ISSN 2723-6471

Machine Learning Models for Predicting Flood Events Using Weather Data: An Evaluation of Logistic Regression, LightGBM, and XGBoost

Maharina[1,*], Tukino Paryono[2], Ahmad Fauzi[3], Jamaludin Indra[4], Sihabudin[5], Muhammad Khoiruddin Harahap[6], Lutfi Trisandi Rizki[7]

[1,2] *Information System Program, Faculty of Computer Science, Universitas Buana Perjuangan Karawang, Indonesia*

[3,4] *Informatics Engineering Program, Faculty of Computer Science, Universitas Buana Perjuangan Karawang, Indonesia*

[5] *Management program, Faculty economic and business, Universitas Buana Perjuangan Karawang, Indonesia*

[6] *Politeknik Ganesha Medan, Indonesia*

[7] *Accounting Research Institute, Universiti Teknologi MARA, Syah Alam, Malaysia*

(Received: September 24, 2024; Revised: October 14, 2024; Accepted: November 16, 2024; Available online: December 30, 2024)

Abstract

This study examines flood prediction in Jakarta, Indonesia, a pressing concern due to its significant implications for public safety and urban management. Machine Learning (ML) presents promising methodologies for accurately forecasting floods by leveraging weather data. However, flood prediction in Jakarta remains challenging due to the city's highly variable weather patterns, including fluctuations in rainfall, humidity, temperature, and wind characteristics. Existing methods often struggle with these complexities, as they rely on traditional ML models such as K-Nearest Neighbors (KNN), which may not capture certain patterns or provide high accuracy and robustness. Therefore, this study proposes three ML methods—Logistic Regression (LR), LightGBM, and XGBoost—to predict floods accurately. Five performance metrics (i.e., accuracy, area under the curve (AUC), precision, recall, and F1-score) were used to measure and compare the accuracy of the algorithms. The proposed method consists of three main processes. The first process involves data preprocessing and evaluation using 14 different ML models. In the second process, additional feature engineering is applied to improve the quality of the data. Finally, the third process combines the previous steps with oversampling techniques and cross-validation methods. This structured approach aims to enhance the overall performance of the analysis. The experimental results show that Process 3 significantly improves performance compared to Processes 1 and 2. The model predicts floods with an accuracy score of 93.82% for LR, 96.67% for XGBoost, and 96.81% for LightGBM, respectively. Thus, the proposed model offers a solution for operational decision-making in flood risk management, including flood mitigation planning.

Keywords: Flood Prediction, Machine Learning, Logistic Regression, XGBoost, LighGBM

1. Introduction

Floods are natural disasters that occur when water from heavy rainfall, overflowing rivers, or tidal waves inundates areas that are typically dry [1]. Floods are often triggered by intense rainfall over a short period, where drainage systems are unable to handle the water volume, or due to infrastructure failures, such as levees or dams [2], [3]. Without more accurate prediction tools, governments and city authorities struggle to identify high-risk areas, leading to ineffective flood mitigation and response efforts [4], [5].

The long-term impacts of flooding include increased risk of property damage, loss of life, economic disruption, and higher post-disaster recovery costs [6], [7]. Frequent flooding can damage infrastructure, disrupt socio-economic activities, and exacerbate environmental and community vulnerabilities [8]. Addressing flood risks is becoming more challenging due to climate change and rapid urbanization. Traditional flood prediction systems often suffer from uncertainty, leading to inadequate preparedness. Therefore, modern systems are urgently needed. Flood prediction approaches have been extensively developed by leveraging various types of data, such as satellite imagery [9], infrared

***Corresponding author: Maharina (maharina@ubpkarawang.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i1.503

sensors [10], UAV-based aerial images [11], weather data [12], [13], etc. The use of satellite imagery, such as that from Landsat or Sentinel, enables the monitoring of large areas and the detection of water surface changes on a broad scale. Infrared sensors can detect surface temperature and soil moisture, which are critical indicators in flood prediction. UAV-based aerial images provide flexibility and high resolution for detailed mapping of flood-prone areas. Weather data, including rainfall, humidity, atmospheric pressure, and wind direction, also serve as the components in flood prediction models. In this study, we utilized weather data to accurately predict floods. To achieve this goal, we introduced machine learning methods as the primary approach. After conducting a series of comprehensive experiments, we found that three machine learning models delivered excellent evaluation performance results.

2. Literature Review

Research on modern flood prediction using Machine Learning (ML) demonstrates the application of various methods and datasets from different geographical locations [14], [15], [16]. Previous studies, such as that conducted by Gauhar et al. [17], applied KNN to a dataset from Bangladesh, achieving an accuracy of 94.91%. While KNN delivered promising results, this method may be less effective in capturing the non-linear complexities inherent in the data. In contrast, the study by Fang et al. [18] employed Long Short-Term Memory (LSTM) in Shangyou County, China, demonstrating its capacity to handle temporal data with 5-fold cross-validation, resulting in an accuracy of 93.75%. In this context, the LSTM approach was chosen due to its advantage in managing the temporal nature of the data.

Furthermore, Motta et al. [4] utilized Random Forest (RF) on a dataset from Lisbon, Portugal, with a 75:25 data split, achieving an accuracy of 96%, highlighting the effectiveness of RF in handling complex variables. The study by Ighile and Tanikawa [19] adopted Artificial Neural Networks (ANN) on a dataset from Nigeria, but with an accuracy of 76.40% and a 70:30 split ratio, it performed less favourably compared to the previously mentioned techniques. Meanwhile, Tso and Pan [20] implemented RF for flood prediction in New York, attaining an accuracy of 91% without cross-validation, suggesting that this method can yield competitive results despite the absence of a validation technique. In Jakarta, Grady et al. [21] achieved an accuracy of 86% with RF, while Hadi et al. [22] applied the C4.5 algorithm with cross-validation, reaching an accuracy of 87.20%. Recent study by Anjireddy and Jaisharma [23] applied KNN on data from Kerala, India, using an 80:20 split for training and test sets, yielding an accuracy of 91.40%. Unfortunately, the majority of the previous studies did not utilise cross-validation techniques, which means the models were unable to generalise effectively, thereby reducing the reliability of their performance in flood prediction.

