Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3681~3688
e-ISSN: 2548-6861

Hybrid PSO-XGBoost Model for Accurate Flood Risk Assessment

Lailatun Nabilah [1], Lukman Hakim [2]
* Teknik Informatika, Universitas Yudharta Pasuruan
** Teknik Informatika, Universitas Yudharta Pasuruan
lailatunnabilah1987@gmail.com [1], lukman@yudharta.ac.id [2]

Article Info

Article history:

Received 2025-09-08
Revised 2025-11-04
Accepted 2025-11-08

Keyword:

XGBoost,
Particle Swarm Optimization,
Flood Prediction,
Parameter Optimization,
Machine Learning.

ABSTRACT

Flood risk prediction is a crucial step in disaster mitigation. This study optimizes the Extreme Gradient Boosting (XGBoost) algorithm using the Particle Swarm Optimization (PSO) method to improve prediction accuracy. The process includes data cleaning, normalization, and classification of risk levels into low, medium, and high. The XGBoost model is trained both before and after parameter optimization of n_estimators, max_depth, and learning_rate. Before optimization, the model achieved 93% accuracy but struggled to identify minority classes. After optimization with PSO, accuracy increased to 97%, with the recall for the low-risk class improving from 21% to 57%. The optimized model also demonstrated more stable performance compared to Support Vector Machine (SVM) and Random Forest. These findings indicate that the combination of XGBoost and PSO can provide more accurate and efficient flood risk predictions.

I. INTRODUCTION

Flooding is one of the most frequent hydrometeorological disasters, causing significant social, economic, and ecological impacts in various countries, including Indonesia. Uncontrolled urbanization, environmental degradation, and increased rainfall intensity due to climate change have exacerbated flood risks in vulnerable areas [1], [2]. Furthermore, future flood risk projections based on the Shared Socioeconomic Pathways (SSP) climate scenarios indicate a global trend of increasing flood-prone areas [3]. Consequently, flood risk mapping and prediction have become urgent needs for early warning systems and disaster mitigation efforts.

Recent evidence indicates that climate change has intensified rainfall patterns, leading to an increase in flood occurrences worldwide. In Indonesia, these events are becoming more frequent in rapidly urbanized regions where inadequate drainage and land-use changes intensify flood impacts. This situation underlines the urgent need for adaptive flood prediction models that can respond to evolving environmental conditions.

Conventional models such as Support Vector Machine (SVM) and Random Forest often struggle with data imbalance and require manual parameter tuning, reducing their generalization capability. Hence, there remains a need for a more efficient approach capable of optimizing parameters automatically while maintaining both accuracy and computational efficiency. To address this limitation, this study integrates Particle Swarm Optimization (PSO) with Extreme Gradient Boosting (XGBoost) for improved flood risk prediction.

However, conventional statistical-based prediction approaches often fail to capture the complex nonlinear relationships among dynamic environmental factors [4]. On the other hand, machine learning algorithms have increasingly been utilized in hydrological disaster modeling due to their ability to handle large and complex datasets [5], [6]. One prominent algorithm in this field is Extreme Gradient Boosting (XGBoost), which is widely recognized for its high accuracy, robustness against overfitting, and flexibility in parameter tuning [7].

Nevertheless, the performance of XGBoost is highly dependent on the configuration of its hyperparameters, such as the number of decision trees (n_estimators), maximum depth (max_depth), and learning rate (learning_rate) [8]. To address this challenge, optimization methods such as Particle Swarm Optimization (PSO) have been adopted to select optimal parameter configurations. PSO is a swarm intelligence algorithm capable of efficiently exploring solution spaces and has been proven to improve classification performance across various domains, including flood risk prediction [9], [10].

Several studies have demonstrated that combining PSO with XGBoost yields superior performance compared to other algorithms, such as Random Forest, Support Vector Machine (SVM), or k-Nearest Neighbor (k-NN) in flood risk classification tasks [11], [12]. Other research has also explored boosting model optimization through alternative approaches, such as the Fick's Law Algorithm (FLA), for urban flood mapping with competitive results [13]. Furthermore, the PSO-XGBoost hybrid approach has been recognized for its advantages in scenarios with limited data or class imbalance, which are commonly encountered in historical flood datasets [14].

Based on the aforementioned background, this study aims to develop a flood risk prediction system by utilizing the XGBoost algorithm optimized with PSO. This model is expected to classify flood risk into three categories: low, medium, and high, while improving the accuracy in detecting minority classes. The evaluation is conducted using multiclass classification metrics, including accuracy, precision, recall, and F1-score.

