JOURNAL OF ENGINEERING TECHNOLOGY AND SCIENCE INTEGRATION

journal homepage: https://nordeoscholar.com/jetsi

E-ISSN: 3164-5208

The Use of Machine Learning and Ensemble Stacking for Classifying Rainfall Intensity in Central Jakarta in 2025

Damar Januar Jorgie Marseno[1], Maysya Amelia Putri[2], Muhammad Favian Gustaf Ahnaf[3*], Muhammad Jawahir Asyrof[4], Kania Laksono[5], Faridhatun Nikmah[6]

[1,2,3,4,5,6] Department Department of Data Science, Faculty of Science and Technology, Raden Mas Said State Islamic University Surakarta, Klaten, Central Java, Indonesia

ARTICLE INFO

Article history:
Received 25 June 2026
Revised 05 July 2026
Accepted 06 July 2026
Published online 30 June 2026

Keywords:
Ensemble Stacking; LassoCV; Machine Learning Algorithm; Rainfall Intensity Classification; Time Series Cross Validation.

Editor: Fandi Achmad
Publisher's note: The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations, while the author(s) bear sole responsibility for the accuracy of content and any legal implications

ABSTRACT

High rainfall intensity in Central Jakarta can trigger inundation and flooding that disrupt infrastructure, mobility, and public activities. This condition highlights the need for a reliable rainfall classification model. This study develops a next-day rainfall classification approach using a Stacking Ensemble that combines Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost as base models, with LassoCV as the meta-model. The dataset was obtained from BMKG meteorological records for Central Jakarta in 2025. The research process follows the CRISP-DM framework, covering data understanding, data preparation, feature engineering, modeling, and evaluation. From 365 daily observations, the feature engineering stage produced 34 derived features. The dataset was then split into 80% training data and 20% testing data according to chronological order, while model validation was conducted using 5-fold Time Series Cross Validation. The evaluation results indicate that Stacking Lasso achieved the best performance, with an F1-Score of 0.774 and a recall of 0.878. During the weighting process, LassoCV retained only LightGBM as the active contributor, with a coefficient of 0.2671. SHAP interpretation showed that average relative humidity was the most influential variable in predicting rainfall. These findings suggest that the Stacking Ensemble approach has potential as a foundation for developing an urban hydrometeorological disaster mitigation support system.

1. Introduction

Weather directly affects many human activities, including agriculture, transportation, health, and urban area management (Koetse & Rietveld, 2009). One of the most important weather elements to observe is rainfall intensity, particularly in densely populated areas such as Central Jakarta. This area holds a strategic position as a center of government and economic activities, yet it is simultaneously vulnerable to inundation and flooding when rainfall increases (BMKG, 2025). Rapid and often unpredictable changes in rainfall make information on rainfall probability important for decision-making processes (Wilks, 2011).

BMKG, as the official meteorological agency, provides daily meteorological data that can be utilized to support weather forecasting, disaster mitigation, transportation safety, and urban planning. Therefore, a data-driven rainfall classification system is needed to help interpret weather patterns in a more measurable and consistent manner.

Weather data routinely collected by BMKG forms a rich historical archive, yet the patterns within it are not always discernible through simple observation alone (BMKG, 2025). To process such data, data mining can be used as an approach to extract knowledge from datasets through statistical and computational techniques (Han et al., 2022; Ian et al., 2017). One of the key tasks in data mining is

*Corresponding Autor: muhammadfavian18@gmail.com (Ahnaf, M. F. G.)

classification, which involves learning patterns from historical data to determine the category of new data (Ian et al., 2017). In this study, classification is directed toward predicting whether rain will occur the following day. The target variable is constructed in binary form, where rain equals 1 and no rain equals 0. Machine learning is selected because this approach is capable of handling complex, non-linear relationships among weather variables involving numerous parameters simultaneously (Safia et al., 2023).

A number of studies on weather classification or prediction still rely heavily on a single algorithm, so the model's ability to capture dynamic meteorological patterns is not always optimal (Ayu Syaharani et al., 2025; Fauziah et al., 2024; Febriansyah Istianto et al., 2024; Hafid et al., 2025; Kurniawan et al., 2024). Single models may perform well under certain patterns but can be less stable when dealing with data characterized by temporal dependencies, noise, and inter-variable variation. On this basis, this study employs ensemble learning so that multiple models can complement each other during the prediction process (Dietterich, 2000; Polikar, 2006). The algorithms used include Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost. All five models are placed as base models within the Stacking Ensemble scheme, and their prediction outputs are subsequently processed by a meta-model. This strategy is expected to produce more stable classifications compared to relying solely on a single model (Sagi & Rokach, 2018). The entire research process is structured following CRISP-DM to ensure that the data analysis stages are systematic, traceable, and replicable (Dwiyanti & Prianto, 2023).

The research data source consists of historical weather data from BMKG monitoring stations in Central Jakarta for the year 2025. The dataset includes daily meteorological parameters such as temperature, air humidity, rainfall, sunshine duration, wind speed, and wind direction (BMKG, 2025). The target label is constructed to indicate rainfall conditions on the following day, so the model does not merely read the conditions of the same day but is directed to perform classification one day ahead. Central Jakarta is selected because this area has dense urban activity and requires faster and more accurate rainfall information to support preparedness. Furthermore, the availability of structured BMKG data provides a suitable foundation for developing a machine learning-based classification model.

Based on the foregoing description, the research problem formulation is how to build a next-day rainfall classification model for the Central Jakarta area by combining Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost within a Stacking Ensemble framework based on Lasso Cross Validation (LassoCV), using BMKG meteorological data from 2025. The objective of this study is to design, implement, and evaluate a rainfall intensity classification system that leverages the strengths of multiple machine learning algorithms simultaneously. By employing LassoCV as the meta-model, this study seeks to obtain predictions that are stronger, more measurable, and more competitive compared to single-model approaches for the Central Jakarta weather data case (Sagi & Rokach, 2018).

