Proceedings of the 9th World Congress on Civil, Structural, and Environmental Engineering (CSEE 2024)
London, United Kingdom – April 14-16, 2024
Paper No. ICGRE 146
DOI: 10.11159/icgre24.146

Predicting Rainfall Using Random Forest and CatBoost Models

Sung-Chi Hsu[1], Alok Kumar Sharma[2, *], Radius Tanone[3] and Yan-Tang Ye[1]
[1]Department of Construction Engineering
Chaoyang University of Technology,
Taichung, Taiwan R.O.C
schsu@mail.cyut.edu.tw; aden363342@gmail.com
[2]Department of Computer Science and Information Engineering
Chaoyang University of Technology,
Taichung, Taiwan R.O.C
rbaloksharma@gmail.com
[3]Department of Information Management
Chaoyang University of Technology,
Taichung, Taiwan R.O.C
s11014903@gm.cyut.edu.tw

Abstract **–** This research offers a detailed examination of forecasting rainfall in Taiwan through the application of tree-based machine learning methods, particularly Random Forest and CatBoost models. The unique weather patterns of Taiwan, marked by frequent typhoons and monsoons, underscore the importance of precise rainfall forecasts for disaster readiness and agricultural strategy. Data for this research was sourced from Ruiyan, Taiwan, specifically from the Department of Atmospheric Sciences at Chinese Culture University, covering the period 1998–2018. The dataset encompasses five principal variables: temperature, humidity, air pressure, wind direction, and wind speed. The Random Forest model was selected for its effectiveness in managing nonlinear data, and the CatBoost model for its adeptness in handling categorical data and mitigating overfitting. Our approach included data pre-processing, adjusting model parameters, and addressing data imbalances through the undersampling technique. The evaluation of both models focused on measures like accuracy, precision, recall, F1 score, and ROC-AUC. Results show that the Random Forest model surpasses CatBoost in accuracy and AUC, reaching a maximum accuracy of 70% and an AUC of 76%. This analysis sheds light on the capabilities of these tree-based machine learning models in rainfall prediction. The study underlines the considerable promise of machine learning in improving meteorological forecasting systems, which is crucial for effectively responding to weather-related challenges in Taiwan.

Keywords**:** Rainfall Predication, Machine Learning, Random Forest and Catboost

1. Introduction

Taiwan, geographically positioned within the subtropical monsoon climate zone, is characterized by its perennial rainfall and maritime surroundings. A dramatic variation in terrain elevation characterizes the island's topography. Consequently, Taiwan is routinely subjected to major catastrophic events, predominantly attributed to heavy rainfall. These meteorological extremes are often associated with severe weather phenomena, such as typhoons [1]. Typhoons are a crucial contributor to the water supply in southern Taiwan, primarily through the rainfall they bring. This intense precipitation, although brief, often results in a plethora of water and triggers natural disasters like river surges, debris flows, and floods in downstream areas [2]. Given these challenges, there is a pressing need in southern Taiwan for sophisticated models that can precisely predict rainfall during typhoon seasons. Such models are essential to mitigate the risks associated with heavy rainfall in localized regions.

In recent times, the realm of machine learning (ML) has witnessed considerable progress. Earlier studies have utilized diverse machine learning techniques to predict rainfall events, including random forest (RF) [3], [4], XGBoost [5], deep learning [6] and artificial neural network (ANN) [7], [8], among others.

This research aims to: a) create a machine learning-based model for predicting rainfall; b) examine suitable ML models for effective rainfall prediction in southern Taiwan; and c) deal with the imbalance class data problem. In this context, we

have applied tree-based ML approaches, particularly RF and Catboost, to forecast rainfall amounts in Taiwan's southern areas. This study used the undersampling method to deal with an imbalanced class. Furthermore, we intend to assess these models' effectiveness using various evaluation metrics, including the ROC curve, accuracy, recall, precision, and F1 score.

