G-Tech : Jurnal Teknologi Terapan
Vol. 9, No. 2, 2025, p. 715-727
E-ISSN: 2623-064X | P-ISSN: 2580-8737

Comparison of Machine Learning Classification Methods for Weather Prediction: A Performance Analysis

Zakha Maisat Eka Darmawan[1]✉, Ashafidz Fauzan Dianta[2], Kholid Fathoni[3], Oktavia Citra Resmi Rachmawati[4], Kevin Ilham Apriandy[5]

[1, 2, 3] Departemen Teknologi Multimedia Kreatif, Politeknik Elektronika Negeri Surabaya, Indonesia
[4, 5] Teknologi Komputer, Politeknik Internasional Tamansiswa Mojokerto, Indonesia

✉ **Corresponding Author :** Zakha Maisat Eka Darmawan (e-mail: zakha@pens.ac.id)

Article Information

Article History

Received : February 18, 2025
Revised : March 09, 2025
Published : April 04, 2025

***Keywords*:**

Machine Learning;
Data Analysis;
Weather;
Hyperparameter Tuning;
Classification.

ABSTRACT

Weather classification is crucial in various sectors, including agriculture, transportation, and disaster management. Accurate weather prediction can help mitigate risks and improve decision-making in these fields. However, classifying weather conditions remains challenging due to the complex and dynamic nature of meteorological data. This study aims to compare different machine learning classification methods to determine the most effective model for weather classification. The research employs a structured methodology consisting of seven key steps: literature study, data understanding, exploratory data analysis, data preparation, modeling, evaluation, and hyperparameter tuning. The study used Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, Gradient Boosting, AdaBoost, and Extra Trees to identify the best-performing classifier. Model evaluation was conducted using accuracy, precision, recall, and F1-score. The results indicate that Gradient Boosting achieved the highest performance, surpassing other models with an accuracy of 90.15%. To optimize the model further, hyperparameter tuning was conducted using GridSearchCV, and feature selection was done using SelectKBest. This process resulted in an improved accuracy of 90.22%, demonstrating the effectiveness of model optimization.

INTRODUCTION

Weather plays an important role in daily life, affecting various aspects such as agriculture, transportation, and disaster planning (Bolan et al., 2023). Accurate weather prediction is urgently needed to reduce the risks of erratic weather conditions (Schultz et al., 2021). Errors in weather forecasting can have significant impacts, ranging from failures in the agricultural sector due to faulty planning to financial losses incurred in the transportation and disaster mitigation sectors (Price et al., 2024). In recent decades, advances in information and computing technology have opened up new opportunities to improve the accuracy of weather prediction (Markovics & Mayer, 2022). One of the growing approaches is using machine learning methods in weather-type classification (Bochenek & Ustrnul, 2022).

Although machine learning has been widely used in weather prediction, various challenges remain in improving classification accuracy and efficiency. Traditional statistical methods, such as multiple linear regression and time series analysis, often struggle with handling non-linear relationships in meteorological data, leading to suboptimal performance (Ahmed et al., 2021). Furthermore, classical machine learning models, such as Decision Tree and Naïve Bayes, tend to

This is an open access article under the CC BY license

be sensitive to noisy data and imbalanced class distributions, reducing their generalization ability in real-world applications (Chen et al., 2020). Several studies have attempted to address these issues by incorporating feature selection techniques and ensemble learning, but challenges related to data preprocessing and model optimization persist (Meenal et al., 2021).

One of the main challenges in weather prediction is how to process data effectively to integrate it with prediction models, thereby improving forecast accuracy. A prediction system with high performance, targeting an accuracy of at least 85%, is essential to minimize forecasting errors and reduce unnecessary anticipation costs. Therefore, this research focuses on comparing various classification methods in weather prediction to identify the most effective algorithm in terms of accuracy and performance. Prior studies have explored various models, including Decision Tree, Naïve Bayes, and Support Vector Machine, to improve prediction accuracy. However, recent developments favor more complex models, such as Random Forest and ensemble learning approaches, which can handle large datasets and yield higher accuracy (Antoniadis et al., 2020).

One of the relevant previous studies is the research conducted by Hamami and Dahlan on weather classification in DKI Jakarta Province using a Random Forest algorithm with an oversampling technique. This study classified weather data into eight categories based on attributes such as temperature, humidity, location, and time. They used the SMOTE technique to overcome class imbalance in the dataset, which successfully improved prediction accuracy, especially for minority weather classes. The results showed that using Random Forest with SMOTE resulted in an accuracy of 70%, higher than without the oversampling technique. This research confirms that combining machine learning algorithms with data balancing techniques can improve model performance in weather classification, contributing to a more accurate and reliable weather prediction system (Hamami & Dahlan, 2022). However, their study did not explore hyperparameter tuning extensively, which could have further optimized the model's performance.

