BORNEO JOURNAL OF SCIENCES AND TECHNOLOGY (BJoST)
VOL. : 7, ISSUE: 2, 99-111
DOI: http://doi.org/10.35370/bjost.2025.7.2-08
e-ISSN: 2672-7439 © 2018, UTS Publisher.

University of Technology Sarawak

ORIGINAL ARTICLE

A Comprehensive Approach to Flood Detection Using a Hybrid Artificial Intelligence Framework and Geographic Information System (GIS)

[1]Siti Maesaroh, [1,3*]Mazlan Mohamed and [2]Afiyati Afiyati

[1]Faculty of Data Science and Computing, Universiti Malaysia Kelantan, Pengkalan Chepa 16100 Kota Bharu Malaysia
[2]Faculty of Computer Science, Mercu Buana University, Joglo, Kecamatan Kembangan, Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta, Indonesia
[3]Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.

ABSTRACT - This study implements the Random Forest model for a flood detection and early warning system by integrating rainfall and water level data. The dataset includes three levels of flood warning status, with a distribution of over 4,000 cases for statuses 1 and 2, and fewer than 500 cases for status 3. A scatter plot visualization shows a clear correlation between rainfall intensity and water level, where status 3 (the highest warning level) predominantly occurs at high rainfall intensity. The optimized Random Forest model, using GridSearchCV, demonstrates excellent performance with 100% accuracy for statuses 1 and 2, and 99% accuracy for status 3. The confusion matrix confirms the model's reliability, with only one misprediction out of 1,757 samples, where one status 3 cases was predicted as status 2. Decision boundary analysis reveals the model's capability to distinguish flood risk zones based on rainfall intensity and water level characteristics. These results demonstrate the effectiveness of a machine learning approach in developing an accurate and reliable flood early warning system.

ARTICLE HISTORY
Received: 15 Jan 2025
Revised: 14 Apr 2025
Accepted: 05 May 2025
Published: 31 July 2025

KEYWORDS
Random Forest,
Geographic Information System,
Flood Detection,
Machine Learning,
Early warning system

INTRODUCTION

Flood events globally often have a dramatic impact on urban communities. Time becomes a critical factor during floods to evacuate vulnerable people, minimize socio-economic, ecological, and cultural impacts, and restore societal conditions quickly [1]. Floods are defined as the overflowing of water onto normally dry land or an increase in water levels that significantly affect human life [2]. This natural phenomenon is one of the most common, often causing significant financial damage to goods and property and endangering human lives [3]. To reduce its impact, society must continue to utilize the latest technological innovations [4]. Based on research findings, a rainfall detection prototype using a tipping bucket sensor can measure heavy and moderate rainfall intensity with an accuracy of up to 81.5%. The proposed system integrates various sensors, such as the DHT22 for monitoring air temperature and humidity, an Ombrometer for measuring rainfall, a Water Flow Sensor for measuring water flow, and an Ultrasonic Sensor for detecting water level changes. Data from these sensors are collected in real time and analyzed to predict potential flooding [5].

Currently, the Internet of Things (IoT) is one of the main focuses of application development because it enables devices to work independently to meet various user needs [6]. In the context of flooding, a real-time flash flood detection system is essential to provide information to the public so they can take appropriate actions. Conventional methods such as newspapers, radio, TV, or public announcements are often too slow to deliver early warnings to the community [7]. However, technological advancements over

*Corresponding Author: Mazlan Mohamed. Universiti Malaysia Kelantan (UMK), email: mazlan.m@umk.edu.my

the past decade have opened significant opportunities, including the use of camera image data and wireless sensors from IoT networks to improve flood management [8].

One approach used in data analysis is machine learning models such as Random Forest. This model is capable of handling large volumes of high-dimensional data and can perform classification and regression effectively [9].Random Forest is an ensemble learning method that constructs multiple decision trees and combines their results to improve accuracy and stability. This approach makes it highly effective in handling large, complex datasets, including flood-related parameters like rainfall intensity and water levels [10]. In recent studies, Random Forest demonstrated exceptional accuracy in flood classification tasks, achieving rates as high as 99.21%. The algorithm's robustness lies in its ability to generalize well even with diverse data inputs [11]. Furthermore, its ability to identify distinct flood levels—ranging from low, moderate, to severe—makes it a vital tool in flood risk management. By integrating IoT technologies and advanced machine learning models like Random Forest, flood management can become more efficient, ensuring timely warnings and reducing disaster impacts.