This study primarily aims to model a natural phenomenon by employing ML techniques to predict flooding accurately. The main contributions of this study are as follows: First, we introduce a feature engineering method utilizing SMOTE, combined with 10-fold cross-validation to enhance the performance of the ML model. Second, we conduct a comprehensive comparison of fourteen algorithms for flood prediction. Finally, we demonstrate that XGBoost outperforms the other models in terms of predictive accuracy.

3. Material and Method

3.1. Dataset

The dataset used in this study includes weather conditions and recorded flood events from various stations in Jakarta during the period from 2016 to 2020 [24]. This dataset consists of 15 features detailing meteorological data as well as information related to floods, enabling a comprehensive analysis of climate trends and their relationship with flood occurrences. Table 1 and table 2 present the details of features and content of a sample dataset. The dataset comprises a total of 6,308 entries, with 476 entries indicating flood occurrences and 5,832 entries recording the absence of flooding.

Table 1. Dataset Structures

No	Features	Description	Data type
1	Date	Date	Date
2	Tn	min temperature (°C)	Float
3	Tx	max temperature (°C)	Float

 ISSN 2723-6471

4	Ta	Avg temperature (°C)	Float
5	RH	avg humidity (%)	Float
6	RR	rainfall (mm)	Float
7	Ss	duration of sunshine(hour)	Float
8	Fx	max wind speed (m/s)	Float
9	Dx	wind direction at maximum speed (°)	Float
10	Fa	max wind speed (m/s)	Float
11	Dc	most wind direction (°)	String
12	St1	station id which records the data	Integer
13	St2	station name which records the data	String
14	Reg	location of the station	String
15	Fl	Flood. Where 1 means True and 0 means false	Integer

Table 2. Sample of Dataset Contents

No	Date	Tn	Tx	Ta	RH	RR	ss	Fx	Dx	Fa	Dc	St1	St2	Reg	Fl
1	2016-01-01	26.00	34.80	28.60	81.00	NaN	5.80	5.00	280.00	2.00	S	96733	Stasiun Klimatologi Banten	Jakarta Selatan	0
2	2016-01-02	25.60	33.20	27.00	88.00	1.60	8.70	4.00	290.00	2.00	W	96733	Stasiun Klimatologi Banten	Jakarta Selatan	1
3	2016-01-03	24.40	34.90	28.10	80.00	33.80	5.40	4.00	280.00	2.00	SW	96733	Stasiun Klimatologi Banten	Jakarta Selatan	1
4	2016-01-04	24.80	33.60	29.20	81.00	NaN	6.60	3.00	200.00	1.00	S	96733	Stasiun Klimatologi Banten	Jakarta Selatan	1
								…							
6308	2018-12-31	25.4	32.08	28.2	69.0	9.9	NaN	14.0	180.0	5.0	SE	96747	Halim Perdana Kusuma Jakarta	Jakarta Timur	0

3.2. Proposed Method

To achieve the goal of obtaining the best ML model performance in flood prediction, we propose three main processes in our methodology. The first process involves data analysis, preprocessing, and prediction using 14 ML models. The second process is similar to the first but includes an additional feature engineering step after preprocessing. This stage we employ fourteen algorithm including Ada Boost Classifier (ADA), Gradient Boosting Classifier (GBC), XGBoost (XGB), LightGBM, Decision Tree Classifier (DT), Ridge, Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), Naive Bayes (NB), KNN, SVM, Extra Trees Classifier (ET), LR, and RF. The third process builds upon the previous steps by adding the SMOTE (Synthetic Minority Over-sampling Technique) to handle data imbalance, along with cross-validation, and then making predictions using the three models (LR, XGBoost, LightGBM). In this study, we compared LR, LightGBM, and XGBoost based on their respective strengths in handling flood prediction. LR is a simple ML model known for its computational efficiency, and it can provide a clear understanding of the relationship between input variables and outputs. However, to address more complex relationships between time-series features, such as fluctuating weather patterns, we also included LightGBM and XGBoost. Both of these algorithms are advanced decision-tree-based models that have proven to be effective in handling datasets with features exhibiting non-linear relationships and complex feature interactions. Furthermore, they have the ability to overcome overfitting issues. LR was also selected because, based on the evaluation results from Process 1, its performance outperformed that of the other models. An overview of the proposed method is illustrated in figure 1. A

more complex approach is taken in the third process. After performing feature engineering, we apply the SMOTE technique to address class imbalance, followed by 10-fold cross-validation. This cross-validation is used to assess the model's stability, reliability, and generalization capability with a more balanced data split. Finally, we employ the LR, LightGBM, and XGBoost algorithms to predict floods more accurately.