The main contribution of this study lies in the application of PSO to optimize XGBoost hyperparameters within the context of flood risk classification based on complex environmental data. Additionally, this study emphasizes training efficiency and improved accuracy in detecting high-risk classes, with the ultimate goal of supporting disaster mitigation systems and risk-based spatial planning.

II. METHOD

The overall research framework is illustrated in Figure 1, which integrates the parameter optimization process using Particle Swarm Optimization (PSO) with the classification model training using Extreme Gradient Boosting (XGBoost).

A. Dataset

The dataset was obtained from the public platform Kaggle, consisting of 50,000 rows of data and 21 environmental features, such as rainfall, soil moisture, temperature, drainage quality, and urbanization. These features have been widely used in previous studies for modeling flood risk in both spatial and temporal contexts [12], [15].

Flood risk classification is categorized into three levels: low (0), medium (1), and high (2) based on flood probability values. This categorization process was carried out using the binning technique, as commonly applied in similar research on flood hazard mapping [3], [16].

The dataset used in this study was obtained from a publicly available Kaggle repository containing 50,000 records and 21 environmental attributes related to flood risk. These attributes include rainfall, temperature, humidity, wind speed, elevation, and soil characteristics. The dataset spans a multi-year period from 2010 to 2020, representing various regional hydrometeorological patterns in Indonesia. All features were preprocessed using normalization and missing-value handling techniques before model training.

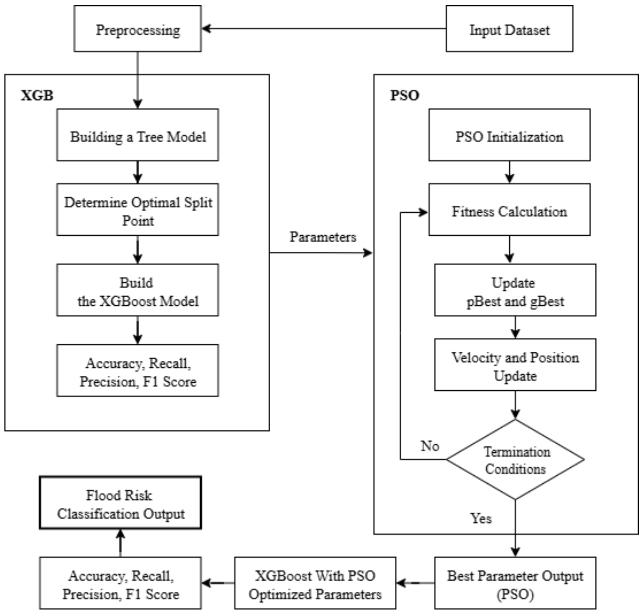

Figure 1 Flowchart XGB-PSO

B. Data Preprocessing

The preprocessing stage began with data cleaning to remove duplicate entries and handle missing values. Next, normalization was applied to numerical features using the StandardScaler method to standardize data scales. This step is essential to ensure that the algorithm does not become biased toward features with large value ranges [13], [17].

Subsequently, the data was split into 80% training data and 20% testing data, following best practices in machine learning. The class distribution within the dataset exhibited significant imbalance, with a dominance in the medium-risk class. Therefore, normalization techniques and proportional data splitting were crucial to maintain the model's generalization capability [4].

C. XGBoost

XGBoost is a tree-based boosting algorithm designed to produce highly accurate predictive models through an ensemble approach. In the initial phase, the XGBoost model was built using default parameters to establish a performance baseline. The parameters tuned include n_estimators, max_depth, and learning_rate, which directly affect the model's generalization capacity [6], [7].

This model is well-known for its built-in regularization, which helps prevent overfitting, and its ability to support feature importance evaluation [8]. These advantages make XGBoost a popular algorithm for disaster classification based on environmental data [14], [18]. The optimization process in XGBoost mathematically utilizes the following objective function:

$$obj(\theta) = \sum_{i=1}^{n} L\,(y_1, \hat{y}_i) + \Omega(\theta) \quad (1)$$

Description:

$obj(\theta)$	=	The overall objective function to be minimized.
$\sum_{i=1}^{n} L\,(y_1, \hat{y}_i)$	=	The sum of the loss function L, which measures the difference be tween the predicted value $\hat{y}_i$ and t he actual value y_1
$\Omega(\theta)$	=	The regularization function that c ontrols model complexity to prev ent overfitting.