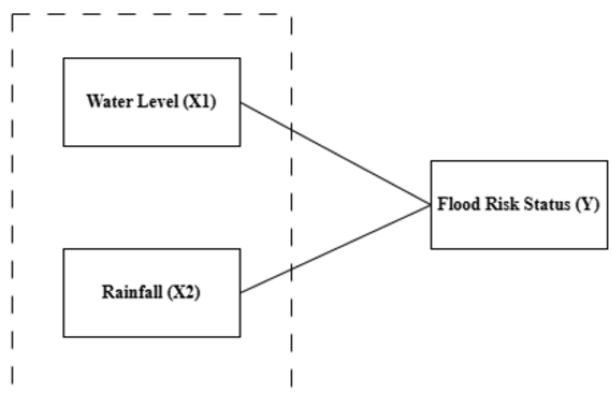

Figure 1. Research Framework

The conceptual framework in this research is designed to understand the relationship between variables that influence the risk of flooding. This research focuses on analyzing several environmental and meteorological factors that contribute to increasing or decreasing flood risk in the observed area. The diagram above shows that there are three independent variables that affect the dependent variable, namely flood risk status. The variables are explained as follows:

1. Independent Variables (X):
   - Water Level (X1)
     Water level is one of the key indicators to recognize potential flooding. When the water surface in rivers or lakes rises, the likelihood of flooding also increases. This information is obtained from routine water level observations.
   - Rainfall (X2)
     A sudden spike in rainfall at a certain time is a major factor that increases river flow. Heavy rain in a short period can accelerate the risk of flooding.

2. Dependant Variable (Y):
   - Flood Risk Status
     This status is the result of the analysis conducted using independent variables. Flood risk levels are divided into several categories, namely low, medium, and high. This variable serves as an indicator of the level of threat faced by the area under study.

Presents significant innovation in flood detection and prediction through the integration of Random Forest models with IoT and GIS sensor data. This study successfully developed a three-level classification system of flood warning status (Safe, Alert, Hazard) with an extraordinary accuracy rate of 99-100%, validated using a comprehensive dataset for one year. The main advantages of this study lie in parameter optimization using GridSearchCV, decision boundary visualization that improves model interpretability, and a hybrid approach that combines IoT technology for real-time data collection, AI algorithms for predictive processing, and GIS for spatial analysis. Although the data distribution for the highest warning status is relatively small (less than 500 cases), the model still shows excellent performance with only one prediction error out of 1,757 test samples, making this system a practical and reliable solution for implementation in various geographic conditions with a real impact on disaster management and public safety.

MATERIALS AND METHODOLOGY

The session covers the components, solar panels and the LDR sensor module used for the study to improve the output, the brighter or higher intensity the light source. The lower the lumen output, the less bright or lower intensity the light source [13]. With a light source of 1,000 *lumens*. If all these 1,000 *lumens* are spread over an area of 1 square meter, there will be an illumination of 1,000 *lux*. If the lumens spread over 10 square meters the illumination or lux would diminish to a less intense and grader 100 *lux*. Equation (1) shows the computation of the illumination. Table 1 presents examples of light levels common to the natural light source released by the Illuminating photovoltaic energy harvest of the solar-powered IoT device. (sample). Three promising research areas for the future: (1) time series data integration for dynamic prediction, (2) transfer learning for model generalization across different regions, and (3) explainable AI approaches to increase trust and adoption in practical disaster management [14]. The main advantage of this approach is its ability to adapt to unpredictable rainfall patterns and changing hydrological conditions in real time. Case studies in three different watersheds demonstrate the generality and robustness of the model across a range of flood scenarios [15]. One of the main challenges in applying deep learning to flood detection is the limited availability of training data, especially in areas that rarely experience flooding or have minimal monitoring infrastructure [16].