This study is important because rainfall is a phenomenon influenced by many interrelated meteorological variables that change over time (Wilks, 2011). In urban areas such as Central Jakarta, heavy rainfall can have impacts on safety, transportation, drainage infrastructure, and economic activities. A single machine learning algorithm may not always be capable of capturing the full range of atmospheric pattern variations. Therefore, Stacking Ensemble is used to combine the strengths of multiple models so that predictions become more stable against data changes (Dietterich, 2000; Polikar, 2006; Sagi & Rokach, 2018). Furthermore, the use of Time Series Cross Validation through Time Series Split is important because weather data has a temporal order. Models must be trained on past data and tested on subsequent data to ensure that the evaluation process does not suffer from data leakage (Bergmeir et al., 2018).

With respect to the novelty of this study, three aspects distinguish it from prior machine learning research on rainfall in Indonesia. First, most existing rainfall classification or prediction studies in Jakarta continue to rely on a single algorithm evaluated in isolation, such as the multiclass rain classification study by Pringandana and Kusnawi (2025), which compared XGBoost and LightGBM independently without combining them in a unified ensemble scheme. Second, while stacking ensembles have been applied to rainfall classification in Indonesia — for example by Hermansyah et al. (2025), who used vertical atmospheric profile data from radiosondes with Random Forest, XGBoost, LightGBM, and SVM as base models and HistGradientBoosting as a nonlinear meta-learner — this study instead uses routine daily surface meteorological data collected by BMKG, combined with a linear, sparsity-inducing LassoCV meta-model. This choice allows the contribution of each base model to be explicitly interpreted through its regression coefficient rather than being treated as a black box, while still relying on data routinely available at BMKG stations. Third, building on the Time Series Cross Validation described earlier, this study specifically targets next-day (H+1) rainfall classification rather than same-day classification, combined with strict chronological validation, which is critical for the operational requirements of early warning systems in flood-prone urban areas such as Central Jakarta. The combination of a stacking ensemble, an interpretable linear meta-model, an H+1 prediction horizon, and locally collected BMKG 2025 data has not been previously reported, and constitutes the primary contribution of this study.

The benefits of this study can be seen from both practical and academic perspectives. Practically, the next-day rainfall classification results can support the government, communities, and relevant parties in preparing anticipatory measures against the risks of inundation, transportation disruption, and drainage management. The developed model can also serve as an initial foundation for web-based or mobile applications to make prediction information more

easily accessible. From an academic perspective, this study contributes to the body of knowledge on the application of machine learning and ensemble learning to Indonesian meteorological data. It also provides an example of the use of time-series-based validation in weather classification, particularly through Stacking Ensemble (Dietterich, 2000; Polikar, 2006; Sagi & Rokach, 2018).

2. Research Methods

This study employs a quantitative approach focused on machine learning-based classification to predict rainfall conditions on the following day in Central Jakarta. The research workflow is structured based on CRISP-DM, encompassing problem understanding, data understanding, data preparation, modeling, evaluation, and deployment planning (Dwiyanti & Prianto, 2023). This framework is used because each stage is clearly defined, allowing the data processing and model development processes to be carried out systematically.

The data used in this study is secondary data obtained from the official BMKG website, consisting of daily meteorological records for Central Jakarta in 2025. The data is analyzed quantitatively to identify patterns associated with rainfall classification using machine learning. The initial dataset is organized in columns A through K and includes daily temperature, humidity, rainfall, sunshine duration, wind speed, and wind direction. This information forms the basis for the data cleaning, feature construction, model training, and research result discussion processes (BMKG, 2025).

The data understanding stage was conducted by reviewing the initial characteristics of the BMKG Central Jakarta 2025 dataset. The raw data contains 365 daily observations with 11 main columns, namely TX, TN, TAVG, RH_AVG, RR, SS, FF_X, FF_AVG, and DDD_CAR, representing temperature, humidity, rainfall, sunshine duration, and wind-related elements (BMKG, 2025). Initial exploration revealed several missing values, particularly in rainfall and wind speed columns, so the data needed to be cleaned before use. In addition, the label distribution indicated an imbalance between the rain and no-rain classes. This condition is an important consideration in selecting modeling and evaluation strategies to ensure that classification results do not disproportionately favor the dominant class.

