Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 18, No. 1, 2026

Rainfall Prediction Based on an Improved XGBoost Model Using Climate Data

Junyi Li *

Department of Computer Science and Technology, BEIJING NORMAL-HONG KONG BAPTIST UNIVERSITY, Zhuhai, Guangdong, China

*** Corresponding author Email:** t330026077@mail.bnbu.edu.cn

Abstract: Rainfall prediction is an important classification problem in meteorology, with practical applications in agriculture, water management, flood warning, and urban planning. Predicting rainfall is difficult because variables such as temperature, humidity, pressure, wind, and time are related in complex and often nonlinear ways. In this study, the WeatherAUS dataset was used to develop a machine-learning framework for predicting rainfall on the following day. The workflow includes data preprocessing, meteorological feature construction, treatment of class imbalance, and parameter adjustment for an Extreme Gradient Boosting (XGBoost) classifier. Logistic Regression, Decision Tree, Random Forest, and standard XGBoost models were used for comparison. Their performance was examined using accuracy, precision, recall, F1 score, ROC-AUC, and confusion matrices. Because missing a rainy day is important in practical forecasting, particular attention was given to the recall of the rain class. On the test set, the improved XGBoost increased rainy-day recall from 0.524 to 0.670, while false negatives fell from 839 to 581. However, this change also resulted in lower precision and accuracy than the standard XGBoost. The results therefore show a clear trade-off: the modified model detects more rainfall events, but it also produces more false alarms.

Keywords: Rainfall Prediction; Climate Data; XGBoost; Feature Engineering; Class Imbalance; Machine Learning.

1. Introduction

Rainfall forecasting plays an important role in meteorology and hydrology. Its results affect agricultural planning, reservoir management, flood prevention, urban drainage, and many other practical activities. As weather conditions become more variable and extreme rainfall events receive greater attention, reliable short-term rainfall prediction has become increasingly important. Previous studies have shown that heavy precipitation can cause considerable damage to infrastructure and social systems [1], [2]. For this reason, improving rainfall forecasting is not only a technical problem but also a practical requirement.

These characteristics make rainfall prediction a suitable problem for data-driven machine learning methods. Rainfall does not depend on a single meteorological variable. Temperature, humidity, atmospheric pressure, wind speed, cloud conditions, and temporal patterns can all affect whether precipitation occurs. More importantly, these variables do not act independently; their relationships are often nonlinear and change over time [18]. Numerical weather prediction and remote-sensing methods remain central to operational forecasting, but accurately describing local or extreme rainfall can still be difficult [1]. Long-term studies also note that precipitation forecasts are affected by uncertainty in both atmospheric conditions and model assumptions [2]. These difficulties provide a reasonable motivation for testing data-driven machine-learning approaches.

Machine-learning methods have become increasingly common in rainfall studies because historical weather records contain patterns that can be learned directly from data. Common approaches include Logistic Regression, Decision Trees, Random Forests, Support Vector Machines, neural networks, and gradient-boosting algorithms [12]. For tabular climate data, tree-based ensemble models are particularly useful because they do not require simple linear relationships between predictors and the target. They can also handle threshold effects and interactions among variables more naturally than a basic linear model. Earlier studies have reported promising results from Random Forest, neural-network, and boosting-based methods in precipitation forecasting [3], [4].

XGBoost is a gradient-boosting method in which decision trees are added sequentially so that later trees correct errors made by earlier ones [13]. The algorithm also includes regularization and efficient handling of sparse data, which makes it suitable for structured classification problems [5]. In rainfall research, XGBoost has been used both on its own and as part of larger forecasting systems. For instance, the VMD-PCA-XGBoost framework combines signal decomposition, dimensionality reduction, and XGBoost for precipitation prediction [1]. Other work has paired optimized XGBoost models with interpretable machine-learning methods to examine how different forecast variables contribute to short-term rainfall prediction [3].

