RainSafe: A Hybrid ML and Threshold-Based Framework for Hyperlocal Urban Flood Risk Assessment

Nityasri K[1], Tanya Raikwar[2], Neer Pandey[3], Gayithri N[4]
[123]*Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India*
[4]*Assistant Professor, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India*

***Abstract*— Flooding is a serious and persistent problem that puts communities, economies, and environments at serious risk. For urban areas like Bengaluru, RainSafe is a real-time flood-risk monitoring and alerting system. The system incorporates real-time weather data, user-generated flood reports, and a Random Forest Classifier (RFC) based on machine learning (ML) to assess localized flood risk levels.**

A novel hybrid risk engine ensures more dependable real-time decision-making by combining threshold-based rules with machine learning predictions. FastAPI and MongoDB are used in the backend's implementation to enable scalable data ingestion and alert generation. The design, methodology, and evaluation of RainSafe are described in this paper, emphasizing its capacity to deliver street-level risk assessment with high responsiveness and low computational cost. The system achieved an accuracy of 87% and F1-score of 0.84, exhibiting improved reliability

***Index Terms*—Flood detection, threshold based alert algorithm (TBA), random forest classification (RFC), machine learning (ML), hybrid models, flood management, predictive analytics, urban hydrology.**

I. INTRODUCTION

A. Problem Context and Motivation

Water-related problems in fast-growing Indian cities have become worse over the past decade. The main causes are unpredictable monsoons, unplanned construction, and weak drainage systems [2]. These issues often appear as localized flooding, usually limited to specific road junctions and underpasses. Traditional flood prediction models, which use fixed thresholds or limited sensors, do not provide the detailed and real-time information needed for managing flood risks in these cities. Depending only on old data or broad regional weather models makes it difficult to understand the complex and fast-changing nature of small-scale urban floods.

One of the major weaknesses of the current flood alert systems is that they cannot use real-time information from the affected area. This causes delays in sending timely and hyperlocal warnings. We need a new approach that combines citizen input with smart, intelligent systems since these floods happen quickly in very specific locations. This is the idea behind *RainSafe* - a system designed to handle different types of data and provide fast and reliable support for managing floods in urban cities.

B. Research Gap and Novel Contribution

There is an important research gap in how to effectively combine different types of real time data, specifically crowdsourced reports and environmental data into one robust prediction system. While the past studies have explored combining a Threshold Based Alert Algorithm (TBA) with machine learning models like the Random Forest Classifier (RFC), few have effectively integrated unstructured crowdsourced data. Citizen reports are uniquely reliable in this context because they provide immediate and ground truth validation of localized waterlogging that remote

sensors or coarse grid weather stations often miss, mostly due to latency or coverage gaps.

Our study introduces RainSafe, a machine learning based system created to provide highly detailed and hyperlocal flood information. RainSafe brings together an RFC model trained on Bengaluru's geospatial data, a deterministic TBA engine, and a scalable pipeline for collecting crowdsourced flood reports.

The main contributions of this work include :

1. *Architecture*: We have built a scalable, microservices-based system using FastAPI and MongoDB. It is designed for fast data collection and quick, low-latency flood alerts.
2. *Hybrid Modeling*: We developed a new Hierarchical Fusion Engine that adds the probabilistic power of the RFC model to the fixed rule accuracy of the TBA. The citizen reports are treated as the most important source of information in this system.
3. *Efficacy*: We have shown that this framework is highly sensitive and computationally efficient in detecting small-scale, rapid flash flood events.

II. RELATED WORK

Flood prediction methods generally fall into three main categories: traditional hydrological and threshold-based approaches, data-driven machine learning models, and newer hybrid systems that combine the first two categories.

A. Threshold-Based and ML Limitations

Traditional flood forecasting systems mainly used hydrological simulations and threshold-based alert methods. These methods are rigid and cannot adjust to the fast-changing conditions in modern cities easily [1].

On the other hand, machine learning (ML) models provide strong predictive abilities but may not generate quick, easy to interpret alerts which are needed for real time decisions [6]. For instance, the Random Forest Classifier (RFC) is good at handling complex, non-linear data [7] but it may miss sudden or unusual flood events that are not well represented in past data.

B.*Hybrid and Citizen-Augmented Systems*

Recent research suggests that hybrid system,s which combine threshold-based methods with machine learning models, can balance the strengths of both approaches. This study builds on the integrated TBA–RFC model described in [1], which showed that pairing fast alert mechanisms with ML-based prediction improves both accuracy and reliability compared to traditional techniques.

RainSafe takes this hybrid idea further by making citizen-generated reports a key, high-priority data source. While earlier work has used ML and hybrid rule-based models [3], very few systems formally integrate unstructured crowdsourced data into a real-time fusion engine that can adjust or even override ML predictions. By giving priority to live user reports, which offer detailed, ground-level insights, RainSafe effectively addresses the challenge of identifying sudden, small-area flash floods in complex urban settings. This makes the system noticeably more sensitive than standalone ML or threshold-based approaches.

