JPPIPA 9(9) (2023)

Jurnal Penelitian Pendidikan IPA

Journal of Research in Science Education

http://jppipa.unram.ac.id/index.php/jppipa/index

Hyperparameters Optimization in XGBoost Model for Rainfall Estimation: A Case Study in Pontianak City

Auriwan Yasper[1], Djati Handoko[1*], Maulana Putra[2], H. K. Aliwarga[3], Mohammad Syamsu Rosid[1]

[1]Department of Physics, Universitas Indonesia, Depok, Indonesia.
[2]Centre for Instrumentation Calibration and Engineering, Indonesia Agency for Meteorology Climatology and Geophysics, Jakarta, Indonesia.
[3]President Commissioner, UMG Idealabs Indonesia, Jakarta, Indonesia.

Received: May 15, 2023
Revised: May 19, 2023
Accepted: September 25, 2023
Published: September 30, 2023

Corresponding Author:
Djati Handoko
djati.handoko@ui.ac.id

DOI: 10.29303/jppipa.v9i9.3890

Abstract: Estimating rainfall accurately is crucial for both the community and various institutions involved in managing water resources and preventing disasters. The XGBoost model has demonstrated its effectiveness in predicting rainfall, but it still requires fine-tuning of hyperparameters to enhance its performance. This study seeks to determine the optimal learning rate for rainfall prediction while keeping the max_depth and n_estimator parameters fixed. The hyperparameter optimization process was carried out using a two-step approach: an initial coarse search using RandomizedSearchCV followed by a more detailed fine-tuning using GridSearchCV. The model's foundation relied on historical rainfall data gathered over three months from the Automated Weather Observed System (AWOS) at the Pontianak Meteorological Station, recorded on an hourly basis. To assess the model's performance, several metrics were employed, including accuracy, precision, recall, F1 score, and ROC-AUC. The model demonstrated promising results, with accuracy, precision, recall, and F1 score all reaching 95%, indicating its ability to effectively predict rainfall. However, the ROC-AUC score was somewhat lower at 62%. After conducting the hyperparameter search, the optimal learning rate determined for the model, utilizing the 2040 dataset, was found to be 0.204.

Keywords: GridSearchCV; Hyperparameter; Rainfall; RandomizedSearchCV; XGBoost.

Introduction

High rainfall greatly affects human life in various sectors including agriculture, transportation, and can also result in natural disasters such as drought, floods, and landslides (Anwar et al., 2021; Ayasha et al., 2020; Palamakumbura et al., 2021; Wang et al., 2022), especially Pontianak City, where the geographical conditions are in the equator, close to the sea, has many watersheds, and has few hills, can affect the intensity of rainfall in this region (Ferijal et al., 2021; Herawati et al., 2015). An increase in average rainfall significantly reduces the level of food insecurity in agricultural activities in an area, especially in rural areas (Tankari, 2020). This has a positive impact on farmers in terms of the fertility of their agricultural land. This is different from urban areas which are more at risk of flooding. Likewise in hilly areas, rainfall greatly influences landslides. According to EM-DAT data, from 1908 to 2022, landslides have caused great damage to society and have claimed 67,169 lives and economic losses of more than 11 billion dollars (Pham et al., 2022). Therefore, weather forecasting is very important to deal with disasters that may occur.

Developing technology with systems that can analyze and estimate rainfall data can provide the right solution to overcome the various negative impacts of rainfall. Machine learning models are one of the alternative technologies that can apply (Feng et al., 2021). There are quite a few machine learning models used to process this data, resulting in accurate estimation values, including Bayesian methods, SVM (Support Vector Machine), ANN (Artificial Neural Network), or variations such as Support Vector Regression (SVR) (Xiang et al., 2018; Zhou et al., 2022).

How to Cite:
Yasper, A., Handoko, D., Putra, M., Aliwarga, H. K., & Rosid, M. S. R. (2023). Hyperparameters Optimization in XGBoost Model for Rainfall Estimation: A Case Study in Pontianak City. *Jurnal Penelitian Pendidikan IPA*, *9*(9), 7113–7121. https://doi.org/10.29303/jppipa.v9i9.3890

There need to be behavior adjustments for the machine learning model to obtain estimation values with good performance. In machine learning, a value can control the model's behavior, namely parameters, and hyperparameters. During the learning process, the model learns patterns in the data and continuously updates the value that controls this behavior, called a parameter. Meanwhile, hyperparameters are used in specific models, and their values are set before the model learns data patterns (Kavzoglu & Teke, 2022). During the learning process, the value of the hyperparameter is used to update the parameters in the model, so the value of the hyperparameter greatly affects the value of the parameters in a machine learning algorithm.