D. PSO (Particle Swarm Optimization)

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the collective behavior of birds and fish, which is used to find the optimal values for the hyperparameters of XGBoost [9], [10]. It has been proven to be stable in handling climate and environmental data and is also applicable under imbalanced data conditions [19], [20]. In PSO, each particle has a position and velocity that are updated in every iteration using the following formula:

$$v_i\,(t) = v_i\,(t-1) + c_1.r_1 \; + \; c_2.r_2\left(G_{best} - x_i(t-1)\right) \quad (2)$$

Description:

$v_i\,(t)$	=	Velocity of particle i at iteration t
x_i	=	Position of particle i at iteration i
P_{best}	=	Best position of particle i so far
G_{best}	=	Global best position among all particles
$c_1.c_2$	=	Acceleration constants
$r_1.r_2$	=	Random numbers between 0 and 1

In this study, PSO was implemented to optimize key XGBoost hyperparameters such as the number of estimators, maximum depth, and learning rate. The swarm consisted of 10 particles and was iterated for a maximum of 5 generations. The objective of PSO was to minimize the classification error, where the fitness function was defined as the model's validation loss. Through this iterative process, PSO guided the search toward the global optimum combination of parameters that yielded the best prediction accuracy.

E. Evaluation

In this stage, the performance of the applied algorithm is evaluated. The evaluation utilizes the Confusion Matrix, followed by the calculation of accuracy, recall, F1-score, and precision based on the applied algorithm and its performance measurement matrix.

The hybrid PSO-XGBoost model was trained and validated using a 5-fold cross-validation scheme to ensure robust generalization across different data partitions. Each fold maintained the same class distribution to address potential data imbalance issues.

All experiments were conducted using Python programming language on Google Colab. The implementation utilized open-source libraries including Scikit-learn for preprocessing and evaluation, XGBoost for model training, and PySwarms for PSO optimization.

III. Results And Discussion

A. Preprocessing

Before training and testing the prediction model, the data must first be prepared through several preprocessing stages. The purpose of this step is to ensure that the data is clean, standardized, and ready to be processed by machine learning algorithms.

1. Data Cleaning

The dataset was examined for the presence of missing values. The dataset used consists of 50.000 rows and 21 features. After the validation process, no missing values were found in any of the columns.

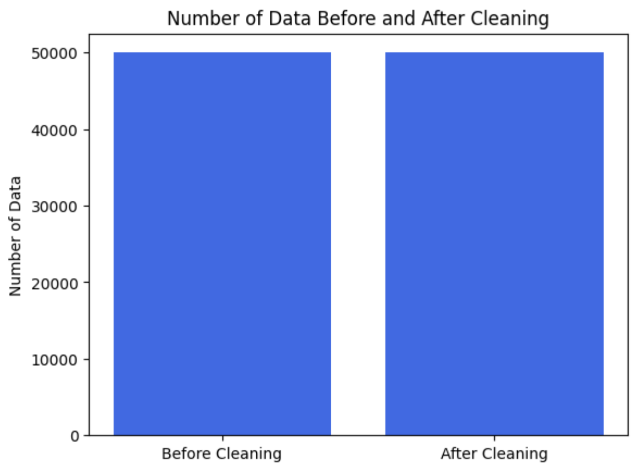

Figure 2 Dataset Before and After Cleaning

Although no missing data were found, this step was still carried out to ensure the quality of the data to be used in the model. The presence of invalid or missing data could directly impact the model's predictive performance, potentially leading to reduced accuracy or overfitting.

2. Data Normalization

The normalization method used in this study is StandardScaler, which transforms each feature value based on the z-score method. Since this approach calculates the difference from the mean, the resulting transformed values may be negative, especially when the original values are below the mean. This behavior is normal and inherent in the standardization process.

TABLE 3
DATA BEFORE NORMALIZATION

No	Monsoon Intensity	Topography Drainage	Urbanization	…	Flood Probability
1	3	8	4	…	0.65
2	8	4	3	…	0.45
3	3	10	2	…	0.62
…	…	…	…	…	…
49998	4	5	6	…	0.48
49999	7	4	3	…	0.66
50000	3	8	5	…	0.54

TABLE 4
DATA AFTER NORMALIZATION

No	Monsoon Intensity	Topography Drainage	Urbanization	…	Flood Probability
1	-0.41	1.05	0.03	…	0.91
2	1.89	-0.70	-0.25	…	-0.41
3	-0.41	2.00	-0.78	…	0.74
…	…	…	…	…	…
49998	0.24	-0.02	0.44	…	-0.31
49999	1.37	-0.70	-0.25	…	0.95
50000	-0.41	1.05	0.21	…	0.01

Based on the table, the scale of values across features has been standardized. This normalization process was conducted using the StandardScaler method, which converts each feature value into a z-score. This transformation is performed using the following formula:

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

z = Normalized value (z-score result)
x = Original feature value
μ = Mean of all values in the feature
σ = Standard deviation of the values in the feature

3. Flood Risk Label Categorization

The flood risk labels in the initial dataset were in the form of continuous values (probabilities), ranging from 0.28 to 0.73. To simplify the classification process, a binning method was applied to categorize the data into three classes:

- Low Risk (0): 0.28 – 0.40
- Moderate Risk (1): 0.41 – 0.55
- High Risk (2): 0.56 – 0.73

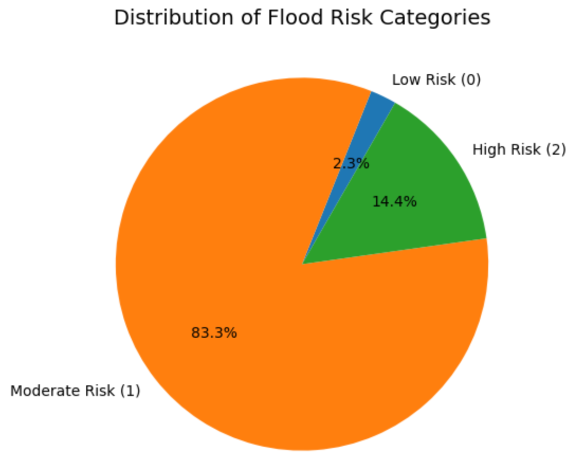

Figure 3 Distribution of Flood Risk Categories

The majority of the data falls into the Moderate Risk category (83.3%), followed by High Risk (14.4%), and Low Risk (2.3%). This imbalanced distribution may affect the model's performance, particularly in recognizing the minority classes. Therefore, the model evaluation in this study will place greater emphasis on metrics such as recall and F1-score.

4. Splitting Data

The dataset was divided into two parts: training data and testing data. The proportion used was 80% for training and 20% for testing. From a total of 50,000 data rows available, this division resulted in 40,000 training data and 10,000 testing data, as shown in Table 2 below:

TABLE 5
DISTRIBUTION OF TRAINING AND TESTING DATA

Data Type	Number of Data
Training	10000
Testing	40000

B. Model Performance Evaluation of XGBoost Before and After PSO

To measure the effectiveness of parameter optimization on the XGBoost model, evaluations were conducted using the metrics of accuracy, precision, recall, and F1-score. These evaluations were carried out both before and after the application of the Particle Swarm Optimization (PSO) algorithm.

TABLE 6
EVALUATION OF XGBOOST MODEL PERFORMANCE BEFORE AND AFTER PSO

Metric	Before PSO	After PSO
Accuracy	93%	97%
Precission	89%	95%
Recall	63%	81%
F1 Score	70%	87%

After being optimized with PSO, there was a significant improvement across all evaluation metrics. The 4% increase in accuracy indicates that the model became more precise in mapping flood risks. Furthermore, the improvement in recall demonstrates that the model became more sensitive in identifying flood risk cases, especially within minority classes that were previously often overlooked.

The overall metric improvement demonstrates that PSO effectively discovered a more optimal set of hyperparameters for the XGBoost model. By minimizing classification errors through iterative parameter updates, PSO improved the model's ability to generalize across different flood risk levels. The increase in recall, particularly for the minority class, reflects higher model sensitivity in recognizing less frequent flood events that were previously misclassified.

C. Feature Importance

After the XGBoost model was optimized using PSO, an analysis was conducted to evaluate the contribution weight of each feature using the feature importance method. This process aimed to identify which features had the most significant impact on flood risk prediction.

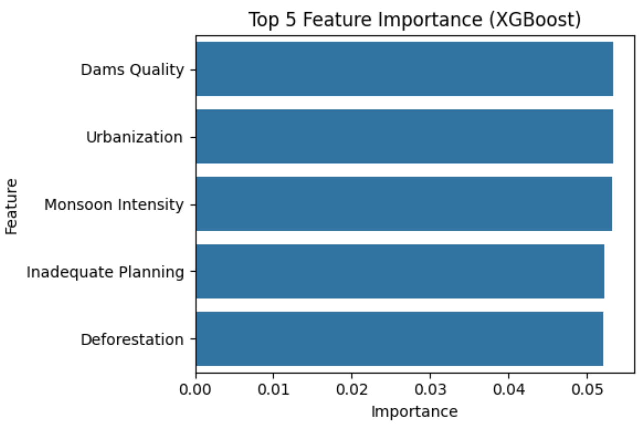

Figure 4 Top 5 Feature Importance

The graph illustrates that Rainfall is the dominant factor in the prediction, followed by the level of Urbanization and the quality of Drainage. This finding is logical, as areas with high rainfall and dense development tend to be more vulnerable to flooding. Meanwhile, features such as River Management and Soil Moisture also play an important role, as they influence the environment's natural ability to absorb water.

