JEECS (Journal of Electrical Engineering and Computer Sciences)
Vol. 11, No. 2, December 2026, pp. 161-171
e-ISSN: 2579-5392 p-ISSN: 2528-0260

Rainfall Classification in Palembang Using Machine Learning Methods

M Faris Al Amin, Keviano Alessandro Delpiero, Rendra Gustriansyah

Department of Informatics Engineering, Faculty of Computer Science, Universitas Indo Global Mandiri, Jl. Jenderal Sudirman Km.4 No. 629, Kec. Ilir Timur I, Palembang, South Sumatra, 30129, Indonesia

Article Info

Article history:
Received: 9 June 2026
Revised: 27 July 2026
Accepted: 18 August 2026

Keyword:
Data mining
Machine learning
Palembang
Rainfall
Rainfall classification

Abstract

Rainfall is an important weather parameter that affects various sectors, including agriculture, transportation, and disaster mitigation. As a tropical region, Palembang experiences varying rainfall patterns, making effective rainfall classification essential. However, previous studies have generally focused on specific machine learning methods, used a limited number of weather variables, or classified rainfall into only two categories. These differences make it difficult to understand how various machine learning methods perform when applied to multiclass rainfall classification using more complete meteorological data. Therefore, this study aims to classify rainfall in Palembang using machine learning methods, namely Random Forest, Decision Tree, Naïve Bayes, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). The dataset used in this study consists of meteorological data obtained from the Sultan Mahmud Badaruddin II Meteorological Station, Palembang, with an initial total of 365 records. The research stages included data preprocessing, rainfall labeling into three categories (No Rain, Rain, and Extreme Rain), data splitting into training and testing sets, and classification using five machine learning algorithms. The results indicate that the Support Vector Machine (SVM) achieved the best performance, with an accuracy of 72,72%, precision of 74.58%, recall of 48.96%, and F1-score of 72.61%. Based on these findings, SVM is considered more effective than the other methods for classifying rainfall using meteorological data in Palembang.

Corresponding author:
M Faris Al Amin, 2023110020@students.uigm.ac.id

DOI: https://doi.org/10.54732/jeecs.v11i2.6

1. Introduction

Rainfall is one of the most important parameters in climate analysis, affecting various sectors such as agriculture, transportation, and disaster mitigation. High rainfall intensity over a certain period can increase water discharge and cause rivers to overflow, leading to flooding events[1]. In addition, heavy or extreme rainfall may trigger natural disasters such as floods and landslides, resulting in both material and non-material losses. In Indonesia, rainfall prediction is particularly challenging because rainfall patterns are strongly influenced by global climate variability phenomena, such as the El Niño–Southern Oscillation (ENSO)[2]. Therefore, analyzing rainfall characteristics and patterns is essential to support effective disaster mitigation and risk reduction efforts.

Data mining is a process that utilizes statistical techniques, mathematics, artificial intelligence, and machine learning to extract and identify useful information and knowledge from large datasets[3]. Data mining can be categorized into several tasks, including description, estimation, prediction, classification, clustering, and association[4]. Among these tasks, classification is one of the most widely used techniques, which aims to assign data into predefined classes based on their characteristics and attributes. Classification is the process of developing a model or function that describes and distinguishes data concepts or classes, with the objective of predicting the class label of an unknown object[5]. Various algorithms can be applied in classification tasks, one of which is the Naïve Bayes algorithm. Naïve Bayes is a machine learning algorithm commonly used for classification problems, particularly in text classification involving high-

dimensional training datasets[6]. This algorithm has been widely adopted due to its strong modeling performance and classification accuracy. In addition, Naïve Bayes is recognized for its computational efficiency, requiring relatively short training times while maintaining competitive accuracy, making it one of the most effective algorithms in terms of model construction and execution speed[7].

Previous studies on rainfall classification using the Naïve Bayes and C4.5 algorithms reported the highest accuracy values of 100% and 99.83%, respectively[8]. However, the study did not compare the performance of these methods with other machine learning algorithms. Another study employing Support Vector Machine (SVM) and Recursive Feature Elimination (RFE) achieved a maximum accuracy of 79%[9], but the rainfall data were classified into only two categories, namely rain and no rain. The other research conducted feature selection to reduces training time, when classifies using Random Forest and Neural Network achieves accuracy up to 99.25% [10], estimating rainfall rate and kinetic energy [11], and applying machine learning for estimate uneven rainfall [12]. Furthermore, a study using Random Forest and Naïve Bayesian obtained an accuracy of 86.55% and 36.61%[13]. Nevertheless, the study utilized only 4 weather attributes, including average temperature, average humidity, sunshine duration, and wind direction at maximum wind speed. Although previous studies have reported encouraging results, their findings cannot be directly compared because they used different weather variables, rainfall categories, and machine learning methods. In addition, most studies evaluated only one or two algorithms. Therefore, a comparison of several machine learning methods using the same dataset and rainfall categories is still needed to better understand the strengths and limitations of each method.