In this study, hyperparameter tuning of the machine learning model will be conducted to estimate rainfall. The selected model is a popular model for estimating historical data sets with optimal performance and highly accurate results, known as XGBoost (Agata & Jaya, 2019; Anand & Kannan, 2022; Azizah et al., 2022; Bansal et al., 2023; Dalal et al., 2022; Hasan et al., 2021; Kaushik & Birok, 2021; Kavzoglu & Teke, 2022; S. Li & Zhang, 2020; X. Li et al., 2022; Ma et al., 2020; Nguyen et al., 2021; Pham et al., 2022; Qin et al., 2021; Shahani et al., 2021; Y. Zhang, 2022). XGBoost is a tree-based model that uses decision trees to make predictions and is very powerful in dealing with large amounts of data, even with decision trees alone it can make very good predictions (Muhsi et al., 2023). Adjusting these hyperparameters aims to find the proper set of values to optimize the XGBoost model. The set of hyperparameter values is intended to optimize the model's performance, reduce the loss function, and obtain the best results with less error (Dalal et al., 2022; Kavzoglu & Teke, 2022; Navas, 2022; Qin et al., 2021).

Method

Dataset

The dataset used is based on physical factors that can affect rainfall intensity, such as air humidity (Wardani et al., 2023), air temperature, atmospheric pressure, wind speed, and wind direction (Ferijal et al., 2021). High humidity can increase the chance of rain because wet air can hold more water vapor, decreasing air temperature can cause condensation of water vapor in the air which eventually forms clouds and rain, low atmospheric pressure can cause air convection and result in the formation of rain clouds, and high wind speeds and directions can carry water vapor over long distances and trigger the formation of rain clouds (Yu et al., 2022).

The source of the dataset in this study is historical weather data obtained from Automated Weather Observed System (AWOS) equipment at the Pontianak Meteorology Station. The data were collected from December 1, 2022, to February 23, 2023, with hourly time resolution. The data consists of five parameters as features, such as air temperature, air humidity, air pressure, wind direction and wind speed, the precipitation data. The numerical rainfall data will be classified into two categories: rain and no rain. The classification is based on the numerical rainfall data; if the precipitation value is greater than zero, it is classified as rain. If the precipitation value is zero, it is categorized as no rain. The full feature can be seen in Table 1.

Table 1. Weather dataset feature table and description.

Feature	Type Data	Description
Air Tmp (C)	Numeric	Air Temperature
Precip 1Hr (mm)	Numeric	Rainfall
QNH (hPa)	Numeric	Air Pressure
RH (%)	Numeric	Relative Humidity
WS (Kt)	Numeric	Wind Speed
Rain Clasification	Category	Rain or No Rain
Mag WD 60 Min (deg)	Numeric	Wind direction
True WD 60 Min (deg)	Numeric	True Wind Direction

The dataset will go through stages as shown in Figure 1. The initial stage in the research flow is understanding what data will be used to build the model, this data is collected in a historical data set. Next, data set preparation is carried out first, in the form of handling missing values, encoding category features, selecting features (Kaushik & Birok, 2021), dividing training data and test data, and standardizing the data set. After that, the data set is trained on the XGBoost model, and hyperparameter tuning is carried out to find the best learning rate. Finally, the model is evaluated with the default model as a comparison of whether the model is optimal or not based on the increase in evaluation metrics.

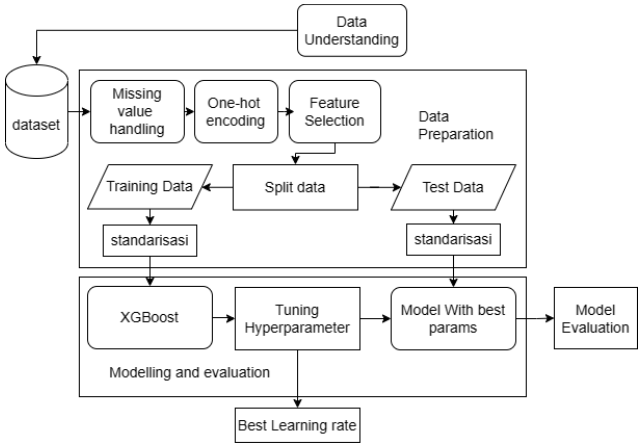

Figure 1. Research flow

Missing Value Handling

The collected data will be preprocessed to make it understandable by the model. In this process, the missing or incomplete data will be evaluated. Several missing data points can affect the training process. A total of 2040 historical data points were collected, while there were 134 missing data points. Several methods can be used to handle missing values, including deleting the missing data points or filling them with the average value of their respective columns. In this study, the missing data points will be deleted.

One-hot Encoding

In machine learning, category data in the form of text or string data types cannot be processed by the model. Therefore, this category of data needs to be converted into a data type that can be recognized and processed by the model, namely into numerical data (Dahouda & Joe, 2021). In this case, one method that can be used is one-hot encoding. The way this method works is by converting category data into a bit vector, with the values 0 and 1. Each bit vector represents one possible value, meaning that the length of this vector is equal to the number of possible values or the number of existing categories (Erjavac et al., 2022).

Feature Selection

There are only four independent or input feature data that will be used, namely, air temperature, air humidity, atmospheric pressure, and wind speed, while the wind direction in the input data is not used because it has a low correlation with rain events and has an important feature score lowest in the model. The dependent features or target features are category data that have been done one-hot encoding, namely the classification of rain and non-rain classification for binary classification.

Split Data

In this research, the existing data set is divided into two parts, namely train data and test data. Train data is used to train the model, while test data is not yet known to the model and is used to test the model that has been designed. Additionally, we will split the dataset into training and testing data using the *train_test_split* function from the library scikit-learn, with 90% of the data reserved for training and the remaining for testing.