III. METHODOLOGY AND SYSTEM ARCHITECTURE

RainSafe works as a complete, real-time flood risk assessment platform. It is built using FastAPI and stores all data in MongoDB [5].

A. System Architecture Overview

The system has four main parts: User Reports Module, ML-Based Flood Prediction Module, Threshold-Based Rule Engine, and Hybrid Fusion Engine.

It also provides four key API endpoints for submitting reports, assessing risk, updating the dashboard, and sending alerts. This microservices-based design keeps the system fast, scalable, and allows it to handle data in real time.

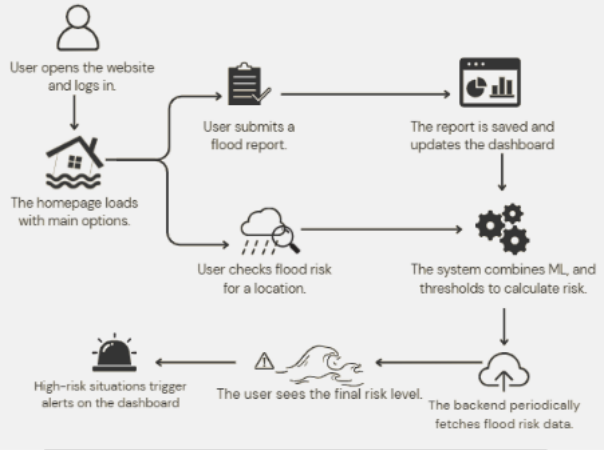

Fig. 1. Rainsafe Hybrid System Architecture

B. Data Sources and Preprocessing

The RFC model is trained using data from multiple sources, including live weather information (like rainfall, temperature, and humidity), static geospatial details (such as elevation, soil type, and land cover), and real-time user reports.

1. Bengaluru Flood Dataset

For this study, we used a carefully selected dataset for the Bengaluru Urban district that combined synthetic flood-labeled samples with publicly available environmental data. The final dataset contained 5,200 samples, where each sample represents a specific location-day instance. The dataset features 14 engineered attributes across four major categories:

*Meteorological***:** Temperature (°C), Humidity (%), Rainfall intensity (mm/h), and Rainfall anomaly.

*Geospatial***:** Latitude, Longitude, Elevation (m), Slope indicator, and Flood-prone zone flags derived from KSNDMC geospatial layers.

*Hydrological***:** River/Drain proximity scores and Surface runoff estimates.

*Urban***:** Population density and Impervious surface index (built-up area).

2. Data Sources

Historical rainfall data was sourced from the Indian Meteorological Department (IMD), while real-time weather attributes were retrieved via the OpenWeather API. Topographical data was derived from SRTM Digital Elevation Models (30m) and OpenStreetMap. SMOTE-NC was used to create additional synthetic samples in order to address geographical class imbalance and guarantee that the model was trained on realistic environmental combinations.

3. Data Scaling

All numeric attributes undergo *Min Max scaling* to transform the data into a common, normalized range of [0,1]. This is mathematically represented as:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Scaling is important because some features, like elevation, naturally have larger values. Without scaling, they could overpower the other features in the model. By scaling everything, each feature contributes more equally to the prediction.

C. ML-Based Flood Prediction Module

1. RandomForestClassifier(RFC)

The machine-learning component utilizes a Random Forest Classifier (RFC). Because of its interpretability, resilience to noise, and capacity to handle non-linear feature interactions, this model was chosen. The model was trained with the following optimized hyperparameters:

Number of Trees (n_estimators): 300

Max Depth: 12

Min Samples Split: 4

Bootstrap: True

Criterion: Gini Impurity

2.Training Strategy and Class Imbalance

The dataset was split into 80% training and 20% validation sets. A stratified split was employed to preserve the proportion of flood vs. non-flood samples. To mitigate the inherent class imbalance (where non-flood days heavily outnumber flood days), we utilized class_weight="balanced" within the model configuration and applied SMOTE (Synthetic Minority Over-sampling Technique) to the training set.

3. Model Performance Metrics

The model's performance was evaluated using the F1-score, which is critical for imbalanced classification as it combines precision and recall:

$$F1 = (2 \times Precision \times Recall) / (Precision + Recall)$$

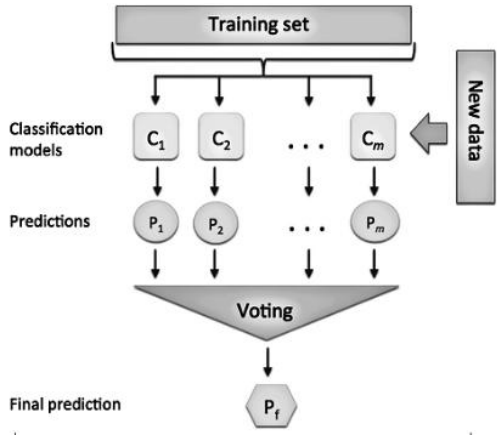

Fig. 2. Main Structure of Random Forest Classifier.