Even so, several practical issues remain. Some studies apply standard machine-learning algorithms directly, without making adjustments for the characteristics of meteorological data. Class imbalance is another problem because dry days usually occur more often than rainy days. A classifier can therefore obtain fairly high overall accuracy while still missing a substantial number of rainfall events. More complicated hybrid models may improve performance by adding decomposition methods, physical data sources, or deep-learning components, but this usually comes with extra computational cost and a less transparent modelling process. These concerns matter in rainfall prediction because a missed rainfall event may be more costly than an unnecessary warning. The classification threshold therefore has a direct effect on the balance between false alarms and missed events. In this work, a public dataset, explicitly defined feature engineering, and a cost-sensitive XGBoost model are used so

that this trade-off can be examined in a relatively transparent and reproducible way.

This study uses the publicly available WeatherAUS dataset to investigate next-day rainfall classification with an improved XGBoost model. RainTomorrow is treated as the target, so the model predicts whether rainfall will occur on the following day. The work has four main parts. First, a consistent preprocessing and feature-engineering pipeline is constructed for the meteorological data. Second, several common machine-learning models are trained under the same experimental setting to provide a fair comparison. Third, class weighting and threshold adjustment are introduced into XGBoost to increase sensitivity to rainy days. Finally, performance is examined with several metrics rather than accuracy alone. The experiments show that the modified model detects more actual rainy days and produces fewer false negatives, although this improvement comes with lower precision. In other words, the model changes the balance between missed rainfall and false alarms instead of improving every metric at the same time.

2. Literature Review

2.1. Traditional and Data-Driven Rainfall Prediction

Traditional rainfall forecasting is mainly based on physical models, statistical methods, and empirical knowledge. Numerical weather prediction remains one of the most important tools because it describes atmospheric processes through physical equations. However, its predictions can still be affected by model structure, parameterization, uncertainty in initial conditions, and regional differences [2]. Rainfall itself is generated through interactions among many atmospheric processes, so representing every relevant mechanism accurately in one forecasting system remains difficult.

Data-driven approaches aim to explore the relationship between historical meteorological variables and rainfall outcomes. This approach is particularly valuable when large amounts of weather records are available in tabular form. Machine learning methods cannot replace the physical understanding of meteorological phenomena, but they can provide an efficient complement to pattern recognition, post-processing, and classification. In recent years, rainfall forecasting frameworks have increasingly combined data preprocessing, feature selection, ensemble learning, and model comparison to improve forecast reliability [2].

2.2. Machine Learning Models for Rainfall Prediction

Rainfall prediction in machine learning typically involves both simple baseline models and more complex nonlinear models [18]. Logistic regression is simple and easy to interpret, but it has relatively strict limitations on the form of the relationship between the predictor and target variables [2, 17]. Decision trees are easy to understand but can be unstable when the training data changes. Random forests reduce variance by averaging the results of multiple decision trees and are often used as a strong baseline model for table prediction tasks. Support Vector Machines and neural networks can model more complicated nonlinear relationships, but they generally require more attention to preprocessing and parameter selection.

Recent studies have also explored ensemble and hybrid approaches rather than relying on one algorithm. One multi-model study combined MLP, SVR, BMA, Random Forest, AdaBoost, XGBoost, and LightGBM with bias-corrected numerical weather prediction data [2]. The combined model performed better than the original individual models. This result suggests that rainfall prediction benefits from direct model comparison and systematic adjustment. Simply choosing a sophisticated algorithm does not guarantee that it will perform well on a particular meteorological dataset.

2.3. XGBoost-Based Precipitation Prediction

XGBoost is widely used for structured prediction because it combines gradient boosting with regularization and an efficient training procedure [5]. These properties are useful for climate datasets, where predictors may be nonlinear, correlated, or incomplete. Its role in precipitation forecasting varies between studies. In some cases, XGBoost is the main predictor; in others, it is used to calibrate or combine the outputs of other models. One short-term precipitation study, for example, optimized an XGBoost model using several types of forecast indices and compared it with benchmark algorithms [3]. Permutation importance and SHAP were then used to examine which indices influenced the predictions. This is useful because it shows that XGBoost can provide not only predictions but also some information about how the model is making them.