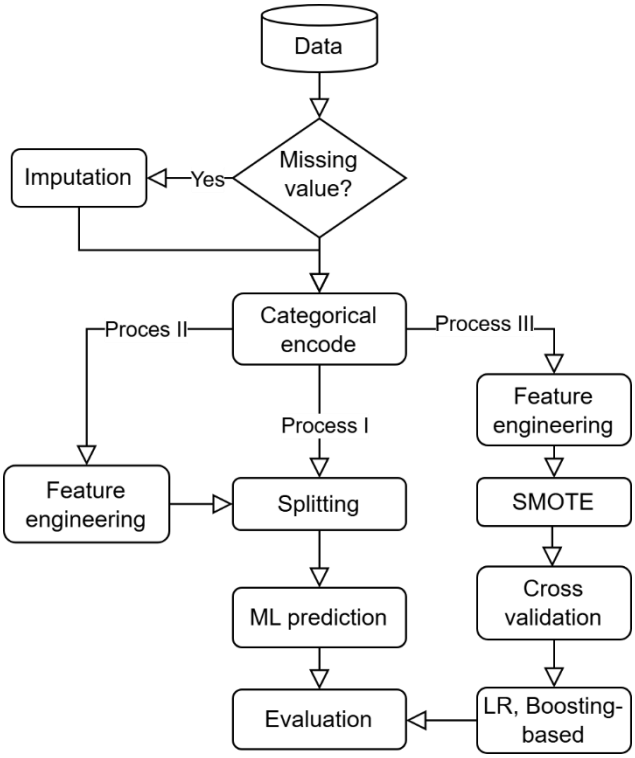


Figure 1. Flowchart of the proposed method

In evaluating the performance of the flood prediction models, we use several key metrics: accuracy (ACC), precision (PRE), recall (REC), and F1-score are calculated as shown in equations 1 to 4, where TP stands for true positive, TN for true negative, FP for false positive, and FN for false negative. Accuracy measures the proportion of total samples that the model has correctly classified. Precision indicates the model's ability to identify truly positive samples among all samples predicted as positive. The research experiment was conducted using Anaconda with Jupyter Notebook [25], and other libraries included Pandas [26], scikit-learn [27], and Matplotlib [28].

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{1}$$

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{2}$$

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{3}$$

$$\mathrm{F1-Score} = 2 \times \frac{\mathrm{PRE} \times \mathrm{REC}}{\mathrm{PRE+REC}} \tag{4}$$

3.3. Preprocessing

In the preprocessing stage, the data is examined to detect missing values. Missing values in weather variables are imputed using the mean method, assuming that the data distribution is stable and the missing values do not exhibit any specific pattern. This method is effective as the amount of missing data is relatively small. In the context of time-series data, such as weather data measured at specific time intervals, mean imputation can help maintain the stability of data patterns and prevent distortion in the analysis. For example, for variables like minimum temperature (Tn), maximum temperature (Tx), and humidity (RH), mean imputation estimates the missing values based on the historical distribution of the available data. Although it does not capture seasonal fluctuations or long-term trends, this approach is sufficient as its impact on the analysis results is minimal. Subsequently, categorical variables are transformed into numerical

representations to ensure compatibility with ML algorithms. In the second and third processes, following the categorical encoding of the data, the next step involves feature engineering, which is elaborated upon in the feature engineering subsection.

3.4. Feature Engineering

In this study, we implemented feature engineering techniques to extract information from the raw data and generate new features, aiming to maximize model performance [29]. A novel binary feature, "se" (season), was introduced, where "0" denotes the dry season and "1" represents the rainy season1. This classification is particularly relevant in Jakarta, Indonesia, which experiences only two distinct seasons annually: the dry season and the rainy season.

The feature "ss" (duration of sunshine) contains missing values, which we will address based on the new "se" feature. When "se" equals "1" (indicating the rainy season), we will impute the missing values in "ss" with the lowest recorded values of "ss". This approach is justified as the duration of sunshine tends to be significantly lower, potentially approaching zero, during the rainy season compared to the dry season. In addition to the "ss" feature, we also perform feature engineering on the "RR" (rainfall) variable, Whika also has missing values. When "ss" equals "0" (indicating the dry season) and the value of "RR" exceeds the mean, we will impute those missing values with "0". Furthermore, interaction features were created to capture potential relationships between key variables. Feature "RR_RH_interaction" was designed to explore the interaction between rainfall (RR) and relative humidity (RH), while "Tn_ss_interaction" and "Tx_ss_interaction" aimed to uncover the combined effects of minimum temperature (Tn) and maximum temperature (Tx) with the duration of sunshine (Ss). These transformations were integral in enhancing the dataset's informativeness for the machine learning model.

3.5. ML Prediction

3.5.1. Logistic Regression

LR algorithm is a supervised learning method that can be used for binary classification tasks [30]. LR model performs binary classification with probabilistic outputs of 0 and 1, where 1 represents 'flood' and 0 represents 'no flood'. The LR model is used to examine and calculate the correlation between features. It also includes a sigmoid function that transforms numerical data into a probability expression between 0 and 1, with a threshold of 0.5, where the first class includes values greater than 0.5, and the other class includes all values equal to or less than 0.5. The sigmoid function is illustrated in figure 2. Logistic function is fundamental to LR model It formulated as presented in Equation 5.

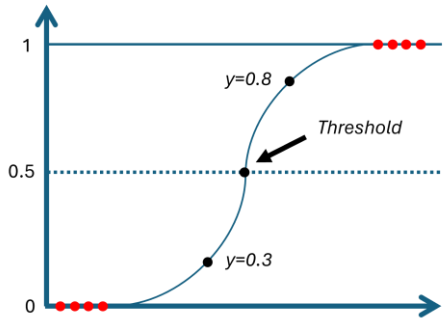


Figure 2. Sigmoid function on LR

$$\frac{1}{(1+e^{-value})} \tag{5}$$

Where $value$ refers to the actual numerical value we want to transform using the sigmoid function, and e represents the base of the natural logarithm.