The main objective of this research is to develop a weather-type prediction system with an accuracy of at least 85% to assist in making more accurate decisions in various sectors affected by weather conditions. With a better prediction system, it is expected that more appropriate and effective anticipatory measures can be taken, thereby reducing negative environmental and economic impacts. To achieve this goal, this research proposes a solution by comparing seven classification algorithms presented in the form of an evaluation table based on performance metrics. From the comparison results, the algorithm with the best performance will be selected for hyperparameter tuning to improve its accuracy further. With this approach, a more optimized weather classification method is expected to be obtained and applied in a more reliable weather prediction system.

The benefits of this research are expected to be felt in various aspects. From an academic perspective, this research can be a reference for developing a more accurate and efficient machine learning-based weather prediction model. From a practical perspective, the results of this research can be used by relevant agencies, such as meteorological agencies, transportation companies, and the agricultural sector, to improve the accuracy of weather predictions to reduce the risks and costs associated with forecast errors. In addition, with a more accurate prediction system, the general public can be better prepared for extreme weather conditions, thereby reducing the negative impacts on daily life. Thus, this research is expected to make a real contribution to the development of more thoughtful and more adaptive weather prediction technology in the future.

RESEARCH METHODS

In this research, the method used consists of seven main steps, namely literature study, data understanding, exploratory data analysis, data preparation, modeling, evaluation, and hyperparameter tuning. Each step is carried out systematically to ensure optimal performance in the resulting weather classification model. This approach aims to understand the data characteristics better, select the most suitable model, and improve prediction accuracy with proper parameter optimization.

Figure 1. Research Methodology

Literature Study

The first step in this research is to conduct a literature study to understand the latest developments in weather classification using machine learning methods. The literature study was conducted by reviewing various previous studies that have implemented classification algorithms such as Decision Trees, Random Forests, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gradient Boosting, AdaBoost, and Extra Trees. In addition, this research also examines techniques for handling imbalanced data, such as oversampling and undersampling methods, as well as evaluation metrics used to assess model performance (Rachmawati et al., 2024).

Data Understanding

After gaining insights from the literature study, the next step is understanding the dataset's characteristics used in this research. The weather dataset was collected from a trusted source and contains various features such as temperature, humidity, air pressure, rainfall, and weather conditions, categorized into several classes. The data is understood by evaluating the number of samples in each class, the distribution of values in each feature, and potential problems such as missing data or data duplication (Rachmawati & Darmawan, 2024). A good understanding of the dataset is essential for optimizing the next steps (Popenoe et al., 2021).

Exploratory Data Analysis

Exploratory data analysis is performed to understand patterns in the dataset and identify relationships between variables (Javed et al., 2021). EDA is conducted in two main approaches, namely univariate and multivariate analysis. Univariate analysis aims to evaluate the distribution of each feature individually (Peng et al., 2021), for example, by using histograms to look at the temperature distribution or boxplots to detect outliers in humidity data. Meanwhile, using scatter plots or correlation heatmaps, multivariate analysis is used to understand the correlation between variables (Feld et al., 2022), such as the relationship between temperature and humidity. By conducting EDA, the understanding of the data becomes more profound, which can help in the data preparation and modeling stages.

Data Preparation

Before a model can be built, the data needs to be prepared appropriately. This stage involves several steps, such as handling missing data by imputing values, normalizing or standardizing numerical features, and encoding categorical variables so machine learning models can use them. In addition, if the dataset has a class imbalance, data balancing techniques are performed, such as oversampling using Synthetic Minority Over-sampling Technique (SMOTE) or undersampling the majority class. The processed data is then divided into training and testing data to evaluate the model's performance objectively.

Modelling

The next stage is modeling, where various machine-learning algorithms are applied to build weather classification models. The models used in this research include Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gradient Boosting, AdaBoost, and Extra Trees. Decision Tree divides data based on the most informative features (Charbuty & Abdulazeez, 2021), while Random Forest improves accuracy and reduces overfitting by combining multiple decision trees (Pal & Parija, 2021). SVM searches for the optimal hyperplane to separate classes in a high-dimensional space (Abdullah & Abdulazeez, 2021), while KNN classifies data based on proximity to data points that have already been labeled (Uddin et al., 2022). Gradient Boosting and AdaBoost are boosting methods that build more muscular models by correcting previous model errors (Demir & Sahin, 2022). At the same time, Extra Trees

are similar to Random Forest but use more random feature selection techniques to improve generalization (Sharma et al., 2022).

Evaluation

After the model is built, the model is evaluated using several key metrics: accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correct predictions compared to all predictions, precision indicates how many positive predictions are correct, recall measures the model's ability to detect positive classes overall, and F1-score balances precision and recall (Fatihia et al., 2023). By comparing the performance of each model using these metrics, the most optimal algorithm for weather classification can be identified (Fatihia et al., 2024).

$$Accuracy = \frac{True\ Positive+True\ Negative}{True\ Positive+True\ Negative+False\ Positive+False\ Negative} \quad (1)$$

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (2)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (3)$$

$$F1\ Score = 2 \times \left(\frac{precision \times recall}{precision+recall}\right) \quad (4)$$

Hyperparameter Tuning

Once the model has been evaluated, the final step is to fine-tune the model performance through hyperparameter tuning (Darmawan & Dianta, 2023). The techniques used in this research are Grid Search and Random Search, which aim to find the best combination of parameters for each model. With optimal hyperparameter tuning, the model can provide more accurate and reliable results in weather classification (Belete & Huchaiah, 2021). By following this seven-step methodology, this research is expected to produce a weather classification model that is accurate and reliable to be applied in a broader weather prediction system.