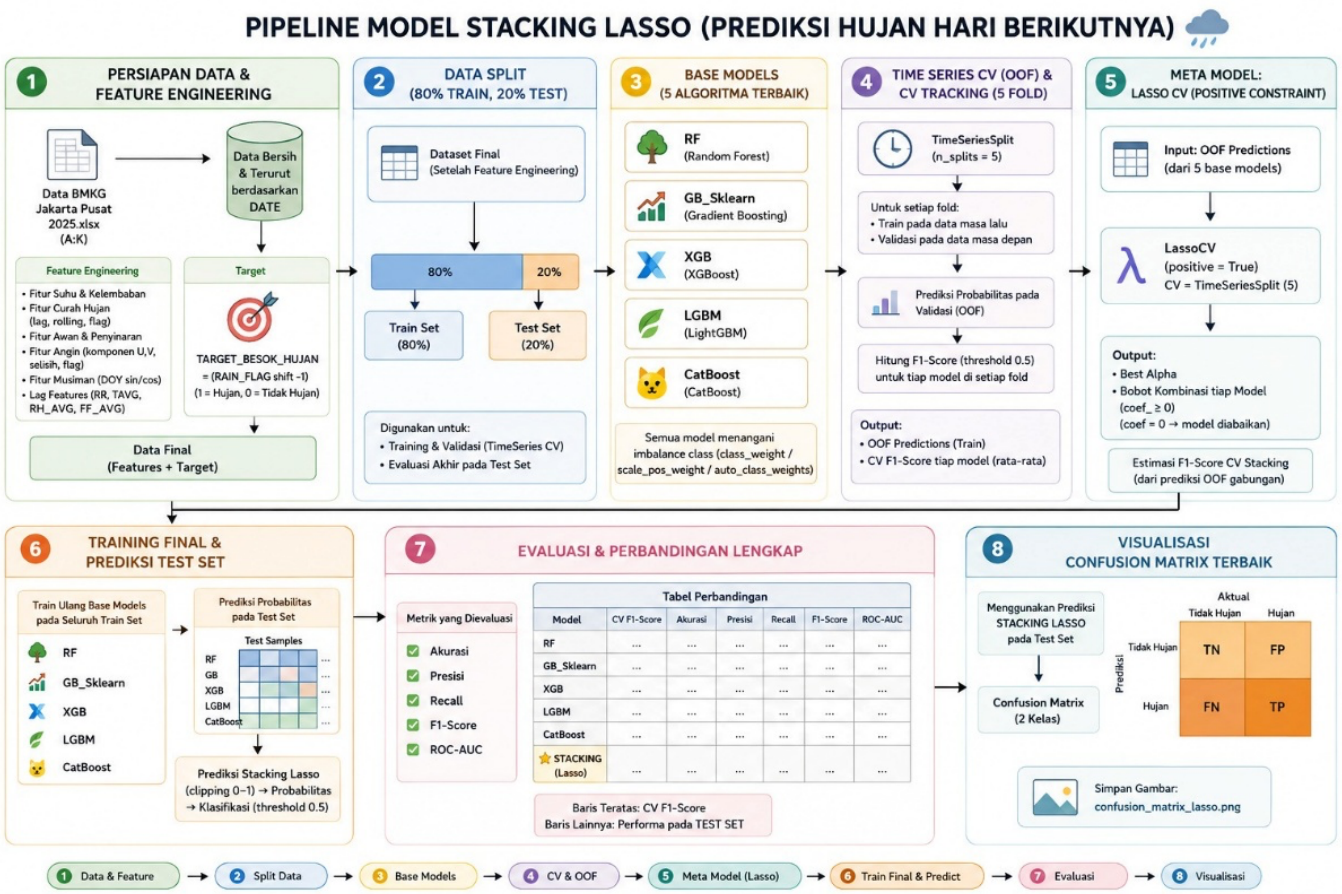

Figure 1. Stacking Lasso Model Pipeline

Prior to modeling, the BMKG data was processed through feature engineering. The features constructed include temperature and humidity derivatives, lag features for rainfall, rolling statistics, specific condition indicators, sunshine duration features, and wind U and V components. Seasonal elements were also incorporated using sine and cosine transformations of the Day of Year (DOY) to represent annual patterns continuously. Key variables such as RR, TAVG, RH_AVG, and FF_AVG were created in lag form so that the model could capture the influence of conditions from several days prior. This type of time-series-based feature approach is widely used because it helps models capture temporal and seasonal patterns in rainfall data (Cawood & van Zyl, 2021). The TARGET_BESOK_HUJAN variable was created by shifting RAIN_FLAG by -1, so that the label indicates whether the following day will have rain or not. After the features were constructed, the data was split into 80% training and 20% testing based on chronological order to preserve the temporal structure and avoid data leakage (Cawood & van Zyl, 2021).

The feature engineering process yielded a total of 34 derived features grouped into seven categories. (1) Temperature derivative features, namely TAVG_minus_TN and TX_minus_TAVG, capturing the daily thermal range. (2) Temperature-humidity interaction features, such as TEMP_RH_PRODUCT, combining temperature and humidity into a single indicator. (3) Lag features, shifting RR, TAVG, RH_AVG, and FF_AVG by one and two days (e.g., RR_lag_1, RR_lag_2, TAVG_lag_1, FF_AVG_lag_2), enabling the model to learn from prior atmospheric conditions. (4) Rolling statistics, such as RR_3d_sum and moving averages of temperature and humidity over several days, summarizing short-term accumulation and trends. (5) Special condition indicators, binary flags for days with extreme temperature, extreme humidity, and calm wind. (6) Sunshine duration features, including raw SS values along with their lag and rolling variants. (7) Wind-related features, comprising U and V components of wind speed and direction (FF_X_U, FF_X_V), along with seasonal encodings using sine and cosine transformations of Day of Year (DOY_sin, DOY_cos) to continuously represent the annual rainfall cycle. These 34 features were derived from eight raw BMKG variables (TX, TN, TAVG, RH_AVG, RR, SS, FF_X, FF_AVG, and DDD_CAR) to help the model capture temporal dependencies, seasonal patterns, and inter-variable interactions characteristic of tropical rainfall (Cawood & van Zyl, 2021).

Regarding these features, no explicit feature selection procedure at the feature level — such as correlation filtering, Recursive Feature Elimination, or variance thresholding — was applied prior to model training. All 34 engineered features, together with the original meteorological variables, were retained and passed directly to the five base learners, because tree-based ensemble algorithms such as Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost are capable of performing implicit feature selection through split-based importance mechanisms within the model (Breiman, 2001; Chen & Guestrin, 2016). Feature importance was therefore examined post hoc using SHAP on the LightGBM model, while model-level selection was performed by LassoCV at the meta-learner stage, where L1 regularization suppresses the contribution of base models whose out-of-fold predictions do not provide unique information beyond what has already been contributed by other base models (Sagi & Rokach, 2018; Song & Zhang, 2024).