Dataset Explanation

	timestamp	rainfall_mm/hour	water_height_m	status
0	1/1/2024 0:00	0.000000	656.181018	1
1	1/1/2024 1:00	0.000000	742.607146	2
2	1/1/2024 2:00	0.000000	709.799091	2
3	1/1/2024 3:00	0.000000	689.798773	2
4	1/1/2024 4:00	0.000000	623.402796	1
...	...	...	...	...
8779	12/31/2024 19:00	0.000000	650.845489	1
8780	12/31/2024 20:00	0.000000	621.462209	1
8781	12/31/2024 21:00	0.000000	728.683378	2
8782	12/31/2024 22:00	0.000000	646.329135	1
8783	12/31/2024 23:00	4.153221	698.828311	2

Figure 2. Dataset Explanation

This data includes information on rainfall levels, water levels, and flood warning status taken hourly over a one-year period (2024). The timestamp column records the measurement time, rainfall_mm/hour depicts rainfall in millimeters per hour, and water_height_m shows the water height in meters. The status column is a label that reflects the flood warning level, consisting of several categories such as "safe", "alert", or "hazard" defined by numbers 1 to 3. This data is crucial to analyze the prediction of flood conditions based on rain patterns and water level height.

Introduction of Random Forest Algorithm

Random Forest is an ensemble learning method designed to improve the performance and robustness of classification models by combining multiple decision trees. Unlike a single decision tree, which may suffer from overfitting or sensitivity to noise, a Random Forest leverages the diversity of multiple trees to create a model that generalizes better to unseen data. This algorithm can also produce lower errors, provides excellent accuracy in classification, can handle very large training datasets, and is effective in addressing incomplete data [12]. This is achieved through two key techniques: bootstrap aggregating (bagging) and random feature selection. Bagging involves training each tree on a different subset of the data, created by random sampling with replacement. Random feature selection ensures that only a random subset of features is considered at each split in the tree, reducing correlation between trees and enhancing the overall accuracy of the model.

A fundamental concept in the splitting process of decision trees, which is central to Random Forest, is the Gini index. The Gini index measures the impurity of a node, providing a criterion to decide the best split at each step. The formula for the Gini index is given as:

$$Gini\ Index = 1 - \sum_{i=1}^{n} (P_i)^2$$

Where n is the total number of classes, and P_i is the proportion of instances belonging to class i at a given node that represent the probability of positive and negative on that class. Therefore, we can write the formula on the selected class with the expression below:

$$Gini\ Index_i = 1 - \left[(P_+)^2 + (P_-)^2\right]$$

A node with a Gini index of 0 is considered pure, meaning all data points at that node belong to a single class. During the construction of a tree, the algorithm evaluates all possible splits and selects the one that minimizes the Gini index, thereby reducing impurity and enhancing the separation between classes.

The Concept of How to Process the Data

The Random Forest algorithm is utilized to analyze data featuring two key predictors: rainfall intensity (rainfall_mm/hour) and water level (water_height_m). These features are critical indicators for predicting flood warning levels that classified into three categories: 1 (Safe), 2 (Alert), and 3 (Hazard). Random Forest, as an ensemble learning method, constructs multiple decision trees to explore the relationship between these variables and their contribution to flood risk. This approach leverages the importance of each feature to identify thresholds and patterns associated with each warning class. Using measures such as the Gini index, the model determines optimal splits at each decision node, hence improving the classification process. The ensemble nature of Random Forest aggregates predictions from individual trees to enhance accuracy and reliability. By systematically combining rainfall intensity and water level data, this method offers a robust framework for flood warning level status prediction that can support timely interventions and informed decision-making in flood management systems.

RESULTS AND DISCUSSION

This approach offers a comprehensive solution to the problem of early detection and flood prediction that is more accurate and proactive. By utilizing appropriate sensors and machine learning algorithms, the system is able to provide flood warnings with greater precision and reduce adverse impacts on society.

