International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

AI-Based Flood Prediction and Alert System Using Machine Learning and Web Technologies

NAGIDI UDAY SHANKAR

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

A. Durga Devi

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Floods are among the most destructive natural disasters, causing significant loss of life, property damage, and environmental degradation worldwide. Early prediction and timely dissemination of alerts are crucial to minimizing the adverse impacts of such disasters. This project presents an AI-based Flood Prediction and Alert System that leverages machine learning techniques and web-based technologies to provide real-time flood risk assessment and location-based alert notifications.The system is designed using the Django web framework and integrates multiple components including user authentication, flood prediction modeling, alert management, and historical data tracking. Users can register and specify their geographic location, enabling the system to deliver personalized flood alerts based on city-specific conditions. Administrators can generate and broadcast alerts with varying severity levels such as low, medium, high, and critical.The core of the system is a machine learning model trained on environmental parameters such as rainfall, monthly rainfall, temperature, humidity, river level, and drainage capacity. These features are processed using a scaler and fed into a trained classification model to predict the likelihood of flood occurrence. The prediction results are categorized into risk levels and displayed to users through an intuitive dashboard.Additionally, the system maintains a history of user predictions, allowing for performance tracking and analysis. The admin dashboard provides insights into model performance metrics, user statistics, and alert history, enhancing decision-making capabilities. Error handling mechanisms ensure robustness, particularly in cases of model version mismatch or missing files.This solution aims to bridge the gap between predictive analytics and real-world disaster management by offering an accessible, scalable, and efficient platform. It can be extended with real-time data integration from weather APIs, IoT sensors, and satellite data for improved accuracy. The system contributes to smart city initiatives and disaster preparedness strategies by empowering both authorities and citizens with actionable insights.

Keywords: Flood Prediction, Machine Learning, Django Web Application, Disaster Management, Alert System, Environmental Data, Risk Assessment, Smart Cities

I. INTRODUCTION

Natural disasters have increasingly become a global concern due to climate change and rapid urbanization. Among these, floods are one of the most frequent and devastating hazards, affecting millions of people each year. In countries like India, floods result in extensive damage to infrastructure, agriculture, and human life. Traditional flood management systems often rely on manual monitoring and delayed communication, which limits their effectiveness in emergency situations.Advancements in artificial intelligence and machine learning have opened new avenues for predicting natural disasters with greater accuracy. By analyzing historical and real-time environmental data, machine learning models can identify patterns and forecast potential flood events. This enables proactive decision-making and timely evacuation measures.The proposed system, an AI-Based Flood Prediction and Alert System, aims to provide a comprehensive solution that combines predictive modeling with a user-friendly web interface. The system allows users to input environmental parameters such as rainfall, temperature, humidity, river levels, and drainage capacity. Based on these inputs, the machine learning model predicts whether a flood is likely to occur.One of the key features of this system is its location-based alert mechanism. Users register with their city, and administrators can send targeted alerts to specific regions. This ensures that warnings are relevant and actionable. The system also categorizes alerts based on severity, helping users understand the level of risk and take appropriate precautions.The integration of Django as the backend framework ensures scalability, security, and efficient data handling. The system includes separate dashboards for users and administrators. Users can view predictions, alerts, and their historical data, while administrators can monitor system performance, manage users, and send alerts.Furthermore, the system addresses challenges such as model compatibility and error handling by incorporating safeguards during model loading and prediction. This enhances reliability and user trust.Overall, this project demonstrates how modern technologies can be utilized to improve disaster preparedness and response. It serves as a step toward building intelligent systems that can mitigate the impact of natural disasters and contribute to safer communities.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Flood prediction has been an active area of research for decades, with various approaches ranging from traditional hydrological models to modern machine learning techniques. Early methods relied heavily on physical and statistical models that used rainfall-runoff relationships and river flow data. While these methods provided valuable insights, they often required extensive domain expertise and were limited in handling complex nonlinear relationships.With the advancement of computational power, machine learning techniques have gained popularity in flood prediction. Algorithms such as Decision Trees, Support Vector Machines (SVM), Logistic Regression, and Artificial Neural Networks

