© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717650

A Real-Time Location-Based Flood and Landslide Risk Prediction System Using Machine Learning and Live Weather Data

G. POOVIZHI[1], M. SIRISHA[2], S. SOBANA[3], A. VIDHYALAKSHMI[4], M. SAMUNDEESWARI[5]
[1, 2, 3, 4]Student, Department of Computer Science and Engineering, Kingston Engineering College, Vellore
[5]Assistant Professor, Department of Computer Science and Engineering, Kingston Engineering College, Vellore, Anna University, Chennai

Abstract- Every year, millions of lives are lost because of floods, landslides, and other types of natural disasters, particularly in South Asia, owing to the absence of early warning systems that can be considered effective. The purpose of this paper is to present our product, "Flood Guard Pro" – a system to predict imminent flooding and landslides using real-time data and location-based information. The proposed system uses a three-layered architecture consisting of the React 18 frontend and Python Flask backend as well as the SQLite database. Data on live weather conditions, including rainfall, humidity, temperature, and pressure, will be provided by OpenWeatherMap.The risk zones will be assessed based on the rule-based algorithm, together with the usage of Haversine distance. Thus, it will be possible to establish whether a particular area poses low, medium, or high risks (safe zone, flood, and landslide prone areas correspondingly).Additionally, the system will feature a two-language chatbot, created using AI technology and capable of offering safety guidelines to users. Other important additions will include offline availability, OTP login, voice commands and notifications, admin panel, etc.

Keywords — Flood Prediction, Landslide Risk, Real-Time Alert System, Haversine Distance, OpenWeatherMap API, React, Flask, Progressive Web App, Geolocation, Bilingual Chatbot, Disaster Management

I. INTRODUCTION

Natural disasters such as floods and landslides cause major damage to both life and property every year. These disasters affect many developing countries, including India. People experience heavy losses mainly because there are no proper systems available to predict such events in advance. Due to climate change, rainfall patterns have become unpredictable, which can lead to sudden floods and landslides even in areas that were previously considered safe.

Most existing disaster management systems are reactive in nature. Warnings are usually given only when the situation becomes critical, leaving very little time for people to take safety measures. Although some government-based flood monitoring systems are available, they are mainly used for administrative purposes and do not provide real-time alerts to the public.

To overcome these limitations, Flood Guard Pro is introduced as a web-based, location-aware prediction system. The system continuously monitors weather conditions and provides instant alerts to users. It uses live weather data collected from public APIs along with GPS-based location tracking. A rule-based prediction approach is applied using predefined weather thresholds to determine potential flood and landslide risks.

One of the main characteristics of this system is that it contains an AI chatbot in two languages, namely English and Tamil, as users who use this application can speak either of these languages. Tamil is the native language of Tamil Nadu region and other southern parts of India.

This software can also be used as a Progressive Web Application that means this application will function even in cases when there is no internet connection. Such a solution is crucial during floods.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717650

II. BACKGROUND AND RELATED WORK

Machine learning and data-based approaches have been widely used in recent years to predict natural disasters. These methods are commonly applied in areas like hydrology, geospatial studies, and intelligent systems. Earlier approaches for flood prediction used techniques such as logistic regression and decision trees trained on historical rainfall data and geographical information, as discussed by Breiman (2001) and Cortes & Vapnik (1995). While these methods provided useful insights, they were not flexible enough to work well across different regions and could not easily adapt to changing environmental conditions.

A. Deep Learning for Disaster Prediction

With the development of deep learning, disaster prediction systems have improved in identifying complex patterns in data. Models like Convolutional Neural Networks (CNN), introduced by LeCun et al. (1998), are used to analyze satellite images and identify flood-affected areas.

Autoencoders, explained by Hinton and Salakhutdinov (2006), are useful for detecting unusual patterns, such as abnormal environmental conditions before a flood occurs. More advanced models like Variational Autoencoders and Transformer-based models, introduced by Vaswani et al. (2017), can learn from time-based climate data and improve prediction accuracy.