Results of Data Processing Visualization

```
#scatter plot based on class
sns.scatterplot(data=data, x='rainfall_mm/hour', y='water_height_m', hue=label, palette='viridis',
alpha=0.7)
plt.title("Flood Warning Level Status in Area X")
plt.xlabel("Rainfall (mm/hour)")
plt.ylabel("Water Level (m)")
plt.legend(title="Status")
plt.show()
```

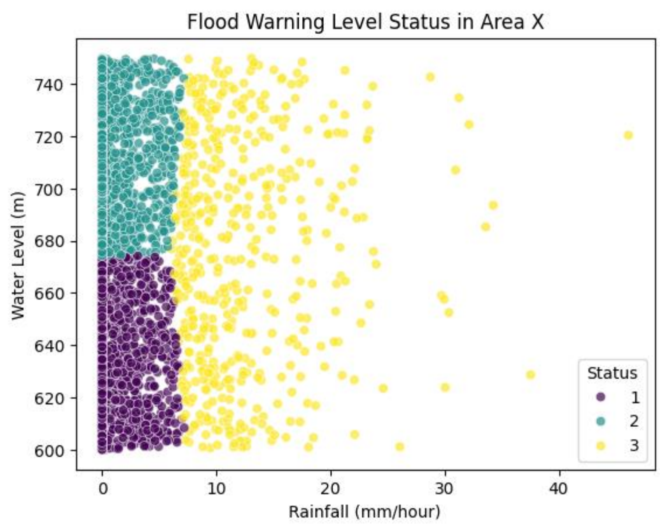

Figure 3. Graphic Flood Warning level Status in Area X

This scatter plot visualization shows the relationship between rainfall (mm/h) and water level (m) based on flood warning conditions in Area X. Each status category (1, 2, and 3) is represented by a different color at the points in the plot. From the illustration, it appears that status 1 (purple) is more common at low water elevation, while status 2 (green) is at the intermediate level, and status 3 (yellow) tends to appear at high rainfall intensity despite varying water elevation. Most of the data with low precipitation levels show a more varied distribution of status compared to high precipitation, where the dominance of status 3 is more pronounced. This shows that rain has a crucial role in determining the flood warning status.

```
#class distribution bar chart
class_counts = data[label].value_counts()
class_counts.plot(kind='bar', color=['skyblue', 'salmon', 'lime'], alpha=0.7)
plt.title("Distribution of Flood Warning Level Status")
plt.xlabel("Flood Warning Level")
plt.ylabel("Cases")
```

```
plt.show()
```

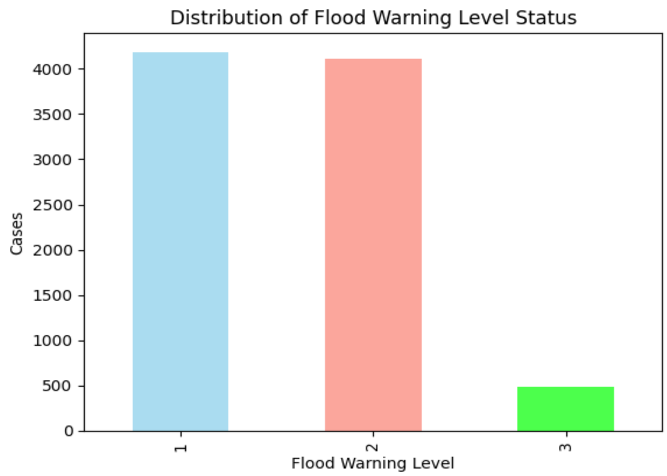

Figure 4. Distribution of Flood Warning Level Status

This visualization utilizes a bar chart to illustrate the distribution of cases by flood warning category. Alert level 1 recorded the highest number of cases of more than 4000, while alert level 2 had almost the same number, around 4000 cases. On the other hand, alert level 3 recorded a much smaller number of cases, below 500 cases. This diagram shows that the majority of incidents occurred at lower alert levels (levels 1 and 2), while incidents at level 3 occurred much less.

```
#scatter Plot Decision Boundary
if X_scaled.shape[1] == 2:  #scatter plot is only possible if the 2D data
  fig, ax = plt.subplots(figsize=(12, 6))

  #create an area plot for the decision boundary model
  DecisionBoundaryDisplay.from_estimator(
    best_rf,
    X_test,
    response_method='predict',
    grid_resolution=500,
    cmap=plt.cm.viridis,
    ax=ax,
    alpha=0.8
  )

  #scatter plot of test data and original label
  scatter = ax.scatter(
    X_test[:, 0],
    X_test[:, 1],
    c=y_test,
    edgecolor='k',
    cmap=plt.cm.viridis,
    label='Test Data'
```

```
    )

    ax.set_title('Decision Boundary of Flood Warning Level in Area X')
    ax.set_xlabel('Rainfall (mm/hour)')
    ax.set_ylabel('Water Level (m)')
    ax.legend(loc='lower right')
    plt.show()
else:
    print("Decision boundary plot can only be done for data with 2 features.")
```