RESULTS AND DISCUSSION

In the data understanding stage, understanding the dataset's characteristics is very important to ensure the feasibility of the data in the analysis and modeling process. The dataset used in this study consists of various weather variables that can be categorized as numerical and categorical. These variables include information on temperature, humidity, wind speed, rainfall, cloud conditions, atmospheric pressure, UV index, season, visibility, location type, and weather type. Each variable has an important role in classifying weather conditions that are the focus of this research.

Table 1. Data Understanding

Column Name	Type	Variable	Description
Temperature	Continous Numerical	Dependent	Temperature in degrees Celsius
Humidity	Discrete Numerical	Dependent	Humidity percentage
Wind Speed	Continous Numerical	Dependent	Wind speed in kilometers per hour
Precipitation	Discrete Numerical	Dependent	Rainfall percentage
Cloud Cover	Ordinal Categorical	Dependent	Cloud conditions
Atmospheric Pressure	Continous Numerical	Dependent	Atmospheric pressure in hPa
UV Index	Discrete Numerical	Dependent	Ultraviolet radiation power
Season	Nominal Categorical	Dependent	Seasonal conditions
Visibility	Continous Numerical	Dependent	The maximum distance at which an object can be seen and recognized with the naked eye
Location	Ordinal Categorical	Dependent	Location Type
Weather Type	Nominal Categorical	Independent	Weather Condition

The numerical variables in the dataset are divided into two types, namely continuous numerical and discrete numerical. Continuous numeric variables include Temperature, Wind Speed, Atmospheric Pressure, and Visibility. These variables have values that can fall within a specific range without round number restrictions so that they can have many possible values. On the other hand, discrete numerical variables in this dataset include Humidity, Precipitation, and UV Index, which have integer values within a specific range. In addition to numerical variables, this dataset contains several categorical variables that play a role in weather classification. Cloud Cover and Location variables are categorized as ordinal, as their values have a certain level or order.

This research uses Weather Type as an independent variable (predictor), while the other variables act as dependent (response). In other words, this research aims to understand how various meteorological factors such as temperature, humidity, air pressure, and other environmental factors affect weather conditions classified by Weather Type. Therefore, a deep understanding of each variable in the dataset is an essential first step before conducting further analysis, such as Exploratory Data Analysis (EDA) and modeling using various classification methods.

In this weather data analysis, the 1.5×Interquartile Range (IQR) Rule method is used to identify the presence of outliers in numerical variables. This method works by determining the interquartile range (IQR), which is the difference between the third quartile (Q3) and the first quartile (Q1). Based on the formula, any data that falls outside the range between the lower and upper bounds will be considered outliers.

$$Q_1 = \frac{1}{4}(n+1)^{th}\ data \tag{5}$$

$$Q_3 = \frac{3}{4}(n+1)^{th}\ data \tag{6}$$

$$IQR = Q_3 - Q_1 \tag{7}$$

$$Lower\ bound = Q_1 - (1.5 \times IQR) \tag{8}$$

$$Upper\ bound = Q_3 + (1.5 \times IQR) \tag{9}$$

Calculation results using the 1.5×IQR Rule method and visualization through violin plots revealed that four variables in this dataset contain outliers: Atmospheric Pressure, Wind Speed, Temperature, and Visibility. The presence of outliers in these variables can be seen from the distribution of data with a long tail and extreme points outside the predetermined interquartile range.