2. Material and Methods

Fig. 1 shows a flowchart describing the methodology for a ML workflow for analysing rainfall data, starting with pre-processing the dataset and then addressing class imbalance through data sampling. Next, the data is divided into training and testing sets. Models are trained on the former and evaluated on the latter to measure performance and accuracy.

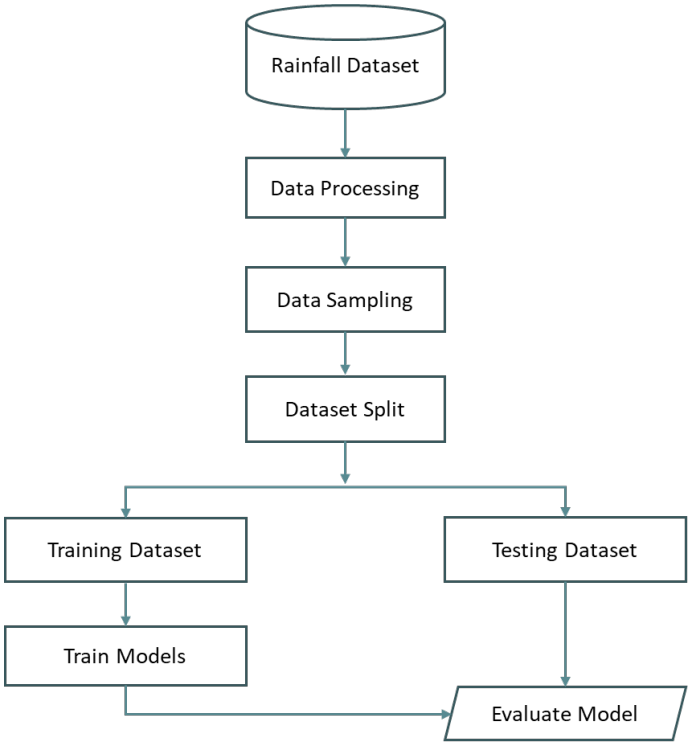


Fig. 1: Research method process.

2.1. Data and Data Pre-Processing

In this research, Ruiyan, Taiwan's precipitation data was utilized to forecast the rainfall. This data, sourced from the Chinese Culture University's Department of Atmospheric Sciences, comprises 173,657 entries spanning from 1998 to 2018. The dataset encompasses six meteorological variables, including temperature, humidity, air pressure, wind direction, and wind speed. During data pre-processing, any missing or null values were eliminated. The study initially transformed the continuous rainfall variable into categorical labels, namely 'Rain' and 'No Rain', in accordance with the guidelines from the MANOBS surface weather observation manual. A measurement of 0 mm indicates no rainfall, whereas any value greater than 0 mm is classified as rainfall. Prior to model development, the dataset was normalized to a 0–1 scale, and the target variable was converted from a string representation to a numerical one, assigning '0' for no rain and '1' for rain conditions.

Fig. 2 represents a comprehensive scatterplot matrix, encompassing a range of atmospheric measurements: air pressure, humidity, temperature, wind speed, and wind direction, compared against instances of rainfall, categorized as 'Rain' or 'No Rain'. Each non-diagonal subplot illustrates a bivariate relationship, allowing us to discern potential dependencies or patterns between the variables under different weather conditions. The plots on the diagonal are density plots, which replace the redundant univariate scatterplot with a distributional view for each variable, segregated by rainfall status.