More recent studies have continued to explore rainfall classification using machine learning. Utari et al. compared Decision Tree, K-Nearest Neighbor (KNN), and Logistic Regression for rainfall classification in Yogyakarta and found that the Decision Tree model achieved the best performance for their dataset. The study also highlighted that the selection of weather variables plays an important role in classification performance[14]. Similarly, Pringandana and Kusnawi evaluated XGBoost and LightGBM for multiclass rainfall classification using meteorological data from Jakarta. Their results showed that model performance can vary depending on the characteristics of the dataset and the machine learning algorithm applied, emphasizing the importance of evaluating different methods before selecting the most appropriate model for a particular region[15].

This study aims to implement and compare several machine learning methods for classifying rainfall in Palembang into multiple categories, namely No Rain, Rain, and Extreme Rain. The comparison is conducted to evaluate the performance of each algorithm and identify the most effective method for rainfall classification based on meteorological data. The results of this study are expected to contribute to the development of rainfall prediction and support decision-making processes related to weather monitoring and disaster mitigation. The results of this study are expected to assist the Meteorological, Climatological, and Geophysical Agency (BMKG) in understanding rainfall patterns in Palembang and support decision-making processes related to regional planning and disaster mitigation. Furthermore, this study provides a comparative performance analysis of several machine learning methods, namely Random Forest, Decision Tree, Naïve Bayes, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), which may serve as a valuable reference for future studies in selecting appropriate classification methods for rainfall-related applications.

2. Research Methodology

The research was conducted through several stages, including BMKG data collection, data preprocessing, training and testing data splitting, implementation of classification algorithms, and model performance evaluation. The overall research workflow is illustrated in Figure 1.

2.1. Data Collection

The dataset used in this study was obtained from the official website of the Meteorological, Climatological, and Geophysical Agency (BMKG), Sultan Mahmud Badaruddin II Meteorological Station, Palembang, South Sumatra. The dataset consists of daily meteorological observations collected from January 1, 2025, to December 31, 2025, resulting in an initial total of 365 records. After the data cleaning process, 339 records were retained for further analysis. Seven meteorological variables were used as input attributes: minimum temperature, maximum temperature, average temperature, average humidity, sunshine duration, maximum wind speed, and average wind speed. The recorded rainfall values were used only to generate the target class labels and were not included as input features during model training. Based on the recorded rainfall intensity, the data were classified into three categories: No Rain, Rain, and Extreme Rain.

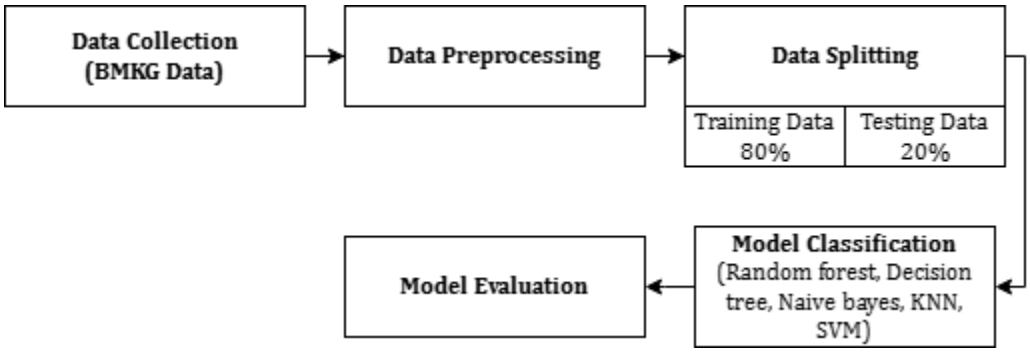

Figure 1. Research Workflow

2.2. Data Preprocessing and Data Splitting

Data preprocessing was performed to prepare the dataset before the classification process. This stage aims to improve data quality and ensure that the dataset can be effectively processed by machine learning algorithms. Data preprocessing is a crucial step because it can significantly influence the performance of the resulting classification models.

During the variable selection stage, the variables used in this study consisted of minimum temperature, maximum temperature, average temperature, average humidity, sunshine duration, maximum wind speed, and average wind speed. The rainfall variable was not used as an input attribute because it served as the basis for generating the target class labels during the data labeling process. Subsequently, a data cleaning process was performed to handle incomplete records and missing values. Records containing empty or invalid attributes were removed to maintain dataset quality and ensure effective processing during the classification stage. Data cleaning is an essential step for reducing inconsistencies and improving dataset quality prior to model development[16].