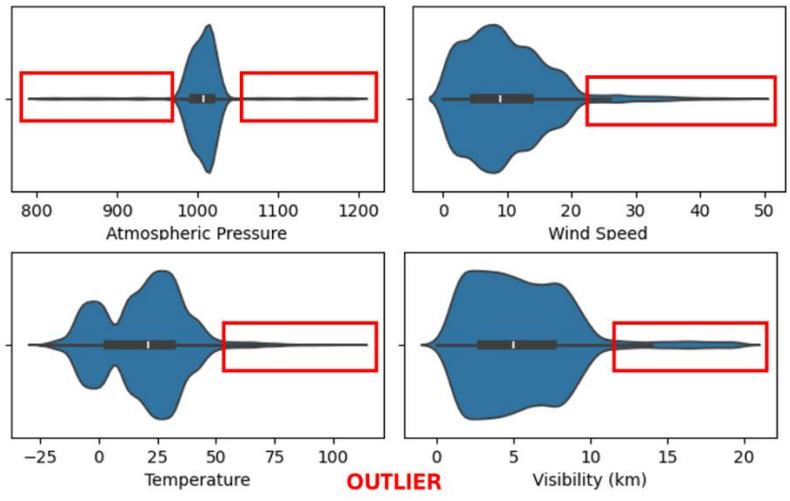

Figure 2. Violin Plot of Outlier Data

Based on the graphs above, Atmospheric Pressure has some outlier values on both sides of the distribution, indicating atmospheric pressure is much lower or higher than its normal range. This could indicate unusual weather conditions or measurement errors. Furthermore, Wind Speed shows many extreme values, which means that some events with wind speeds are much greater than normal, possibly caused by storms or certain meteorological phenomena. In addition, Temperature also has some outliers in the very high-temperature range, which could indicate the presence of extreme temperature conditions at some points in time in the dataset. Meanwhile, Visibility shows an outlier on the right side of the distribution, indicating very high visibility compared to normal conditions. Outliers in the dataset can affect the analysis and modeling results, especially when using algorithms sensitive to extreme values such as k-Nearest Neighbors (KNN) and Support Vector Machine (SVM). Therefore, it is necessary to handle outliers before the modeling process is carried out, either by data transformation techniques, winsorizing, or deleting outlier data if it does not represent the actual conditions.

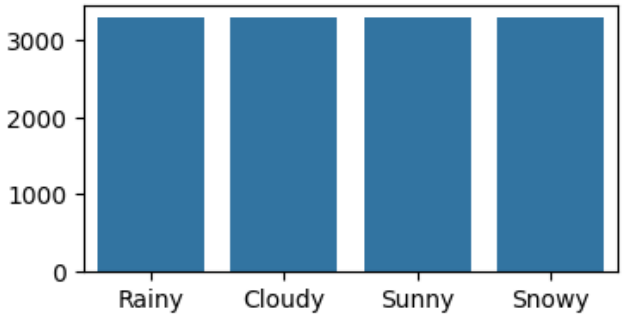

Figure 2. Bar Chart of Independent Variable

This research uses the Weather Type variable as the target or dependent variable in the weather classification process. One important aspect of data processing is using a balanced dataset, which ensures that the number of samples for each weather category is relatively balanced. This is crucial to prevent overfitting to the majority class, where the model tends to be more accurate in predicting the most frequently occurring classes but fails to recognize the less frequent classes. With a balanced dataset, the machine learning model can learn equally from all weather categories, improving its ability to perform a more accurate classification of minority classes. This strategy also helps improve evaluation metrics such as precision, recall, and F1-score, key indicators in assessing overall model performance.

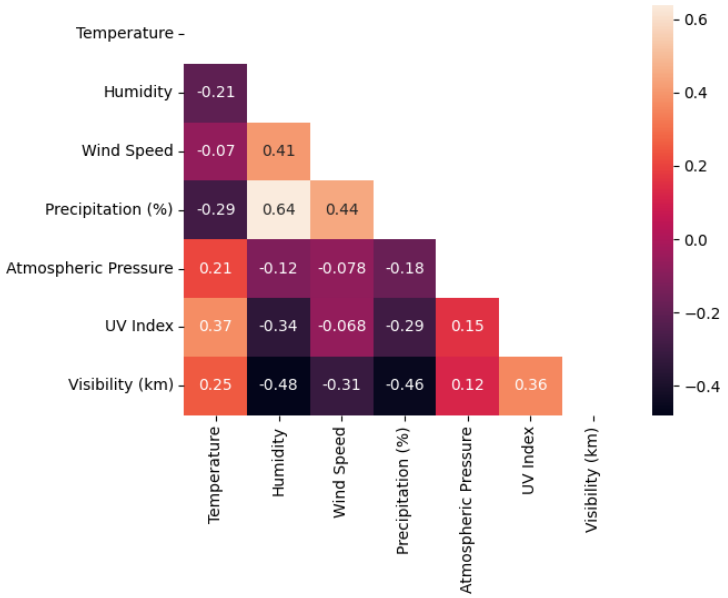

Figure 3. Correlation Heatmap

Based on the correlation heatmap aboce, it can be seen that the relationship between variables in this dataset shows a diverse correlation pattern. Humidity and Precipitation have a positive correlation with a score of 0.638631, indicating that precipitation also tends to increase as humidity increases. This is due to the meteorological concept where more humid air often leads to a higher probability of rain. Meanwhile, Wind Speed and UV Index show no correlation with a score of -0.068147, which means that wind speed does not have a significant linear relationship with ultraviolet radiation intensity, possibly because other factors, such as atmospheric conditions and cloud cover, are more influential in determining UV Index values. Additionally, Humidity and Visibility (km) have a negative correlation with a score of -0.479969, indicating that visibility tends to decrease as humidity increases, which also makes sense as more humid air is often associated with fog or atmospheric conditions that reduce visibility. This correlation helps understand the relationship between variables and can be used as a basis for further predictive model development analysis.