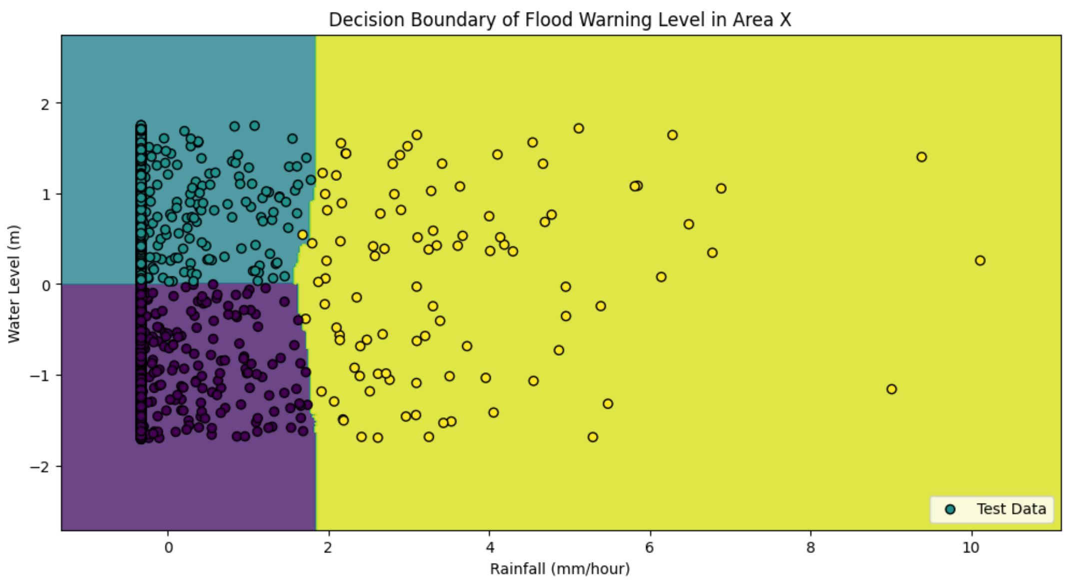

Figure 5. Graphic Decision Boundary of Flood Warning Level In Area X

This visualization utilizes a scatter plot with a colormap to illustrate the decision boundary of the flood warning level prediction model in Area X. The graph shows the relationship between rainfall (mm/h) on the X-axis and water level (m) on the Y-axis. The dots on the graph represent the experimental data, with colors reflecting the original labels. The existence of decision limits shows how the model differentiates data into categories based on rainfall and water height features. This visualization describes the model's ability to classify data based on its characteristics.