(ANN) have been widely used. These models can learn patterns from historical data and make predictions based on multiple environmental factors.Recent studies have shown that ensemble methods and deep learning approaches, such as Random Forests and Recurrent Neural Networks (RNN), offer improved accuracy in flood forecasting. These models can capture temporal dependencies and complex interactions among variables. However, they often require large datasets and significant computational resources.Another important area of research is real-time flood monitoring using IoT and sensor networks. Sensors placed in rivers and drainage systems collect real-time data on water levels, flow rates, and weather conditions. This data can be integrated with machine learning models to provide dynamic predictions. However, the deployment and maintenance of such infrastructure can be costly.Web-based flood alert systems have also been explored to improve communication between authorities and the public. These systems use GIS mapping and mobile notifications to disseminate information quickly. While effective, many existing systems lack integration with predictive analytics, limiting their ability to provide early warnings.The proposed system builds upon these existing approaches by combining machine learning prediction with a web-based alert system. It uses a classification model trained on environmental parameters and integrates it into a Django application for real-time user interaction. Unlike traditional systems, it provides both prediction and personalized alerts in a single platform.Additionally, the system addresses common challenges such as usability, scalability, and error handling. By incorporating features like user authentication, prediction history, and admin dashboards, it offers a comprehensive solution that bridges the gap between research and practical implementation.

III. EXISTING SYSTEM

Existing flood prediction and alert systems primarily rely on traditional monitoring techniques and centralized communication methods. Government agencies typically use hydrological models and manual data collection to assess flood risks. These systems depend on parameters such as rainfall intensity, river discharge, and reservoir levels. While they are effective to some extent, they often lack real-time responsiveness and scalability.Many current systems use Geographic Information Systems (GIS) and satellite data to monitor flood-prone areas. Although these technologies provide accurate spatial analysis, they require significant infrastructure and are not always accessible to the general public. Additionally, the dissemination of alerts is often done through mass media channels such as television, radio, and SMS, which may not be timely or location-specific.Some modern solutions incorporate IoT devices and sensor networks to collect real-time environmental data. However, these systems are expensive to deploy and maintain, making them less feasible for widespread adoption, especially in developing regions.Another limitation of existing systems is the lack of user interaction and personalization. Most platforms do not allow users to input their own data or receive customized predictions. Furthermore, they do not maintain user-specific history or provide insights into past predictions.Machine learning-based systems have been proposed in research, but many of them remain theoretical or lack integration with user-friendly interfaces. As a result, there is a gap between predictive capabilities and practical usability.The proposed system addresses these limitations by providing a cost-effective,

scalable, and interactive solution. It integrates machine learning predictions with a web-based interface, enabling real-time analysis and personalized alerts. This makes it more accessible and practical for everyday use.

IV. PROPOSED METHOD

The proposed system is an **AI-based Flood Prediction and Alert System** that integrates machine learning techniques with a web-based platform to provide real-time flood risk assessment and targeted alert dissemination. The system is designed to overcome the limitations of traditional flood monitoring systems by offering predictive capabilities, automation, and user-specific notifications.The system consists of three major modules: **user module, admin module, and prediction engine**. Users register with their location (city), enabling the system to deliver personalized flood alerts. They can input environmental parameters such as rainfall, monthly rainfall, temperature, humidity, river level, and drainage capacity. These inputs are processed by a trained machine learning model to predict flood risk.The prediction engine uses a pre-trained classification model along with a scaler to normalize input data. Based on the prediction result, the system categorizes the risk into levels such as low or high. The results are displayed on a user-friendly dashboard along with historical prediction records.The admin module allows administrators to monitor system performance, manage users, and send alerts to specific cities. Alerts are categorized into severity levels (low, medium, high, critical) and are delivered only to users in the affected regions.Unlike existing systems, this solution combines **machine learning prediction with location-based alerting**, ensuring both accuracy and relevance. Machine learning models are capable of handling complex nonlinear relationships in environmental data, improving prediction accuracy .The system is scalable and can be extended with real-time data from IoT devices and weather APIs, making it suitable for smart city applications and disaster management systems.