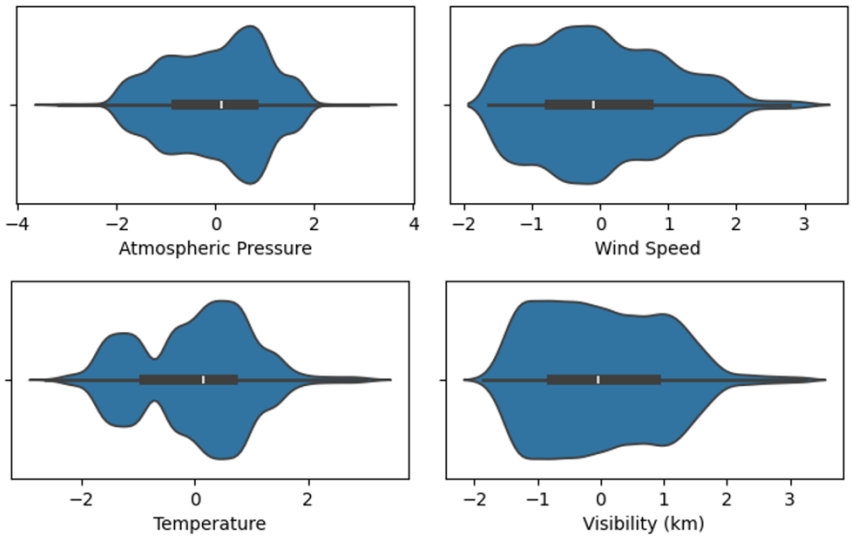

Figure 4. Violin Plot of Clean Data

To handle outliers in the Atmospheric Pressure, Wind Speed, Temperature, and Visibility columns, a K-Nearest Neighbors (KNN) based imputation method is used. This method replaces outlier values with more representative values based on proximity to other data in the feature space. In this case, each outlier value will be replaced with the average of k nearest neighbors with similar values, thus maintaining the data distribution pattern and reducing the impact of extremes that can affect the performance of machine learning models. KNN imputation is superior to simple methods such as mean or median because it considers the relationship between features in the dataset. Hence, the imputation results are more accurate and do not significantly alter the data's original structure. This approach improves data quality as the distribution is preserved while problems caused by outliers are minimized.

In the data pre-processing process, categorical variables are encoded so they can be used in machine learning models. The Location and Cloud Cover columns use Label Encoding because the values in these columns have an order or level that needs to be maintained, such as Location, which indicates an area's elevation level, and Cloud Cover, which represents the level of cloud density in the sky. Label Encoding converts the categories in these columns into numerical values that reflect that hierarchy. Meanwhile, the Season column uses One-Hot Encoding, as its categories have no specific order. Seasons such as Winter, Spring, Summer, and Fall are nominal and do not have a hierarchical relationship, so a one-hot representation is more appropriate to avoid assuming an incorrect ordinal relationship. The model can more easily understand and process categorical data by applying appropriate encoding techniques without losing important information.

In the data pre-processing stage, normalization is performed using Standard Scaler to homogenize the range of values for each variable in the dataset. Standard Scaler converts each value in a column into a distribution with a mean of zero and a standard deviation of one so that the values in each variable have a comparable scale. This normalization is essential for machine learning algorithms sensitive to differences in data scale, such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). With a Standard Scaler, variables with an extensive range of values, such as Atmospheric Pressure or Wind Speed, will not dominate other variables with a smaller range, so the model can perform the learning process more effectively and produce more accurate predictions.

Table 2. Data Scaling

Column Name	Mean	
	Before Scaling	After Scaling
Temperature	19.127576	-5.167584e-17
Humidity	68.710833	3.445056e-16
Wind Speed	9.832197	3.229740e-18
Precipitation	53.644394	1.098112e-16
Atmospheric Pressure	1005.827896	9.689219e-14
UV Index	4.005758	1.315634e-18
Visibility	5.462917	4.844610e-16

Splitting the data into two types using the Train-Test Split technique is an important step in the machine-learning modeling process to ensure the model can be evaluated objectively. In this case, the data was split into two parts: 90% for training data, which amounts to 11,880 rows, and 10% for testing data, which amounts to 1,320 rows. The training data is used to train the model to learn patterns from the data. In contrast, the test data is used to measure the model's performance after training, ensuring that the model can generalize well to data it has never seen before. This division helps avoid overfitting and ensures a more accurate evaluation of the model's capabilities.

Table 3. Data Splitting

Data	Persentase	Amount
Train	90%	11880
Test	10%	1320

After going through the data preprocessing stage, the next step is to choose the best method to predict the weather with optimal accuracy. In this research, experiments are conducted using seven machine learning methods, namely Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gradient Boosting, AdaBoost, and Extra Trees. Each method will be tested for performance based on various evaluation metrics to determine the most effective algorithm for classifying weather types. By doing this comparison, it is expected to obtain a model that is accurate and can generalize well to new data.