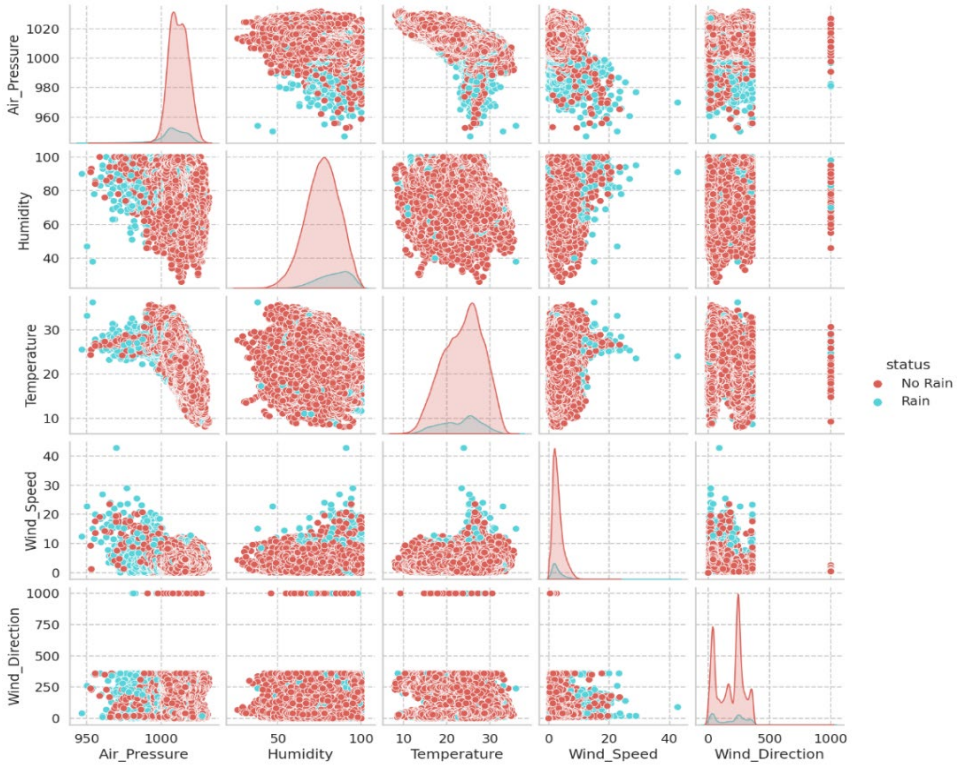


Fig. 2: Features distribution based on rainfall status

2.2. Data Sampling

Our research encountered a dataset with uneven class sizes, as depicted in Figure 3, where the number of 'no rainfall' instances was significantly higher than that of 'rainfall' instances. To address this imbalance, we implemented undersampling to decrease the size of the predominant 'no rainfall' class to match the 'rainfall' class, thereby achieving a balanced dataset. This approach enhances the model's fairness and accuracy, although it results in a reduction of the total data size. After undersampling, our dataset comprised 37,672 records, with 18,836 'no rain' and 18,836 'rain' records.

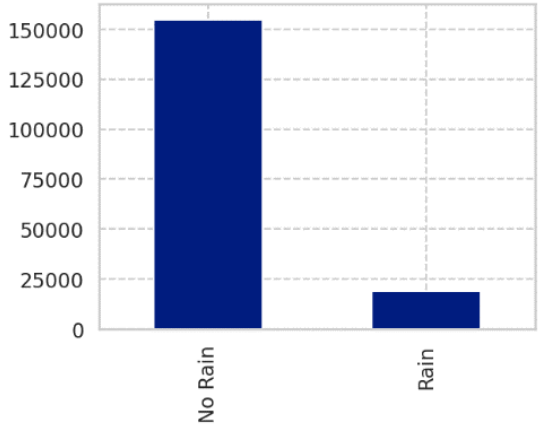


Fig. 3: Class imbalance

2.3. Data Split

In the development of ML models, we strategically separated our dataset into distinct sets for training set and testing set purposes. The training set, which is used to educate the models, contains 26,370 data points, amounting to 70% of the entire dataset. This allows the models to learn from the majority of the data, ensuring they capture the underlying patterns and relationships. The testing set is composed of 11,302 entries, making up the remaining 30% of the data. This set is crucial for evaluating the models' accuracy and generalizability to new, unseen data. By maintaining this 70-30 split, we aim to

achieve a balance between thorough learning during the training phase and rigorous evaluation during the testing phase, which is essential for validating the models' predictive power in real-world scenarios.

2.4. Models

In our study, we employed two different ML algorithms for predicting rainfall events, specifically RF and Catboost.