In the class labeling stage, rainfall data were categorized into three classes based on rainfall intensity, No Rain, Rain, and Extreme Rain. This process was conducted to assign a target class label to each record, which was later used in the machine learning classification process. For dataset partitioning, the data were divided into training data and testing data using an 80:20 ratio. The training data were used to build the classification models, while the testing data were utilized to evaluate model performance[17].

The final preprocessing step involved data normalization to ensure that all attributes were measured on a comparable scale. Normalization is particularly important because several machine learning algorithms, such as K-Nearest Neighbor (KNN) and Support Vector Machine (SVM), are sensitive to differences in feature scales[18]. By applying normalization, each attribute can contribute more equally to the classification process, thereby improving model performance.

2.3. Model Classification

A. Random Forest

Random Forest is an ensemble-based machine learning algorithm that constructs multiple decision trees and combines their predictions to improve classification accuracy[19]. This algorithm is widely used in weather and rainfall classification tasks because it can effectively handle large datasets and achieve strong classification performance[20].

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \dots, h_k(x)\} \quad (1)$$

Explanation:

- $\hat{y}$ = final prediction result
- $h_1(x)$, $h_2(x)$, ..., $h_k(x)$ = prediction results generated by each decision tree
- k = number of trees in the Random Forest model
- mode = the class label that appears most frequently among all predictions (majority voting)

B. Decision Tree

Decision Tree is a flowchart-like structure in which each internal node (a node that is not a leaf node) represents a test on an attribute, each branch represents the outcome of that test, and each leaf node represents a class label[21]. The process of constructing a decision tree begins by calculating the entropy value to measure the impurity or uncertainty of the dataset. The entropy formula can be expressed as follows:

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \quad (2)$$

Explanation:
- S = dataset (set of data instances)
- p_i = probability of class *i*
- n = total number of classes

Subsequently, the information gain value is calculated to determine the best attribute to be selected as a node in the decision tree. The information gain formula can be expressed as follows:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (3)$$

Explanation:
- Gain (S, A) = information gain value of attribute *A*
- Entropy(S) = entropy of the original dataset
- S_v = subset of data corresponding to attribute *A*
- $|S_v|$ = number of instances in the subset
- |S| = total number of instances in the dataset

C. Naïve Bayes

Naïve Bayes is a probabilistic classification method that performs classification based on probability calculations derived from the frequency and combination of values within a dataset. The Naïve Bayes algorithm assumes that each attribute is independent of the others with respect to the target class. It can also be described as a classification method based on probability theory and statistics, first introduced by the English mathematician Thomas Bayes, which predicts future outcomes based on prior observations and evidence[22].The fundamental formula of Bayes' Theorem can be expressed as follows:

$$P(H \mid X) = \frac{P(X \mid H) \cdot P(H)}{P(X)} \quad (4)$$

Explanation:
- P(H|X) = probability of hypothesis *H* given condition *X*
- P(X|H) = probability of observing data *X* given hypothesis *H*
- P(H) = prior probability of hypothesis *H*
- P(X) = probability of observing data *X*

In the classification process, Naïve Bayes assigns a data instance to the class with the highest posterior probability based on its attributes. The Naïve Bayes classification formula can be expressed as follows:

$$C_{NB} = \arg\max_{C_i} P(C_i) \prod_{j=1}^{n} P(x_j \mid C_i) \quad (5)$$

Explanation:
- $C_{n\beta}$ = classification result produced by the Naïve Bayes algorithm
- C_i = *i-th* class
- $P(C_i)$ = prior probability of class C_i
- $P(x_j|C_i)$ = probability of attribute x_j given class C_i
- n = number of attributes

D. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a classification method that determines the class of a data instance based on the majority class among its nearest neighbors using a distance calculation approach. The algorithm performs classification by comparing new data instances with training data according to their level of similarity or proximity.

KNN is widely used because of its simplicity, ease of implementation, and ability to produce satisfactory classification results across various types of datasets. Furthermore, the classification process in KNN is based on the similarity between data instances, making it an effective approach for classification

tasks[23]. In this study, the distance between data instances was calculated using the Euclidean Distance metric, which is defined as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (6)$$

Explanation:
- d(x,y) = distance between data instances *x* and *y*
- x_i = value of the *i-th* attribute of the testing instance
- y_i = value of the *i-th* attribute of the training instance
- n = number of attributes

After calculating the distances, the classification process is performed by assigning the class that appears most frequently among the k nearest neighbors. The KNN classification formula can be expressed as follows:

$$y = \text{mode}(k\text{-nearest neighbors}) \quad (7)$$

Explanation:
- y = classification result
- k = number of nearest neighbors
- mode = the class label that appears most frequently among the *k* nearest neighbors

E. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification method that works by finding the optimal hyperplane to separate data into different classes. This algorithm determines the optimal decision boundary by maximizing the margin between classes, enabling it to achieve strong classification performance[24].