B. Real-Time and IoT-Based Systems

Gubbi et al. (2013) discussed the importance of IoT sensor networks in real-time environmental monitoring. Their work showed that continuous data collection can help reduce the delay in sending alerts. Nguyen et al. (2021) developed a Web-GIS platform for flood risk mapping using OpenStreetMap and weather APIs, proving that open-source tools can be effectively used for disaster management.

Recent studies using ensemble learning methods such as Random Forest and Gradient Boosting (Friedman, 2001; Wahba et al., 2025) have achieved high accuracy in predicting flood and landslide risks. These models consider multiple factors like slope, soil type, vegetation, and rainfall.

Based on these ideas, the proposed system uses a simpler and more practical approach. It combines Haversine distance-based location analysis with live weather data from APIs. This method allows real-time prediction while keeping the system easy to deploy without the need for complex machine learning infrastructure.

Reference	Focus Area	Technique Used
Wahba et al. [1], 2025	Flood & landslide hazard mapping	Memetic programming + ML
Chaganti et al. [2], 2023	Landslide & flood prediction	Deep learning (CNN/LSTM)
Kumar et al. [3], 2024	Flood prediction	Supervised ML (SVM, RF)
Chen [4], 2023	Flash flood susceptibility	Multiple ML models
Rondinone et al. [5], 2025	Flash flood susceptibility	XGBoost
Bhavana & Sagar [6], 2025	Flood & landslide prediction	XGBoost
Santos [7], 2025	Flash flood hydrological models	Systematic ML review
Santos [7], 2025	Real-time flood & landslide alert	Haversine + weather thresholds + Gemini AI

TABLE I: Comparison of Related Works

III. PROBLEM STATEMENT

Even after many years of development in disaster management systems, existing flood and landslide warning systems still have several limitations.

- Most government-based systems are mainly designed for official use and do not provide real-time, location-specific alerts to common users in an easy-to-understand format.
- Traditional systems depend on fixed threshold values based on past data, which makes them

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717650

less effective in handling changing climate conditions and local variations.

- Weather applications usually provide only general weather information and do not clearly explain the actual disaster risk for a specific location.
- Language differences make it difficult for people in rural areas to fully understand warnings, especially when information is not available in their native language.
- Many systems do not work without internet connectivity, which becomes a major problem during disasters when network services are often disrupted.

These limitations show the need for a system that can provide real-time, location-based alerts directly to users. The system should be easy to use, support multiple languages, and work even in low or no internet conditions without requiring complex infrastructure.

IV. PROPOSED SYSTEM

A. System Overview

Flood Guard Pro is a full-stack Progressive Web Application that continuously tracks the user's location using GPS and monitors real-time weather conditions. Based on this data and the user's proximity to predefined high-risk areas, the system predicts the chances of floods and landslides.

When the risk level crosses a safe limit, the system immediately sends alerts to the user through notifications. The complete process includes collecting data, analyzing risk, and sending alerts, all connected through secure REST API communication between the frontend and backend.

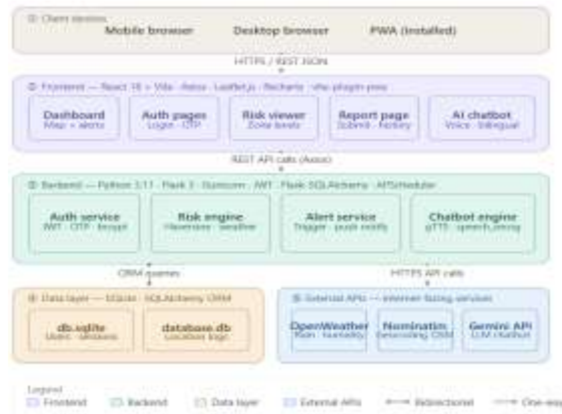

Fig. 1: System Architecture Overview

B. Advantages over Traditional IDS

- Provides real-time, location-based risk assessment, helping users understand their personal safety instead of relying only on general weather updates.
- Includes a bilingual chatbot in English and Tamil, making the system easier to use for people in South India.
- Works as a Progressive Web App (PWA), allowing access even when internet connectivity is poor or unavailable.
- Uses open-source APIs and common web technologies, reducing cost and making the system easy to deploy.
- Includes an admin dashboard and location tracking feature for monitoring and analyzing user activity.