XGBoost Model design

XGBoost is a development of gradient boosting with better results (Kapoor & Perrone, 2021; D. Zhang & Gong, 2020). This method needs an objective function to evaluate how much the resulting model fits the training data. This objective function has two essential parts, namely the missing value in training and the regularization value, as seen in equation (1).

$$obj(\vartheta) = L(\vartheta) + \Omega(\vartheta) \quad (1)$$

Where L is the loss function, Ω is the regularization function, and ϑ is the parameter of the model. The loss function in general, can be written as in equation (2).

$$L(\theta) = \sum_{l=1}^{n} l(y_i, p_i) \quad (2)$$

Where y_i is the actual data value and p_i is the predicted value, while n is the number of iterations in the model. It is assumed that the data is $D = \{(x_i, y_i)\}(|D| = n, x_i \in R^m, y_i \in R$, so there are n observations. Each observation has m features and y variables. Then the predicted value to be obtained by the model is $\hat{y}_i$, which is like equation (3).

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{k} f_k(x_i) \quad (3)$$

$$f_k(x_i) = F_{k-1}(x) + \eta_m * h_m(x; \theta_m) \quad (4)$$

Where η_m is the learning rate, $h_m(x; \theta_m)$ is the output value for each leaf in the eXtreme Gradient Boost decision tree. Here f_k is the regression tree and $f_k(x_1)$ is the score provided by the k-tree for the i-th observation. The f_k function is chosen to minimize the value of the objective function like equation (5).

$$L(\phi) = \sum_{l} l(y_{i,} p_{i,}) + \sum_{k} \Omega(f_k) \quad (5)$$

Where l is the loss function and Ω denotes the regularization function as in equation (6).

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||w||^2 \quad (6)$$

Where T is the number of leaves and w is the weight of the leaves. To avoid overfitting and simplify the model, the penalty for T is set by γ, and the penalty for w is set by λ. The thing that differentiates it from the usual gradient boosting is the unique value penalty. The iteration method is used to reduce the objective function. In the t-th iteration, f_t is added to reduce the objective function like equation (7).

$$L^t = \sum_{l=1}^{n} l\left(y_i, p_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (7)$$

Taylor expansion is used to simplify this equation. From Taylor's equation we can derive the formula for reducing loss after the tree is divided from the given nodes (Ma et al., 2020), the formula can be seen in equation (8).

$$L_{split} = \left[\frac{(\sum_{n\in I_R} g_n)^2}{\sum_{n\in I_R} h_n + \lambda}\right] + \frac{1}{2}\left[\frac{(\sum_{n\in I_L} g_n)^2}{\sum_{n\in I_L} h_n + \lambda}\right] + \left[\frac{(\sum_{n\in I} g_n)^2}{\sum_{n\in I} h_n + \lambda}\right] - \gamma \quad (7)$$

Where 1 represents the subset of observations available at the current node. I_L is the subset of observations available at the left node and I_R is the subset of observations available at the right node after the split. The g_n and h_n functions are used to find the best split and are defined as equations (9) and (10).

$$g_n = \partial_{p_n{}^{(j-1)}}\, l\left(y_n, p_n{}^{(j-1)}\right) \quad (9)$$

$$h_n = \partial^2{}_{p_n{}^{(j-1)}}\, l\left(y_n, p_n{}^{(j-1)}\right) \quad (10)$$

The final objective function depends only on the first and second-order gradients of the loss function at each data point and the regularization parameter γ.

Model Implementation

The machine learning model used is XGBoost. Three hyperparameters have values set in the model, namely max_depth, learning_rate, and n_estimator. Before the hyperparameter was adjusted, the value was determined from a previous rainfall estimation study by MT Anwar et al. (Anwar et al., 2021). With a max_depth value of 6, the n_estimator is 100, and learning_rate is 0.3. The value to be adjusted in this study is learning_rate, and the value with the best performance is selected.

Hyperparameter Tuning

Several methods are often used in the tuning process. In this study, the methods used were Random Search Cross Validation and Grid Search Cross-Validation. This method is straightforward: looking for hyperparameters randomly and selecting the parameter with the best performance. A coarse to refined technique will be applied to optimize the hyperparameter search, which optimizes the search space in a specific space. The basic principle is that the value of the first iteration is assigned to the entire search space. After obtaining the best hyperparameter, the search space is focused on a finer space in the zone with the best value to find hyperparameters with better performance than before (Kapoor & Perrone, 2021; Lee et al., 2018).