3.5.2. Extreme Gradient Boosting

The XGBoost algorithm is a supervised learning algorithm widely used in ML [31]. XGBoost enhances gradient boosting by incorporating regularization (Ω) into the differentiable loss function (L), aimed at improving performance and preventing overfitting. This regularization is represented by the following equation 6:

 ISSN 2723-6471

$$L(\theta) = \sum_i l(y_i, \hat{y}_i) + \Omega(f) \quad (6)$$

Where $l(y_i, \hat{y}_i)$ represents the loss function measuring the difference between the predicted y_i and actual values $\hat{y}_i$.

3.5.3. Light Gradient Boosting Machine

LightGBM is an ML algorithm that operates by constructing trees in a leaf-wise manner [32], where deeper tree growth occurs in areas with greater error reduction. LightGBM is designed to deliver more accurate predictions, but without appropriate regularization, it carries a higher risk of overfitting. LightGBM has the capability to handle large datasets, limited memory, and deliver accurate predictions in a shorter time compared to traditional boosting algorithms (such as Gradient Boosting and AdaBoost).

3.6. Cross Validation Technique

We applied 10-fold cross-validation in this study with the aim of evaluating the model's performance more robustly and reducing evaluation bias. The 10-fold cross-validation technique essentially divides the flood dataset into 10 subsets of approximately equal size. Each time, 9 subsets are used to train the model, while the remaining 1 subset is used to test the model. This process is repeated 10 times, so that each subset serves as the test data once and as the training data 9 times. By implementing this 10-fold cross-validation technique, we can minimize the potential for overfitting, improve the model's generalization, and ensure that the evaluation results are more representative of the model's performance on unseen data. In this study 10-fold cross-validation provides a more reliable evaluation compared to hold-out validation and is less computationally intensive than leave-one-out cross-validation.

4. Experimental Results

4.1. Dataset Analysis

Figure 3 shows the weekly rainfall graph from several meteorological stations in the Jakarta area and its surroundings. The Tanjung Priok Maritime Meteorological Station recorded several instances of extreme rainfall, particularly in 2017 and 2020, with intensities exceeding 200 mm in a single week. This high level of rainfall poses a significant potential for flooding, especially in low-lying areas around the port that are susceptible to water accumulation. Nevertheless, most other weeks indicate low rainfall, signifying an uneven distribution of precipitation. The Banten Climatology Station also exhibits a variable rainfall pattern, with peaks exceeding 100 mm/week, especially in 2017 and from late 2019 to early 2020. The spikes in rainfall during this period are closely related to flood risks, particularly in urban areas such as South Jakarta. Additionally, data from this station highlight prolonged dry periods, which also emphasize the characteristic seasonal fluctuations in the region. At the Kemayoran Meteorological Station, the highest recorded rainfall spike occurred in 2020, with intensities surpassing 200 mm/week. This indicates a flood potential in Central Jakarta, which includes Kemayoran. Most other weeks showed low rainfall; however, the sudden onset of heavy rain poses a significant threat regarding flood risks. Meanwhile, the Halim Perdana Kusuma Jakarta Station recorded relatively consistent rainfall, although there was a notable peak in 2017 that reached over 150 mm in one week. This high intensity of rainfall also has the potential to cause localized flooding in the Halim area.