V. IMPLEMENTATION

The implementation of the flood prediction system is carried out using **Django**, a high-level Python web framework, along with machine learning libraries such as **Scikit-learn**, **NumPy**, and **Joblib**.

1. Backend Development

The backend is developed using Django's MVC architecture. Models such as UserProfile, FloodAlert, and PredictionHistory are defined to manage user data, alerts, and prediction records. Django's built-in authentication system is used for secure login and registration.

2. User Module

Users can register by providing username, email, password, and location. Upon login, they access a dashboard where they can:

- Enter environmental parameters
- View flood prediction results
- Check recent alerts
- Access prediction history

3. Admin Module

The admin dashboard provides functionalities such as:

- Viewing total users and predictions
- Sending alerts to specific cities
- Monitoring system metrics
- Managing alert history

4. Machine Learning Integration

The core prediction functionality is implemented using a trained machine learning model stored as a .pkl file. A scaler is also used to normalize input features before prediction.

Steps:

1. Input data is collected from the user
2. Data is converted into numerical format
3. Scaling is applied using a pre-trained scaler
4. Model predicts flood risk (binary classification)
5. Output is displayed as “Flood Likely” or “No Flood Risk”

Machine learning models such as SVM, Decision Trees, and Logistic Regression are widely used in flood prediction due to their ability to model complex relationships .

5. Error Handling

The system includes robust error handling:

- Handles missing model files
- Detects version mismatch in libraries
- Validates user input

6. Database Management

SQLite (default Django DB) is used to store user data, alerts, and prediction history. Queries are optimized using Django ORM.

7. Frontend

HTML, CSS, and Bootstrap are used for UI design, ensuring responsiveness and usability.

VI. ALGORITHMS

The system utilizes machine learning classification algorithms to predict flood occurrence based on environmental parameters.

1. Logistic Regression

A statistical model used for binary classification. It predicts the probability of flood occurrence using a sigmoid function.

2. Support Vector Machine (SVM)

SVM is widely used for flood prediction due to its high accuracy in classification tasks. It separates data points using optimal hyperplanes and performs well with nonlinear data .

3. Decision Tree

Decision Trees classify data based on feature thresholds. They are easy to interpret and suitable for environmental datasets.

4. Feature Scaling Algorithm

A scaler (e.g., StandardScaler) is used to normalize input data, ensuring that all features contribute equally to prediction.

Working Process

1. Input features are collected
2. Features are scaled
3. Model processes data
4. Prediction is generated

Machine learning algorithms are preferred because they can handle nonlinear and complex environmental relationships better than traditional models .

VII. SYSTEM DESIGN

The system architecture follows a **three-tier design**:

1. Presentation Layer

This layer includes the user interface developed using HTML, CSS, and Bootstrap. It provides:

- Registration and login forms
- Prediction input forms
- Dashboards for users and admins

2. Application Layer

This layer is implemented using Django and contains:

- Business logic
- Authentication system
- Alert management
- Prediction processing

3. Data Layer

This layer consists of:

- Database (SQLite)
- Machine learning model files (.pkl)
- Metrics JSON file

System Architecture Flow

1. User inputs environmental data
2. Data is sent to the server
3. Backend processes input
4. ML model predicts outcome
5. Result stored in database
6. Output displayed to user

Data Flow Diagram (Conceptual)