Model Evaluation

Model performance will be measured by the following evaluation metrics (Canayaz, 2021). In binary classification, positive data is rain data and negative data is no rain data, so TP means the correct prediction for rain data, FP is the wrong classification prediction for rain data, FN is the wrong classification prediction for no rain data, and TN is the classification prediction true for no rain data. From these values, 4 model performances are calculated, namely accuracy, precision, recall, f1-score, and AUC-ROC. Accuracy is the percentage of correct predictions made by the model out of all predictions. Can be written in equation (11).

$$accuracy = \frac{T_p + T_n}{F_p + F_p + T_n + F_n} \quad (11)$$

Precision is the ratio of true positives to the total number of positive predictions made by the model. Calculated in equation (12).

$$precision = \frac{T_p}{T_p + F_p} \quad (12)$$

Recall (sensitivity) is the ratio of true positives to the total number of actual positives in a data set. Calculated as equation (13).

$$recall = \frac{T_p}{T_p + F_n} \quad (13)$$

F1-score is harmonic average of precision and recall. This gives equal importance to precision and recall. This is calculated as equation (14).

$$F1_{score} = 2 * \frac{precision * recall}{precision + recall} \quad (14)$$

AUC-ROC is the area under the receiver operating characteristic curve (ROC curve) is a measure of the trade-off between sensitivity (recall) and specificity. It is used to evaluate binary classification performance.

Result and Discussion

Data Analysis

The research dataset is numerical data, with data distribution as shown in Figure 2. The numerical data pattern in the time series data set has a certain pattern, so the model learns this data pattern in making estimates. From Figure 2, the temperature data varies according to the seasons that occur in Pontianak City. Based on Figure 2, temperature data from 22 to 27°C has a higher intensity than other temperature data. Meanwhile, the rainfall data has unbalanced or imbalanced data, where the zero value dominates the entire data. With unbalanced data, forecasting methods

or regression methods that use numerical data as targets will be difficult, so to estimate, the classification method is the right choice (Depto et al., 2023; Johnson & Khoshgoftaar, 2019; Y. Zhang et al., 2023). Furthermore, for atmospheric pressure, the majority of data has normal values, namely between 1000 and 1014. Then for humidity data, the number of samples with high humidity increases as the air humidity increases, this can be seen clearly in the graph which has increased. While the wind speed is dominated by wind with a low speed of 1-6 kt. For magnetic wind direction and true wind direction, the distribution of the amount of data does not have a particular pattern, because the amount of data and wind direction are evenly distributed at high and low values.

Next, the data is analyzed to find out how significant the correlation is between each piece of data. The correlation score between rainfall and other features can be seen in Table 2, while the correlation matrix between features can be seen in Figure 3. Correlation coefficient values range from -1 to 1, with 1 indicating a perfect positive correlation, -1 indicating a perfectly negative correlation, 0 indicating no correlation, and the closer to 1 or -1 the stronger the correlation (Ramadhan et al., 2022). It can be concluded from the table and the correlation matrix that this imbalanced data has a minimal correlation with the rainfall data.

Table 2. Weather dataset feature table and description.

Feature	Correlation score
Air Tmp (C)	0.01
Precip 1Hr (mm)	1.00
QNH (hPa)	-0.11
RH (%)	0.02
WS (Kt)	0.01

The small value of the correlation coefficient is caused by some zero-value data, where the data is not balanced. The highest percentage of data is in the no rain category, while the percentages in other categories are very small, so the model is more likely to understand no rain data compared to other categories. This shows that real events in the field are indeed the case, so data imbalance will be maintained because some methods for handling unbalanced classes will damage data patterns (Depto et al., 2023; Johnson & Khoshgoftaar, 2019; Y. Zhang et al., 2023).