The Support Vector Machine (SVM) method is widely used in classification tasks because it has strong capability in handling complex and high-dimensional data, and can achieve high classification performance. SVM works by constructing an optimal hyperplane that separates different classes while maximizing the margin between them[25]. The hyperplane equation in SVM can be expressed as follows:

$$w \cdot x + b = 0 \quad (8)$$

Explanation:
- w= weight vector
- x= input data
- b= bias
- $w \cdot x$= dot product between the weight vector and the input data

The classification process is carried out using a decision function to determine the position of a data point relative to the hyperplane. The SVM classification function is expressed as follows:

$$f(x) = \text{sign}(w \cdot x + b) \quad (9)$$

Explanation:
- $f(x)$= classification result
- sign = class decision function
- Jika $f(x) > 0$, the data is classified into the positive class.
- Jika $f(x) < 0$, the data is classified into the negative class.

Furthermore, SVM works by maximizing the margin between classes to obtain an optimal decision boundary. The margin formula in SVM can be expressed as follows:

$$Margin = \frac{2}{\| w \|} \quad (10)$$

Explanation:
- $Margin$= distance between the hyperplanes
- $\| w \|$= magnitude (norm) of the weight vector

2.4. Evaluation

The model evaluation stage was conducted to measure the performance of each classification algorithm in classifying rainfall data. The evaluation was performed using a confusion matrix, which produces accuracy, precision, recall, and F1-score values. A confusion matrix is an evaluation method used to compare the model's prediction results with the actual class labels. The confusion matrix consists of several components, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)[26].

Accuracy is used to measure the overall correctness of a classification model by calculating the proportion of data instances that are correctly predicted out of the total number of instances[27]. The formula for calculating accuracy is presented in Equation (11).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Precision is used to measure the accuracy of the model in predicting positive instances compared to all instances predicted as positive[28]. The calculation of precision is presented in Equation (12).

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

Recall measures the proportion of positive cases that are successfully identified by the model, indicating how many instances from the positive class are correctly detected. Recall is particularly important in situations where False Negative (FN) errors pose a greater risk[29]. The recall formula can be seen in Equation (13).

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

F1-score is a metric used to measure the harmonic mean between precision and recall[30]. The formula for the F1-score is shown in Equation (14).

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

3. Results and Discussions

3.1 Data Preprocessing Results

The dataset used in this study was obtained from online data provided by the Meteorology, Climatology, and Geophysics Agency (BMKG) of South Sumatra. The dataset consists of 365 records with 9 variables. A data cleaning process was then performed to handle missing values or empty data. Records with missing values in the rainfall variable were removed, resulting in a reduced dataset of 339 records.

After that, an attribute selection process was carried out, resulting in 7 variables used in this study, minimum temperature, maximum temperature, average temperature, average humidity, sunshine duration, maximum wind speed, and average wind speed. The date variable was not used because it does not directly affect the classification process, while the rainfall variable was used as the basis for labeling or forming rainfall class categories. The rainfall labelling categories used in this study are presented in Table 1.

Furthermore, the distribution of the number of data points in each rainfall category was obtained. The data distribution can be seen in Table 2. The data splitting was performed using an 80:20 ratio, where 80% of the data was used as training data and 20% as testing data. From the total dataset used, 273 records were allocated for training and 66 records for testing. Subsequently, data normalization was applied to the numerical attributes using the Min-Max Scaling method.

Table 1. Rainfall Categories

Category	Rainfall (mm)
No Rain	0 mm
Rain	1–49 mm
Extreme Rain	≥ 50 mm

Table 2. Data Distribution

Category	Number of Data
No Rain	133
Rain	188
Extreme Rain	18

3.2 Training Data Evaluation Results

In the training phase, the model was trained using five machine learning methods, namely Random Forest, Decision Tree, Naïve Bayes, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). Model performance on the training data was evaluated using a confusion matrix with accuracy, precision, recall, and F1-score metrics. The evaluation results of the training data for each method are presented in Table 3.

Based on Table 3, the Random Forest method shows the highest performance, with accuracy, precision, recall, and F1-score reaching 100%. These results indicate that the method is able to recognize patterns in the training data very well; however, they also suggest a potential risk of overfitting. Meanwhile, Decision Tree, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM) show relatively good performance with accuracy values above 75%. On the other hand, the Naïve Bayes method achieved the lowest performance compared to the other methods.