- Input → Processing → Prediction → Storage → Output

Design Features

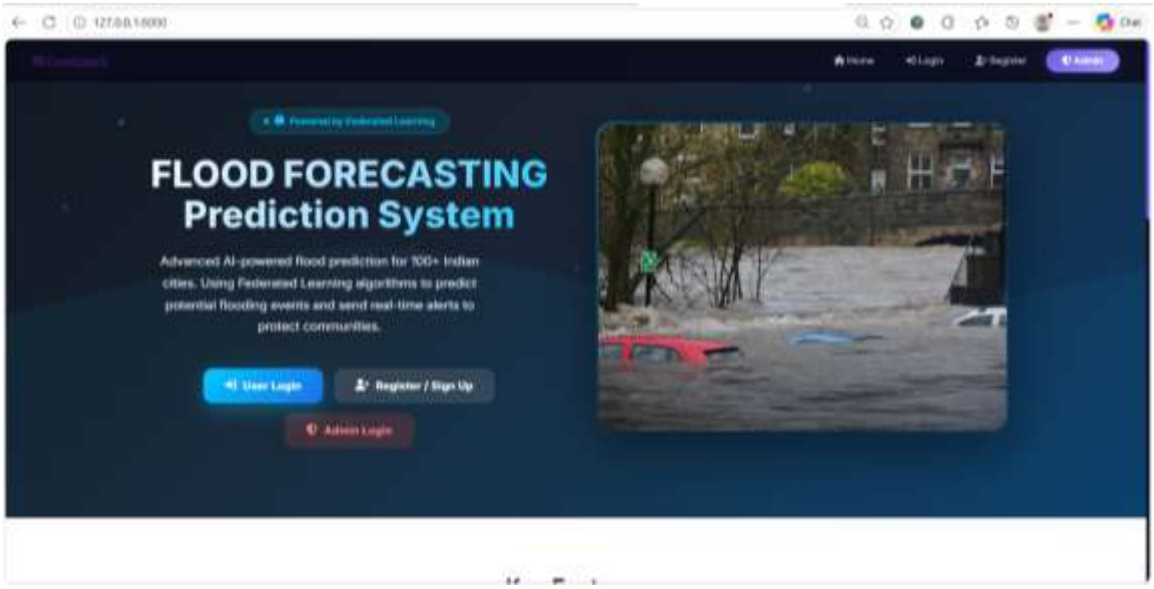
- Modular architecture
- Scalability
- Secure authentication
- Real-time prediction
- Location-based alerts

Advantages

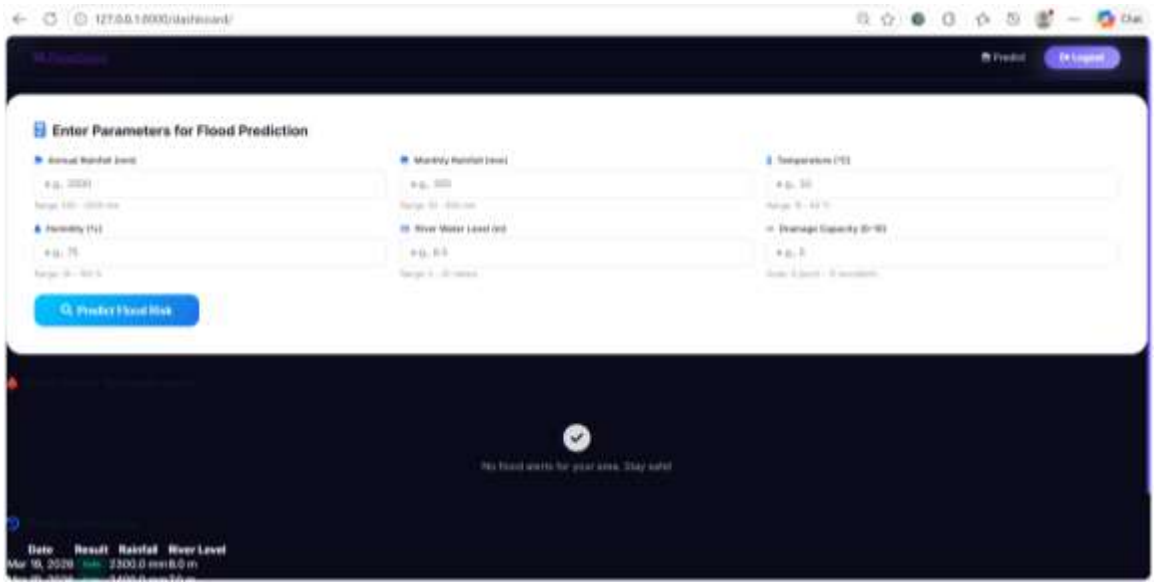
- User-friendly interface
- Accurate predictions
- Real-time alerts
- Scalable design