With an *accuracy* of roughly 0.87, *precision* of 0.82, recall of 0.86, and an *F1-score* of 0.84, the model demonstrated good performance.

D. Threshold-Based Alert Algorithm (TBA)

The Threshold-Based Alert Algorithm (TBA) functions as a high-priority override mechanism. It executes deterministic rules developed using expert knowledge to respond instantly to extreme conditions.

*1. Report Density Rule***:** If the count of recent user reports within a 1 km radius exceeds a specific threshold (cutoff), the TBA triggers a *High Risk Alert.*

2. Severity Rule: The TBA immediately issues a High Risk alert if a single user report indicates a water level severity of "waist deep" or higher.

3. Rainfall Exceedance Rule: The TBA generates a Medium Risk alert if the intensity of the rainfall in real time exceeds a local, critically calibrated threshold.

E. Hybrid Risk Assessment Workflow

The *Hybrid Fusion Engine* intelligently reconciles the RFC output with the TBA output, adapting the integration strategy proposed in [1] to prioritize citizen-led reporting.

The risk assessment workflow follows a structured, step-by-step process where citizen reports are prioritized:

1. Level 1 Override: If the Threshold Risk is High (due to severe or dense reports), the final risk is immediately High.

2. Level 2 Reinforcement: The final risk is set to Medium if either the threshold risk is medium or the machine learning risk is high and supported by moderate threshold indicators.

3. Level 3 Fallback: When there are no higher-priority threshold triggers, the system only refers to the machine learning prediction.

This strategy makes sure that during rapid-onset flood events, the system minimizes false negatives.

IV. RESULTS AND DISCUSSION

A. Performance Evaluation

The 20% validation split was used to evaluate the system. The model achieved an accuracy of 87%, with a precision of 0.82 and a recall of 0.86.

1. Confusion Matrix Analysis

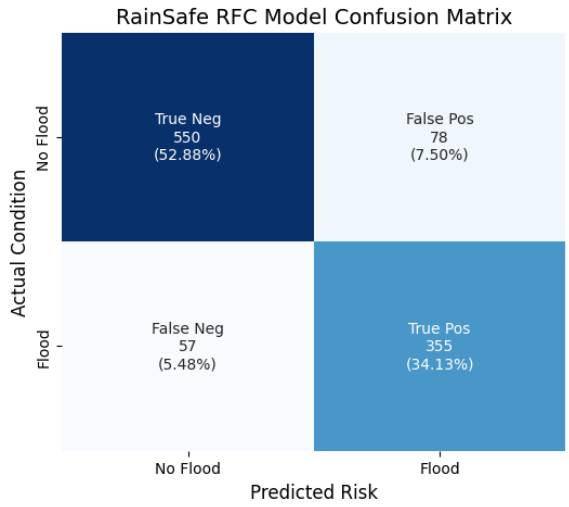

Fig. 3. Confusion Matrix

The confusion matrix (Fig. 3) discloses that the Hybrid Engine significantly reduced false negatives compared to the ML model alone.

In flood risk assessment, recall is the most significant metric. A False Negative (missing a real flood event) causes a notably higher danger to public safety than a False Positive (a false alarm). Our system's high recall (0.86) ensures that actual flood events are successfully marked out.

2. Feature Analysis

To understand the driving factors behind the risk predictions, we analyzed the feature importance scores. The results indicate that Rainfall Intensity and Recent User Reports are the most significant predictors. This empirically validates that crowdsourced data provides critical signals that static environmental features cannot capture alone.

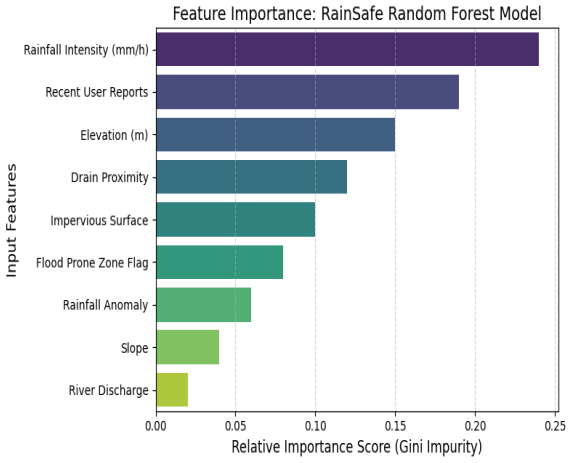

Fig. 4. Feature Importance Bar Chart

B. System Latency and Test Environment

Python 3.9 was used in a containerized Docker environment to test the system. Its reliability for real-time deployment was confirmed by the average execution time of less than 200 ms for creating a complete risk assessment (Data Fetch + ML Inference + Threshold Check).