XGBoost has also been included in more complex hybrid systems. Kong et al. developed a VMD-PCA-XGBoost approach that first decomposes the signal, then reduces its dimensionality, before applying gradient boosting [1]. Their experiments reported better performance than standard XGBoost and related decomposition-based variants. Zhu and Zhang used XGBoost in a different way, applying it to correct residual errors from LSTM- and CNN-based precipitation models [4]. They reported an increase in R^2 and a narrower distribution of prediction errors after correction. Together, these studies indicate that XGBoost is flexible enough to serve either as a primary model or as a correction component for nonlinear precipitation patterns.

2.4. Class Imbalance and Evaluation in Rainfall Classification

Rainfall datasets often contain many more non-rainy observations than rainy ones. If this imbalance is ignored, a classifier can become biased toward the majority class and predict “no rain” too frequently [6]. This is a practical concern because, in some warning applications, failing to detect an actual rainfall event may have greater consequences than producing a false alarm. Common approaches for dealing with the problem include resampling, cost-sensitive learning, and adjustment of the classification threshold [6], [7].

For this reason, accuracy alone is not sufficient for evaluating the models. Precision, recall, F1 score, ROC-AUC, and the confusion matrix are considered together because each metric describes a different part of classification performance. Recall is particularly important when the research objective is to reduce missed rainfall events, while precision helps control false alarms. The F1 score balances precision and recall, while ROC-AUC measures ranking ability at different thresholds. Therefore, this study evaluates the model using multiple metrics, with particular focus on recall and F1 score, consistent with the research objectives stated in the proposal.

2.5. Research Gap

The literature reviewed identified three main gaps relevant to this study. First, while complex hybrid frameworks can achieve excellent performance, they often require specialized data sources or complex modeling components, which may reduce the reproducibility of first-time experimental research projects. Second, standard XGBoost is often used as a high-performing benchmark model, but its adaptability to rainfall category imbalances and meteorological feature structures requires further experimental validation. Third, many studies only report overall performance but fail to adequately distinguish the trade-off between detecting rainfall events and controlling false alarms. These gaps prompted this study to construct a transparent and reusable XGBoost rainfall classification framework using publicly available climate datasets.

3. Materials And Methods

3.1. Dataset and Prediction Target

The experiment used the weatherAUS dataset, which contains daily weather observations from multiple locations in Australia [9]. The dataset includes weather variables such as minimum and maximum temperature, rainfall, evaporation, sunshine, wind direction and speed, humidity, pressure, cloud cover, temperature at different times of day, RainToday, and RainTomorrow. RainTomorrow is the target variable, representing whether it will rain the next day. This task is modeled as a binary classification problem, where RainTomorrow = 1 represents the rain class and RainTomorrow = 0 represents the no-rain class.

The original file contained 145,460 rows and 23 columns. Rows lacking target labels were removed before modeling, leaving 142,193 valid target records. After excluding incomplete 2007 data, 142,132 rows were ultimately retained for time series modeling. The dataset was split chronologically rather than by random sampling: 2008 to 2015 were used for training, 2016 for validation, and 2017 was reserved for testing. Due to the limited scope of the dataset, the testing period ended on June 25, 2017. This design reduced time leakage and better simulated the prediction scenario—that is, not using future observations to train early predictions.

Table 1. Chronological Data Split and Class Distribution

Split	Years	Rows	Negative	Positive	Positive Rate
Training	2008-2015	116,158	90,246	25,912	22.31%
Validation	2016	17,508	13,325	4,183	23.89%
Test	2017	8,466	6,703	1,763	20.82%

3.2. Data Preprocessing and Feature Engineering

Preprocessing was performed before model training. RainTomorrow and RainToday were converted from Yes/No labels to binary values. Numerical features were imputed using the median fitted only to the training set. Categorical variables, including location and wind direction variables were imputed using the "missing" class and one-hot encoded according to the classes in the training set. For tree-based models, no feature scaling is required. For logistic regression, the standardized numerical features are saved separately to ensure the stability of the optimization process. After encoding, the final tree model matrix contains 138 features.

The feature engineering design is based on meteorological significance while also considering the need to avoid using future information. Date-based features include year, month, day, dates within a year, and season. Difference features include temperature range, air pressure difference, humidity difference, and temperature difference. Within each location, lag and sliding indices were calculated using only prior observational data, including rainfall lag, three-day average rainfall, lagged afternoon humidity, lagged afternoon pressure, lagged afternoon temperature, lagged RainToday, and the number of rainy days in the previous three observation periods. The original "Date" column was removed from the final modeling matrix after creating date-based derived variables.

