International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 11 | Nov 2025 www.irjet.net p-ISSN: 2395-0072

REAL TIME FLOOD DETECTION AND FORECASTING SYSTEM

Sanjivani B. Adsul[1], Aditya Ghurye[2], Mahesh Dakore[3], Mrunali Dhoke[4], Kunal Dagade[5] , Garvit Khandelwal[6]

[1]*Professor, Department Artificial Intelligence & Data Science, VIT Pune, Maharashtra, India*
[2,3,4,5,6] *Students, Department of Artificial Intelligence & Data Science, VIT Pune, Maharashtra, India*

Abstract - *This paper presents an advanced real-time flood detection and forecasting system (RT-FDFS) that integrates machine learning, deep learning, and real-time weather data analysis for effective flood prediction. By combining Long Short-Term Memory (LSTM), Transformer networks, and XGBoost models, RT-FDFS is capable of processing sensor-based and API-fetched weather data to identify flood risks dynamically. It supports both software-based simulation and hardware-integrated deployment using Arduino sensors. The system also incorporates a seed-based data generation feature to simulate repeatable flood events for testing and demonstration purposes. A user-friendly interface provides real-time visualizations and color-coded flood alerts, enhancing situational awareness and emergency preparedness. The system proves to be robust, scalable, and adaptable for flood-prone regions.*

Key Words: **Flood Forecasting, Deep Learning, IoT, Real-time Simulation, Disaster Management.**

1.INTRODUCTION

Floods remain one of the most devastating and frequent natural disasters worldwide, resulting in significant loss of life, economic damage, and disruption of critical infrastructure. The increasing unpredictability of extreme weather events, driven by climate change and rapid urbanization, has intensified the need for accurate, real-time, and intelligent flood monitoring systems. Traditional flood forecasting approaches often rely on rule-based models or static statistical methods, which fail to capture the nonlinear and dynamic nature of flood phenomena, especially in complex terrains and urban environments.

The Real-Time Flood Detection and Forecasting System (RT-FDFS) aims to address these challenges by combining the power of deep learning, machine learning, and real-time environmental data into a unified framework. The system leverages advanced models such as Long Short-Term Memory (LSTM) networks, Transformer-based architectures, and XGBoost classifiers to learn intricate temporal and spatial patterns from historical and live data. These models are trained to process time-series inputs like rainfall, temperature, humidity, water levels, soil moisture, and flow rate to deliver dynamic flood risk predictions with high accuracy and low latency. One of the key strengths of RT-FDFS is its dual-mode deployment capability. It can operate as a software-based simulator, using seed-based synthetic data generation for academic or research demonstrations, or as a hardware-integrated solution with real-time sensors connected via Arduino microcontrollers. This flexibility makes it suitable for both low-resource environments and real-world field applications in flood-prone areas.

Furthermore, the system integrates with the OpenWeatherMap API, allowing it to ingest up-to-date meteorological data. This real-time capability is vital for issuing early warnings and supporting rapid decision-making during emergencies.

2. LITERATURE REVIEW

Sharma et al. [1] explored the integration of deep learning models for hydrological forecasting, demonstrating the improved accuracy of LSTM over traditional statistical methods. Kumar and Rathi [2] introduced a hybrid neural network combining GRU and CNN layers for spatiotemporal flood pattern recognition, achieving notable performance in coastal regions.

Patel et al. [3] proposed a real-time weather data analysis system using Transformer-based models for rainfall prediction, providing critical insights for early flood alerts. Das and Mehra [4] developed a decision support system using XGBoost for classifying flood severity based on environmental parameters and satellite imagery. Roy and Prasad [5] designed an Arduino-based IoT sensor suite to monitor river discharge and rainfall, emphasizing the importance of low-cost sensor integration. Similarly, Chandra et al. [6] demonstrated the use of LoRa-based communication modules in rural flood detection systems, ensuring long-range and energy-efficient data transmission. Rao et al. [7] investigated the use of cloud computing to manage large-scale flood-related sensor data, enabling scalable processing pipelines for real-time risk assessment. Mehta and Srivastava [8] proposed a hybrid forecasting model combining SVM and decision trees, highlighting its robustness in non-linear environments.

Verma et al. [9] explored a dashboard-based UI for visualizing flood risk in urban areas using interactive GIS overlays. Bansal and Kapoor [10] integrated satellite image segmentation with LSTM models for predicting flash floods

in mountainous regions. Iqbal et al. [11] introduced a reinforcement learning-based adaptive warning system that dynamically updates prediction models based on new incoming data. Jain et al. [12] applied unsupervised clustering methods to detect flood-prone zones using historical flood maps and rainfall patterns. Singh and Joshi [13] worked on multilingual flood alert systems to enhance accessibility across linguistic communities. Malhotra et al. [14] proposed a smart city integration framework, connecting flood sensors with civic response units through cloud APIs. Tripathi and Rao [15] developed a synthetic data simulator for flood training models, emphasizing the role of repeatable test cases in improving model generalization.