Although the PSO–XGBoost model achieved stable results, several classification errors were still observed. Most misclassifications occurred in areas with high rainfall but no actual flooding. This condition is likely due to elevated terrain or efficient drainage systems, which prevent inundation even under heavy precipitation. These findings indicate that additional topographical and hydrological parameters could further enhance prediction accuracy.

D. The Effect of Performance from PSO Result

To examine the model's sensitivity to the number of decision trees, an experiment was conducted by increasing the n_estimators parameter value from 300 to 550. The other parameters (max_depth and learning_rate) were kept constant, as they showed no significant effect on accuracy during testing.

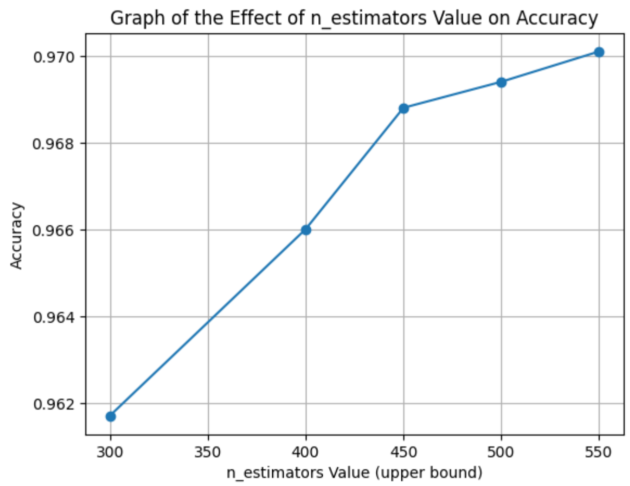

Figure 5 Graph of the Effect of n_estimators Value on Accuracy

The graph shows that increasing the n_estimators value directly impacts the improvement of the model's accuracy. This can be technically explained by the fact that the more trees (estimators) are used, the greater the model's opportunity to capture complex patterns within the data. However, after reaching around 500 estimators, the accuracy improvement began to slow down and tended to stagnate at 97.01% when the value reached 550. This indicates that the model has reached an optimal point, where adding more trees only increases computation time without providing significant accuracy gains.

E. Extebded Model Evaluation Using AP. Map AND FPR

1. Macro Precision-Recal Curve Graph

To complement the precision–recall evaluation, the Receiver Operating Characteristic (ROC) curve was also analyzed to examine the balance between true positive and false positive rates. The PSO–XGBoost model achieved an average Area Under the Curve (AUC) score above 0.95 across all classes, indicating excellent discriminative performance in differentiating flood risk categories.

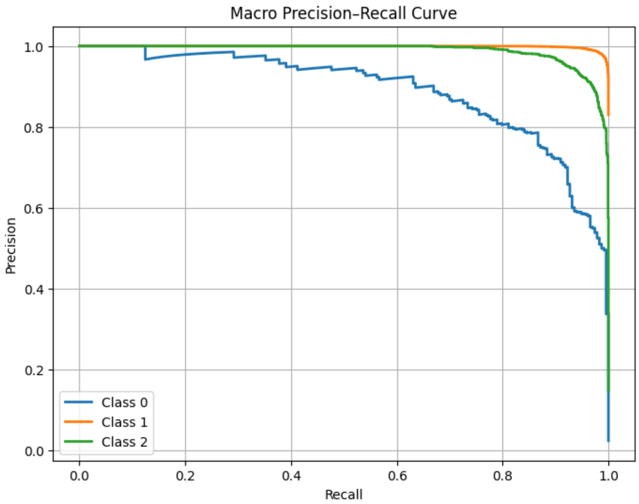

Figure 6 Macro Precision–Recall and ROC Curve for Flood Risk Classification

The Macro Precision-Recall (PR) Curve illustrates the relationship between precision and recall for each class. Classes 1 and 2 produce stable curves that approach the top-right corner, indicating excellent performance. Class 0 shows a slight decline in the middle section, suggesting that the model is slightly less precise in distinguishing the low-risk class. However, overall, the model successfully maintains a good balance between precision and recall.