3.3. Baseline Models and Improved XGBoost

Five comparative model settings were evaluated before selecting the final recommended model. Logistic Regression, Decision Tree, Random Forest, and Standard XGBoost were used as baseline models [8], [12]. Standard XGBoost was trained using the original engineered feature set and a default classification threshold of 0.50. Furthermore, a standard XGBoost model with threshold tuning was evaluated to distinguish the impact of threshold adjustment from that of class weight adjustment. The proposed improved XGBoost uses the same feature set as the standard XGBoost but adds class weights and validation set-based threshold tuning.

The final improved XGBoost model was designed as an XGBoost classifier with 600 estimators, a maximum depth of 6, a learning rate of 0.05, a column sampling ratio of 0.8, a gamma of 0.3, a minimum child node weight of 3, an L2 regularization coefficient of 1, and a scale_pos_weight value of 4.0. The threshold was selected on the validation set and then fixed at 0.64 during test evaluation. By employing class weights, the penalty for misclassifying rainy days was increased. Subsequently, the operating point of the probabilistic classifier was optimized through threshold adjustment, thus enabling the final decision rule to better reflect the goal of reducing missed rainfall events.

An enhanced-feature and Optuna-tuned trial was also tested during development. This trial introduced additional engineered features and performed hyperparameter search, but it was not selected as the final model because it reduced F1-score, ROC-AUC, and PR-AUC despite increasing recall. Therefore, this paper only reports a final proposed model, named "Improved XGBoost", while the "Enhanced Experiment" is regarded as an unsuccessful extension in the discussion section.

The threshold selection process is also crucial for the reproducibility of the results. The probabilistic classifier generates scores between 0 and 1, but the default threshold of 0.50 is merely a convention. In tasks with unbalanced rainfall, this default value may be too conservative, as many real-world rainfall cases, while having a probability below 0.50, still have a higher probability of rainfall than most non-rainfall cases. Therefore, thresholds were sequentially selected from low to high probability levels on the validation set, without using the test set during this selection process. This design ensures that the selected thresholds reflect the actual application goals of the research without revealing the final evaluation information. Furthermore, this also facilitates the reproducibility and verification of the experiment.

3.4. Evaluation Metrics

These models are evaluated using accuracy, rainy day category precision, rainy day category recall, F1 score, F2 score, ROC-AUC, PR-AUC, and confusion matrix analysis. Accuracy measures the overall proportion of correct predictions [14]. but its results can be misleading when non-rainy-day samples dominate the dataset. Rainy day category precision measures the proportion of predicted rainy days that were actually rainy, while rainy day category recall measures the proportion of actual rainy days detected by the model [16]. The F1 score balances precision and recall, while the F2 score focuses more on recall, making it very useful when considering that missed rainfall events are more prevalent than false positives. ROC-AUC evaluates ranking ability at different thresholds, while PR-AUC is particularly useful for imbalanced classification [15].

Validation sets are used in the development of thresholds and model comparisons. The test set is used only for the assessment of the final report. This distinction is important because the use of test sets to select thresholds would overestimate performance. The final reading focused on whether the proposed improved XGBoost could reduce the number of false negatives while maintaining acceptable accuracy and overall differentiation.

4. Experimental Results and Analysis

4.1. Dataset Split and Class Distribution

In chronological order, 116,158 training samples, 17,508 validated samples and 8,466 test samples were produced. The share of the rain-day category was 22.31 per cent in the training set, 23.89 per cent in the validation set and 20.82 per cent in the test set. This distribution indicates an imbalance in the classification task, with non-rainy days belonging to the majority category and rainy days belonging to the minority category. Although the imbalance is moderate, it is wide enough that accuracy can be misleading. For example, a model that is too conservative in predicting rainfall may have a high accuracy rate but fail to detect many actual rainy days.