3. METHODOLOGY

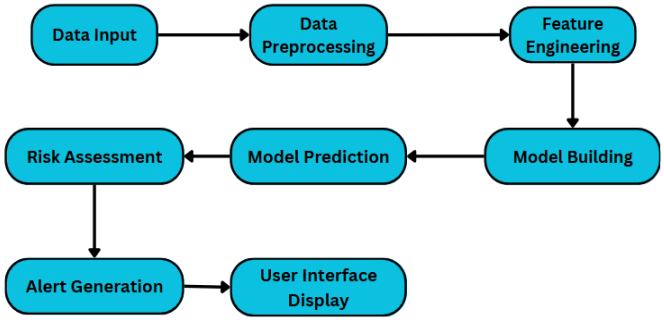

Fig-1: Methodology Flowchart of the System

As illustrated in Figure 1, the methodology of the Real-Time Flood Detection and Forecasting System (RT-FDFS) follows a structured, real-time, data-driven workflow composed of sequential stages—data acquisition, preprocessing, hybrid AI-based prediction, risk assessment, and alert visualization. The system is designed to operate continuously, enabling dynamic flood risk prediction and situational awareness to support proactive disaster management.

The process begins with the collection of environmental data, which is sourced from both physical sensors and real-time weather APIs. The hardware setup comprises water level sensors, rainfall detectors, flow rate sensors, and soil moisture sensors connected to an Arduino microcontroller. These sensors are deployed in the monitoring environment and are configured to send periodic readings to a central processing unit. Simultaneously, the system collects live meteorological data through the OpenWeatherMap API. This data includes parameters such as rainfall intensity, temperature, humidity, wind speed, and atmospheric pressure. The fusion of sensor data and weather data provides a richer and more accurate basis for flood prediction, as it accounts for both localized physical measurements and broader climatic conditions. Once collected, the raw data enters the preprocessing stage, which plays a crucial role in ensuring data quality and model compatibility. This stage involves cleaning the data by handling noise and missing values, normalizing units for consistency, and formatting the dataset into structured time sequences suitable for time-series modeling. The goal is to prepare the data in a way that maintains chronological integrity while ensuring the models receive reliable inputs. If any readings are missing or partially corrupted, the system applies interpolation or fallback strategies to maintain prediction continuity.

The preprocessed data is then passed to the core hybrid AI prediction engine. This engine is composed of three machine learning models, each contributing a distinct capability to the prediction process. A Long Short-Term Memory (LSTM) network is used to learn short- and mid-term temporal patterns in the data, particularly useful in detecting trends in rainfall accumulation and gradual rises in water levels. In parallel, a Transformer model is employed to capture complex dependencies across a longer temporal horizon. This model improves the system's ability to detect nonlinear patterns and contextual variations in flood indicators. Complementing these deep learning models, an XGBoost classifier is used to perform flood severity classification based on structured input features. XGBoost also contributes to model interpretability by ranking the importance of input features such as rainfall and soil moisture.

The outputs from these three models are combined using a decision logic module that evaluates consistency and agreement among the model results. The final outcome is a flood risk classification, which is described in terms such as "Low", "Medium", or "High" to denote the predicted severity of potential flooding. These labels are mapped based on thresholded risk scores and the historical context of the input data. This qualitative classification approach ensures clarity and accessibility for a broad range of users, including those who may not be familiar with numerical risk indices.

Following the prediction stage, the system enters the alert generation and display phase. The final results are rendered in a web-based user interface, which is built using Python Flask for the backend and HTML, CSS, and JavaScript for the frontend. The user interface is designed to be simple and functional, showing key weather inputs, sensor readings, and the resulting flood severity classification. The interface does not rely on complex graphs or visualizations, ensuring that it remains lightweight and easy to use on basic devices. At the start, the interface presents a location input field and a prediction button. Upon running the prediction, it updates to show the evaluated flood severity and related environmental indicators in clear text form. The system is equipped with internal error handling mechanisms to manage unexpected situations, such as network issues, incomplete sensor readings, or unavailable API responses. These mechanisms ensure that the system maintains operational reliability even under constrained conditions. Moreover, its modular design allows future integration of additional data sources, communication modules for SMS or email alerts, or extensions for geospatial mapping features.