ISSN 2723-6471

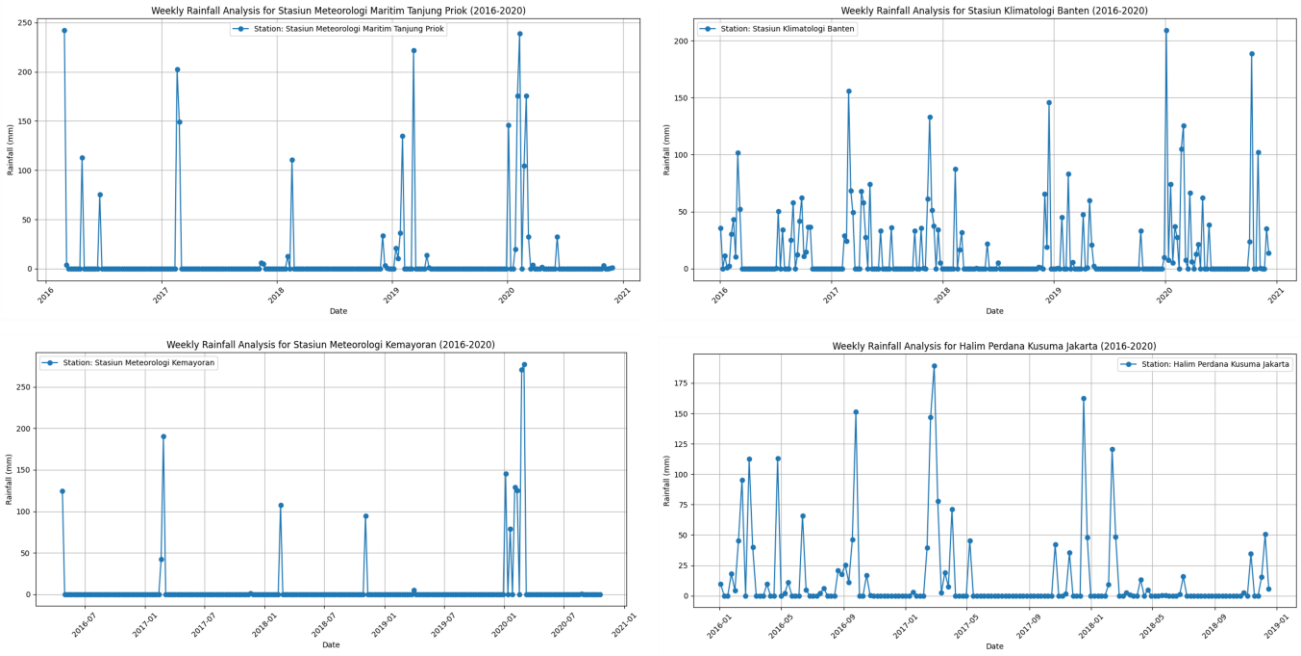


Figure 3. Graph rainfall during flood events in Jakarta area

4.2. Performance Prediction of Flood

The experiment was conducted in three processes as outlined in the methodology section. In Process 1, as shown in table 3, LR model achieved the highest accuracy of 0.9234, with an AUC of 0.8396. In Process 2, following the application of feature engineering, the LR model exhibited a slight improvement, achieving an accuracy of 0.9237 and an AUC of 0.8412, alongside improvements in recall (0.0478) and precision of 0.4748. Notably, the XGBoost model demonstrated a significant enhancement in recall, increasing from 0.1743 in Process 1 to 0.3275 in Process 2, with a corresponding improvement in the F1-score. Overall, the incorporation of feature engineering resulted in modest gains in model performance across several metrics.

Table 3. Performance Metrics of ML Models

Models	Process 1					Process 2				
	ACC	AUC	RE	PR	F1	ACC	AUC	RE	PR	F1
LR	0.9234	0.8396	0.0447	0.3845	0.0777	0.9237	0.8412	0.0478	0.4748	0.0852
ET	0.9207	0.8279	0.3184	0.4547	0.3711	0.9207	0.8277	0.3214	0.4579	0.3747
KNN	0.9203	0.6652	0.0510	0.3791	0.0881	0.9203	0.6652	0.0510	0.3791	0.0881
RF	0.9175	0.8003	0.3306	0.4313	0.3711	0.9182	0.8023	0.3307	0.4401	0.3744
ADA	0.9119	0.7902	0.3424	0.3993	0.3671	0.9123	0.7898	0.3394	0.4008	0.3648
GBC	0.9112	0.7111	0.3425	0.3963	0.3659	0.9114	0.6978	0.3455	0.3999	0.3692
LightGBM	0.9108	0.7788	0.3304	0.3893	0.3561	0.9110	0.7720	0.3275	0.3892	0.3547
DT	0.9103	0.7676	0.3275	0.3849	0.3528	0.9101	0.6412	0.3246	0.3853	0.3508
Ridge	0.9101	0.6412	0.3246	0.3853	0.3508	0.9101	0.8256	0.1743	0.3232	0.2230
XGB	0.9101	0.8256	0.1743	0.3232	0.2230	0.9099	0.7745	0.3213	0.3805	0.3471
QDA	0.8831	0.8211	0.3996	0.2923	0.3363	0.8831	0.8211	0.3996	0.2923	0.3363
LDA	0.8815	0.7973	0.3515	0.2721	0.3059	0.8815	0.7972	0.3515	0.2721	0.3059
NB	0.8689	0.8157	0.4205	0.2647	0.3228	0.8689	0.8156	0.4205	0.2647	0.3228
SVM	0.8395	0.5407	0.1000	0.0075	0.0139	0.8395	0.5407	0.1000	0.0075	0.0139

 ISSN 2723-6471

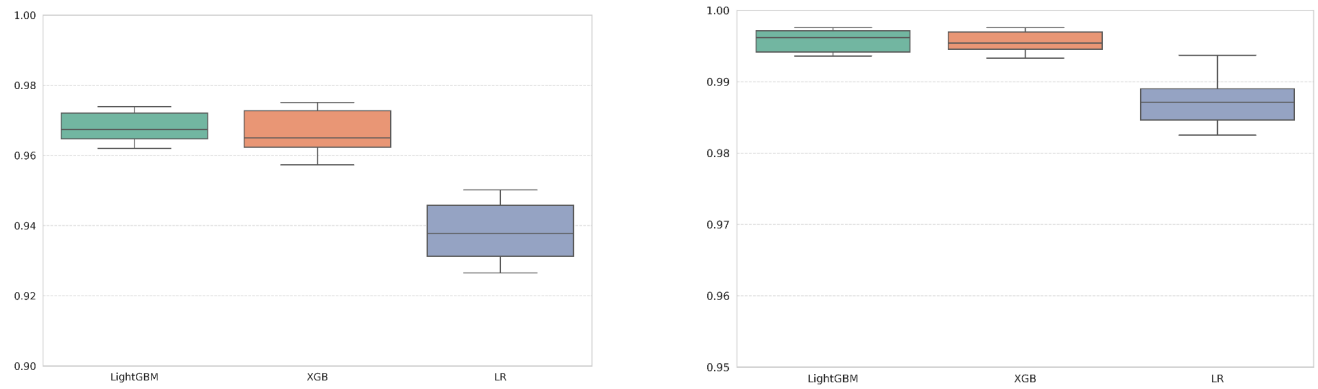
In Process 3, various strategies were employed to enhance model performance. As shown in table 3, LightGBM achieved the highest mean accuracy of 0.9681, followed by XGBoost with a mean accuracy of 0.9667, and LR with a mean accuracy of 0.9382. Table 4 highlights the performance in terms of the AUC, where LightGBM again outperformed the other models with a mean AUC of 0.9957, while XGBoost secured the second position with a mean AUC of 0.9956. LR, although slightly trailing, still exhibited performance with a mean AUC of 0.9874.