2. Evaluation of Average Precision (AP) and Mean Average Precission (mAP)

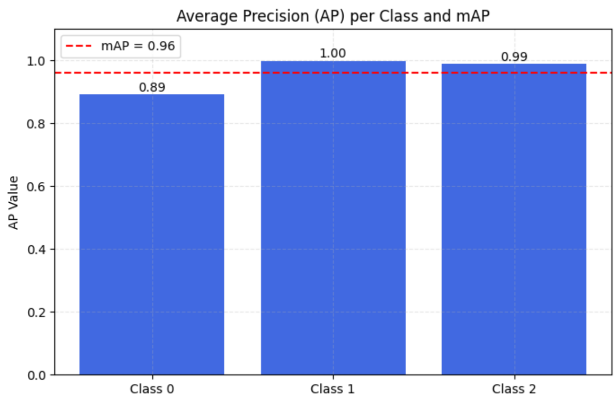

Figure 7 Average Precision (AP) per Class and Map

TABLE 7
AVERAGE PRECISION (AP) PER CLASS AND MEAN AVERAGE PRECISION (mAP)

Kelas	Nilai AP
Kelas 0	85.09
Kelas 1	99.80
Kelas 2	97.78

Additional evaluation was conducted using the Average Precision (AP) metric for each class, along with the mean Average Precision (mAP) as the overall average score. Based on the testing results, the AP scores for classes 0, 1, and 2 were 85.09%, 99.80%, and 97.78%, respectively. The mean mAP reached 94.22%, indicating that the model does not only perform well for a single class but also demonstrates stability across all flood risk categories. The bar chart visualization shows that the AP score for class 1 is nearly perfect, while class 0 remains challenging due to its underrepresented data. The mAP line on the graph serves as a general reference for overall model performance. These results further reinforce the conclusion that the XGBoost model optimized with PSO is capable of delivering consistent and balanced multi-class classification performance.

3. False Positive Rate (FPR)

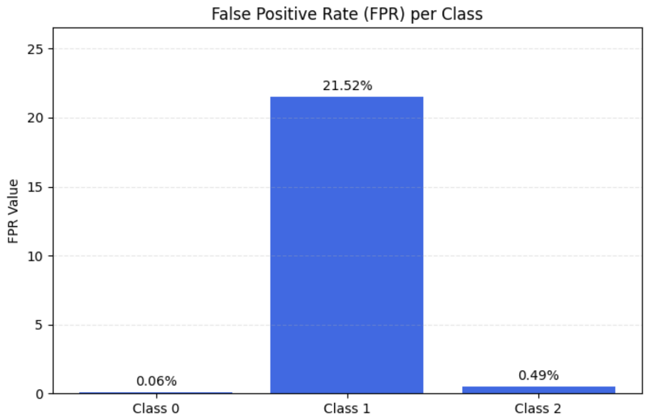

Figure 8 False Positive Rate (FPR) per Class

The False Positive Rate (FPR) is calculated from the confusion matrix to determine how often the model incorrectly classifies a negative class as positive. The evaluation results show that the FPR for Class 1 is higher compared to Classes 0 and 2. This indicates that the model tends to misclassify instances of Class 1 slightly more often than the others. However, in general, the FPR values remain within acceptable limits and do not significantly impact the overall performance of the model.

4. Combined Evaluation

The model's performance was evaluated using two complementary approaches to obtain a comprehensive understanding. First, a classical evaluation was performed using metrics such as precision, recall, and F1-score, which are calculated based on the final predicted class against the actual labels using a confusion matrix. This method provides assessment based on a single classification decision threshold.

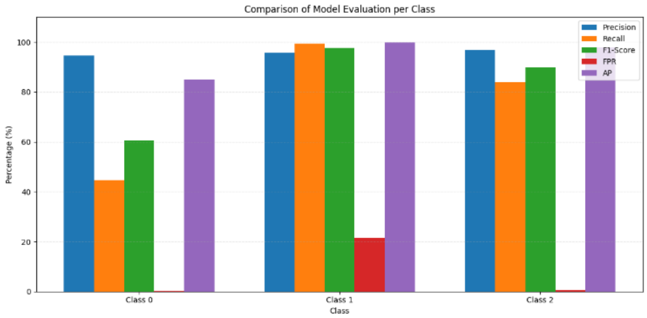

Figure 9 Comparison of Model Evaluation per Class

TABLE 8
COMPREHENSIVE MODEL EVALUATION PER CLASS

Kelas	Precission	Recall	F1-Score	FPR	AP
Kelas 0	94.55	44.64	60.64	0.06%	85.09
Kelas 1	95.75	99.42	97.55	21.52%	99.80
Kelas 2	96.70	83.86	89.82	0.49%	97.78

Second, a curve-based evaluation was conducted by calculating the Average Precision (AP) for each class and the overall Mean Average Precision (mAP). AP takes into account the area under the precision–recall curve, thereby reflecting the model's performance across various prediction thresholds.