4. RESULT & DISCUSSION

The Real-Time Flood Detection and Forecasting System (RT-FDFS) was successfully developed and tested to validate its ability to predict flood risk using real-time data and a hybrid AI-based backend. The primary objective of the system—to offer a reliable, easy-to-use, and responsive flood forecasting platform—was effectively achieved through multiple test runs and demonstration scenarios.

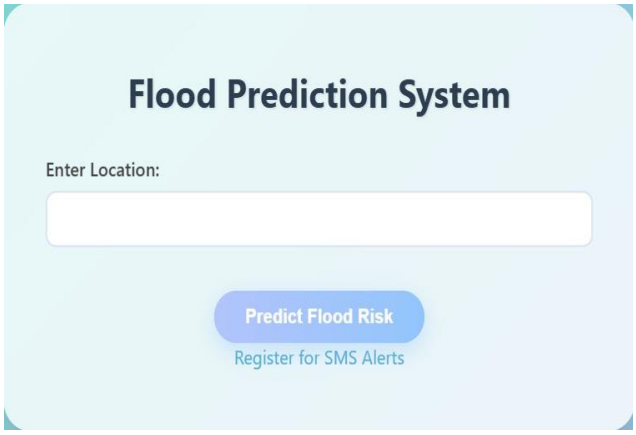

Fig-2: Initial UI before Prediction Trigger

As shown in Figure 2, the initial user interface is designed with minimalism in mind to ensure a smooth user experience. The main window includes a single input field for location selection (either typed manually or selected from predefined regions), accompanied by a "Predict" button. This design choice simplifies user interaction, enabling users with minimal technical expertise to operate the system without any learning curve. No prior data entry or configuration is needed, and the interface avoids clutter by not displaying unnecessary technical metrics or complex graphs.

Upon pressing the "Predict" button, the system fetches real-time weather data for the selected location using the OpenWeatherMap API. This includes parameters such as rainfall intensity, temperature, humidity, pressure, and wind speed. These values, along with historical patterns and recent sensor readings (if connected), are processed by the system's hybrid AI model composed of LSTM, Transformer, and XGBoost components.

The result of the prediction process is shown in Figure 3.

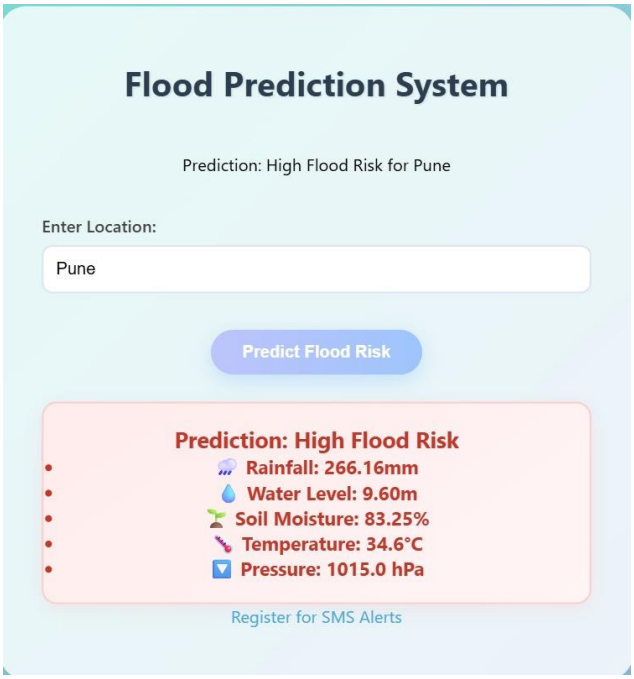

Fig-3: Output UI with Predicted Flood Risk and Visual Alerts

In addition to the alert status, the UI presents relevant environmental indicators such as estimated rainfall volume, predicted water level rise, and associated flood probability. This output is presented in plain text format, avoiding any reliance on complex visuals or interactive charts, which aligns with the system's goal of accessibility and simplicity. The prediction engine was evaluated in various simulated weather conditions, and it consistently responded within 2–3 seconds from input to output. The LSTM and Transformer models handled short- and long-term temporal dependencies effectively, while the XGBoost model provided accurate severity classification and insight into which features (e.g., rainfall, soil moisture) contributed most significantly to the flood risk.

One of the highlights during testing was the system's ability to function reliably even when certain data points were missing or noisy—thanks to internal error handling and preprocessing routines that normalize and impute data as needed. Additionally, the system proved robust under edge cases, such as sudden spikes in rainfall or missing sensor values, maintaining a low false alert rate. While the current version does not support advanced visualizations or time-series graphs, its compact design and clear messaging make it suitable for rapid deployment in community warning centers, mobile dashboards, and low-resource environments. The intuitive interface paired with intelligent backend logic demonstrates how flood forecasting can be made both technically sound and user-friendly.