Table 4. Metrics Accuracy and AUC of LightGBM, XGBoost, and LR

Metrics	LR	XGB	LightGBM
ACC	0.9382 ± 0.0082	0.9667 ± 0.0061	0.9681 ± 0.004
AUC	0.9874 ± 0.0034	0.9956 ± 0.0014	0.9957 ± 0.0015

Accuracy metrics indicate how often the model makes correct predictions, which is highly relevant in the context of flood classification, where accurate decisions are crucial for disaster mitigation. Although LightGBM stands out with the highest average accuracy of 0.9681, AUC provides additional insights into the model's ability to handle class imbalance. In terms of AUC, LightGBM also shows superiority with an average value of 0.9957, demonstrating its ability to distinguish between classes with a high level of confidence.

The application of the SMOTE technique significantly enhances the model's performance. For the LR, XGB, and LightGBM models, the application of SMOTE improves accuracy (ACC) by approximately 1.6%, 6.2%, and 6.3%, respectively, with AUC increasing by around 17.6%, 20.5%, and 21.7%. SMOTE was chosen because, rather than using downsampling, which removes data and may hinder the model's learning with limited data, SMOTE generates synthetic samples for the minority class, improving the model's ability to make more accurate predictions and reducing bias towards the majority class. Figure 4 present boxplots that visualise the distribution of accuracy and AUC, further illustrating the robustness and consistency of LightGBM and XGBoost across the folds, while LR demonstrated somewhat higher variability.



(a) Accuracy (b) AUC

Figure 4. Comparison of LightGBM, XGBoost, and LR Models

Figure 5 illustrates the SHAP (Shapley Additive Explanations) visualization, which explains the impact of each feature on the predictions. The feature Ta has the most significant influence on the model, followed by Dx, month, and Ra. The colors on the plot indicate feature values, with red representing high values and blue representing low values. The SHAP value points on the horizontal axis show the direction and magnitude of each feature's impact on the predictions: positive values increase predictions, while negative values decrease them. This chart helps identify the most influential features on prediction outcomes and the direction of their impact within the model. Some features, such as Tx_ss_interaction, Tn_ss_interaction, RR_RH_interaction, ffa, and Dc, exhibit varied contributions to the model's predictions. The feature Ra stands out as one of the most important features, with higher values tending to increase model predictions.

 ISSN 2723-6471

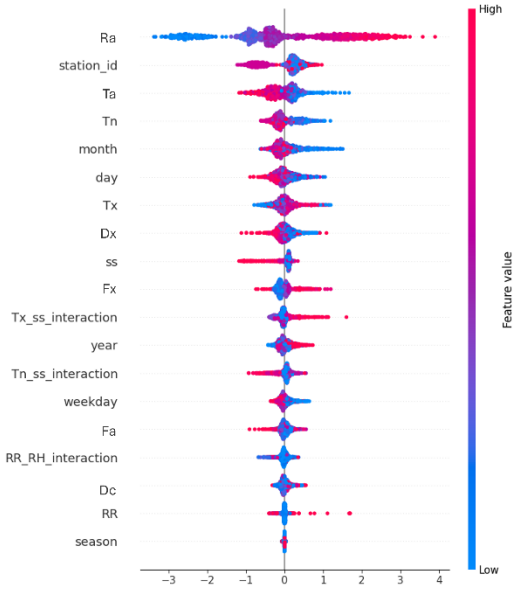


Figure 5. Visualization of Features Through SHAP

Additionally, the interactions Tx_ss_interaction and Tn_ss_interaction, representing the interaction between maximum temperature and sunshine duration, and minimum temperature and sunshine duration, respectively, show distinct patterns. High values of these interactions tend to have negative SHAP values, indicating that high combinations of temperature and sunshine reduce the model's predictions. The most important features in the model are located higher in the SHAP plot and exhibit a wide distribution of SHAP values around zero. The feature Ra emerges as the most significant, indicating that average humidity has a major influence on the model's predictions, both positively and negatively. This is followed by station_id, Ta, Tn, and Tx, which also show substantial impacts. Furthermore, interaction features such as Tx_ss_interaction (interaction between maximum temperature and sunshine duration), RR_RH_interaction (interaction between rainfall and humidity), and Tn_ss_interaction (interaction between minimum temperature and sunshine duration) also demonstrates significant influence, often with negative SHAP values. Overall, these features contribute substantially to the predictions, indicating that the model leverages environmental variables to generate more accurate results. Features located lower on the SHAP plot, such as weekday and RR, have smaller impacts, showing they are less critical compared to the top-ranked features.

5. Conclusion

This study aims to improve flood prediction accuracy using ML techniques. First, we demonstrate the effectiveness of various ML models by comparing the performance of 14 models. LightGBM, XGBoost, and LR were identified as the top-performing models, with LightGBM achieving the highest average accuracy of 96.81% and the highest AUC at 0.9957. These results confirm the superior predictive capability of LightGBM, followed by XGBoost, which exhibited comparable performance in terms of both accuracy and AUC. Additionally, this study implements the SMOTE strategy, along with 10-fold cross-validation, to enhance the model's ability to generalize and maintain consistent performance across various data splits. This methodological enhancement significantly contributes to improving the stability of the model. However, this study has certain limitations that must be acknowledged. The dataset employed records only one data point per day, while in reality, weather conditions can fluctuate every second. Consequently, this dataset does not adequately capture the complexities of real-world scenarios. Furthermore, the analysis relied on just 15 variables, despite the possibility that numerous other significant factors may also impact flood prediction outcomes. Future research should prioritise the use of more comprehensive and granular data to enhance the accuracy and relevance of flood predictions. Furthermore, it is essential to address the performance of the model in real-world scenarios. The proposed model demonstrates potential for use in flood prediction systems; however, further testing is required to evaluate its scalability and real-time applicability. Implementing the model in real-time would necessitate supportive data infrastructure and efficient algorithms capable of handling high-density data and temporal variability. The utilisation of hybrid models, such as combining RNN or LSTM with tree-based models or ensemble methods, could

offer a promising solution to simultaneously manage temporal and non-linear data patterns. This strategy could enhance predictive performance, particularly in capturing the complex dynamics of continually changing environmental factors.