2.4.1. Random Forest

Random Forest (RF) [9], [10] is an ensemble ML algorithm used for both regression tasks and classification. It functions by building numerous decision trees (DT) while training and producing the class's most common outcome (for classification) or the average prediction (for regression) from these individual trees. For classification, RF combines the results from multiple decision trees to decide the final class of the object. The strength of RF lies in its ability to reduce overfitting by averaging multiple decision trees, making it more robust and accurate than individual decision trees. The algorithm performs well in a wide range of data types and is particularly useful in handling large datasets with higher dimensionality. The mathematical equation is as follows:

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2 \quad (1)$$

Where p_i denotes the proportion of the observed class in the dataset, while c signifies the total count of distinct classes.

2.4.2. CatBoost

CatBoost [11] emerges as a notable advancement in the ML arena, developed by Yandex. It distinguishes itself with an exceptional ability to handle categorical data, a common challenge in many ML tasks. Traditional methods often require extensive pre-processing, such as one-hot encoding, which is both time-consuming and prone to data leakage. CatBoost innovatively processes categorical variables inherently, significantly simplifying the data preparation process. The strong point of CatBoost lies in its robustness and accuracy, maintained even with default parameter settings. This user-friendliness is beneficial for both novices and experts, as it allows for achieving optimal performance without extensive parameter tuning. Versatile in nature, CatBoost is adept at handling various ML tasks containing regression, classification, and ranking. Its efficiency in managing categorical data makes it a preferred choice among data scientists and ML engineers, especially when dealing with complex datasets.

2.5. Evaluation Matrix

A confusion matrix (CM) [12], depicted in Table 1, is commonly utilized to assess the effectiveness of a classification model (or 'classifier') on test data with known actual values. It facilitates the visual representation of an algorithm's performance, usually in supervised learning. In the matrix, each row corresponds to instances in a real class, and each column to instances in a predicted class. The following is the layout of a CM:

Table 1: Confusion matrix.

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

Where,

- TP: The term "TP" stands for True Positive, representing the count of positive instances correctly classified.
- TN: TN, or True Negative, reflects the total correctly classified negative cases.
- FP: False Positive, denoted by "FP", is the count of actual negative instances wrongly labeled as positive.
- FN: FN, meaning False Negative, signifies the number of actual positive instances that were incorrectly classified as negative.

Table 2: Formulas for evaluation metrics.

Name	Formulas
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-Score	$2 * \left(\frac{Precision * Recall}{Precision + Recall}\right)$

3. Results

Fig. 4 shows the correlation matrix heat map to represent the correlation coefficients between pairs of variables within a dataset. The matrix includes variables such as Air Pressure, Humidity, Temperature, Wind Speed, Wind Direction, and status, each correlated with one another. The coefficients range from -1 to 1, with 1 signifying a perfect positive correlation and -1 indicating a perfect negative correlation. The diagonal, where variables intersect with themselves, predictably shows a correlation of 1. Notably, Air Pressure and Temperature have a relatively strong negative correlation of 0.68, suggesting that as air pressure increases, temperature tends to decreases. Conversely, there are pairs with negligible correlations, such as Wind speed and Temperature, which exhibit a coefficient close to 0, indicating no significant linear relationship.