Table 4. Modelling

Method	Advantage	Disadvantage	Parameter
Decision Tree	Able to handle non-linear relationships between features and targets	Structures become complex and slow to build if handling big data	criterion='gini', max_depth=None, min_samples_split=2
Random Forest	Reduces variance and improves model generalization by combining	Requires a lot of memory because it generates many decision trees	n_estimators=100, max_depth=None, max_features='sqrt'

	predictions from multiple trees		
Support Vector Machine	Has flexibility in handling complex data due to the use of kernel tricks	Less optimal on unbalanced datasets as it tends to focus on the margin that maximizes the majority class	C=1.0, kernel='rbf', gamma='scale'
K Nearest Neighbors	Able to handle multiclass data without the need for special modifications	easily affected by noisy data and outliers that can reduce the accuracy of the model	n_neighbors=5, metric='minkowski', weights='uniform'
Gradient Boosting	Generate feature importance estimates during the learning period	Sensitive to uncorrelated features	max_features=None, learning_rate=0.1, loss='log_loss', max_depth=3
Ada Boost	Easily adapt to new and changing data over time due to its iterative nature	Sensitive to noise and outliers because it affects the assignment of inappropriate weights in the next iteration.	learning_rate=0.1, n_estimators=50, estimator=None
Extra Trees	Faster in training because it splits nodes based on randomly selected split points without searching for the optimal split.	Suboptimal on unbalanced datasets without additional adjustments	n_estimators=100, max_features='sqrt', max_depth=None

In the modeling stage, the process of finding the superior machine learning algorithm in predicting the weather type in the dataset is carried out. Looping is done using various machine learning algorithms with default parameters to achieve this. The algorithms tested include Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gradient Boosting, AdaBoost, and Extra Trees. Each model is evaluated based on performance metrics such as accuracy, precision, recall, and F1-score. With this approach, the algorithm that provides the best results can be identified before further optimization through hyperparameter tuning techniques.

Table 5. Evaluation

Method	Accuracy	Precision	Recall	F1 Score
Decision Tree	89,16%	89,16%	89,16%	89,16%
Random Forest	89,84%	89,84%	89,84%	89,84%
Support Vector Machine	89,77%	89,77%	89,77%	89,77%
K Nearest Neighbors	88,18%	88,18%	88,18%	88,18%
Gradient Boosting	90,15%	90,15%	90,15%	90,15%
Ada Boost	88,33%	88,33%	88,33%	88,33%
Extra Trees	90,08%	90,08%	90,08%	90,08%

Based on the model evaluation results, the performance of each algorithm is measured using accuracy, precision, recall, and F1-score metrics. Of the seven methods tested, Gradient Boosting showed superior performance with an accuracy of 90.22%, followed by Extra Trees, which obtained an accuracy of 90.17%. Meanwhile, methods such as Decision Tree, Random Forest, and Support Vector Machine (SVM) had quite competitive accuracy, ranging from 89.16% to 89.84%. On the other hand, K-Nearest Neighbors (KNN) and AdaBoost showed slightly lower performance with accuracy below 89%. Gradient Boosting achieved an optimal balance between model complexity and generalization to data, resulting in more accurate predictions than other

methods. This suggests that the boosting technique in Gradient Boosting effectively improves model performance by iteratively correcting prediction errors.

After determining the best-performing machine learning algorithm, the next step is performing hyperparameter tuning using GridSearchCV to find the optimal combination of parameters to improve model accuracy. In addition, feature selection is performed using SelectKBest to identify features with the most significant contribution to the prediction process. In this way, the model is optimized not only in terms of parameters but also in terms of more relevant input data. This tuning and feature selection process is applied simultaneously using Pipeline so all stages can be carried out in one structured and efficient flow. With this approach, the resulting model is expected to be more accurate, efficient, and better generalize the data.

Table 6. Hyperparameter Tuning

Parameter	Values	Module
k	10, 11, 12, 13	SelectKBest
learning_rate	1,0; 0,1	GradientBoostingClassifier
criterion	'friedman_mse', 'squared_error'	GradientBoostingClassifier
max_features	'sqrt', 'log2'	GradientBoostingClassifier
loss	'log_loss', 'exponential'	GradientBoostingClassifier
max_depth	3, 4	GradientBoostingClassifier

Gradient Boosting proved to have the most superior performance compared to other machine learning methods in the experiments conducted. Therefore, efforts to improve model performance through hyperparameter tuning focus on the Gradient Boosting algorithm to achieve more optimal weather prediction accuracy. To improve the model's performance, various parameters were combined using the GridSearchCV technique, which allows the exploration of various configurations to find the best combination. Some key parameters tested in this tuning include the number of best features (k) in the SelectKBest method, with values ranging from 10 to 13 to determine the most influential features. In addition, the core parameters of the GradientBoostingClassifier were also optimized, including the learning_rate, which was tested with values of 1.0 and 0.1, and the criterion which was compared between 'friedman_mse' and 'squared_error' to determine the best separation criterion. Furthermore, the max_features parameter was varied between 'sqrt' and 'log2', which controls the number of features each decision tree uses in boosting. In addition, the loss function parameters, namely 'log_loss' and 'exponential', were tested to determine the most optimal loss function in model learning. Finally, max_depth, which controls the maximum depth of the tree in boosting, was tested with values of 3 and 4 to find a balance between model complexity and good generalization. By exploring these parameters, the Gradient Boosting model is expected to achieve the best performance in classifying weather types with higher accuracy.

