] Badan Pusat Statistik, Luas Panen, Produksi, dan Produktivitas Padi Menurut Provinsi, 2024, Accessed: Sep. 17, 2024. https://www.bps.go.id/id/statistics-table/2/MTQ5OCMy/luas-panen--produksi--dan-produktivitas-padi-menurut-provinsi.html.

[2] Badan Pusat Statistik, Produksi Padi dan Beras Menurut Kabupaten/Kota di Provinsi Jawa Tengah (Ton), 2023, Accessed: Sep. 17, 2024. https://jateng.bps.go.id/id/statistics-table/2/NDY1IzI=/produksi-padi-dan-beras-menurut-kabupaten-kota-di-provinsi-jawa-tengah.html.

[3] R.A. Purba, Peramalan Curah Hujan Bulanan Di Kota Bandung Menggunakan Metode Extreme Gradient Boosting, Thesis, Universitas Padjadjaran, 2020. https://repository.unpad.ac.id/handle/kandaga/140610160069.

[4] V. Kumar, N. Kedam, K.V. Sharma, K.M. Khedher, A.E. Alluqmani, A Comparison of Machine Learning Models for Predicting Rainfall in Urban Metropolitan Cities, Sustainability 15 (2023), 13724. https://doi.org/10.3390/su151813724.

[5] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, USA, 2016, pp. 785-794. https://doi.org/10.1145/2939672.2939785.

[6] XGBoost Developers, XGBoost Parameters, Accessed: Sep. 20, 2024. https://xgboost.readthedocs.io/en/stable/parameter.html.

[7] N. Alfarisi, Comparison of Convolutional Neural Network and Extreme Gradient Boosting Performance in Predicting Dengue Hemorrhagic Fever Incidence in DKI Jakarta by Considering Climate Factors, Thesis, Universitas Indonesia, Depok, 2023. https://lib.ui.ac.id/detail?id=9999920543724&lokasi=lokal.

[8] M.I. Azhar, W.F. Mahmudy, Prediksi Curah Hujan Menggunakan Metode Adaptive Neuro Fuzzy Inference System (ANFIS), J. Pengembangan Teknol. Inform. Ilmu Komput. 2 (2018), 4932-4939.

[9] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time series analysis: Forecasting and Control, John Wiley & Sons, (2015).

[10] I.G.P. Bahari, Hyperparameter Pptimization of XGBoost: A Case Study on Insurance Claim Prediction, Thesis, Universitas Indonesia, 2020. https://lib.ui.ac.id/detail?id=20509606&lokasi=lokal.

[11] R.P. Dhenanta, I.B. Kholifah, Prediksi Curah Hujan Bulanan Kabupaten Trenggalek Tahun 2022 Dan 2023 Menggunakan Metode Arima, Semin. Nas. Off. Stat. 2022 (2022), 1135-1144. https://doi.org/10.34123/semnasoffstat.v2022i1.1368.

[12] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A Survey on Ensemble Learning, Front. Comput. Sci. 14 (2019), 241-258. https://doi.org/10.1007/s11704-019-8208-z.

[13] A. Fauziah, H. Hermanto, M.A. Sukmarini, Extreme Gradient Boosting Pada Peramalan Pola Curah Hujan Bulanan Kabupaten Banyuwangi, J. Kridatama Sains Teknol. 6 (2024), 430-440. https://doi.org/10.53863/kst.v6i02.1154.

[14] D.N. Gono, H. Napitupulu, Firdaniza, Silver Price Forecasting Using Extreme Gradient Boosting (XGBoost) Method, Mathematics 11 (2023), 3813. https://doi.org/10.3390/math11183813.

[15] A.P. Hadi, Ann Method Implementation To Predict Rainfall in Case of Dengue Fever Anticipation in Malang District, Thesis, Institut Teknologi Sepuluh Nopember, 2018. https://repository.its.ac.id/52608.