The distribution of missing values also reflects the complexity of real weather data. Variables such as Sunshine, Evaporation, Cloud9am, and Cloud3pm contained substantial missingness, while core variables such as temperature, humidity, and pressure had fewer missing values. Pre-processing processes address these issues through the use of training-set-fitted imputation methods and one-hot encoding, avoiding the direct deletion of a large number of records. This preserves the time-series coverage required for model training and assessment while avoiding data loss.

4.2. Overall Model Comparison

Table 2 lists the final test results. The standard XGBoost model achieved the highest accuracy of 0.865 and the highest rainy-class precision of 0.750. It also achieved excellent ROC-AUC and PR-AUC values of 0.885 and 0.727, respectively. However, its rainfall recall was only 0.524, meaning that it detected only slightly more than half of the actual rainfall events in the test set. This confirms the main limitation of the standard model: excellent overall performance does not necessarily mean effective detection of rainfall events.

Logistic regression achieved the highest rainy-class recall (0.749), but its precision was only 0.519 and its overall accuracy was 0.803. This means it identified many rainy days but also generated many false positives. Decision trees showed a similar pattern, with high recognition rates but low precision and weak overall discriminative power. Random forests provided a more balanced result, with an F1 score of 0.637 and an ROC-AUC of 0.875. Standard XGBoost, with threshold adjustment, improved the accuracy of rainfall category identification from 0.524 to 0.669 and the F1 score from 0.617 to 0.650, demonstrating that simply adjusting the threshold can significantly improve the detection performance of rainfall categories.

The proposed improved XGBoost algorithm achieved an accuracy of 0.849, a recall of 0.670 for the rainy day category, an F1 score of 0.649, an F2 score of 0.662, a ROC-AUC of 0.885, and a PR-AUC of 0.726. Its F1 score is slightly lower than the threshold-adjusted standard XGBoost, but its recall and F2 score are slightly higher. Since this study focuses on reducing rain leakage, the weighted and threshold-adjusted model was chosen as the proposed improved XGBoost. This result should be interpreted with caution: the proposed model improves the detection ability of rainy weather categories but does not improve the overall ranking ability measured by ROC-AUC or PR-AUC.

Table 2. Test performance comparison of baseline and XGBoost models.

Model	Acc.	Prec.	Recall	F1	F2	ROC-AUC	PR-AUC
Logistic Regression	0.803	0.519	0.749	0.613	0.688	0.868	0.678
Decision Tree	0.795	0.506	0.719	0.594	0.663	0.836	0.632
Random Forest	0.840	0.603	0.674	0.637	0.659	0.875	0.697
Standard XGBoost	0.865	0.750	0.524	0.617	0.558	0.885	0.727
Standard XGBoost + Threshold Tuning	0.850	0.632	0.669	0.650	0.661	0.885	0.727
Improved XGBoost	0.849	0.629	0.670	0.649	0.662	0.885	0.726

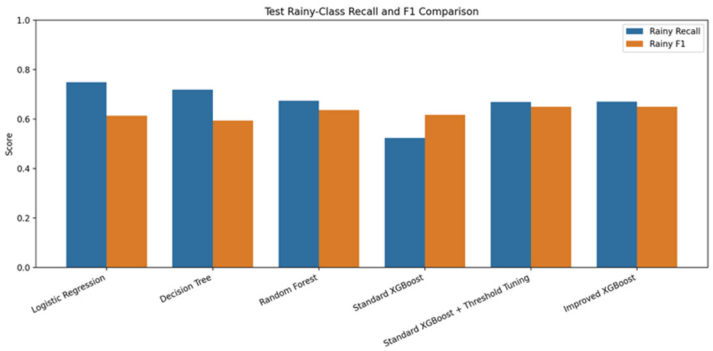


Fig 1. Test rainy-class recall and F1-score comparison

4.3. Confusion Matrix Analysis

The confusion matrix more clearly illustrates the actual performance of the proposed model. The standard XGBoost model produced 6395 true negatives, 308 false positives, 839 false negatives, and 924 true positives. The improved XGBoost model produced 6006 true negatives, 697 false positives, 581 false negatives, and 1182 true positives. Therefore, the proposed model reduced the number of missed rainfall events from 839 to 581, reducing false negatives by 258. Meanwhile, the number of false positives increased from 308 to 697. This illustrates the key precision-recall tradeoff in the experiment.