6. Declarations

6.1. Author Contributions

Conceptualization: M.; Methodology: M.; Software: M.; Validation: A.F., S., T.P., and J.I.; Formal Analysis: M., J.I., and L.T.R.; Investigation: T.P., A.F., J.I., S., M.K.H.; Resources: A.F., and S.; Data Curation: S.; Writing Original Draft Preparation: M., A.F., S., and L.T.R.; Writing Review and Editing: T.P., A.F., J.I., S., M.K.H., and L.T.R; Visualization: M., A.F., and T.P. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ma, G. Zhao, B. He, Q. Li, H. Dong, S. Wang, and Z. Wang, "XGBoost-based method for flash flood risk assessment," *Journal of Hydrology*, vol. 598, no. 1, pp. 1-12, Jul. 2021, doi: 10.1016/j.jhydrol.2021.126382.

[2] P. J. Ward, M. C. de Ruiter, J. Mård, K. Schroter, A. Van Loon, T. Veldkamp, N. von Uexkull, N. Wanders, A. AghaKouchak, K. Arnbjerg-Nielsen, L. Capewell, M. C. Llasat, R. Day, B. Dewals, G. Di Baldassarre, L. S. Huning, H. Kreibich, M. Mazzoleni, E. Savelli, C. Teutschbein, H. van den Berg, A. van der Heijden, J. M. R. Vincken, M. J. Waterloo, and M. Wens, "The need to integrate flood and drought disaster risk reduction strategies," *Water Security*, vol. 11, no. 1, pp. 1-11, Dec. 2020, doi: 10.1016/j.wasec.2020.100070.

[3] A. Fekete, "Critical infrastructure and flood resilience: Cascading effects beyond water," *WIREs Water*, vol. 6, no. 5, pp. 1-9, Sep. 2019, doi: 10.1002/wat2.1370.

[4] M. Motta, M. De Castro Neto, and P. Sarmento, "A mixed approach for urban flood prediction using Machine Learning and GIS," *International Journal of Disaster Risk Reduction*, vol. 56, no. 1, pp. 1-11, Apr. 2021, doi: 10.1016/j.ijdrr.2021.102154.

[5] M. Jehanzaib, M. Ajmal, M. Achite, and T.-W. Kim, "Comprehensive Review: Advancements in Rainfall-Runoff Modelling for Flood Mitigation," *Climate*, vol. 10, no. 10, pp. 1-17, Oct. 2022, doi: 10.3390/cli10100147.

[6] I. Johnston, W. Murphy, and J. Holden, "A review of floodwater impacts on the stability of transportation embankments," *Earth-Science Reviews*, vol. 215, no. 1, pp. 1-15, Apr. 2021, doi: 10.1016/j.earscirev.2021.103553.

[7] S. Zhong, L. Yang, S. Toloo, Z. Wang, S. Tong, X. Sun, D. Crompton, G. FitzGerald, and C. Huang, "The long-term physical and psychological health impacts of flooding: A systematic mapping," *Science of The Total Environment*, vol. 626, no. 1, pp. 165–194, Jun. 2018, doi: 10.1016/j.scitotenv.2018.01.041.

[8] M. S. G. Adnan, Z. S. Siam, I. Kabir, Z. Kabir, M. R. Ahmed, Q. K. Hassan, R. M. Rahman, and A. Dewan, "A novel framework for addressing uncertainties in machine learning-based geospatial approaches for flood prediction," *Journal of Environmental Management*, vol. 326, no. 1, pp.1-14, Jan. 2023, doi: 10.1016/j.jenvman.2022.116813.

[9] H. S. Munawar, A. W. A. Hammad, and S. T. Waller, "Remote Sensing Methods for Flood Prediction: A Review," *Sensors*, vol. 22, no. 3, pp. 1-21, Jan. 2022, doi: 10.3390/s22030960.

[10] M. Mousa, X. Zhang, and C. Claudel, "Flash Flood Detection in Urban Cities Using Ultrasonic and Infrared Sensors," *IEEE Sensors J.*, vol. 16, no. 19, pp. 7204–7216, Oct. 2016, doi: 10.1109/JSEN.2016.2592359.

[11] H. S. Munawar, F. Ullah, S. Qayyum, and A. Heravi, "Application of Deep Learning on UAV-Based Aerial Images for Flood Detection," *Smart Cities*, vol. 4, no. 3, pp. 1220–1242, Sep. 2021, doi: 10.3390/smartcities4030065.

[12] C. Rossia, F. S. Acerbo, K. Ylinen, I. Juga, P. Nurmi, A. Bosca, F. Tarasconi, M. Cristoforetti, and A. Alikadic, "Early detection and information extraction for weather-induced floods using social media streams," *International Journal of Disaster Risk Reduction*, vol. 30, no. 1, pp. 145–157, Sep. 2018, doi: 10.1016/j.ijdrr.2018.03.002.

[13] M. J. Subashini, R. Sudarmani, S. Gobika, and R. Varshini, "Development of Smart Flood Monitoring and Early Warning System using Weather Forecasting Data and Wireless Sensor Networks-A Review," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Tirunelveli, India: IEEE, vol. 3, no. 1, pp. 132–135, 2021. doi: 10.1109/ICICV50876.2021.9388418.