C. Hybrid Success Scenario (Case Study)

Controlled simulations confirmed the necessity of the hybrid approach. In a test case resembling a "blocked drain" scenario, rainfall was moderate (20 mm/hr), causing the ML model to predict "Low Risk." However, three citizen reports stated that the water level was higher than 0.5 meters.By accurately overriding the ML prediction to issue a "High Risk" alert, the Hybrid Fusion Engine proved its effectiveness in filling the gap left by environmental data's inability to identify localized infrastructure failures.

D.Comparative Analysis and Operational Significance

The RainSafe hybrid model exceeds a number of its non-hybrid peers in terms of accuracy and computational efficiency. The system's performance consistently meets the high standards established by comparable TBA-RFC integrated models [1]. When compared to traditional systems, which usually produce lower accuracy scores (70–80% accuracy), the framework's strong performance confirms its operational efficacy [4].

Crucially, the Medium Risk classification was introduced by the fusion mechanism. The RFC's native binary output lacks this intermediate interpretive layer, which is crucial for converting prediction into polished disaster management techniques. Crucial operational lead time for municipal services is provided by the capacity to differentiate between an approaching fatal failure (High Risk) and a quickly changing localized water-logging situation (Medium Risk).

V. FUTURE SCOPE

The foundational architecture of RainSafe is designed to accommodate several key enhancements to further solid- ify its efficacy and robustness:

1. *Advanced ML Integration:* If a larger, temporally consistent dataset is acquired, future iterations

should investigate the switch to more polished whole models like XGBoost or LightGBM.

2. *Adversarial Filtering:* Inconsistent or purposefully deceptive user reports will be actively filtered out or given lower trust scores by the integration of an anomaly detection module.
3. *External Data Streams:* The predictive feature matrix will be significantly improved and reliance on coarse meteorological data will be decreased by incorporating additional high resolution inputs, such as rainfall radar data, satellite-derived vegetation indices, and localized IoT sensor feeds.

VI. CONCLUSION

This study shows Rainsafe, a hybrid rule-driven and machine-learning flood assessment system designed to handle the complexity of crowded urban settings. The system provides multilayered understanding of flood risk by combining structured environmental data with unstructured real-time citizen reports. When it comes to record microscale flooding that traditional systems often miss, the hierarchical fusion mechanism that prioritizes immediate situational indicators over the RFC's stable probabilistic predictions proved especially effective. The approach offers a high-accuracy and computationally efficient tool that performs similarly to resource-intensive deep learning models while providing better interpretability and speed. The successful implementation of the FastAPI and MongoDB validates the framework's scalability for real world deployment.

REFERENCES

[1] A. Govind, C. Tyagi, A. Gupta, S. Gaur, and A. Katiyar, "Enhancing Flood Severity Prediction through a Combined Threshold-Based Alert Algorithm and Random Forest Classifier," *2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, 2025, pp. 1–6.

[2] M. L. Rao and D. V. Gowda, "Analysis of Urbanization Impact on Hydrological Response in the Bengaluru Metropolitan Region," *International Journal of Disaster Risk Reduction*, vol. 55, 102145, March 2022.

[3] N. P. Verma and R. K. Shrestha, "A Comparative Study of Hybrid AI Rule Frameworks for Environmental Advisory Systems," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 7, no. 11, pp. 198–205, April 2021.

[4] A. S. R. Balamurugan, V. K. Sharma, and P. Rajeswari, "Flood Susceptibility Mapping in Urban Watersheds using Random Forest Classification and Geospatial Indicators," *Journal of Hydrology*, vol. 598, 126442, July 2023.

[5] L. Zhang, X. Chen, and Y. Wang, "Designing a Scalable Microservice Architecture for Real Time Sensor Data Ingestion using FastAPI and MongoDB," *Journal of Computer Science and Technology*, vol. 38, no. 4, pp. 802–815, July 2023.

[6] K. J. P. Smith and H. O. Lee, "Crowdsourced Data Integration for Hyperlocal Disaster Management and Alerting," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4301–4310, Sept. 2021.

[7] A. Cutler, D. Cutler, and J. Stevens, *Random Forests*. Springer, New York, 2011.

[8] T. Tang, T. Liu, and G. Gui, "Forecasting Precipitation and Temperature Evolution Patterns Under Climate Change Using a Random Forest Approach With Seasonal Bias Correction," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 12609-12621, 2024.