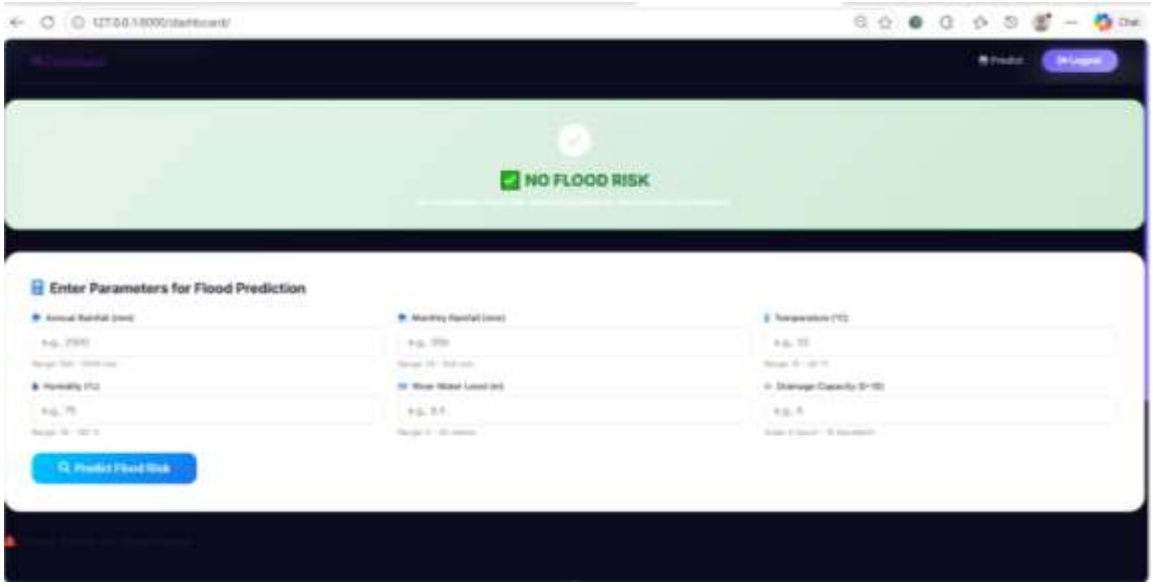
Machine learning-based systems improve prediction efficiency by handling large datasets and complex relationships effectively .