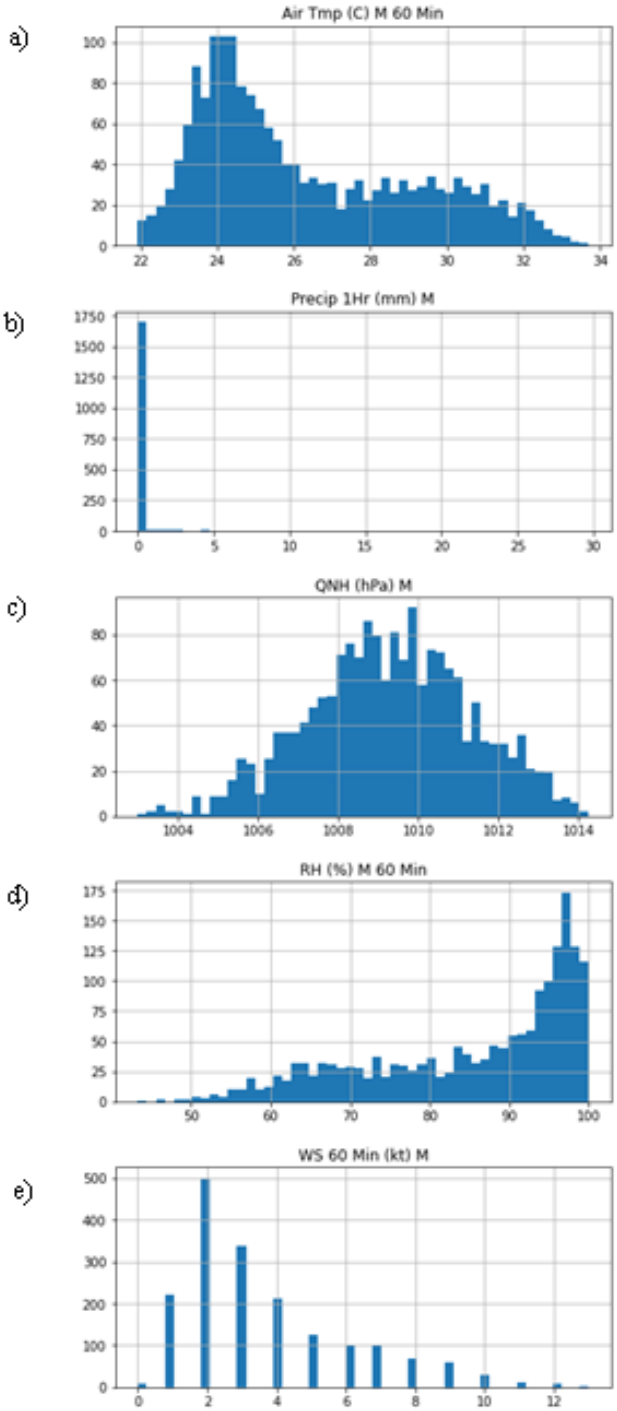

Figure 2. Distribution of research data in the form of air temperature (a), rainfall (b), air pressure (c), air humidity (d), and wind speed (e).

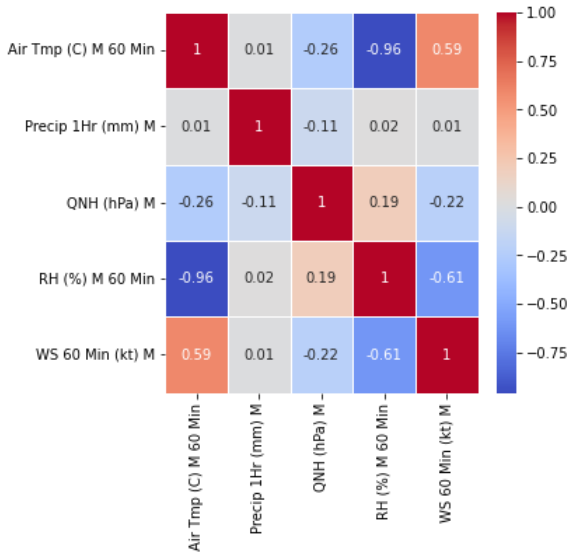

Figure 3. Correlation matrix all dataset

Evaluation metrics

Evaluation metrics measure model performance measures. In the problem of classification, performance is measured by accuracy, F1-score, precision, sensitivity, and ROC_AUC metrics (Canayaz, 2021). In this case, the confusion matrix will be used to determine each value of the evaluation metric (Deng et al., 2016; Jakka & Vakula Rani, 2019). The metric evaluation table on XGBoost before and after tuning can be seen in Table 3, and the tuning result confusion matrix can be seen in Figure 4. Table 3. Comparison table without tuning (max depth = 6, n_estimator = 100, learning_rate = 0.3) and with tuning.

Table 3. Comparison without tuning

Metric	Without Tuning	With Tuning
Akurasi	0.94	0.95
Precision	0.94	0.95
Recall	0.94	0.95
F1-score	0.94	0.95
ROC-AUC	0.61	0.62

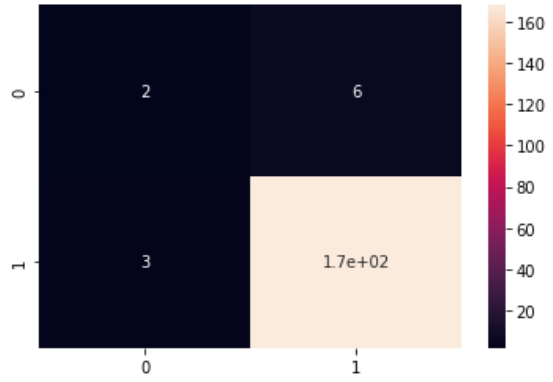

Figure 4. Confusion matrics

Model evaluation on the accuracy, recall, precision, sensitivity, F1, and AUC ROC metrics before tuning and after tuning only increased by 1%. The ability of the model to discriminate between positive and negative data or the AUC ROC curve gives a general picture that the performance of the model is slightly better than the random model because the AUC ROC score is more excellent than 50% and still needs to be improved. The AUC ROC curve can be seen in Figure 5.

Results of Tuning Learning Rate Hyperparameter

This study aimed to optimize the performance of XGBoost in predicting the rainfall category by tuning its learning rate hyperparameter. We employed the random grid coarse to refined technique, which uses a combination of random and grid search to explore the hyperparameter space effectively. We found that the best learning rate was 0.204, and using this value, the resulting model achieved an accuracy of 95% in predicting the rainfall category.

The tuning of hyperparameters is a crucial step in improving model performance (Dalal et al., 2022; Kavzoglu & Teke, 2022). In our study, we focused on the learning rate hyperparameter of XGBoost, a vital hyperparameter affecting model convergence and overfitting. By utilizing the random grid coarse to refined technique, we efficiently searched for the best learning rate hyperparameter. Random search allowed us to explore the hyperparameter space effectively, while grid search enabled us to perform a more detailed search for optimal hyperparameters.