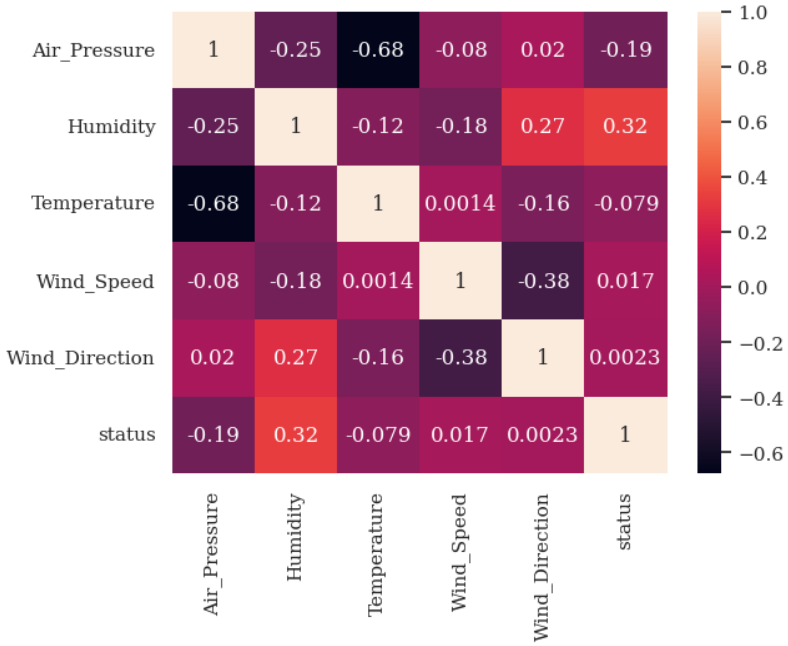
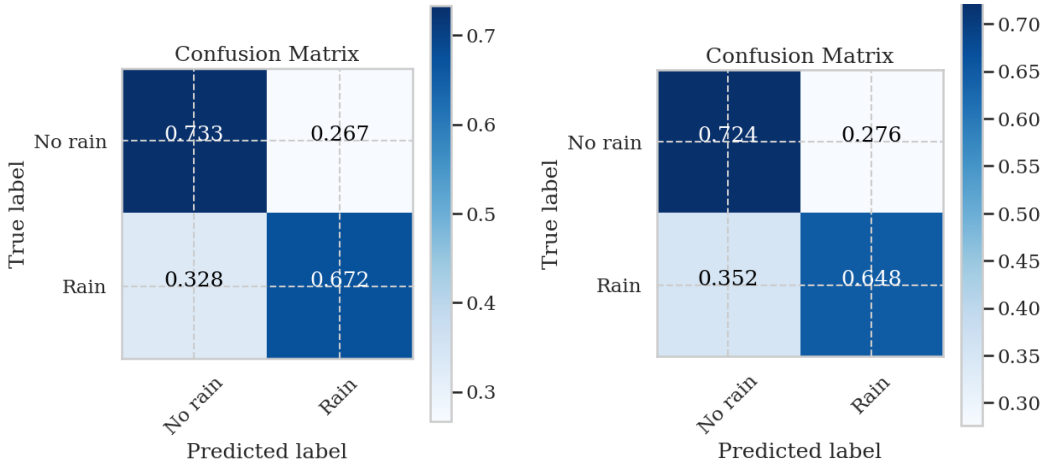


Fig.4: Pearson correlation

The CM for the RF model (shown in Fig.5(a)) in our study of rainfall prediction indicates that it successfully predicted the absence of rain with an accuracy of 73.3%, and correctly forecasted rainfall occurrences with a precision of 67.2%. The model exhibited a higher tendency to misclassify actual rainfall instances, evidenced by a false negative rate of 32.8%, suggesting that it occasionally failed to detect rain. False positives were less common, at a rate of 26.7%. This implies that the RF model is fairly adept at forecasting rain but requires improvements to lower the incidence of missed rain predictions.

CatBoost model's CM (shown in Fig.5(b)) reveals a true positive rate of 64.8%, meaning it accurately predicted rain roughly two-thirds of the time. It was slightly more accurate in predicting non-rain events, with a true negative rate of 72.4%.

Nevertheless, the model demonstrated a false negative rate of 35.2%, overlooking a substantial proportion of rain occurrences. False positives were recorded 27.6% of the time, indicating incorrect rain predictions when it was dry. These findings demonstrate the CatBoost model's solid prediction capabilities, yet they also emphasize the necessity of enhancing its performance in terms of reducing false negatives to ensure more dependable rain forecasts.



(a) RF (b) CatBoost

Fig.5: Confusion Matrix

In this research, we conducted a comparative analysis of two ML models, the RF and Catboost, focusing on their performance in predicting rain. The evaluation metrics of Precision, Recall, F1-Score and accuracy, were used to assess the effectiveness of these models.