The Implementation of Random Forest with Python

```
#import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, ConfusionMatrixDisplay
from sklearn.preprocessing import StandardScaler
from sklearn.inspection import DecisionBoundaryDisplay
```

```
#import dataset
from google.colab import files
uploaded = files.upload()

#read and display data
flooddetection = pd.read_csv('data_clean_flood_detection.csv')
print(flooddetection)

#view data-related information
flooddetection.info()

#use relevant numeric features and labels
data = flooddetection[['rainfall_mm/hour', 'water_height_m', 'status']]
features = ['rainfall_mm/hour', 'water_height_m']
label = 'status'
X = data[features]
y = data[label]

#scatter plot based on class
sns.scatterplot(data=data, x='rainfall_mm/hour', y='water_height_m', hue=label, palette='viridis',
alpha=0.7)
plt.title("Flood Warning Level Status in Area X")
plt.xlabel("Rainfall (mm/hour)")
plt.ylabel("Water Level (m)")
plt.legend(title="Status")
plt.show()

#class distribution bar chart
class_counts = data[label].value_counts()
class_counts.plot(kind='bar', color=['skyblue', 'salmon', 'lime'], alpha=0.7)
plt.title("Distribution of Flood Warning Level Status")
plt.xlabel("Flood Warning Level")
plt.ylabel("Cases")
plt.show()

#fine Tune Model Random Forest
#data normalization
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

#split data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42,
stratify=y)
```

```
# Grid Search with Cross-Validation
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

rf = RandomForestClassifier(random_state=42)
grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='f1_weighted', n_jobs=-1)
grid_search.fit(X_train, y_train)

best_rf = grid_search.best_estimator_
print("Best Parameters:", grid_search.best_params_)

#confusion Matrix and Classification Report
y_pred = best_rf.predict(X_test)
print("Classification Report:\n", classification_report(y_test, y_pred))

ConfusionMatrixDisplay.from_estimator(best_rf, X_test, y_test, cmap='Blues')
plt.title("Confusion Matrix")
plt.show()

#scatter Plot Decision Boundary
if X_scaled.shape[1] == 2:  #scatter plot is only possible if the 2D data
    fig, ax = plt.subplots(figsize=(12, 6))

    #create an area plot for the decision boundary model
    DecisionBoundaryDisplay.from_estimator(
        best_rf,
        X_test,
        response_method='predict',
        grid_resolution=500,
        cmap=plt.cm.viridis,
        ax=ax,
        alpha=0.8
    )

    #scatter plot of test data and original label
    scatter = ax.scatter(
        X_test[:, 0],
        X_test[:, 1],
        c=y_test,
        edgecolor='k',
        cmap=plt.cm.viridis,
```

```
        label='Test Data'
    )

    ax.set_title('Decision Boundary of Flood Warning Level in Area X')
    ax.set_xlabel('Rainfall (mm/hour)')
    ax.set_ylabel('Water Level (m)')
    ax.legend(loc='lower right')
    plt.show()
else:
    print("Decision boundary plot can only be done for data with 2 features.")


#save the best model and its scaler
import joblib
joblib.dump(best_rf, 'best_rf_model.pkl')
joblib.dump(scaler, 'scaler.pkl')
print('Model and Scaler Saved')
```

The code analyzes and predicts flood alert levels based on rain and water level data. After importing the required libraries, the 'data_clean_flood_detection.csv' dataset is downloaded and loaded, then analyzed to show general information as well as data distribution. The associated numerical features are selected to be used as input to the model. Scatter plots are created to illustrate the distribution of data by class, while bar graphs show the distribution of the number of cases in each category. The data is normalized with the 'StandardScaler' and separated into training data and test data. The Random Forest model is applied to perform predictions, with optimal parameter search via GridSearchCV using F1-weighted metrics. After the training process, the model evaluation is carried out using a confusion matrix and a classification report. The decision limit model is visualized to observe the separation between classes on the test data. Finally, the best models and scalers are saved with a 'joblib' to be used again in the future.

Processing Results from Data Mining

```
#confusion Matrix dan Classification Report
y_pred = best_rf.predict(X_test)
print("Classification Report:\n", classification_report(y_test, y_pred))

ConfusionMatrixDisplay.from_estimator(best_rf, X_test, y_test, cmap='Blues')
plt.title("Confusion Matrix")
plt.show()
```

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	837
2	1.00	1.00	1.00	823
3	1.00	0.99	0.99	97
accuracy			1.00	1757
macro avg	1.00	1.00	1.00	1757
weighted avg	1.00	1.00	1.00	1757

Figure 6. Code Classification Report

The results of the classification report show that the applied Random Forest model shows very good performance in predicting flood warning status. All of the main metrics, namely precision, recall, and F1-score, almost got an ideal score (1.00) for each class, especially in grades 1 and 2, which had a higher amount of data (837 and 823 data). For class 3, the metric is somewhat lower (F1-score 0.99), but it still shows excellent performance despite the smaller amount of supporting data (97 data). The total accuracy of the model is 1.00, which indicates that the model can accurately predict almost any sample in the test data. The average values (macro avg and weighted avg) are also very high, indicating that the model is balanced in managing all classes. This performance indicates that the model has successfully identified patterns in the data without showing any signs of overfitting.