Our findings showed that the optimal learning rate was 0.204, indicating that a higher learning rate can cause overfitting, while a lower learning rate can lead to slower convergence and underfitting. The resulting model achieved a 95% accuracy in predicting the rainfall category, which significantly improved compared to the default model. Our study demonstrated the effectiveness of the random grid coarse-to-fine technique in hyperparameter tuning for XGBoost. The results of the tuning graph can be seen in Figure 6. In the figure, the blue dot is the search point for refined search, while the red dot is a search with rough search, and the best learning rate parameter is 0.204.

In conclusion, our study provides insights into the importance of hyperparameter tuning in machine learning and the effectiveness of the random grid coarse-to-fine technique in achieving optimal results. By combining random and grid search techniques, we efficiently searched for the best learning rate hyperparameter, which significantly improved the model's performance in predicting the rainfall category.

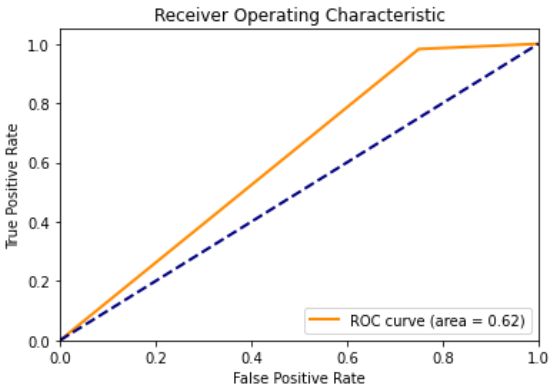

Figure 5. ROC AUC Graphics

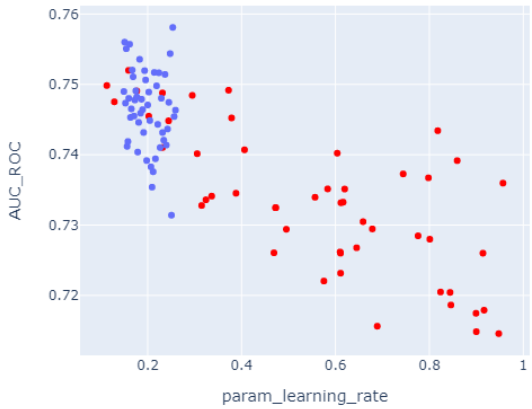

Figure 6. Results of Hyperparameter Optimization

Conclusion

The results of the estimated rainfall show quite good performance on the binary classification problem of 2040 rainfall data, with accuracy, precision, recall, f1-score, and a sensitivity of 95%. In this study, the hyperparameters in the XGBoost model were optimized using the Random Grid Coarse to Fine technique which focused on finding the best learning rate. The result is the best learning rate value of 0.204. The research results show that the learning rate depends on the data set that is trained in making the decision tree so that in different proportions of the data set, this value will produce different accuracy even though the learning rate is the same and a small learning rate is more accurate in making classification predictions than with a large learning rate, however, model learning will take relatively longer. In future research, it is hoped that there will be tuning for all existing hyperparameters with a larger sample, and unbalanced data can be balanced with oversampling or under-sampling techniques (Y. Zhang et al., 2023).

Acknowledgments
We would like to thank all those who have contributed to the success of this research, the research advisors from the physics department at the University of Indonesia. We thank UMG Idealabs that supporting this research and the provider of the necessary data sources from BMKG. In addition, we appreciate constructive input from reviewers and editors who have helped improve the quality of this manuscript.

Author Contributions
Conceptualization: Djati, Yasper; supervision: Djati, Maulana; data curation: Maulana; funding acquisition: Kiwi, Djati, Maulana; methodology: Djati, Syamsu, Maulana; visualization: Yasper; writing-original draft: Yasper; writing–review & editing: Djati & Maulana

Funding
This research was independently funded by researchers.

Conflicts of Interest
No Conflicts of interest.

References

Agata, R., & Jaya, I. G. N. M. (2019). A comparison of extreme gradient boosting, SARIMA, exponential smoothing, and neural network models for forecasting rainfall data. *Journal of Physics: Conference Series*, *1397*(1). https://doi.org/10.1088/1742-6596/1397/1/012073

Anand, A., & Kannan, S. R. (2022). Rain/no-rain classification from combined radar- Radiometer data using machine learning. *Remote Sensing Applications: Society and Environment*, *25*. https://doi.org/10.1016/j.rsase.2021.100682

Anwar, M. T., Winarno, E., Hadikurniawati, W., & Novita, M. (2021). Rainfall prediction using Extreme Gradient Boosting. *Journal of Physics: Conference Series*, *1869*(1). https://doi.org/10.1088/1742-6596/1869/1/012078

Ayasha, N., Ryan, M., & Fadlan, A. (2020). Study of atmosphere dynamics in the event of very heavy rain causing flood in Supadio International Airport Pontianak using WRF-ARW Model and Himawari-8 Satellite Imagery (Case study: November 11, 2017). *IOP Conference Series: Earth and Environmental Science*, *561*(1). https://doi.org/10.1088/1755-1315/561/1/012032

Azizah, M., Yanuar, A., & Firdayani, F. (2022). Dimensional Reduction of QSAR Features Using a Machine Learning Approach on the SARS-Cov-2 Inhibitor Database. *Jurnal Penelitian Pendidikan IPA*, *8*(6), 3095–3101. https://doi.org/10.29303/jppipa.v8i6.2432