[16] A.M. Huda, A. Choiruddin, O. Budiarto, S. Sutikno, Peramalan Data Curah Hujan dengan Seasonal Autoregressive Integrated Moving Average (SARIMA) dengan Deteksi Outlier Sebagai Upaya Optimalisasi Produksi Pertanian di Kabupaten Mojokerto, Seminar Nasional: Kedaulatan Pangan dan Energi, Fakultas Pertanian Universitas Trunojoyo Madura, Jun 2012, Bangkalan, Indonesia. ffhal-01677093. https://hal.science/hal-01677093v1

[17] Ishak, Dampak Curah Hujan Terhadap Produktivitas Tanaman Padi Sawah Pada Masyarakat Petani di Desa

Lambo-Lemo Kecamatan Samaturu Kabupaten Kolaka, J. Penelit. Pendidik. Geogr. 3 (2018), 210-223. https://doi.org/10.36709/jppg.v3i4.4834.

[18] S.F.N. Islam, A. Sholahuddin, A.S. Abdullah, Extreme Gradient Boosting (xgboost) Method in Making Forecasting Application and Analysis of Usd Exchange Rates Against Rupiah, J. Phys.: Conf. Ser. 1722 (2021), 012016. https://doi.org/10.1088/1742-6596/1722/1/012016.

[19] R.S. Kalaksita, Daily Rainfall Forecasting in Semarang City Using Adaptive Neuro Fuzzy Inference System (ANFIS), Thesis, Institut Teknologi Sepuluh Nopember, 2016.

[20] N. Khikmah, Rainfall Forecasting of Kudus District Using VAR (Vector Autoregressive) Method, Thesis Universitas Islam Negeri Walisongo, 2021.

[21] C.D. Usman, A.P. Widodo, K. Adi, R. Gernowo, Rainfall Prediction Model in Semarang City Using Machine Learning, Indones. J. Electr. Eng. Comput. Sci. 30 (2023), 1224. https://doi.org/10.11591/ijeecs.v30.i2.pp1224-1231.

[22] S. Makridakis, S. Wheelwright, R. Hyndman, Forecasting: Methods and Applications, John Wiley & Sons, New York, 1997.

[23] M.A. Mukid, S. Sugito, Model Prediksi Curah Hujan Dengan Pendekatan Regresi Proses Gaussian (Studi Kasus Di Kabupaten Grobogan), Media Stat. 6 (2013), 113-122. https://doi.org/10.14710/medstat.6.2.103-112.

[24] B.N.A. Nareswari, Estimation of Convective Rainfall Based on Weather Radar Measurements Using a Tree-based Machine Learning Approach, Thesis, Universitas Indonesia, 2023.

[25] H.N. Pradani, Forecasting of Dengue Fever Extraordinary Event (Klb) in Malang District Using Extreme Gradient Boosting (Xgboost), Thesis, Institut Teknologi Sepuluh Nopember, 2020.

[26] T. Toharudin, R.E. Caraka, I.R. Pratiwi, Y. Kim, P.U. Gio, et al. Boosting Algorithm to Handle Unbalanced Classification of PM2.5Concentration Levels by Observing Meteorological Parameters in Jakarta-Indonesia Using AdaBoost, XGBoost, CatBoost, and LightGBM, IEEE Access 11 (2023), 35680-35696. https://doi.org/10.1109/access.2023.3265019.

[27] Y. Rombe, The Use of the XGBoost Method for Classification of Obesity Status in Indonesia, Thesis, Universitas Hassanudin, 2021.

[28] R. Amelia, Improving Rainfall Classification Accuracy in The XGBoost Algorithm Through Hyperparameter Tuning, Thesis Universitas Indonesia, 2024.

[29] S.G.A. Rasyad, Comparison of LSTM and GRU Methods in Forecasting Rainfall in Bogor Regency, Thesis, Universitas Padjadjaran, 2023.

[30] N. Sijabat, Application of Transfer Function Models for Monthly Rainfall Prediction in Grobogan Regency, Thesis, Universitas Padjadjaran, 2022.

[31] B. Warsito, Tarno, A. Sugiharto, Prediksi Curah Hujan Sebagai Dasar Perencanaan Pola Tanam Padi dan Palawija Menggunakan Model General Regression Neural Network, J. Litbang Provinsi Jawa Teng. 7 (2009), 1-12.

[32] A. Yasper, Hyperparameter Tuning of the Extreme Gradient Boosting Model for Rainfall Classification: A Case Study in Pontianak City, Thesis, Universitas Indonesia, 2023.