Table 3 From an application perspective, this trade-off may be acceptable when the goal is to provide early warnings of rainfall events or mitigate risks. False alarms may cause unnecessary preparation, but a missed rainfall event can be more serious in applications such as flood response, agricultural planning, or outdoor operations. The purpose of the improved model is therefore not simply to raise the total number of correct predictions. Instead, it changes the model's decision behaviour so that more actual rainy days are identified. This point is important because the main benefit of the proposed approach lies in reducing missed events rather than increasing overall accuracy.

Table 3. Confusion matrix comparison between standard and improved XGBoost.

Model	TN	FP	FN	TP
Standard XGBoost	6,395	308	839	924
Improved XGBoost	6,006	697	581	1,182

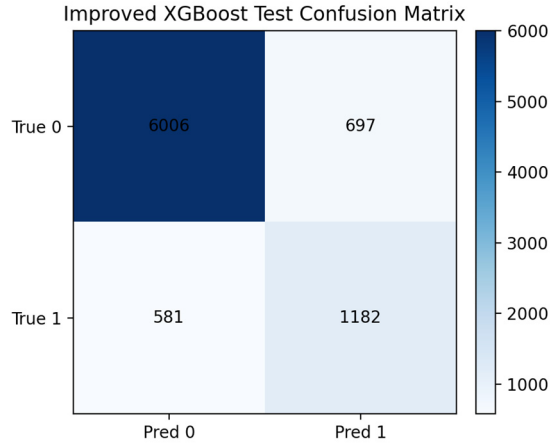


Fig 2. Confusion matrix obtained from the final Improved XGBoost model on the test set.

4.4. ROC and PR Curve Interpretation

The ROC-AUC values of standard XGBoost, threshold-tuned XGBoost, and the improved XGBoost are nearly the same. This result is reasonable because changing the classification threshold affects the final class labels but does not change the ordering of the predicted probabilities. The difference between standard and improved XGBoost is also very small, which suggests that adding class weights did not noticeably strengthen the model's overall ranking ability. The main change is instead at the decision stage: the selected threshold makes the classifier more willing to label an observation as rainfall.

A similar result can be seen from PR-AUC. Standard XGBoost obtained 0.727, whereas the improved model obtained 0.726. Therefore, the higher recall and F2 score should not be interpreted as evidence that the improved model performs better in every respect. What changed is much more specific: under the selected operating threshold, the model detects a larger proportion of rainfall events. This matches the purpose of the study, but the corresponding reduction in precision and accuracy also needs to be reported rather than ignored.

4.5. Feature Importance

Gain-based feature importance was calculated for the final improved XGBoost model. Humidity3pm had the largest importance value, followed by RainToday, Sunshine, Rainfall, WindGustSpeed, Cloud3pm, Pressure3pm, and TempRange. Most of these variables have a reasonable relationship with rainfall. High afternoon humidity, for example, represents relatively moist atmospheric conditions. RainToday and Rainfall describe recent precipitation and may therefore capture short-term persistence in weather conditions. Sunshine and Cloud3pm provide information related to cloudiness and radiation, while pressure-related variables can reflect changes in weather systems. WindGustSpeed may also provide information about unstable or changing atmospheric conditions.

Several encoded location and wind-direction variables also appeared near the top of the ranking. This is understandable because rainfall patterns differ across regions, and local wind direction may be associated with particular weather conditions in the WeatherAUS dataset. However, location variables should be interpreted carefully. A high importance value does not mean that a location itself causes rainfall; rather, the encoded location can represent regional climate differences that are not fully described by the other variables. Overall, the importance ranking indicates that the model mainly uses recognizable meteorological information rather than relying on features with no clear connection to weather conditions.

Table 4. Top ten gain-based features of the final Improved XGBoost model.