Bansal, N., Singh, D., & Kumar, M. (2023). Computation of energy across the type-C piano key weir using

gene expression programming and extreme gradient boosting (XGBoost) algorithm. *Energy Reports*, *9*, 310–321. https://doi.org/10.1016/j.egyr.2023.04.003

Canayaz, M. (2021). C+EffxNet: A novel hybrid approach for COVID-19 diagnosis on CT images based on CBAM and EfficientNet. *Chaos, Solitons and Fractals*, *151*. https://doi.org/10.1016/j.chaos.2021.111310

Dahouda, M. K., & Joe, I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access*, *9*, 114381–114391. https://doi.org/10.1109/ACCESS.2021.3104357

Dalal, S., Seth, B., Radulescu, M., Secara, C., & Tolea, C. (2022). Predicting Fraud in Financial Payment Services through Optimized Hyper-Parameter-Tuned XGBoost Model. *Mathematics*, *10*(24). https://doi.org/10.3390/math10244679

Deng, X., Liu, Q., Deng, Y., & Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, *340–341*, 250–261. https://doi.org/10.1016/j.ins.2016.01.033

Depto, D. S., Rizvee, M. M., Rahman, A., Zunair, H., Rahman, M. S., & Mahdy, M. R. C. (2023). Quantifying imbalanced classification methods for leukemia detection. *Computers in Biology and Medicine*, *152*. https://doi.org/10.1016/j.compbiomed.2022.106372

Erjavac, I., Kalafatovic, D., & Mauša, G. (2022). Coupled encoding methods for antimicrobial peptide prediction: How sensitive is a highly accurate model? *Artificial Intelligence in the Life Sciences*, *2*, 100034. https://doi.org/10.1016/j.ailsci.2022.100034

Feng, Y., Duan, Q., Chen, X., Yakkali, S. S., & Wang, J. (2021). Space cooling energy usage prediction based on utility data for residential buildings using machine learning methods. *Applied Energy*, *291*. https://doi.org/10.1016/j.apenergy.2021.116814

Ferijal, T., Batelaan, O., & Shanafield, M. (2021). Spatial and temporal variation in rainy season droughts in the Indonesian Maritime Continent. *Journal of Hydrology*, *603*. https://doi.org/10.1016/j.jhydrol.2021.126999

Hasan, M. K., Jawad, M. T., Dutta, A., Awal, M. A., Islam, M. A., Masud, M., & Al-Amri, J. F. (2021). Associating Measles Vaccine Uptake Classification and its Underlying Factors Using an Ensemble of Machine Learning Models. *IEEE Access*, *9*, 119613–119628. https://doi.org/10.1109/ACCESS.2021.3108551

Herawati, H., Suripin, & Suharyanto. (2015). Impact of climate change on streamflow in the tropical Lowland of Kapuas River, West Borneo, Indonesia. *Procedia Engineering*, *125*, 185–192. https://doi.org/10.1016/j.proeng.2015.11.027

Jakka, A., & Vakula Rani, J. (2019). Performance evaluation of machine learning models for diabetes prediction. *International Journal of Innovative Technology and Exploring Engineering*, *8*(11), 1976–1980. https://doi.org/10.35940/ijitee.K2155.0981119

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0192-5

Kapoor, S., & Perrone, V. (2021). *A Simple and Fast Baseline for Tuning Large XGBoost Models*. http://arxiv.org/abs/2111.06924

Kaushik, S., & Birok, R. (2021). Heart Failure prediction using Xgboost algorithm and feature selection using feature permutation. *2021 4th International Conference on Electrical, Computer and Communication Technologies (ICECCT),* 1-6. https://doi.org/10.1109/ICECCT52121.2021.9616626

Kavzoglu, T., & Teke, A. (2022). Advanced hyperparameter optimization for improved spatial prediction of shallow landslides using extreme gradient boosting (XGBoost). *Bulletin of Engineering Geology and the Environment*, *81*(5). https://doi.org/10.1007/s10064-022-02708-w

Lee, H. H., Tang, Y., Bao, S., Abramson, R. G., Huo, Y., & Landman, B. A. (2018). Rap-Net: Coarse-To-Fine Multi-Organ Segmentation With Single Random Anatomical Prior. *In 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI),* 1491-1494. Retrieved from https://github.com/MASILab/coarse_to_fine_prior_seg.

Li, S., & Zhang, X. (2020). Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm. *Neural Computing and Applications*, *32*(7), 1971–1979. https://doi.org/10.1007/s00521-019-04378-4

Li, X., Shan, G., & Shek, C. H. (2022). Machine learning prediction of magnetic properties of Fe-based metallic glasses considering glass forming ability. *Journal of Materials Science and Technology*, *103*, 113–120. https://doi.org/10.1016/j.jmst.2021.05.076

Ma, Z., Chang, H., Sun, Z., Liu, F., Li, W., Zhao, D., & Chen, C. (2020). Very Short-Term Renewable Energy Power Prediction Using XGBoost Optimized by TPE Algorithm. *2020 4th International Conference on HVDC, HVDC 2020*,