Table 7. Tuned Model Specification

Parameter	Values	Module
k	12 (Temperature, Humidity, Wind Speed, Precipitation, Cloud Cover, Atmospheric Pressure, UV Index, Visibility, Location, Spring, Summer, Winter)	SelectKBest
learning_rate	0,1	GradientBoostingClassifier
criterion	'squared_error'	GradientBoostingClassifier
max_features	'sqrt'	GradientBoostingClassifier
loss	'log_loss'	GradientBoostingClassifier
max_depth	3	GradientBoostingClassifier

After combining various parameters 288 times in the hyperparameter tuning process using GridSearchCV, the best parameter configuration is obtained that can improve the performance of the Gradient Boosting model in predicting weather types more accurately. From the feature selection results using SelectKBest, the 12 best features were selected, namely Temperature, Humidity, Wind Speed, Precipitation (%), Cloud Cover, Atmospheric Pressure, UV Index, Visibility (km), Location, Spring, Summer, and Winter, indicating that these variables have significant contributions in determining the weather type. In addition, the best parameters for GradientBoostingClassifier were also found, among which was the optimal learning_rate of 0,1, which balances learning speed and model generalization. The best criterion selection falls on 'squared_error', reducing the prediction error in quadratic regression. The best max_features parameter is 'sqrt', which allows the model to select a smaller subset of features on each tree, improving efficiency and reducing the risk of overfitting. For the loss function, the best value obtained was 'log_loss', which worked optimally in the classification scenario. Finally, the optimal max_depth found was 3, which ensures that the model is not too complex and can still generalize well to new data.

Table 8. Tuned Model Performance

Metric	Origin Model	Tuned Model
Accuracy	90,15%	90,22%
Precision	90,15%	90,22%
Recall	90,15%	90,22%
F1-Score	90,15%	90,22%

After hyperparameter tuning of the Gradient Boost model, there was an improvement in performance on various evaluation metrics, albeit on a small scale. Before tuning, the model had an accuracy, precision, recall, and F1-score of 90,15%, while after tuning, the value increased to 90,22%. This increase shows that selecting the optimal parameters through GridSearchCV can improve the model's ability to classify the data accurately. Although the difference is not very large, this tuning still has a positive impact on improving the stability of the model and its ability to deal better with new data. In addition, this process also includes feature selection using SelectKBest, which helps the model focus on the most relevant features for weather prediction, thereby reducing calculation complexity and improving efficiency. With the combination of hyperparameter tuning and feature selection performed in a single pipeline, the resulting model is more optimized, more accurate, and better able to avoid overfitting, allowing it to be used more reliably in various weather conditions.

CONCLUSION

Based on the research results, the solution developed has successfully answered the problem statement: to build a weather prediction model based on available historical data. The resulting model can accurately classify weather types, so it can be used as a tool for understanding future weather patterns. In addition, this research has achieved the expected goals, namely creating a model with more than 85% accuracy. This shows that the method used is quite effective in performing classification.

Furthermore, this research also significantly impacted the selection of the most suitable machine learning model for the categorical prediction task on this dataset. Gradient Boosting proved to provide the best performance after hyperparameter tuning of the various algorithms tested. The parameter tuning and feature selection process using GridSearchCV and SelectKBest improved the model's accuracy, resulting in a more optimized solution. With these results, the research can be a basis for developing more accurate and reliable weather prediction systems.

ACKNOWLEDGEMENTS

The researchers would like to express their most profound appreciation to PENS for their generous financial support, which has been instrumental in completing this research. Their assistance has provided the necessary resources and opportunities to conduct in-depth analysis and experimentation, ultimately contributing to advancing knowledge in weather classification using machine learning. The researchers are sincerely grateful for the institution's continuous encouragement and commitment to academic excellence.

REFERENCES

Abdullah, D. M., & Abdulazeez, A. M. (2021). Machine Learning Applications based on SVM Classification A Review. *Qubahan Academic Journal*, 1(2), 81–90. https://doi.org/10.48161/qaj.v1n2a50

Antoniadis, A., Lambert-Lacroix, S., & Poggi, J. (2020). Random forests for global sensitivity analysis: A selective review. *Reliability Engineering & System Safety*, 206, 107312. https://doi.org/10.1016/j.ress.2020.107312

Belete, D. M., & Huchaiah, M. D. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, 44(9), 875–886. https://doi.org/10.1080/1206212x.2021.1974663

Bochenek, B., & Ustrnul, Z. (2022). Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives. *Atmosphere*, 13(2), 180. https://doi.org/10.3390/atmos13020180