The RF model demonstrated a balanced performance in predicting both 'No Rain' and 'Rain' scenarios shown in Table 1. For the 'No Rain' class, it showed a precision of 0.69, indicating that 69% of its predictions in this category were accurate. Its recall was slightly higher at 0.73, meaning it correctly identified 73% of the actual 'No Rain' instances, leading to an F1-Score of 0.71. The 'Rain' class predictions had a slightly better precision of 0.72 but a lower recall of 0.67. The F1-Score for this class was 0.69. The overall accuracy of the RF model stood at 0.70, with a macro average of 0.70 across all metrics, showing a consistent performance across the two classes.

In comparison, the Catboost model exhibited closely aligned results shown in Table 2. For the 'No Rain' class, its precision matched that of the RF model at 0.69, but with a marginally lower recall of 0.72, resulting in an F1-Score of 0.70. The 'Rain' class had a precision of 0.70 and a recall of 0.65. The F1-Score was 0.69. The overall accuracy of the Catboost was 0.69, a tad lower than the RF model. Both the macro average and weighted average across metrics were 0.69, indicating a uniform performance but slightly trailing behind the RF model.

Table 1: RF classification result.

RF	**Precision**	**Recall**	**F1-Score**	**Support**
No Rain	0.69	0.73	0.71	5652
Rain	0.72	0.67	0.69	5650
Accuracy			0.70	11302
Macro Avg	0.70	0.70	0.70	11302
Weighted Avg	0.70	0.70	0.70	11302

Table 2: Catboost classification result.

Catboost	**Precision**	**Recall**	**F1-Score**	**Support**
No Rain	0.69	0.72	0.70	5652
Rain	0.70	0.65	0.67	5650
Accuracy			0.69	11302
Macro Avg	0.69	0.69	0.69	11302

Weighted Avg	0.69	0.69	0.69	11302

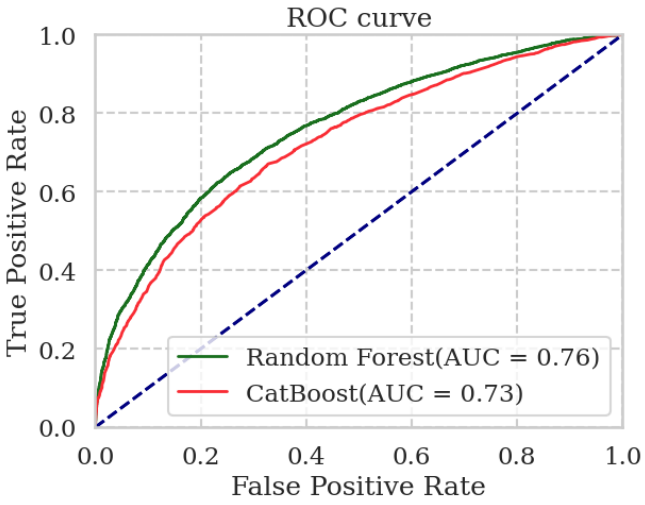


Fig.6: ROC curve.

Fig. 6 displays the ROC curves for two models: the RF with an AUC of 0.77, and the Catboost with an AUC of 0.73. These AUC scores suggest both models are effective classifiers, with the RF model being slightly more discriminative.

In conclusion, while both the RF and Catboost models showed competent and similar capabilities in rain prediction, the RF model exhibited a marginal advantage in overall accuracy and a balanced performance across the 'Rain' and 'No Rain' categories. The differences between the two models, however, were not substantial, suggesting that either could be effectively utilized for this prediction task.

4. Conclusion

This study has used tree-based ML, such as RF and CatBoost models, to predict the rainfall using the Taiwan dataset. The choice of the Random Forest model was made because of its recognized strength in handling nonlinear datasets efficiently. On the other hand, the CatBoost model was selected for its excellence in processing categorical data and its potential to lower the chances of overfitting. Our study was grounded in an extensive analysis of a dataset spanning two decades (1998-2018), obtained from the Department of Atmospheric Sciences at the Chinese Culture University, Ruiyan, Taiwan. The research process involved rigorous steps of data pre-processing, fine-tuning of model parameters, and employing the undersampling technique to address the issue of data imbalance. The Random Forest model demonstrated superior performance, achieving a peak accuracy of 70% and an AUC of 76%. This outcome is particularly noteworthy as it not only validates the effectiveness of tree-based ML models in the complex task of rainfall prediction but also highlights the specific strengths of the Random Forest model in this regard.