Five ensemble learning algorithms are used as base models in the Stacking Ensemble (Sagi & Rokach, 2018). The models are as follows:

- Random Forest, a method that builds multiple decision trees from randomly selected data samples using bootstrap techniques. At each node split, only a subset of features is considered, resulting in more diverse trees. The final result is determined through majority voting. This mechanism helps reduce variance and the risk of overfitting. Due to its ability to handle non-linear relationships and a relatively large number of features, Random Forest is well-suited for meteorological data (Breiman, 2001).
- Gradient Boosting Machine (GBM). Unlike Random Forest, which builds trees in parallel, GBM constructs trees sequentially. Each new tree is directed to correct the errors of the previous tree by minimizing the loss function. The update process is carried out through the gradient descent principle, allowing the model to iteratively adjust its predictions. In this way, GBM is capable of learning complex non-linear patterns in weather data (Friedman, 2001).
- Extreme Gradient Boosting (XGBoost), developed as a boosting version with higher computational efficiency. XGBoost uses L1 and L2 regularization to suppress overfitting, and supports parallel computation for faster training on larger datasets. This algorithm also has the ability to handle missing values internally and performs more controlled tree pruning. These characteristics make XGBoost relevant for rainfall classification involving many meteorological variables (Chen & Guestrin, 2016).
- Light Gradient Boosting Machine (LightGBM), introduced by Ke et al. (2017). LightGBM is known for its efficiency in terms of training time and memory usage. While some methods grow trees level-wise, LightGBM uses a leaf-wise strategy, expanding the leaf that provides the greatest loss reduction. This algorithm also employs Gradient-based One-Side Sampling (GOSS), allowing the learning process to focus more on data with large gradients. These advantages make LightGBM suitable for identifying varied weather data patterns without requiring heavy computation (Ke et al., 2017).
- Categorical Boosting (CatBoost), developed by Yandex. CatBoost is designed to handle categorical features more conveniently and reduce the need for complex preprocessing. This algorithm uses Ordered Boosting to suppress prediction shift that commonly occurs in

conventional boosting methods. With this approach, CatBoost tends to be stable when facing noise and can deliver good performance even without highly complex hyperparameter tuning (Prokhorenkova et al., 2018).

After all five base models are prepared, all models are combined in the Stacking Ensemble scheme (Sagi & Rokach, 2018). The first stage involves generating Out-of-Fold (OOF) predictions using Time Series Split with 5 folds. In each fold, the model is only trained on past data and validated on subsequent data. This approach preserves temporal order and reduces the risk of data leakage. OOF predictions from all five base models are then used as input for LassoCV as the meta-model. LassoCV is applied with positive coefficient constraints and 5-fold Time Series Split validation. L1 regularization in LassoCV helps select models that truly contribute by assigning small or zero weights to less relevant models. In the final stage, base models are retrained using all training data, probabilistic predictions on the testing data are combined by LassoCV, and the results are converted into rain or no-rain classes using a threshold of 0.5.

Regarding missing values, the raw BMKG dataset contained data gaps primarily in the RR (rainfall) and FF_AVG (average wind speed) columns. Rows with unrecorded rainfall values were treated as zero rainfall, following common practice for BMKG station data, while missing wind speed values were imputed using linear interpolation based on the chronological order of observations to preserve the time series structure (Cawood & van Zyl, 2021). Rows that still contained missing values after lag and rolling window feature construction, particularly at the beginning of the series, were dropped, reducing the dataset from 365 to 361 usable rows. Regarding class imbalance, this study did not apply resampling techniques such as SMOTE or random undersampling. Instead, the imbalance between rain and no-rain classes was addressed implicitly through the choice of evaluation metrics — specifically recall and F1-Score, which are more sensitive to minority class detection than accuracy alone — and through the non-negative coefficient constraint on LassoCV, which tends to select base models with the most discriminative probability estimates. This approach differs from related studies such as Hermansyah et al. (2025) and Pringandana and Kusnawi (2025), which applied oversampling to handle imbalance in similar rainfall classification tasks; this methodological difference is acknowledged as a limitation and is further discussed in the Discussion section.

Beyond missing values and class imbalance, hyperparameter tuning for the five base models was conducted through manual grid search guided by the F1-Score from 5-fold Time Series Split cross-validation, rather than automated search procedures such as GridSearchCV or RandomizedSearchCV. The primary hyperparameters adjusted include the number of estimators, maximum tree depth, and learning rate for boosting-based models (Gradient Boosting, XGBoost, LightGBM, and CatBoost), and the number of estimators and maximum depth for Random Forest. For the LassoCV meta-model, the regularization strength (alpha) was selected automatically from a predefined range using its built-in cross-validated search, yielding an optimal alpha of 0.00010. Finally, the probability threshold of 0.5 used to convert Stacking Lasso output into binary rain/no-rain classes was adopted as a neutral default value, not an optimized one. This threshold treats false positives and false negatives symmetrically, whereas in the context of flood early warning, the cost of missing an actual rain event is generally higher than the cost of a false alarm. Because systematic threshold optimization based on the precision-recall curve was not performed in this study, the reported precision and recall values should be understood as model performance under a neutral decision rule; lowering the threshold would likely further increase recall at the expense of precision, as discussed further in the Discussion section.

Model evaluation was conducted using accuracy, precision, recall, F1-Score, and ROC-AUC on the 20% testing data. These five metrics are used so that model performance can be assessed from multiple dimensions, not just from the number of correct predictions. Recall is important for assessing the model's ability to detect rainfall events, while precision indicates how accurate the given rainfall predictions are. F1-Score is used to balance precision and recall, while ROC-AUC helps assess the model's ability to distinguish between the two classes. In addition to numerical metrics, a confusion matrix is also created to show the number of correct and incorrect predictions for both the rain and no-rain classes. The use of diverse metrics is important in the evaluation of meteorological and time series classification data (Tyralis et al., 2021).

At the deployment stage, this study has not yet reached the development of an application directly used by the public. Nevertheless, the tested model can be further developed into a web-based or mobile system. Such a system could be connected to real-time BMKG data, allowing the rainfall classification results to be utilized by the community, local government, and other parties with an interest in weather risk mitigation.