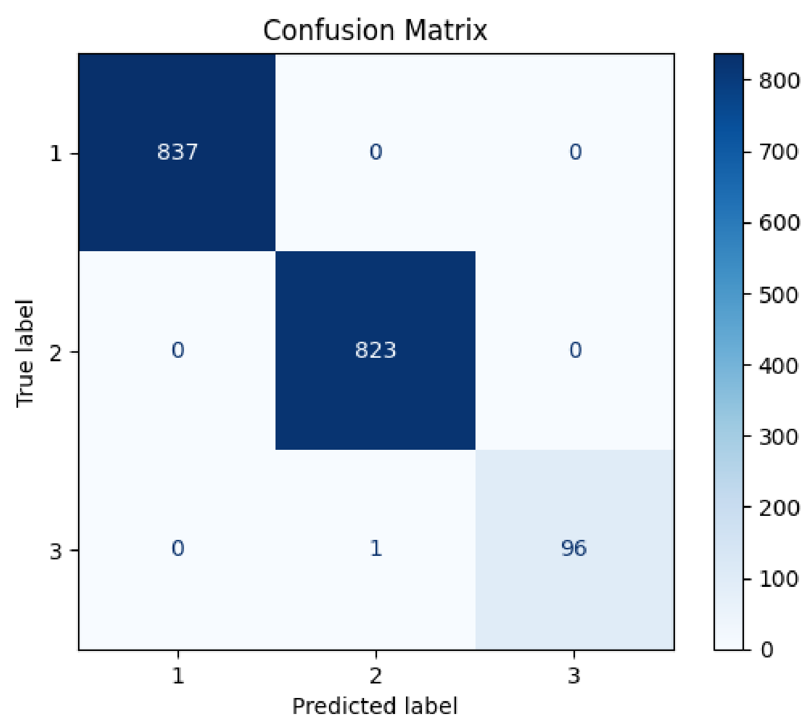

Figure 7. Graphic Confusion Matrix

The Confusion Matrix describes the performance of the Random Forest model in forecasting flood warning status. The main diagonals of the matrix (837, 823, and 96) reflect the total correct predictions for each class: 1, 2, and 3. There are no prediction errors for classes 1 and 2 (0 on the other rows), but for class 3 there is 1 prediction error where the actual state is 3 but is predicted as 2. Overall, this matrix shows that the model has a very high level of accuracy, with only one prediction error out of 1757 samples, which reflects the model's ability to distinguish flood warning status well.

CONCLUSION

This study successfully developed and validated a machine learning-based flood detection system with the Random Forest model which showed excellent performance with an accuracy rate of 100% in level 1 and 2 warning status, and 99% in level 3 status which only experienced one prediction error from a total of 1,757 test samples. Further analysis through the visualization of decision boundaries on the pattern of the relationship between the two variables to the level of flood risk has revealed that the combination of rainfall and water level variables has high significance in determining the flood warning status. The system offers a reliable solution for flood early warning systems and can provide a high level of confidence in supporting decision-making on disaster management. Nonetheless, the uneven distribution of data, particularly in high-risk status, indicates the need for additional data collection to improve the representation of such cases. Further research is recommended to consider the addition of other variables, such as topographic data, land use patterns, or additional meteorological parameters, to increase the complexity and reliability of the model. The success of this study shows the great potential of the application of machine learning in flood risk mitigation and makes a significant contribution to community protection through the optimization of artificial intelligence technology. This comprehensive approach combines the power of IoT, AI, and GIS to create a flood prediction system that is not only accurate but also practical and can be implemented across a wide range of geographic conditions, with a significant impact on disaster management and public safety.