Rank	Feature	Interpretation
1	Humidity3pm	Afternoon humidity and moisture condition
2	RainToday	Current-day rainfall indicator
3	Sunshine	Sunshine duration and cloudiness proxy
4	Rainfall	Current-day rainfall amount
5	WindGustSpeed	Wind gust intensity
6	Cloud3pm	Afternoon cloud cover
7	Pressure3pm	Afternoon atmospheric pressure
8	TempRange	Daily temperature range
9	Location__Wollongong	Regional climatological effect
10	WindDir9am__NNE	Morning wind-direction pattern

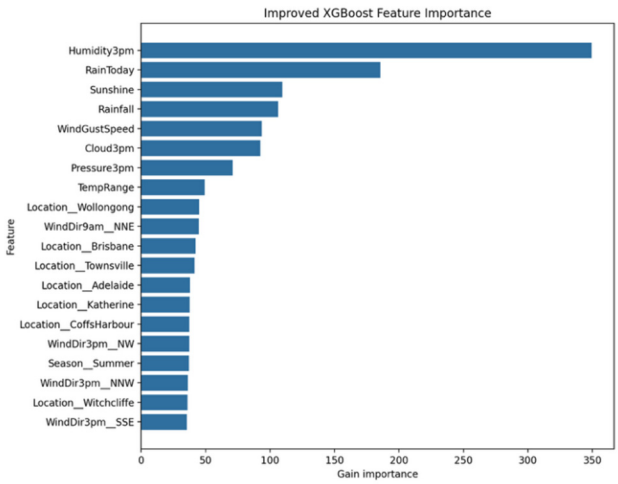


Fig 3. Top 20 feature-importance values of the final Improved XGBoost model.

5. Discussion

The experiments show that the proposed XGBoost modification mainly changes the model's ability to detect rainy days. Recall increased from 0.524 with standard XGBoost to 0.670 with the improved model, and false negatives decreased from 839 to 581. F1 also rose slightly, from 0.617 to 0.649, while F2 increased from 0.558 to 0.662. These changes are consistent with what would be expected from class weighting and threshold adjustment: the classifier becomes less conservative when identifying the positive class. For a task where missed rainfall is considered important, this behaviour is useful because fewer actual rainfall events are overlooked.

However, the results do not mean that the improved model is better on every metric. Its accuracy decreased from 0.865 to 0.849, and precision dropped from 0.750 to 0.629. ROC-AUC and PR-AUC also remained almost unchanged. In practical terms, the model detects more rainy days but pays for this by generating additional false alarms. The improvement therefore comes mainly from cost-sensitive training and the threshold selected on the validation set, rather than from a major improvement in probability ranking. This distinction matters when interpreting the experiment. Depending on the application, a model can still be useful even when it does not dominate every evaluation metric. A model may be more useful for a specific operational objective, rather than being universally superior across all metrics.

The comparison with the threshold-adjusted standard XGBoost is particularly valuable. Threshold adjustment alone achieved an F1 score of 0.650, slightly higher than the 0.649 score achieved by the improved XGBoost, which also performed slightly better in recall and F2 score. The numerical differences are small, therefore the effect of class weights should be interpreted with caution. The main conclusion is not that class weights are significantly superior to threshold adjustment, but rather that cost-sensitive and threshold-aware XGBoost frameworks can drive the model toward the desired recall-oriented behavior. Since this study prioritizes reducing leakage, a weighted model was chosen as the final proposed model.

The enhanced features and the experiment tuned via Optuna provide another useful insight [11]. While it improved the recall to 0.695, it simultaneously reduced the F1 score, ROC-AUC, and PR-AUC.

These results suggest that a larger feature set and a broader parameter search do not necessarily produce a better model. Extra engineered variables can introduce noise or make the fitted model more sensitive to the validation data. They may also push the decision boundary toward a more aggressive classification of rainfall, which increases recall but can create many additional false positives. For a public tabular dataset such as WeatherAUS, the simpler cost-sensitive model therefore offers a more stable and easier-to-explain alternative.

Previous precipitation studies have also reported gains from hybrid systems such as VMD-PCA-XGBoost or neural-network models followed by residual correction [1], [4]. Such methods can be effective, but they normally require additional processing stages, more data, or greater computational effort. The present study deliberately uses a simpler framework. Its results show how much model behaviour can change even when the underlying classifier is kept relatively simple and the main adjustments are class weighting and the decision threshold.