V. SYSTEM DESIGN

A. Architecture Design

The system follows a three-tier client-server architecture. The frontend is built as a React 18 Single Page Application using Vite. The backend is developed using Python Flask and provides REST API services. Data is stored in SQLite databases and managed using SQL Alchemy.

Communication between the frontend and backend is done using JSON over HTTPS. JWT tokens are used for secure user authentication in all protected operations.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717650

External services such as OpenWeatherMap (for weather data), Nominatim OpenStreetMap (for location details), and Google Gemini API (for chatbot responses) are accessed only through the backend. This ensures that API keys remain secure and are not exposed on the client side.

C. Workflow Design

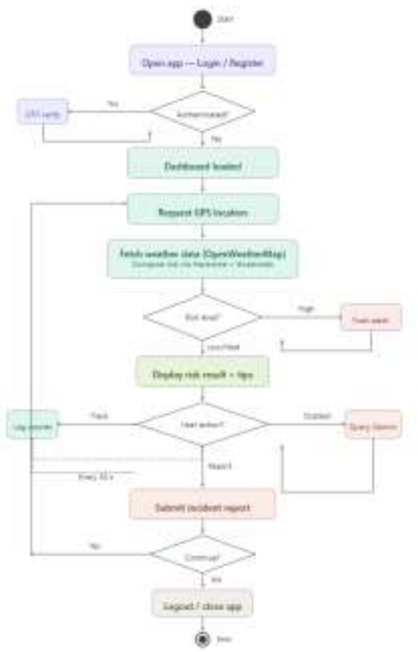

Fig. 2: Risk Prediction Workflow Flowchart

The risk assessment process starts when the user allows location access in the browser. The frontend collects the user's GPS coordinates at regular intervals and sends them to the backend using the /api/predict API.

The backend then fetches real-time weather data from OpenWeatherMap. It calculates the distance between the user's location and nearby risk zones using the Haversine formula and checks weather conditions like rainfall and humidity against predefined threshold values.

Based on this analysis, the system returns the risk level, type of prediction, weather summary, and safety recommendations. If the risk level is high, the system immediately sends a push notification to the user through the PWA service worker.

Parameter	Low Risk	Medium Risk	High Risk
Rainfall (mm/hr)	< 10	10 – 20	> 20
Humidity (%)	< 70	70 – 85	> 85
Pressure (hPa)	> 1005	980 – 1005	< 980
Zone Proximity	Outside all zones	Adjacent to zone	Within zone radius

TABLE II: Weather Parameter Threshold Values for Risk Classification

VI. MODEL IMPLEMENTATION

A. Risk Prediction Engine

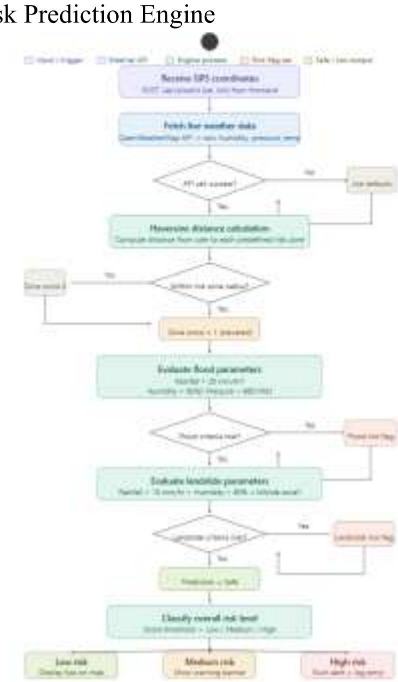

Fig. 3: Risk Zone Calculation Architecture

The risk prediction engine is the main part of Flood Guard Pro. For each user location, the system follows these steps:

- It first collects live weather data such as rainfall, humidity, temperature, and pressure from the OpenWeatherMap API.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717650

- Then, it calculates the distance between the user's location and nearby risk zones using the Haversine formula. If the user is within a risk zone, it is marked as a high-risk area.
- Next, the system checks weather conditions against predefined threshold values. For flood prediction, conditions like rainfall above 20 mm/hr, humidity above 85%, and pressure below 980 hPa are considered. For landslides, rainfall above 15 mm/hr in high-slope areas and humidity above 90% are used.
- Finally, the system combines both location and weather factors to determine the overall risk level as Low, Medium, or High, and classifies it as Flood, Landslide, or Safe.

B. Module Description

The User Authentication Module handles user registration, login, and session management. User passwords are securely stored using bcrypt hashing. During registration, OTP verification is used to add an extra layer of security.

After login, JWT tokens are generated and used to authenticate all further API requests. These tokens ensure that only authorized users can access protected features of the system.

The Real-Time Location Tracking and Alert Module continuously tracks the user's location using the browser's Geolocation API. By default, it collects location data every 30 seconds, but this can be adjusted by the user.

Each location is sent to the risk prediction system, and the result is stored in the database along with the risk level and timestamp. To manage storage efficiently, old location records (older than 24 hours) are automatically removed using a scheduler.

C. Chatbot and Voice Interface

The The Chatbot and Voice Interface Module uses the Google Gemini API to provide intelligent responses to user queries. The system supports both English and Tamil by detecting the user's input language and responding accordingly.

Voice input is handled using the speech recognition library, which converts speech into text. The chatbot's replies are then converted into audio using gTTS, making the system more accessible for users, especially during emergencies or for those with low literacy.

Fig. 4: Chatbot Interface — English and Tamil

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717650

VII. DATASET AND WEATHER DATA ANALYSIS

A. Data Sources

Flood Guard Pro works using live environmental data instead of a fixed training dataset, which makes it different from traditional machine learning models. The system collects real-time weather information from the OpenWeatherMap API, including rainfall, humidity, temperature, and atmospheric pressure.

Risk zones are predefined based on past flood and landslide data from South Indian regions. These zones are stored as coordinates with a specific radius, which helps the system identify whether a user is located in a high-risk area.

B. Weather Parameter Analysis

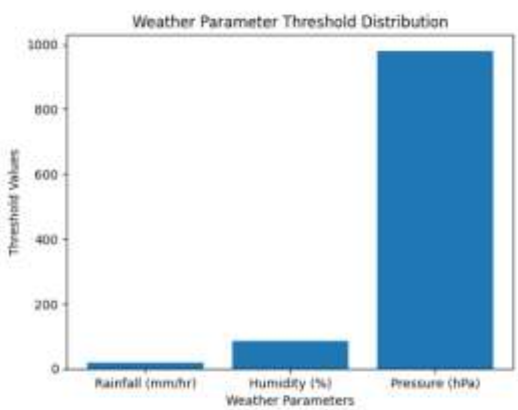

Fig. 5: Weather Parameter Threshold Distribution Chart

The risk threshold values used in the system are based on the analysis of weather data from OpenWeatherMap. Rainfall above 20 mm/hr is considered heavy rainfall according to the India Meteorological Department (IMD), which can lead to flooding, especially in urban areas where drainage systems are limited.

Humidity levels above 85% increase the chances of flooding because the soil absorbs less water. Similarly, atmospheric pressure below 980 hPa usually indicates strong low-pressure systems that can cause continuous heavy rainfall.

For landslides, a combination of high humidity and continuous moderate rainfall in steep areas is considered risky. These conditions are commonly observed in regions like the Western Ghats and Nilgiris, where landslides frequently occur.