ACKNOWLEDGEMENT

This research was made possible through the support and assistance of various parties. The author would like to express sincere and profound gratitude to all individuals and institutions who have contributed directly or indirectly to the successful completion of this study. Special appreciation is extended to the funding body for their generous support under the research grant No: R/IMAP01/A1500/00929A/006/2024/01438 or IMAP/2/2024/TK10/UMK/1, which provided the necessary resources and facilities to carry out this work effectively.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Van Ackere, S., Verbeurgt, J., De Sloover, L., Gautama, S., De Wulf, A. and De Maeyer, P. (2019). A review of the Internet of Floods: Near real-time detection of a flood event and its impact. *Water (Basel)*, 11(11), p.2275. https://doi.org/10.3390/w11112275

[2] Hashi, A.O., Abdirahman, A.A., Elmi, M.A., Hashi, S.Z.M. and Rodriguez, O.E.R. (2021). A real-time flood detection system based on machine learning algorithms with emphasis on deep learning. *International Journal of Engineering Trends and Technology*, 69(5), pp.249–256. https://doi.org/10.14445/22315381/IJETT-V69I5P232

[3] Suwarno, I., Ma'arif, A., Raharja, N.M., Nurjanah, A., Ikhsan, J. and Mutiarin, D. (2020). IoT-based lava flood early warning system with rainfall intensity monitoring and disaster communication technology. *Emerging Science Journal*, 4(Special Issue), pp.154–166. https://doi.org/10.28991/esj-2021-SP1-011

[4] Iqbal, U., Bin Riaz, M.Z., Zhao, J., Barthelemy, J. and Perez, P. (2023). Drones for flood monitoring, mapping and detection: A bibliometric review. *Drones*, 7(1), p.32. https://doi.org/10.3390/drones7010032

[5] Rizal, H.M., Warni, E., Angriawan, R., Hariadi, M., Arif, Y.M. and Maulina, D. (2024). Design of flood early detection based on the Internet of Things and decision support system. *Ingenierie des Systemes d'Information*, 29(3), pp.1183–1193. https://doi.org/10.18280/isi.290335

[6] Darwis, M., Al Banna, H.A., Aji, S.R., Khoirunnisa, D. and Natassa, N. (2023). IoT based early flood detection system with Arduino and ultrasonic sensors in flood-prone areas. *Jurnal Teknik Informatika*, 16(2), pp.133–140. https://doi.org/10.15408/jti.v16i2.32161
[7] Rashid, A.A., Ariffin, M.A.M. and Kasiran, Z. (2021). IoT-based flash flood detection and alert using TensorFlow. In: *2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, pp.80–85. https://doi.org/10.1109/ICCSCE52189.2021.9530926
[8] Arshad, B., Ogie, R., Barthelemy, J., Pradhan, B., Verstaevel, N. and Perez, P. (2019). Computer vision and IoT-based sensors in flood monitoring and mapping: A systematic review. *Sensors (Switzerland)*, 19(22). https://doi.org/10.3390/s19225012
[9] Sankaranarayanan, S., Prabhakar, M., Satish, S., Jain, P., Ramprasad, A. and Krishnan, A. (2020). Flood prediction based on weather parameters using deep learning. *Journal of Water and Climate Change*, 11(4), pp.1766–1783. https://doi.org/10.2166/wcc.2019.321
[10] Purwati, S.E. and Pristyanto, Y. (2024). Model Random Forest and Support Vector Machine for flood classification in Indonesia. *Sinkron*, 8(4), pp.2261–2268. https://doi.org/10.33395/sinkron.v8i4.13973
[11] Qamarani, L.S. and Riasetiawan, M. (n.d.). Klasifikasi level banjir menggunakan Random Forest dan Support Vector Machine. *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, x(x), pp.1–5.
[12] Primajaya, A. and Sari, B.N. (2018). Random forest algorithm for prediction of precipitation. *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)*, 1(1), pp.27–31. https://doi.org/10.24014/ijaidm.v1i1.4903
[13] Liu, X., et al. (2023). GAN-based hybrid approach for flood detection with limited training samples. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16, pp.5642–5654.
[14] Tehrany, M.S. and Kumar, L. (2022). Flood detection and flood risk mapping using machine learning, GIS, and remote sensing: Current advances and future perspectives. *Remote Sensing*, 14(5), p.1119.
[15] Chen, Y., et al. (2023). Deep reinforcement learning for real-time flood control and mitigation. *Water Resources Research*, 59(2), e2022WR033601.