Table 3. Evaluation Metrics Result

No	Metode	Accuracy	Precision	Recall	F1-Score
1	Random Forest	1.0000	1.0000	1.0000	1.0000
2	Decision Tree	0.7912	0.7957	0.5521	0.8101
3	Naïve Bayes	0.6959	0.6261	0.6239	0.6249
4	KNN	0.7838	0.7750	0.6050	0.6358
5	SVM	0.7582	0.7884	0.5177	0.7667

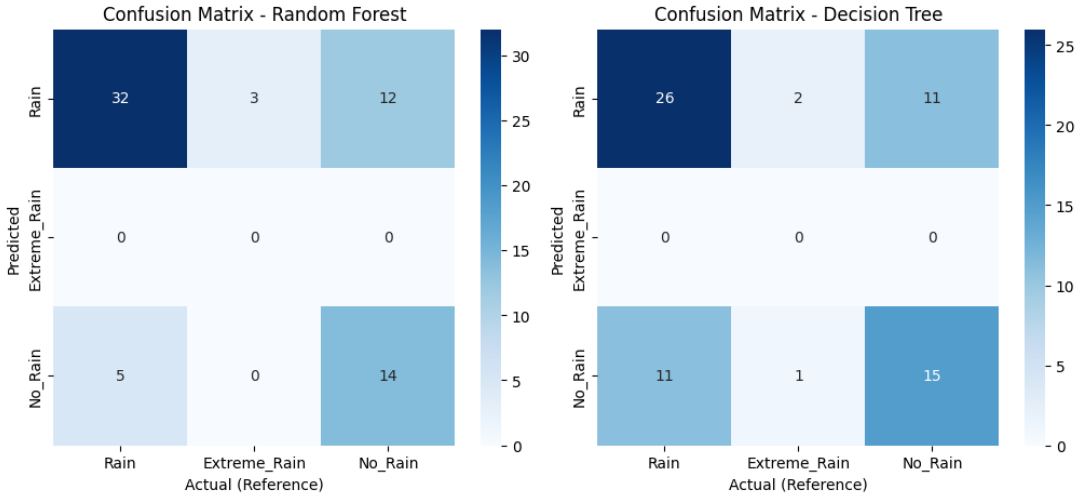

Figure 2. Confusion Matrix of Random Forest and Decision Tree

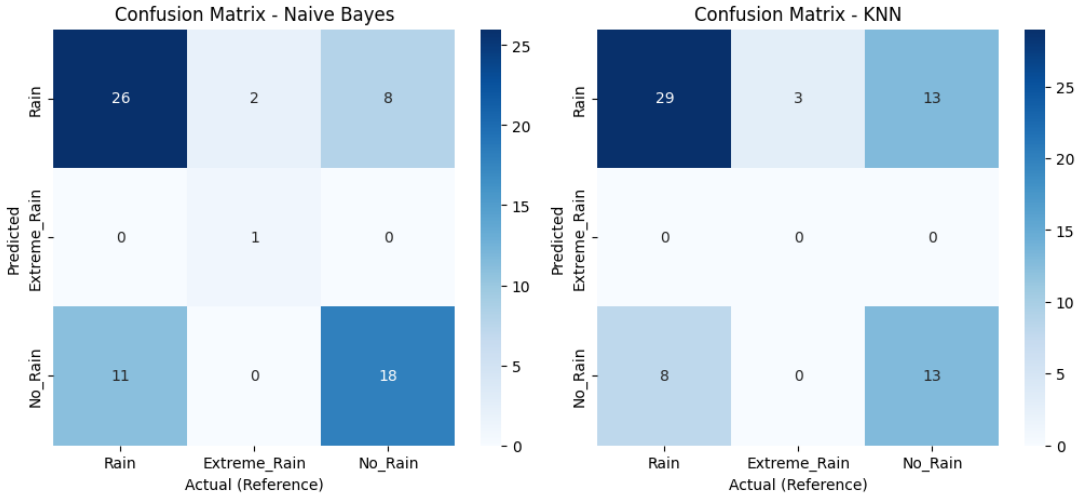

Figure 3. Confusion Matrix of Naïve Bayes and KNN

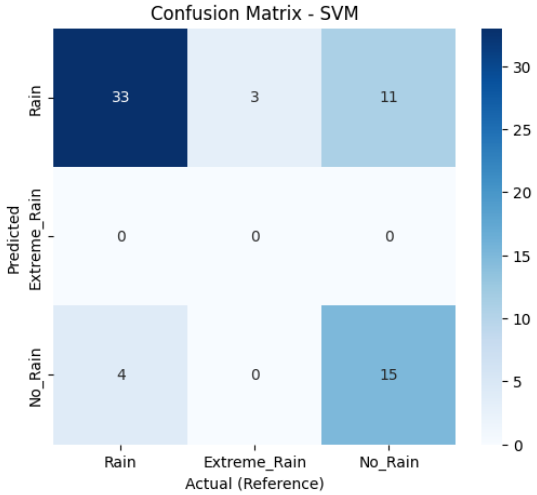

Figure 4. Confusion Matrix of SVM

Table 4. Evaluation Metrics Result

No	Metode	Accuracy	Precision	Recall	F1-Score
1	Random Forest	0.6969697	0.7088466	0.4677755	0.6920635
2	Decision Tree	0.6212121	0.6111111	0.4265419	0.6251241
3	Naïve Bayes	0.6818182	0.7809706	0.5761146	0.6222914
4	KNN	0.6363636	0.631746	0.4279279	0.6302543
5	SVM	0.7272727	0.7458007	0.489605	0.7261905

3.3 Testing Data Evaluation Results

The testing phase was conducted using a confusion matrix to evaluate the ability of each method to classify rainfall categories. The confusion matrix is used to compare the predicted results with the actual classes so that classification error patterns produced by each method can be identified. The confusion matrices for the five methods are shown in Figure 2, Figure 3, and Figure 4.

Based on Figures 2, 3, and 4, most of the evaluated methods still have difficulty recognizing the Extreme Rain category. This is likely influenced by the imbalanced distribution of the dataset, where the Extreme Rain class contains substantially fewer samples than the No Rain and Rain classes. As a result, the models receive limited information to learn the characteristics of this minority class during training, making it more difficult to correctly identify Extreme Rain events during testing. In contrast, the Rain category contains the largest number of samples, causing several models to predict this class more frequently. Consequently, the classification performance for the minority class remains lower than that of the majority classes.

Furthermore, model testing was conducted using testing data to evaluate the ability of each method in classifying new data. The evaluation was performed using a confusion matrix with accuracy, precision, recall, and F1-score metrics. The evaluation results on the testing data are presented in Table 4. Based on Table 4, the Support Vector Machine (SVM) method shows the best performance among the evaluated methods, achieving an accuracy of 72.72% and an F1-score of 72.61%. These results indicate that SVM provides better generalization when classifying unseen rainfall data. In contrast, Random Forest showed a noticeable decrease in accuracy compared to its performance on the training data. This difference may indicate that the model learned patterns that were highly specific to the training data and therefore did not generalize as well to new data. One possible reason is the relatively small dataset