[14] H. Mojaddadi, B. Pradhan, H. Nampak, N. Ahmad, and A. H. B. Ghazali, "Ensemble machine-learning-based geospatial approach for flood risk assessment using multi-sensor remote-sensing data and GIS," *Geomatics, Natural Hazards and Risk*, vol. 8, no. 2, pp. 1080–1102, Dec. 2017, doi: 10.1080/19475705.2017.1294113.

[15] T. Xie, C. Hu, C. Liu, W. Li, C. Niu, and R. Li, "Study on long short-term memory based on vector direction of flood process for flood forecasting," *Sci Rep*, vol. 14, no. 1, pp. 21446-21458, Sep. 2024, doi: 10.1038/s41598-024-72205-5.

[16] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, and A. Krishnan, "Flood prediction based on weather parameters using deep learning," *Journal of Water and Climate Change*, vol. 11, no. 4, pp. 1766–1783, Dec. 2020, doi: 10.2166/wcc.2019.321.

[17] N. Gauhar, S. Das, and K. S. Moury, "Prediction of Flood in Bangladesh using k-Nearest Neighbors Algorithm," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, DHAKA, Bangladesh: IEEE, vol. 2, no. 1, pp. 357–361, 2021. doi: 10.1109/ICREST51555.2021.9331199.

[18] Z. Fang, Y. Wang, L. Peng, and H. Hong, "Predicting flood susceptibility using LSTM neural networks," *Journal of Hydrology*, vol. 594, no. 1, pp. 1-20, Mar. 2021, doi: 10.1016/j.jhydrol.2020.125734.

[19] E. H. Ighile, H. Shirakawa, and H. Tanikawa, "Application of GIS and Machine Learning to Predict Flood Areas in Nigeria," *Sustainability*, vol. 14, no. 9, pp. 1-33, Apr. 2022, doi: 10.3390/su14095039.

[20] J. Tso and H. Pan, "A Novel Machine Learning Approach for Flood Prediction with Local Interpretable Explanations," in *2023 IEEE MIT Undergraduate Research Technology Conference (URTC)*, Cambridge, MA, USA: IEEE, vol. 2023, no. 1, pp. 1–5, 2023. doi: 10.1109/URTC60662.2023.10534980.

[21] F. Grady, J. K. Tarigan, J. R. Wahidiyat, and A. Prasetyo, "Classification of Flood Alert in Jakarta with Random Forest," in *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)*, Yogyakarta, Indonesia: IEEE, vol. 7, no. 1, pp. 1–6, 2022. doi: 10.1109/ICITDA55840.2022.9971411.

[22] S. Hadi, D. Fitrianah, V. Ayumi, and S. Mooi Lim, "The Data Analysis of Determining Potential Flood-Prone Areas in DKI Jakarta Using Classification Model Approach," *ijaste*, vol. 1, no. 1, pp. 313–323, Feb. 2023, doi: 10.24912/ijaste.v1.i1.313-323.

[23] D. Anjireddy and J. K, "Performance Comparison of Flood Prediction Using Recurrent Time Delay Neural Network and K-Nearest Neighbor Algorithm," in *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India: IEEE, vol. 9, no. 1, pp. 1–5, 2024. doi: 10.1109/ICONSTEM60960.2024.10568880.

[24] Gadzz, "Climate and Flood Jakarta 2016–2020," Dataset on flood and climate in Jakarta from 2016 to 2020. [Online]. Available: https://www.kaggle.com/datasets/christopherrichardc/climate-and-flood-jakarta/data. Accessed: Jan. 21, 2024.

[25] D. Rolon-Mérette, M. Ross, T. Rolon-Mérette, and K. Church, "Introduction to Anaconda and Python: Installation and setup," *TQMP*, vol. 16, no. 5, pp. S3–S11, May 2020, doi: 10.20982/tqmp.16.5.S003.

[26] J. Bernard, "Python Data Analysis with pandas," in *Python Recipes Handbook: A Problem-Solution Approach*, J. Bernard, Ed., Berkeley, CA: Apress, pp. 37–48, 2016. doi: 10.1007/978-1-4842-0241-8_5.

[27] O. Kramer, "Scikit-Learn," in *Machine Learning for Evolution Strategies*, O. Kramer, Ed., Cham: Springer International Publishing, 2016, pp. 45–53. doi: 10.1007/978-3-319-33383-0_5.

[28] E. Bisong, "Matplotlib and Seaborn," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, E. Bisong, Ed., Berkeley, CA: Apress, 2019, pp. 151–165. doi: 10.1007/978-1-4842-4470-8_12.

 ISSN 2723-6471

[29] M. F. Uddin, J. Lee, S. Rizvi, and S. Hamada, "Proposing Enhanced Feature Engineering and a Selection Model for Machine Learning Processes," *Applied Sciences*, vol. 8, no. 4, pp. 646-657, Apr. 2018, doi: 10.3390/app8040646.

[30] A. Bailly, C. Blanc, É. Francis, T. Guillotin, F. Jamal, B. Wakim, and P. Roy, "Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models," *Computer Methods and Programs in Biomedicine*, vol. 213, no. 1, pp. 1-7, Jan. 2022, doi: 10.1016/j.cmpb.2021.106504.

[31] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, vol. 22, no. 1, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

[32] G. V. D. Kumar, V. Deepa, N. Vineela, and G. Emmanuel, "Detection of Parkinson's disease using LightGBM Classifier," in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India: IEEE, vol. 6, no. 1, pp. 1292–1297, 2022. doi: 10.1109/ICCMC53470.2022.9753909.