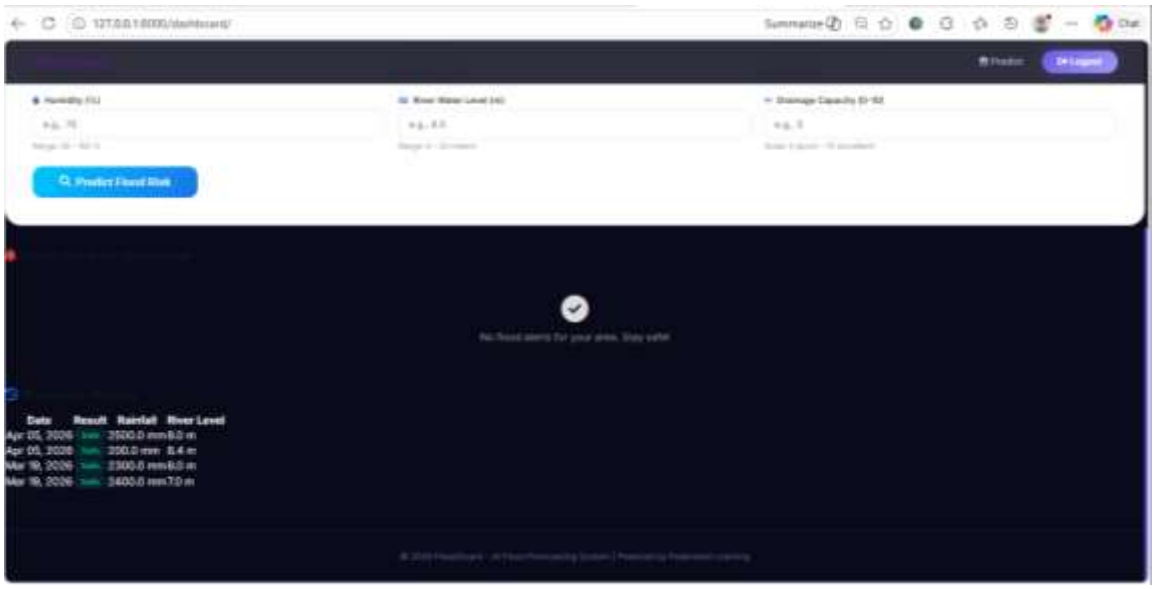
SYSTEM DESIGN IMAGES



International Journal of Engineering Research and Science & Technology







CONCLUSION

The AI-Based Flood Prediction and Alert System provides an efficient and scalable solution for disaster management. By integrating machine learning with a web-based platform, the system enables accurate flood prediction and timely dissemination of alerts to users based on their location.The system successfully addresses the limitations of traditional flood monitoring methods by introducing predictive analytics, automation, and user-specific notifications. The use of machine learning algorithms enhances prediction accuracy by analyzing complex environmental factors such as rainfall, temperature, and river levels.The implementation using Django ensures a secure and scalable architecture, while the integration of user and admin modules enhances usability and control. The system also maintains historical prediction data, allowing for analysis and improvement of model performance over time.Furthermore, the system can be extended with real-time data sources such as IoT sensors, satellite imagery, and weather APIs to improve accuracy and responsiveness. Future enhancements may include deep learning models, mobile application integration, and GIS-based visualization.Overall, this project demonstrates the practical application of artificial intelligence in disaster management and contributes to the development of smart and resilient communities.

REFERENCES

1. · Liu, B., et al. (2025). *Machine Learning Approaches for Flood Depth Estimation*.
2. · Zehra, N. (2020). *Flood Prediction using SVM and NARX*.
3. · Gao, Y. (2024). *Machine Learning in Hydrological Flood Prediction*.
4. · Srilekha, S. et al. (2024). *Comparative Study of ML Algorithms in Flood Prediction*.
5. · Reddy, K. et al. (2023). *Flood Prediction using Machine Learning*.
6. · Zhou, J. et al. (2025). *ML Models in Flood Forecasting*.
7. · Syeed, M. et al. (2022). *Flood Prediction Using ML Models*.
8. · Sidrane, C. et al. (2019). *Flood Susceptibility Prediction*.
9. · Grzesiak, M. (2024). *Quantum ML for Flood Prediction*.
10. · Mosavi, A. et al. (2019). *ML Literature Review for Flood Prediction*.
11. · Chau, K. (2017). *Applications of AI in Hydrology*
12. · Kratzert, F. (2019). *LSTM Networks for Hydrological Modeling*
13. · Abrahart, R. (2012). *Neural Networks in Hydrology*
14. · Dawson, C. (2005). *Hydrological Modeling using Neural Networks*