3. Result

The initial stage of the research results began with data cleaning and feature engineering on the BMKG Central Jakarta 2025 dataset. The initial data contained daily parameters including maximum temperature, minimum temperature, average temperature, humidity, rainfall, sunshine duration, and wind speed and direction. After derived features were constructed, the final dataset contained 361 rows of clean data, as several rows were lost due to lag feature construction and target shifting. A total of 288 rows were used as training data, while 73 rows were used as testing data. The split was performed based on chronological order to preserve the temporal patterns of the weather data.

The testing data contained 41 rain events and 32 non-rain events, so class imbalance needed to be taken into account during the modeling process.

Model validation was conducted using Time Series Split with 5 folds. Each base model was trained on earlier portions of the data and then validated on subsequent data to generate OOF predictions. Based on the average CV F1-Score, Random Forest obtained the highest value of 0.650. XGBoost ranked second with a score of 0.646, followed by CatBoost at 0.607, LightGBM at 0.605, and Gradient Boosting at 0.602. Stacking Lasso obtained a CV F1-Score of 0.622, placing it between the performance of several base models. OOF predictions from all five base models were subsequently used to train the LassoCV meta-model. The best alpha value obtained was 0.00010.

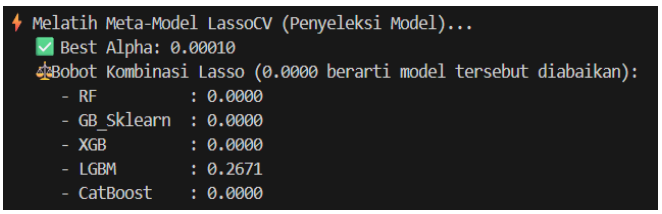

Figure 2. Model Weighting Results on LassoCV Meta-Model

The weighting results showed that L1 regularization in LassoCV retained only LightGBM as the model with a positive contribution, with a coefficient of 0.2671. The four other models, namely Random Forest, Gradient Boosting, XGBoost, and CatBoost, received zero weights and were therefore not used in the final Stacking Lasso predictions. Table 1 presents the evaluation results of all model performances on the testing data.

Table 1. Evaluation Results of All Model Performance

Metric	Random Forest	Gradient Boosting	XGBoost	LightGBM	CatBoost	Stacking Lasso
CV F1-Score	0.650	0.602	0.646	0.605	0.607	0.622
Accuracy	0.726	0.671	0.685	0.671	0.658	0.712
Precision	0.800	0.718	0.714	0.730	0.711	0.692
Recall	0.683	0.683	0.732	0.659	0.659	0.878
F1-Score	0.737	0.700	0.723	0.692	0.684	0.774
ROC-AUC	0.773	0.695	0.751	0.736	0.767	0.736

Based on the evaluation results in Table 1, Stacking Lasso achieved the highest F1-Score of 0.774 and the highest recall of 0.878. Random Forest still outperformed the others in precision with a value of 0.800 and an ROC-AUC of 0.773. Among the base models, XGBoost had the highest recall at 0.732. Meanwhile, CatBoost showed the lowest performance in this test, with an accuracy of 0.658 and an F1-Score of 0.684.

The Stacking Lasso confusion matrix shows that out of 32 actual no-rain data points, 16 were correctly identified as no-rain, while the remaining 16 were predicted as rain. For the actual rain class, the model correctly identified 36 out of 41 data points, while 5 rain events were incorrectly classified as no-rain. These results indicate that the model is more sensitive to the rain class, as reflected by the high number of True Positives in the confusion matrix visualization.

SHAP interpretation was performed on LightGBM since it was the only base model retained by LassoCV. Based on the SHAP summary plot, RH_AVG, or average relative humidity, emerged as the feature with the greatest influence. The next most important features were TAVG_minus_TN, FF_AVG_lag_2, TAVG_lag_1, FF_X_U, RR_3d_sum, FF_X_V, TEMP_RH_PRODUCT, TX_minus_TAVG, and RR_lag_2. High RH_AVG values tend to contribute positively to rainfall predictions, while low RH_AVG values push the model toward a no-rain prediction. The range of SHAP values observed was approximately -1.5 to +1.2.