combined with the imbalanced distribution of rainfall classes, particularly the limited number of Extreme Rain samples. Under these conditions, Random Forest can produce a model that fits the training data very well while showing lower performance on unseen data. Therefore, model selection should consider not only training accuracy but also testing performance, as testing results provide a better indication of the model's generalization ability. In addition, the recall values of several methods remain relatively low, indicating that some rainfall categories are still difficult to identify correctly.

3.4 Discussions

The experimental results demonstrate that the five machine learning methods exhibit different classification performances when applied to the same rainfall dataset. Among the evaluated methods, SVM achieved the highest testing accuracy and F1-score, indicating better generalization ability than the other models. In contrast, Random Forest produced the highest training accuracy but showed a substantial

decrease in testing accuracy, suggesting that excellent performance on training data does not necessarily translate into better performance on unseen data. These findings highlight the importance of evaluating machine learning models using independent testing data rather than relying solely on training performance.

The superior performance of SVM on the testing data can be attributed to its ability to identify an optimal decision boundary that maximizes the separation margin between classes. This characteristic enables SVM to generalize more effectively when applied to unseen data, particularly for datasets with a relatively limited number of samples. In contrast, Random Forest tends to construct highly detailed decision boundaries by combining multiple decision trees. Although this approach can achieve excellent performance on the training data, it may also capture patterns that are too specific to the training dataset, leading to reduced generalization performance on new data. The performance differences observed in this study suggest that model complexity should be balanced with the characteristics of the available dataset to obtain better classification performance.

Another important finding is that all evaluated models showed relatively poor performance in identifying the Extreme Rain category. This can be largely attributed to the imbalanced distribution of the dataset, where the number of Extreme Rain samples was considerably smaller than the other rainfall categories. As a result, the models had fewer examples from which to learn the characteristics of this minority class, making accurate classification more challenging. This finding suggests that class distribution is an important factor affecting rainfall classification performance. Future studies may consider applying data balancing techniques or collecting additional Extreme Rain observations to improve the classification performance for minority classes.