The differences in scores resulting from the two approaches are expected due to the differing evaluation principles. Nonetheless, both methods consistently indicate that the model performance has improved and stabilized after the application of PSO-based optimization.

Overall, the integration of traditional metrics such as accuracy, precision, recall, and F1-score with curve-based evaluations like AUC–ROC and mAP provides a broader view of model performance. The high and consistent values across all metrics confirm that the PSO–XGBoost approach not only improves accuracy but also ensures stability and robustness in multi-class flood risk prediction.

5. Comparison with Other Algoriyhms

The experiments were conducted on four classification models, namely XGBoost, XGBoost with PSO, SVM, and Random Forest. The evaluation results are presented using accuracy, precision, recall, and F1-score metrics, as shown table 9.

TABLE 9
PERFORMANCE COMPARISON OF EVALUATION METRICS BETWEEN ALGORITHMS

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	93%	89%	63%	70%
XGBoost + PSO	97%	95%	81%	87%
SVM	98%	94%	88%	91%
Random Forest	83%	82%	55%	61%

Although SVM achieved the highest accuracy and F1-score, this model presents a drawback in terms of computational efficiency. The feature importance evaluation process using SVM can take more than 30 minutes, making it less ideal for real-time systems. In contrast, XGBoost optimized with PSO provides nearly comparable accuracy while offering superior time efficiency. With 97% accuracy, a high F1-score, and significantly faster execution time, this model is considered more stable and balanced.

TABLE 10
EXECUTION TIME COMPARISON BETWEEN MODELS

Model	Execution Time
XGBoost	2 Seconds
XGBoost + PSO	5 Minutes 30 Seconds
SVM	2 Minutes 40 Seconds
Random Forest	5 Seconds

While SVM excels in terms of accuracy, the high computational cost becomes a significant concern. On the other hand, XGBoost + PSO offers a balance between high accuracy and shorter processing time compared to SVM, as well as more consistent results compared to Random Forest. This makes XGBoost + PSO the most practical model to be implemented in predictive systems that require both speed and accuracy simultaneously.

IV. CONCLUSION

This study successfully developed a flood risk prediction model using the XGBoost algorithm optimized through the Particle Swarm Optimization (PSO) method by searching for optimal values of the n_estimators, max_depth, and learning_rate hyperparameters, resulting in an accuracy improvement of up to 97.01%. In addition to accuracy, the model's performance enhancement is also reflected in the increase of the minority class recall from 21% to 57%, as well as the F1-score from 41% to 70%. Further evaluation shows the Average Precision (AP) scores per class reached 85.09%, 99.80%, and 97.78%, with a mean Average Precision (mAP) of 94.22%. Moreover, the False Positive Rate (FPR) values were very low for Classes 0 and 2. The Macro Precision–Recall Curve visualization further confirms the model's stable performance in multi-class classification. When compared to other algorithms such as SVM and Random Forest, the XGBoost + PSO model offers a balanced performance between classification accuracy and computational efficiency, making it highly potential for implementation in real-time flood detection systems. For future research, this model can be further explored using other optimization methods, tested on real-time datasets, and applied to early warning systems that adapt to climate change and environmental conditions.

REFERENCES

[1] H. Farhadi and M. Najafzadeh, "Flood Risk Mapping by Remote Sensing Data and Random Forest Technique," *Water*, vol. 13, no. 21, p. 3115, Nov. 2021, doi: 10.3390/w13213115.

[2] H. Khodaei, F. Nasiri Saleh, A. Nobakht Dalir, and E. Zarei, "Future flood susceptibility mapping under climate and land use change," *Sci Rep*, vol. 15, no. 1, Apr. 2025, doi: 10.1038/s41598-025-97008-0.

[3] M. Ma *et al.*, "XGBoost-based method for flash flood risk assessment," *Journal of Hydrology*, vol. 598, p. 126382, Jul. 2021, doi: 10.1016/j.jhydrol.2021.126382.

[4] M. M. A. Syeed, M. Farzana, I. Namir, I. Ishrar, M. H. Nushra, and T. Rahman, "Flood Prediction Using Machine Learning Models," in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey: IEEE, Jun. 2022, pp. 1–6. doi: 10.1109/HORA55278.2022.9800023.