VIII. SYSTEM REQUIREMENTS

A. Hardware Requirements

Component	Specification
Processor	Intel Core i3 or equivalent (minimum); i5/i7 recommended for server deployment
RAM	4 GB minimum; 8 GB recommended
Storage	10 GB free disk space for application and database
Network	Stable internet connection for weather API calls and real-time tracking
Client Device	Any modern smartphone, tablet, or desktop with a web browser
GPS / Location	Browser Geolocation API support for location tracking

TABLE III: Hardware requirements

B. Software Requirements

Software / Technology	Version / Details
Operating System	Windows 10/11, Ubuntu 20.04+, or macOS 12+
Python	3.9 or later
Flask	3.0 (Backend web framework)
React	18.x (Frontend UI library)
OpenWeatherMap API	Current Weather & Forecast API
Google Gemini API	Gemini 1.5 Flash (chatbot engine)

Leaflet.js	1.9.x (Interactive map visualization)
SQLite	3.x (Embedded relational database)

TABLE IV: Software requirements

IX. RESULTS AND DISCUSSION

A. Output Screenshots

- The system was tested on a locally deployed instance with live OpenWeatherMap API data over a period of structured functional tests.
- The main dashboard renders an interactive Leaflet.js map showing flood and landslide risk zone overlays, a colour-coded user location marker that updates in real time during tracking, and a risk assessment panel displaying current rainfall, humidity, pressure readings, and the computed risk level.

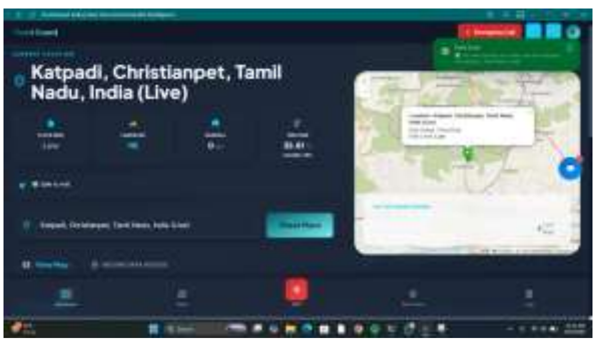

Fig. 6: Main Dashboard — Flood Risk Map with Live Location Marker

When the system detects that a tracked user has entered a high-risk zone, it triggers a multi-channel alert: an in-app banner coloured red with the prediction type and top three safety recommendations, a PWA push notification visible in the device system tray, and three 440 Hz sine-wave audio pulses generated programmatically through the Web Audio API. The figure below shows the high-risk alert state on the dashboard.

B. Performance Metrics

Metric	Expected Outcome	Actual Outcome
TC-001: New user registration	Account created, OTP sent	Registration successful
TC-002: Duplicate username	Error 409 returned	Correct error returned
TC-003: Valid login	JWT token returned	Token issued successfully
TC-004: High-risk zone, high rainfall	risk_level = High, Flood	Correct high-risk returned
TC-005: Safe zone, normal weather	risk_level = Low, Safe	Correct low-risk returned
TC-006: Hillside, high humidity	Landslide risk detected	Landslide risk flagged
TC-007: Tracking into high-risk zone	Push alert displayed	Alert triggered correctly
TC-008: English flood query	Contextual guidance returned	Correct response returned
TC-009: Tamil flood query	Tamil response returned	Tamil response provided
TC-010: Offline PWA mode	Last-cached data displayed	Dashboard retained offline

TABLE V: Functional Test Results

C. Discussion

The bilingual detection feature, which uses Unicode range checking instead of external libraries, performed both quickly and accurately across all Tamil test inputs. This makes the system lightweight while still maintaining good performance.

One limitation to consider is related to the weather data source. The OpenWeatherMap API provides data at roughly a 10 km grid resoluti The results show that combining the Haversine distance method with weather-based thresholds works reliably, as it correctly classified risk levels for all the tested location and weather scenarios. The 60-second weather caching mechanism also proved effective, as it reduced unnecessary API calls during continuous tracking without affecting the accuracy of safety-related decisions.