1236–1241. https://doi.org/10.1109/HVDC50696.2020.9292870

Muhsi, M., Suprapto, S., & Rofiuddin, R. (2023). Node Selection Method for Split Attribute in C4.5 Algorithm Using the Coefficient of Determination Values for Multivariate Data Set. *Jurnal Penelitian Pendidikan IPA*, *9*(7), 5574–5583. https://doi.org/10.29303/jppipa.v9i7.4031

Navas, J. (2022, February 8). *What is hyperparameter tuning Anyscale*. Anyscale. Retrieved from https://www.anyscale.com/blog/what-is-hyperparameter-tuning

Nguyen, H., Vu, T., Vo, T. P., & Thai, H. T. (2021). Efficient machine learning models for prediction of concrete strengths. *Construction and Building Materials*, *266*. https://doi.org/10.1016/j.conbuildmat.2020.120950

Palamakumbura, R., Finlayson, A., Ciurean, R., Nedumpallile-Vasu, N., Freeborough, K., & Dashwood, C. (2021). Geological and geomorphological influences on a recent debris flow event in the Ice-scoured Mountain Quaternary domain, western Scotland. *Proceedings of the Geologists' Association*, *132*(4), 456–468. https://doi.org/10.1016/j.pgeola.2021.05.002

Pham, K., Kim, D., Le, C. V., & Choi, H. (2022). Dual tree-boosting framework for estimating warning levels of rainfall-induced landslides. *Landslides*, *19*(9), 2249–2262. https://doi.org/10.1007/s10346-022-01894-8

Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., & Liu, P. (2021). XGBoost optimized by adaptive particle swarm optimization for credit scoring. *Mathematical Problems in Engineering*, *2021*. https://doi.org/10.1155/2021/6655510

Ramadhan, R., Marzuki, M., Yusnaini, H., Ningsih, A. P., Hashiguchi, H., Shimomai, T., Vonnisa, M., Ulfah, S., Suryanto, W., & Sholihun, S. (2022). Ground Validation of GPM IMERG-F Precipitation Products with the Point Rain Gauge Records on the Extreme Rainfall Over a Mountainous Area of Sumatra Island. *Jurnal Penelitian Pendidikan IPA*, *8*(1), 163–170. https://doi.org/10.29303/jppipa.v8i1.1155

Shahani, N. M., Kamran, M., Zheng, X., Liu, C., & Guo, X. (2021). Application of gradient boosting machine learning algorithms to predict uniaxial compressive strength of soft sedimentary rocks at Thar coalfield. *Advances in Civil Engineering*, *2021*, 1-19. https://doi.org/10.1155/2021/2565488

Tankari, M. R. (2020). Rainfall variability and farm households' food insecurity in Burkina Faso: nonfarm activities as a coping strategy. *Food Security*, *12*, 567–578. https://doi.org/10.1007/s12571-019-01002-0/Published

Wang, X., Xia, J., Zhou, M., Deng, S., & Li, Q. (2022). Assessment of the joint impact of rainfall and river water level on urban flooding in Wuhan City, China. *Journal of Hydrology*, *613*. https://doi.org/10.1016/j.jhydrol.2022.128419

Wardani, A., Akbar, A. J., Handayani, L., & Lubis, A. M. (2023). Correlation Among Rainfall, Humidity, and The El Niño-Southern Oscillation (ENSO) Phenomena in Bengkulu City During the Period from 1985-2020. *Jurnal Penelitian Pendidikan IPA*, *9*(4), 1664–1671. https://doi.org/10.29303/jppipa.v9i4.2971

Xiang, Y., Gou, L., He, L., Xia, S., & Wang, W. (2018). A SVR–ANN combined model based on ensemble EMD for rainfall prediction. *Applied Soft Computing Journal*, *73*, 874–883. https://doi.org/10.1016/j.asoc.2018.09.018

Yu, Y., Zhu, J., Gao, T., Liu, L., Yu, F., Zhang, J., & Wei, X. (2022). Evaluating the influential variables on rainfall interception at different rainfall amount levels in temperate forests. *Journal of Hydrology*, *615*. https://doi.org/10.1016/j.jhydrol.2022.128572

Zhang, D., & Gong, Y. (2020). The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure. *IEEE Access*, *8*, 220990-221003. https://doi.org/10.1109/ACCESS.2020.3042848

Zhang, Y. (2022). Classification of Quasars, Galaxies, and Stars by Using XGBoost in SDSS-DR16. *Proceedings - 2022 International Conference on Machine Learning and Knowledge Engineering, MLKE 2022*, 266–272. https://doi.org/10.1109/MLKE55170.2022.00058

Zhang, Y., Liu, Y., Wang, Y., & Yang, J. (2023). An ensemble oversampling method for imbalanced classification with prior knowledge via generative adversarial network. *Chemometrics and Intelligent Laboratory Systems*, *235*. https://doi.org/10.1016/j.chemolab.2023.104775

Zhou, M., Wang, L., Wu, H., Li, Q., Li, M., Zhang, Z., Zhao, Y., Lu, Z., & Zou, Z. (2022). Machine learning modeling and prediction of peanut protein content based on spectral images and stoichiometry. *LWT*, *169*. https://doi.org/10.1016/j.lwt.2022.114015