Compared with previous studies, the findings of this research provide a broader evaluation of machine learning methods for rainfall classification. The study by [8] reported higher classification accuracy, however, it evaluated only Naïve Bayes and C4.5, making it difficult to assess the relative performance of other machine learning algorithms. Meanwhile, the study in [9] focused on binary rainfall classification, which represents a less complex classification problem than the three-class classification adopted in this study. In addition, the study in [13] used only four meteorological variables, whereas this research employed seven meteorological attributes to represent rainfall conditions more comprehensively. Although the testing accuracy obtained in this study is lower than that reported in some previous studies, the results provide a more comprehensive comparison of five widely used machine learning methods under the same experimental setting. This allows a more objective evaluation of the strengths and limitations of each method for multiclass rainfall classification in Palembang.

This study has several limitations that should be considered when interpreting the results. First, the dataset was collected from a single meteorological station in Palembang and covered only one year of observations, which may limit the generalizability of the findings to other regions or longer observation periods. Second, the imbalanced distribution of rainfall classes, particularly the limited number of Extreme Rain samples, affected the classification performance for the minority class. Despite these limitations, this study addresses the research gap identified in previous studies by providing a comparative evaluation of five machine learning methods using the same dataset, seven meteorological variables, and multiclass rainfall categories. Future research may include larger datasets collected over multiple years, additional meteorological variables, and data balancing techniques to further improve rainfall classification performance.

4. Conclusion

Based on the evaluation results on the testing data, the Support Vector Machine (SVM) method achieved the best performance with an accuracy of 72.72% and an F1-score of 72.61%. Meanwhile, the Random Forest method showed very high performance on the training data but experienced a decline on the testing data, indicating the presence of overfitting. In addition, the imbalanced data distribution also affected the model's ability to recognize all rainfall categories, particularly the extreme rainfall class. These results show that the choice of method and data characteristics significantly influence the performance of rainfall classification using machine learning. Future research may address data imbalance issues by applying techniques such as oversampling or SMOTE to improve classification performance on minority classes.

References

[1] Z. Sembiring, D. Silvani, W. M Habiby, I. A Lizahra, and U. Azmi, "Faktor Penyebab Terjadinya Banjir," *Didaktik: Jurnal Ilmiah PGSD STKIP Subang*, vol. 11, no. 02, pp. 366–371, 2025, doi: 10.36989/didaktik.v11i02.7352.