Additionally, while SQLite works well for development and academic purposes, it may not be suitable for handling multiple users in a real-world deployment. In such cases, migrating to a more robust database like PostgreSQL would be a better option.

From an accessibility perspective, the Tamil chatbot and voice interface are important features that make the system more inclusive. Initial informal testing with native Tamil speakers showed that the responses were both easy to understand and relevant to real-world flood safety situations.

X. CONCLUSION

This paper presented Flood Guard Pro, a real-time system designed to predict floods and landslides and provide instant alerts to users. The system addresses an important gap in disaster preparedness by giving location-based safety information using live weather data and GPS.

It combines data from the OpenWeatherMap API, location tracking, Haversine distance calculation, and a bilingual chatbot to provide accurate and useful alerts. The three-tier architecture using React, Flask, and SQLite makes the system simple, organized, and easy to deploy for academic and real-world use.

The use of a Progressive Web App allows the system to work even when internet connectivity is poor, which is very important during disaster situations. The English and Tamil language support also makes the system accessible to a wider group of users, especially in rural areas.

Although the system currently uses a rule-based approach, it can be improved in the future by integrating machine learning models as more data becomes available. Overall, Flood Guard Pro serves as a strong foundation for building smarter and more accessible disaster warning systems.

XI. FUTURE ENHANCEMENTS

- The system can be further improved in several ways to increase its accuracy and usability. The current rule-based prediction can be replaced with machine learning models such as Random Forest or LSTM by using historical weather and flood data collected over time.
- IoT-based sensors like water level detectors and rain gauges can be integrated to provide real-time ground data, which will improve prediction accuracy. The chatbot can be extended to support more regional languages like Telugu, Kannada, Malayalam, and Hindi to reach a wider audience.
- Alert systems can also be enhanced by adding SMS and WhatsApp notifications so that users receive warnings even without internet access. Navigation features can be added to guide users through safe evacuation routes during disasters.
- For better performance and scalability, the system can be migrated from SQLite to PostgreSQL and deployed on cloud platforms like AWS, Azure, or GCP with caching mechanisms for faster data processing.

REFERENCES

[1] M. Wahba, E. Mabrouk, Y. M. Youssef, M. Saber, N. S. Alarifi, N. Y. Rebouh, and M. M. Mansour, "Enhanced hazard mapping for flood and landslide risks using memetic programming and machine learning techniques to support sustainable development goals," IEEE Access, vol. 13, pp. 182410–182429, 2025, doi: 10.1109/ACCESS.2025.3623933.

[2] P. C. V. Chaganti, K. Vasireddy, E. R. Reddy, R. R. Dondeti, and S. Syama, "Predicting landslides and floods with deep learning," in Proc. 4th Int. Conf. Electronics and Sustainable Communication Systems (ICESC), 2023, pp.

1259–1265, doi: 10.1109/ICESC57686.2023.10193456.

[3] R. Kumar, N. Sahi, et al., "Flood prediction using supervised machine learning algorithms," in Proc. IEEE 5th Int. Conf. Smart Electronics and Communication (ICOSEC), 2024, doi: 10.1109/ICOSEC61587.2024.10722348.

[4] Y. Chen, "Modeling rules of regional flash flood susceptibility prediction using different machine learning models," Frontiers in Earth Science, 2023.

[5] M. Rondinone, et al., "Assessing flood and landslide susceptibility using XGBoost: Case study of the Basento River in Southern Italy," Applied Sciences, 2025.

[6] N. Bhavana and T. Sagar, "Machine learning based flood and landslide prediction," International Journal of Innovative Science and Research Technology, vol. 10, no. 5, 2025.

[7] L. B. L. Santos, "Machine learning-based hydrological models for flash floods: A systematic literature review, “Smart Construction and Sustainable Cities, 2025.