[5] G. Al-Rawas, M. R. Nikoo, M. Al-Wardy, and T. Etri, "A Critical Review of Emerging Technologies for Flash Flood Prediction: Examining Artificial Intelligence, Machine Learning, Internet of Things, Cloud Computing, and Robotics Techniques," *Water*, vol. 16, no. 14, p. 2069, Jul. 2024, doi: 10.3390/w16142069.

[6] S.-E. Ryu, D.-H. Shin, and K. Chung, "Prediction Model of Dementia Risk Based on XGBoost Using Derived Variable Extraction and Hyper Parameter Optimization," *IEEE Access*, vol. 8, pp. 177708–177720, 2020, doi: 10.1109/access.2020.3025553.

[7] A. Ogunleye and Q.-G. Wang, "XGBoost Model for Chronic Kidney Disease Diagnosis," *IEEE/ACM Trans. Comput. Biol. and Bioinf.*, vol. 17, no. 6, pp. 2131–2140, Nov. 2020, doi: 10.1109/TCBB.2019.2911071.

[8] N. Ghatasheh, I. Altaharwa, and K. Aldebei, "Modified Genetic Algorithm for Feature Selection and Hyper Parameter Optimization: Case of XGBoost in Spam Prediction," *IEEE Access*, vol. 10, pp. 84365–84383, 2022, doi: 10.1109/access.2022.3196905.

[9] B. Liu, H. Lin, Y. Chen, and C. Yang, "Prediction of Rock Unloading Strength Based on PSO-XGBoost Hybrid Models," *Materials*, vol. 17, no. 17, p. 4214, Aug. 2024, doi: 10.3390/ma17174214.

[10] D.-V. Hoang and Y.-A. Liou, "Elevating Flash Flood Prediction Accuracy: A Synergistic Approach with PSO and GA Optimization," Dec. 12, 2024, *Copernicus GmbH*. doi: 10.5194/nhess-2024-215.

[11] J. Yu, W. Zheng, L. Xu, L. Zhangzhong, G. Zhang, and F. Shan, "A PSO-XGBoost Model for Estimating Daily Reference Evapotranspiration in the Solar Greenhouse," *Intelligent Automation & Soft Computing*, vol. 26, no. 5, pp. 989–1003, 2020, doi: 10.32604/iasc.2020.010130.

[12] D. Kim, K. Lee, J. G. Hwang-Bo, H. S. Kim, and S. Kim, "Development of the Method for Flood Water Level Forecasting and Flood Damage Warning Using an AI-based Model," *J. Korean Soc. Hazard Mitig*, vol. 22, no. 4, pp. 145–156, Aug. 2022, doi: 10.9798/kosham.2022.22.4.145.

[13] H. C. Kilinc, "Daily Streamflow Forecasting Based on the Hybrid Particle Swarm Optimization and Long Short-Term Memory Model in the Orontes Basin," *Water*, vol. 14, no. 3, p. 490, Feb. 2022, doi: 10.3390/w14030490.

[14] W. Liang, S. Luo, G. Zhao, and H. Wu, "Predicting Hard Rock Pillar Stability Using GBDT, XGBoost, and LightGBM Algorithms," *Mathematics*, vol. 8, no. 5, p. 765, May 2020, doi: 10.3390/math8050765.

[15] M. El Haou, M. Ourribane, M. Ismaili, S. Krimissa, and M. Namous, "Enhancing urban flood hazard assessment: a comparative analysis of frequency ratio and xgboost models for precision risk mapping," *Ecol. Eng. Environ. Technol.*, vol. 26, no. 3, pp. 286–300, Mar. 2025, doi: 10.12912/27197050/200141.

[16] S. Puttinaovarat and P. Horkaew, "Flood Forecasting System Based on Integrated Big and Crowdsource Data by Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 5885–5905, 2020, doi: 10.1109/access.2019.2963819.

[17] X. Dairu and Z. Shilong, "Machine Learning Model for Sales Forecasting by Using XGBoost," in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, Guangzhou, China: IEEE, Jan. 2021, pp. 480–483. doi: 10.1109/iccece51280.2021.9342304.

[18] A. B. Parsa, "Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis," *Accident Analysis and Prevention*, 2020.

[19] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network Intrusion Detection Based on PSO-Xgboost Model," *IEEE Access*, vol. 8, pp. 58392–58401, 2020, doi: 10.1109/ACCESS.2020.2982418.

[20] E. Gul, "Urban flood hazard assessment using FLA-optimized boost algorithms in Ankara, Türkiye," *Appl Water Sci*, vol. 15, no. 4, Apr. 2025, doi: 10.1007/s13201-025-02424-2.