[2] D. P. Diyani, E. R. Sriwijayanti, and R. Gustrianysah, "Prediksi Curah Hujan Berbasis Regreasi PolinomialMenggunakan Data Historis Cuaca," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 14, no. 2, pp. 130–137, 2025, doi: 10.34010/komputa.v14i2.16331.
[3] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.
[4] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
[5] A. P. D. Silva, "Optimization approaches to Supervised Classification," *European Journal of Operational Research*, vol. 261, no. 2, pp. 772–788, Sep. 2017, doi: 10.1016/j.ejor.2017.02.020.
[6] A. Ridwan, "Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, 2020, doi: 10.47970/siskom-kb.v4i1.169.
[7] U. N. Dulhare, "Prediction system for heart disease using Naive Bayes and particle swarm optimization," *Biomedical Research (India)*, vol. 29, no. 12, pp. 2646–2649, 2018, doi: 10.4066/biomedicalresearch.29-18-620.
[8] M. W. Martadiansyah, A. Ghufron, R. A. Hidayah, D. Salzabila, and L. Amanda, "Perbandingan Algoritma C4.5 dengan Naive Bayes Untuk Klasifikasi Curah Hujan," *Jurnal Komputer Antartika*, vol. 3, no. 1, pp. 8–17, 2025, doi: 10.70052/jka.v3i1.648.
[9] A. R. I. Pratama, S. A. Latipah, and B. N. Sari, "Optimasi Klasifikasi Curah Hujan Menggunakan Support Vector Machine (Svm) Dan Recursive Feature Elimination (Rfe)," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 2, pp. 314–324, 2022, doi: 10.29100/jipi.v7i2.2675.
[10] M. S. Sawah, "A novel feature selection-driven framework for sustainable and efficient rainfall classification using machine learning models," *Unconventional Resources*, vol. 9, p. 100286, 2026, doi: 10.1016/J.UNCRES.2025.100286.
[11] B. K. Seela, J. Janapati, P. L. Lin, E. Joseph, S. Vadloori, and C. C. Tu, "Improving the rainfall rate and kinetic energy estimation in Taiwan using machine learning approaches," *Atmospheric Research*, vol. 342, p. 109129, 2026, doi: 10.1016/J.ATMOSRES.2026.109129.
[12] F. M. Chiu, L. J. Y. Liu, U. Hasanah, and C. M. Liu, "Applying machine learning for precipitation forecasting in uneven rainfall regions of Taiwan," *Journal of Hydrology: Regional Studies*, vol. 64, p. 103243, 2026, doi: 10.1016/J.EJRH.2026.103243.
[13] N. A. Prakoso Indaryono, "Analisa Perbandingan Algoritma Random Forest Dan Naïve Bayes Untuk Klasifikasi Curah Hujan Berdasarkan Iklim Di Indonesia," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 1, pp. 158–167, 2024, doi: 10.29100/jipi.v9i1.4421.
[14] D. T. Utari, G. R. P. Palage, F. Fadhlirobby, and A. B. Nuswantoro, "Comparative Analysis Of Machine Learning Models For Rainfall Classification in Yogyakarta," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 19, no. 4, pp. 2765–2776, 2025, doi: 10.30598/barekengvol19iss4pp2765-2776.
[15] C. G. L. Pringandana and K. Kusnawi, "A Comparative Analysis of Hyperparameter-Tuned XGBoost and LightGBM for Multiclass Rainfall Classification in Jakarta," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 4, pp. 2467–2483, 2025, doi: 10.52436/1.jutif.2025.6.4.4965.
[16] E. Alshdaifat, D. Alshdaifat, A. Alsarhan, F. Hussein, and S. M. F. S. El-Salhi, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data*, vol. 6, no. 2, pp. 11–33, 2021, doi: 10.3390/data6020011.
[17] N. N. A. Aryanti and O. Suria, "Analisis Sentimen Terhadap Pemutusan Hubungan Kerja Di Indonesia : Komparasi Indobert Dengan Svm, Random Forest, Dan Decision Tree Dengan Optimasi Tf - Idf," *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 10, no. 2, pp. 1158–1176, 2025, doi: 10.36341/rabit.v10i2.6364.
[18] P. P. Allorerung, A. Erna, M. Bagussahrir, and S. Alam, "Analisis Performa Normalisasi Data untuk Klasifikasi K-Nearest Neighbor pada Dataset Penyakit," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 9, no. 3, pp. 178–191, 2024, doi: 10.14421/jiska.2024.9.3.178-191.
[19] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[20] J. S. Rhodes, A. Cutler, and K. R. Moon, "Geometry- and Accuracy-Preserving Random Forest Proximities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10947–10959, 2023, doi: 10.1109/TPAMI.2023.3263774.
[21] I. Sutoyo, "Implementasi Algoritma Decision Tree untuk Klasifikasi Serangan Jantung," *Jurnal Sistem Informasi dan Ilmu Komputer*, vol. 1, no. 4, pp. 207–212, 2023, doi: 10.59581/jusiik-

widyakarya.v1i4.1895.
[22] A. Afdhaluzzikri, H. Mawengkang, and O. S. Sitompul, "Perfomance analysis of Naive Bayes method with data weighting," *SinkrOn*, vol. 7, no. 3, pp. 817–821, 2022, doi: 10.33395/sinkron.v7i3.11516.
[23] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*, 2014, pp. 372–378, doi: 10.1109/SAI.2014.6918213.
[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.
[25] Y. Guo, X. Yin, X. Zhao, D. Yang, and Y. Bai, "Hyperspectral image classification with SVM and guided filter," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 56–65, 2019, doi: 10.1186/s13638-019-1346-z.
[26] G. B. Firmanesha, S. S. Prasetyowati, and Y. Sibaroni, "Detecting Hoax News Regarding the Covid-19 Vaccine using Levenshtein Distance," *Jurnal Bumigora Information Technology (BITe)*, vol. 4, no. 2, pp. 133–142, 2023, doi: 10.30812/bite.v4i2.2023.
[27] L. M. D. Guswandi, M. Yanto, M. Hafidz, "Analisis Hybrid Decision Support System dalam Penentuan Status Kelulusan Mahasiswa," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 6, pp. 1127–1136, 2021, doi: 10.29207/resti.v5i6.3587.
[28] P. Subarkah, E. Pambudi, and S. O. N. Hidayah, "Perbandingan Metode Klasifikasi Data Mining untuk Nasabah Bank Telemarketing," *Matrik : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 1, pp. 139–148, 2020, doi: 10.30812/matrik.v20i1.826.
[29] S. D. Prasetyo, S. S. Hilabi, and F. Nurapriani, "Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN," *Jurnal KomtekInfo*, vol. 10, pp. 1–7, 2023, doi: 10.35134/komtekinfo.v10i1.330.
[30] A. F Nugraha, R. F. A. Aziza, and Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *Jurnal Infomedia*, vol. 7, no. 1, pp. 39–44, 2022, doi: 10.30811/jim.v7i1.2959.