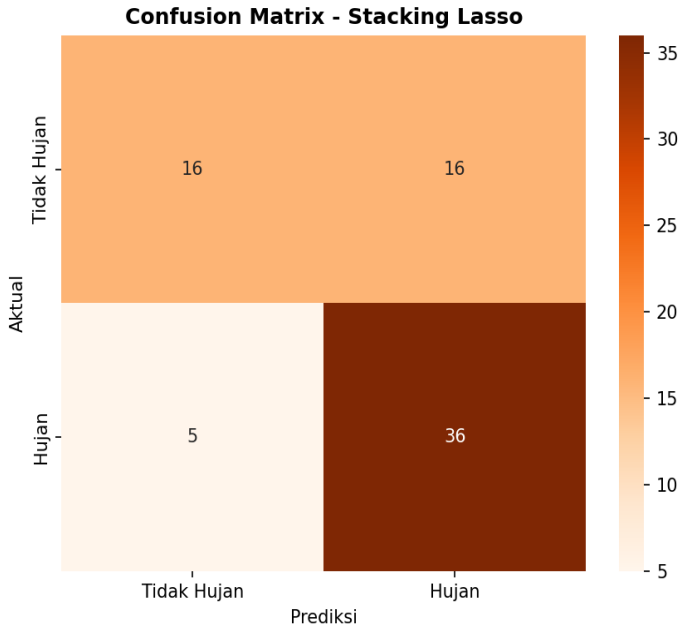

Figure 3. Confusion Matrix of Stacking Lasso Model

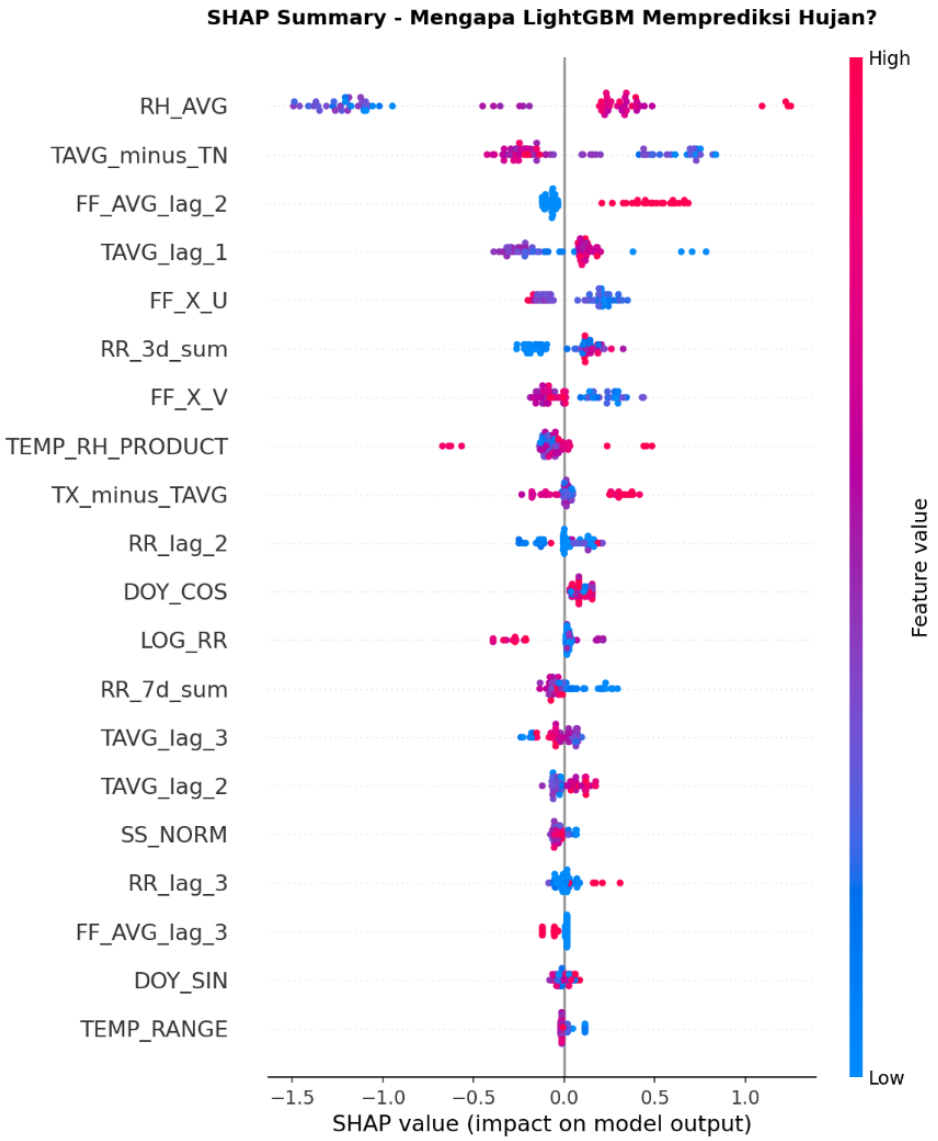

Figure 4. SHAP Summary Plot for Rainfall Prediction Interpretation in LightGBM Model

4. Discussion

The F1-Score of 0.774 achieved by Stacking Lasso demonstrates that the multi-tier ensemble approach is capable of delivering the best results compared to single models in this study. The most notable advantage lies in the recall value of 0.878. This means that out of 41 rain events in the testing data, the model successfully detected 36 and only missed 5. In the context of flood mitigation in Central Jakarta, the ability to recognize rain events is critically important because failure to detect rain can delay community and infrastructure preparedness. Therefore, high recall is the primary advantage of this model, even though accuracy is not the highest.

The selection of LightGBM as the only active model by LassoCV is also an important finding to discuss. At first glance, this result appears different from the CV F1-Score rankings since Random Forest actually has the highest validation score among the base models. However, LassoCV does not select models based on individual scores alone. This meta-model seeks the most efficient combination of OOF predictions to minimize overall error. In this process, the probabilistic predictions from LightGBM were judged to already provide sufficiently useful information, while contributions from other models did not significantly improve the results. Consequently, L1 regularization reduced the weights of Random Forest, Gradient Boosting, XGBoost, and CatBoost to zero.

This behavior can be explained through the way LassoCV handles correlated predictors. Because all five base models are trained on the same feature set and temporal structure, the out-of-fold probabilistic predictions of each

model tend to be highly correlated with one another, particularly among the boosting-based models (Gradient Boosting, XGBoost, and CatBoost) that share similar sequential error-correction mechanisms (Friedman, 2001; Chen & Guestrin, 2016; Prokhorenkova et al., 2018). When predictors are strongly correlated, L1 regularization in Lasso is known to retain one representative predictor from the correlated group while assigning zero weights to the others, rather than distributing weights evenly across all (Song & Zhang, 2024). In this case, LightGBM's leaf-wise growth strategy combined with Gradient-based One-Side Sampling (Ke et al., 2017) appears to produce out-of-fold probability estimates that are both accurate and sufficiently distinct from those of other base models, making it the representative predictor selected by LassoCV. Random Forest, despite achieving the highest individual CV F1-Score, was not selected because its bagging-based decision boundaries likely correlate closely with those of other tree-based models without providing unique additional information once LightGBM is included. This finding underscores an important methodological insight: a meta-learner such as LassoCV evaluates the redundancy among base learner predictions, not the standalone accuracy of each base learner, which is consistent with the general theory of stacked generalization (Wolpert, 1992; Sagi & Rokach, 2018).