These findings have significant implications for rainfall forecasting. Employing tree-based ML methods, this study aids in improving weather prediction models, especially in areas like Taiwan where precise rainfall forecasts are crucial for disaster management and agriculture. The effectiveness of Random Forest and CatBoost models in this research highlights the substantial promise of ML in meteorology, pointing towards future advancements in more accurate and reliable weather forecasting systems.

Future studies will explore different ML models, like neural networks and gradient-boosting machines, to assess their effectiveness in predicting rainfall and determine the best models for various meteorological scenarios.

References

[1] T.-T. Tsai, Y.-J. Tsai, C.-L. Shieh, and J. H.-C. Wang, “Triggering Rainfall of Large-Scale Landslides in Taiwan: Statistical Analysis of Satellite Imagery for Early Warning Systems,” *Water*, vol. 14, no. 21, p. 3358, 2022, doi: 10.3390/w14213358.

[2] S. S. Lin, K. Y. Zhu, J. Y. Wang, and Y. P. Liao, “Integrating ANFIS and Qt Framework to Develop a Mobile-Based Typhoon Rainfall Forecasting System,” *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022.

[3] C. H. Oh, K. S. Choo, C. M. Go, J. R. Choi, and B. S. Kim, “Forecasting of debris flow using machine learning-based adjusted rainfall information and ramms model,” *Water (Switzerland)*, vol. 13, no. 17, pp. 1–23, 2021.
[4] S.-C. Hsu and A. K. Sharma, “Forecast Rainfall Density by Utilizing Machine Learning Models,” Mar. 2023.
[5] R. Aguasca-Colomo, D. Castellanos-Nieves, and M. Méndez, “Comparative analysis of rainfall prediction models using machine learning in islands with complex orography: Tenerife Island,” *Appl. Sci.*, vol. 9, no. 22, 2019.
[6] J. L. Chiang, C. M. Kuo, and L. Fazeldehkordi, “Using Deep Learning to Formulate the Landslide Rainfall Threshold of the Potential Large-Scale Landslide,” *Water (Switzerland)*, vol. 14, no. 20, pp. 1–20, 2022.
[7] Y. Kassem, H. Gökçekuş, H. Çamur, and E. Esenel, “Application of artificial neural network, multiple linear regression, and response surface regression models in the estimation of monthly rainfall in northern Cyprus,” *Desalin. Water Treat.*, vol. 215, no. October 2019, pp. 328–346, 2021.
[8] J. T. Esteves, G. de Souza Rolim, and A. S. Ferraudo, “Rainfall prediction methodology with binary multilayer perceptron neural networks,” *Clim. Dyn.*, vol. 52, no. 3–4, pp. 2319–2331, 2019.
[9] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, pp. 5–32, 2001.
[10] L.-H. Li, A. K. Sharma, R. Ahmad, and R. Chen, “Predicting the Default Borrowers in P2P Platform Using Machine Learning Models,” in *Artificial Intelligence and Sustainable Computing for Smart City*, India: Springer International Publishing, 2021, pp. 267–281.
[11] M. Saber, T. Boulmaiz, M. Guermoui, K. I. Abdrabo, S. A. Kantoush, T. Sumi, H. Boutaghane, D. Nohara, E. Mabrouk, “Examining LightGBM and CatBoost models for wadi flash flood susceptibility prediction,” *Geocarto Int.*, vol. 37, no. 25, pp. 7462–7487, Dec. 2022, doi: 10.1080/10106049.2021.1974959.
[12] R. Susmaga, “Confusion Matrix Visualization,” *Intell. Inf. Process. Web Min.*, pp. 107–116, 2004.