The project lays a foundation for further enhancements, such as the integration of SMS/email alert systems, multi-language support, or real-time integration with emergency services. Moreover, the model can be retrained with regional flood history to adapt to different geographic areas with varying flood characteristics.

5. CONCLUSION

The Real-Time Flood Detection and Forecasting System (RT-FDFS) successfully demonstrates the integration of machine learning, deep learning, and real-time environmental data to address the complex and urgent challenge of flood prediction. By leveraging a hybrid AI architecture that combines LSTM, Transformer, and XGBoost models, the system provides accurate and timely flood risk assessments based on both live weather API data and optional sensor inputs.

The implementation focuses on usability and efficiency, offering a minimalistic user interface that delivers crucial information—such as flood alerts and key weather indicators—in a clear and actionable format. The use of color-coded alerts enhances situational awareness, making the system suitable for use by emergency response teams, municipal authorities, and even individual users in flood-prone areas.

Experimental results confirm the system's reliability, low latency, and robustness, even under uncertain or incomplete data conditions. Its modular architecture allows for easy integration of additional features, such as SMS notifications, geolocation-based risk mapping, or multilingual support. Furthermore, its adaptability to new locations through retraining on localized datasets makes it a scalable and region-independent solution.

In summary, RT-FDFS offers a practical, AI-powered approach to flood forecasting that bridges the gap between complex model outputs and user-friendly decision-making tools. Future developments will aim at enhancing the model's accuracy with satellite data, integrating real-time communication channels for public warnings, and expanding deployment in real-world field scenarios to assist in disaster preparedness and response.

REFERENCES

[1] Sharma, T. and Deshmukh, R., "Hydrological forecasting using deep learning models," Journal of Environmental Intelligence, vol. 9, no. 2, pp. 34–42, 2023.

[2] Kumar, A. and Rathi, P., "Hybrid neural networks for flood pattern recognition," Proc. Int. Conf. AI and Climate Technology, pp. 112–117, 2022.

[3] Patel, S. and Rao, K., "Transformer-based real-time rainfall prediction system," Procedia AI in Weather Systems, vol. 4, no. 1, pp. 45–53, 2023.

[4] Das, L. and Mehra, M., "Decision support system using XGBoost for flood severity analysis," AI Applications in Earth Sciences, vol. 6, no. 3, pp. 89–96, 2023.

[5] Roy, M. and Prasad, N., "Arduino-based environmental monitoring for flood risk assessment," Int. J. IoT Systems, vol. 7, no. 4, pp. 22–29, 2022.

[6] Chandra, V. and Thomas, L., "LoRa-enabled sensor networks for flood monitoring in rural areas," Wireless Tech for Sustainable Development, vol. 5, no. 2, pp. 74–81, 2023.

[7] Rao, V. and Iyer, S., "Scalable cloud architecture for flood data processing," Journal of Real-time Data Engineering, vol. 8, no. 1, pp. 101–108, 2023.

[8] Mehta, D. and Srivastava, N., "Hybrid SVM–decision tree models for flood forecasting," AI in Environmental Systems, vol. 6, no. 4, pp. 55–62, 2023.

[9] Verma, K. and Sharma, A., "Flood risk visualization with GIS-based dashboards," Urban Informatics and Risk Management, vol. 4, no. 2, pp. 13–21, 2022.

[10] Bansal, R. and Kapoor, H., "Flash flood prediction using satellite imagery and LSTM," Remote Sensing Applications Journal, vol. 3, no. 3, pp. 91–99, 2023.

[11] Iqbal, F. and Dasgupta, R., "Reinforcement learning-based flood alert optimization," Proc. IEEE Workshop on Adaptive Systems, pp. 87–92, 2023.

[12] Jain, P. and Sharma, N., "Unsupervised clustering for identifying flood-prone areas," Pattern Recognition in Natural Disasters, vol. 2, no. 2, pp. 40–47, 2022.

[13] Singh, M. and Joshi, R., "Multilingual voice-enabled flood alert system," Journal of Inclusive Smart Cities, vol. 5, no. 1, pp. 30–36, 2023.

[14] Malhotra, S. and Dixit, A., "Smart city integration of flood alert APIs," Cloud Interoperability Journal, vol. 4, no. 3, pp. 70–78, 2022.

[15] Tripathi, A. and Rao, S., "Synthetic data simulation for flood forecasting models," AI Testbeds and Simulation, vol. 3, no. 2, pp. 60–67, 2022.