The Stacking Lasso precision value of 0.692 should be understood as a consequence of high recall. In binary classification, especially when data is imbalanced, an increase in recall is often accompanied by a decrease in precision. The model tends to predict the rain class more frequently to avoid missing rain events. As a result, some data that is actually no-rain is classified as rain, as seen from the 16 False Positives in the confusion matrix. In the context of early warning, this condition is still acceptable because a false rain prediction generally carries lower risk than failing to detect an actual rain event. However, in future development, the classification threshold can be recalibrated to achieve a better balance between precision and recall according to field requirements.

The SHAP results provide insight into why the model makes its rainfall predictions. RH_AVG emerged as the most dominant feature, which is consistent with meteorological understanding that high air humidity supports cloud formation and precipitation. The TAVG_minus_TN feature is also influential because the difference between average temperature and minimum temperature can reflect daily thermal conditions. When temperature variation is relatively small, atmospheric conditions may be associated with more humid and overcast weather. Lag features such as FF_AVG_lag_2 and TAVG_lag_1 indicate that conditions from several days prior contribute to the model's ability to read weather patterns. Furthermore, RR_3d_sum shows that the accumulation of rainfall over the preceding three days is an important signal for predicting rain on the following day.

When compared with related rainfall classification studies in Indonesia, the findings of this study show both points of agreement and distinction. Pringandana and Kusnawi (2025), who classified multiclass rainfall categories in Jakarta using individually tuned XGBoost and LightGBM with RandomizedSearchCV, mean imputation, and oversampling, reported an accuracy of 94% for XGBoost and 91% for LightGBM. These figures are numerically higher than the 71.2% accuracy of Stacking Lasso in this study, but the comparison is not fully equivalent: their study addressed a same-day multiclass problem with balanced classes, whereas this study performs binary next-day classification on the original, imbalanced class distribution, which is inherently more challenging and operationally more realistic for early warning purposes.

Hermansyah et al. (2025), who applied a stacking ensemble of Random Forest, XGBoost, LightGBM, and SVM with HistGradientBoosting as a meta-learner and SMOTE-based class balancing on radiosonde data, reported a precision of 0.91 and an F1-Score of 0.87 for rainfall intensity classification. Although their F1-Score is numerically higher than the 0.774 achieved by Stacking Lasso in this study, their model depends on vertical atmospheric profile data collected through radiosondes, which are not as continuously or widely available as BMKG surface station data, limiting its operational scalability compared to the surface-data-based approach used here. Compared also with regression-based and single-algorithm studies such as Ayu Syaharani et al. (2025), who predicted daily rainfall in Bogor using a Grid-Search-tuned Random Forest and reported an RMSE of 21.92 mm and MAE of 15.21 mm, this study frames the problem as binary classification rather than regression, which is arguably more directly actionable for early warning purposes because the output (rain or no rain) can be used immediately without requiring a separate threshold conversion from a continuous rainfall estimate.

In general, the research results demonstrate that Stacking Ensemble with LassoCV as the meta-model can improve rainfall detection capability compared to single models, particularly in terms of recall and F1-Score. Nevertheless, this study still has limitations. The data used covers only one year with 361 clean samples, which is not yet sufficient to capture long-term climate variation in Central Jakarta. Future research should use data with a longer time range, apply more diverse imbalanced class handling methods such as SMOTE, and attempt threshold optimization so that the model can be adjusted to the risk levels of real-world applications.

5. Conclusion

This study demonstrates that Stacking Ensemble with LassoCV can be used for next-day rainfall classification and produces better performance than single models on several key metrics. Among the five base models used, Stacking Lasso achieved an F1-Score of 0.774 and a recall of 0.878. These results mean the model successfully identified 36 out of 41 rainfall events in the test data and missed only 5

events. In the context of urban flood mitigation, this level of rainfall detection capability is important because failing to detect rain can lead to more serious consequences than false alarms.

The weighting results also show that LassoCV retained only LightGBM with a weight of 0.2671, even though Random Forest had the highest CV F1-Score among the base models. This indicates that LassoCV functions as a contribution selector, not merely a selector of the model with the highest individual score. The meta-model selects the predictions that are most helpful in the final combination and eliminates contributions that are deemed to offer no additional benefit. From an interpretive perspective, SHAP indicates that average relative humidity (RH_AVG) is the variable that most influences rainfall predictions, followed by temperature features and several lag features. This finding supports the assertion that the model not only produces evaluation metrics but also captures meteorological patterns that can be logically explained.

Based on these results, this study provides benefits for several parties. For the public, information on indicators such as high humidity and rainfall patterns from the preceding days can help increase awareness, particularly in flood-prone areas. For BMKG and related institutions, this study demonstrates that historical meteorological data holds great potential when processed using appropriate machine learning methods. Collaboration between data-providing institutions and academic bodies needs to be strengthened so that more complete historical data can be utilized for early warning system development. For other researchers, the Stacking Ensemble and Time Series Cross Validation workflow in this study can serve as a reference for other time-series-based classification problems.