Bolan, S., Padhye, L. P., Jasemizad, T., Govarthanan, M., Karmegam, N., Wijesekara, H., Amarasiri, D., Hou, D., Zhou, P., Biswal, B. K., Balasubramanian, R., Wang, H., Siddique, K. H., Rinklebe, J., Kirkham, M., & Bolan, N. (2023). Impacts of climate change on the fate of contaminants through extreme weather events. *The Science of the Total Environment*, 909, 168388. https://doi.org/10.1016/j.scitotenv.2023.168388

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20–28. https://doi.org/10.38094/jastt20165

Darmawan, Z. M, E., & Dianta, A. F. (2023). Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 13(1), 8–15. https://doi.org/10.26594/teknologi.v13i1.3098

Demir, S., & Sahin, E. K. (2022). An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost. *Neural Computing and Applications*, 35(4), 3173–3190. https://doi.org/10.1007/s00521-022-07856-4

Fatihia, W. M., Fariza, A., & Karlita, T. (2023). CNN with Batch Normalization Adjustment for Offline Hand-written Signature Genuine Verification. *JOIV International Journal on Informatics Visualization*, 7(1), 200. https://doi.org/10.30630/joiv.7.1.1443

Fatihia, W. M., Fariza, A., & Karlita, T. (2024). Convolutional neural network enhancement for mobile application of offline handwritten signature verification. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 22(4), 931. https://doi.org/10.12928/telkomnika.v22i4.25849

Feld, L., Da Silva, V. H., & Strand, J. (2022). Characterization of foamed plastic litter on Danish reference beaches – Pollution assessment and multivariate exploratory analysis. *Marine Pollution Bulletin*, 180, 113774. https://doi.org/10.1016/j.marpolbul.2022.113774

Hamami, F., & Dahlan, I. A. (2022). Klasifikasi Cuaca Provinsi DKI Jakarta menggunakan Algoritma Random Forest dengan Teknik Oversampling. *Jurnal Teknoinfo*, 16(1), 87. https://doi.org/10.33365/jti.v16i1.1533

Javed, U., Ijaz, K., Jawad, M., Ansari, E. A., Shabbir, N., Kütt, L., & Husev, O. (2021). Exploratory Data Analysis Based Short-Term Electrical Load Forecasting: A Comprehensive analysis. *Energies*, 14(17), 5510. https://doi.org/10.3390/en14175510

Markovics, D., & Mayer, M. J. (2022). Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction. *Renewable and Sustainable Energy Reviews*, 161, 112364. https://doi.org/10.1016/j.rser.2022.112364

Meenal, R., Michael, P. A., Pamela, D., & Rajasekaran, E. (2021). Weather prediction using random forest machine learning model. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2), 1208. https://doi.org/10.11591/ijeecs.v22.i2.pp1208-1215

Pal, M., & Parija, S. (2021). Prediction of Heart Diseases using Random Forest. *Journal of Physics Conference Series*, 1817(1), 012009. https://doi.org/10.1088/1742-6596/1817/1/012009

Peng, J., Wu, W., Lockhart, B., Bian, S., Yan, J. N., Xu, L., Chi, Z., Rzeszotarski, J. M., & Wang, J. (2021). DataPrEP.EDA: Task-Centric Exploratory Data Analysis for statistical modeling in Python. *Proceedings of the 2022 International Conference on Management of Data*, 2271–2280. https://doi.org/10.1145/3448016.3457330

Popenoe, R., Langius-Eklöf, A., Stenwall, E., & Jervaeus, A. (2021). A practical guide to data analysis in general literature reviews. *Nordic Journal of Nursing Research*, 41(4), 175–186. https://doi.org/10.1177/2057158521991949

Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T. R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P., Lam, R., & Willson, M. (2024). Probabilistic weather forecasting with machine learning. *Nature*. https://doi.org/10.1038/s41586-024-08252-9

Rachmawati, O. C. R., & Darmawan, Z. M. E. (2024). The comparison of deep learning models for Indonesian political hoax news detection. CommIT (Communication and Information Technology) Journal, 18(2), 123–135. https://doi.org/10.21512/commit.v18i2.10929

Rachmawati, O. C. R., Barakbah, A. R., & Karlita, T. (2024). Programming language selection for the development of deep learning library. *JOIV International Journal on Informatics Visualization*, 8(1), 434. https://doi.org/10.62527/joiv.8.1.2437

Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H., Mozaffari, A., & Stadtler, S. (2021). Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society a Mathematical Physical and Engineering Sciences*, 379(2194), 20200097. https://doi.org/10.1098/rsta.2020.0097

Sharma, D., Kumar, R., & Jain, A. (2022). Breast cancer prediction based on neural networks and extra tree classifier using feature ensemble learning. *Measurement Sensors*, 24, 100560. https://doi.org/10.1016/j.measen.2022.100560

Uddin, S., Haque, I., Lu, H., Moni, M. A., & Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, 12(1). https://doi.org/10.1038/s41598-022-10358-x