There are several limitations to the present experiment. First, the test data cover only the early part of 2017. Seasonal characteristics in this period may affect the reported results, so evaluation over several complete and more recent years would provide stronger evidence of generalization. Second, the model is based only on the variables contained in WeatherAUS. It does not include radar observations, satellite imagery, or outputs from numerical weather prediction models. Such data could provide additional information that is not available in a purely tabular dataset. Third, gain-based feature importance indicates which variables the model uses frequently, but it should not be treated as evidence of causality. Further work could use SHAP analysis, probability calibration, regional subgroup testing, and rolling-origin validation. It would also be useful to examine whether the selected classification threshold remains stable when the model is applied to different years or locations.

6. Conclusion

This study examined next-day rainfall prediction on the public WeatherAUS dataset using an improved XGBoost classifier. RainTomorrow was defined as a binary target, and the data were divided chronologically so that earlier observations were used for model development and later observations for testing. Several baseline classifiers were compared with standard XGBoost and the proposed cost-sensitive version. In the final model, class weighting was combined with a threshold chosen from the validation set. Performance was then evaluated using accuracy, precision, recall, F1, F2, ROC-AUC, PR-AUC, and the confusion matrix.

The standard XGBoost produced strong accuracy and AUC values, but it missed a relatively large proportion of rainy days. After class weighting and threshold adjustment, rainy-class recall increased from 0.524 to 0.670 and F1 increased from 0.617 to 0.649. F2 also rose from 0.558 to 0.662, while false negatives decreased from 839 to 581. These improvements were accompanied by lower precision and overall accuracy, which makes the precision–recall trade-off clear. The feature-importance results also show that the model relied mainly on variables with recognizable meteorological meaning, including afternoon humidity, recent rainfall, sunshine, cloud cover, pressure, gust speed, and temperature range.

Overall, the main objective of the study was met: the proposed model detected a larger proportion of actual rainfall events and reduced the number of missed rainy days.

However, the model does not improve the ROC-AUC or PR-AUC values; therefore, this improvement should be considered an operational gain related to thresholds and costs, rather than a comprehensive enhancement of ranking capabilities. Future research should apply this method to longer time periods, incorporate more meteorological data sources, and examine the model's calibrability and interpretability in more detail.

References

[1] Kong, Q., Awange, J., Li, Q., Mi, X., Bai, Q., Wang, S., Yang, Y., & Bo, G. (2025). A novel optimized hybrid VMD-PCA-XGBoost model for forecasting precipitation: Exemplified by the Beijing-Tianjin-Hebei study region in China. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18, 17154–17165. https://doi.org/10.1109/JSTARS.2025.3586351.

[2] Tadayon, A., Nazari, M., & Kerachian, R. (2025). Enhancing long-lead rainfall forecasting in data-scarce large watersheds using multi-model fusion. Journal of Hydrology: Regional Studies, 62, 102936. https://doi.org/10.1016/j.ejrh.2025.102936.

[3] Gao, Y., Wang, D., Huang, L., & Li, E. (2025). Applying XAI to optimized XGBoost models: Elucidating multi-type predictor impacts for short-term precipitation forecast. Journal of Tropical Meteorology, 31, 603–621. https://doi.org/10.3724/j.1006-8775.2025.040.

[4] Zhu, Y., & Zhang, D. (2025). Application of fusion neural network based on XGBoost correction in precipitation prediction. People's Pearl River. (in Chinese)

[5] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). San Francisco, CA, USA. https://doi.org/10.1145/2939672.2939785.

[6] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239.

[7] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953.

[8] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://doi.org/10.5555/1953048.2078195.

[9] Young, J. (n.d.). Rain in Australia. Kaggle dataset. https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package. Accessed June 3, 2026.

[10] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 2623–2631). https://doi.org/10.1145/3292500.3330701.

[11] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

[12] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451.

[13] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010.

[14] Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning (pp. 233–240). Pittsburgh, PA, USA. https://doi.org/10.1145/1143844.1143874.

[15] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432.

[16] Jolliffe, I. T., & Stephenson, D. B. (2012). Forecast verification: A practitioner's guide in atmospheric science (2nd ed.). Wiley.

[17] Wilks, D. S. (2011). Statistical methods in the atmospheric sciences (3rd ed.). Academic Press.