In terms of practical implementation, the high recall of the Stacking Lasso model suggests that it is best positioned as a conservative early warning trigger rather than a precise rainfall estimator. For instance, the Regional Disaster Management Agency (BPBD) of Central Jakarta could use the model's next-day rain classification as a one-day-ahead alert to prompt preparedness actions such as activating drainage pumps, readying flood barriers, or issuing public advisories, accepting a moderate false alarm rate as a reasonable operational trade-off given the higher cost of missing actual rain events. Since the model relies solely on routine BMKG surface variables rather than specialized instruments such as radiosondes, it is in principle replicable for other BMKG monitoring stations across Jakarta and other Indonesian cities without requiring additional data infrastructure. Furthermore, because LassoCV provides interpretable and non-negative weights for each base model, decision-makers and technical staff without deep machine learning backgrounds can still understand which model most influences the final prediction, supporting transparency when the system is used to underpin public warnings or resource allocation decisions. The primary limitation of this study lies in the restricted data volume, as the dataset covers only one year (2025), which is insufficient to capture long-term climate variability, including the potential influence of El Niño and La Niña on rainfall patterns in Central Jakarta. Future research should therefore utilize historical records spanning five to ten years to enable the model to learn broader rainfall patterns. In addition, class imbalance handling methods such as SMOTE, random undersampling, or a combination of both, as applied in related studies (Hermansyah et al., 2025; Ayu Syaharani et al., 2025), could be tested to achieve a more balanced performance between the rain and no-rain classes. Threshold optimization based on the precision-recall curve could also be performed so that the model's sensitivity can be tailored to specific operational requirements, rather than relying on the default threshold of 0.5 as in this study. From an implementation standpoint, the model can be further developed into a web-based or mobile application connected to real-time BMKG data. Testing on other Indonesian cities such as Surabaya, Medan, or Makassar is also warranted to determine whether this approach remains effective across regions with differing climate characteristics.

Disclosure Statement

The authors declare no conflicts of interest.

Funding Statement

The authors declare that no funding was received for this research

References

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83. https://doi.org/10.1016/j.csda.2017.11.003

BMKG. (2025). *Data dan informasi meteorologi Jakarta Pusat*. Bmkg.Go.Id.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cawood, P., & van Zyl, T. L. (2021). Feature-Weighted Stacking for Nonseasonal Time Series Forecasts: A Case Study of the COVID-19 Epidemic Curves. *2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 53–59. https://doi.org/10.1109/ISCMI53840.2021.9654809

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, 1–15.

Dwiyanti, Z. A., & Prianto, C. (2023). Prediksi Cuaca Kota Jakarta Menggunakan Metode Random Forest. *Jurnal Tekno Insentif*, *17*(2), 127–137. https://doi.org/10.36787/jti.v17i2.1136

Fauziah, A., Hermanto, H., & Sukmarini, M. A. (2024). Extreme Gradient Boosting pada Peramalan Pola Curah Hujan Bulanan Kabupaten Banyuwangi. *JURNAL KRIDATAMA SAINS DAN TEKNOLOGI*, *6*(02), 430–440. https://doi.org/10.53863/kst.v6i02.1154

Febriansyah Istianto, A., Id Hadiana, A., & Rakhmat Umbara, F. (2024). Prediksi Curah Hujan Menggunakan Metode Categorical Boosting (CatBoost). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(4), 2930–2937. https://doi.org/10.36040/jati.v7i4.7304

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Hafid, H., Rais, Z., & Rezky, A. R. R. T. (2025). Klasifikasi Curah Hujan di Kota Makassar Menggunakan Gradient Boosting Machine (GBM). *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, *7*(2), 133–142. https://doi.org/10.35580/variansiunm386

Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

Hermansyah, M., Saikhu, A., & Amaliah, B. (2025). Pemodelan Data Radiosonde Menggunakan Stacking Ensemble untuk Klasifikasi Hujan. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 10(2), 1678–1687. https://doi.org/10.29100/jipi.v10i2.7806

Ian, H. W., Eibe, F., Mark, A., & Chrisopher, J. P. (2017). *Data Mining: Practical machine learning tools and techniques*. Elsevier.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

Koetse, M. J., & Rietveld, P. (2009). The impact of climate change and weather on transport: An overview of empirical findings. *Transportation Research Part D: Transport and Environment*, *14*(3), 205–221. https://doi.org/10.1016/j.trd.2008.12.004

Kurniawan, D., Hidayat, F. O., & Saputra, A. H. (2024). Analisis Performa Model LightGBM dalam Prediksi Intensitas Hujan Wilayah Stasiun Meteorologi Kelas 1 Kualanamu. *Jurnal Penelitian Sains Dan Teknologi Indonesia*, *3*(1), 270–281.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45. https://doi.org/10.1109/MCAS.2006.1688199

Pringandana, C. G. L., & Kusnawi. (2025). A Comparative Analysis of Hyperparameter-Tuned XGBoost and LightGBM for Multiclass Rainfall Classification in Jakarta. Jurnal Teknik Informatika (JUTIF), 6(4), 2467–2483. https://doi.org/10.52436/1.jutif.2025.6.4.4965

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased Boosting with Categorical Features. *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html

Safia, M., Abbas, R., & Aslani, M. (2023). Classification of Weather Conditions Based on Supervised Learning for Swedish Cities. *Atmosphere*, *14*(7), 1174. https://doi.org/10.3390/atmos14071174

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, *8*(4). https://doi.org/10.1002/widm.1249

Song, Y., & Zhang, J. (2024). Enhancing short-term streamflow prediction in the Haihe River Basin through integrated machine learning with Lasso. Water Science & Technology, 89(9), 2367–2383. https://doi.org/10.2166/wst.2024.142

Syaharani, M. A., Dermawan, B. A., & Adam, R. I. (2025). Prediksi Curah Hujan Kota Bogor Menggunakan Algoritma Random Forest. *Jurnal Sistem Informasi dan Aplikasi*, *3*(1), 52-62. https://doi.org/10.52958/jsia.v3i1.12459

Tyralis, H., Papacharalampous, G., & Langousis, A. (2021). Super ensemble learning for daily streamflow forecasting: large-scale demonstration and comparison with multiple machine learning algorithms. *Neural Computing and Applications*, *33*(8), 3053–3068. https://doi.org/10.1007/s00521-020-05172-3

Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences* (Vol. 100). Academic press.

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1