International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Smart Flood Detection and Alert System using Edge Machine Learning

K.Ravi kumar[1], K.kusuma[2], E. Venu kumar[3], K. Mohan[4], J.Harsha vardhan[5]
[1]Assistant professor, Dept of ECE, Gudlavalleru Engineering College, Gudlavalleru, AP
[2,3,4,5] Btech Students, Dept of ECE, Gudlavalleru Engineering College, Gudlavalleru, AP
kusumakolavennu1234@gmail.com[2], venukumarelasagaram@gmil.com[3], mohankuna32@gmail.com[4], harshajangam2004@gmail.com[5]

Abstract
Floods are among the most destructive natural disasters, causing severe loss of human life, damage to infrastructure, and disruption of agricultural activities. Conventional flood detection systems typically rely on fixed threshold values or cloud-based data processing, which can result in delayed responses and depend heavily on continuous internet connectivity. To address these limitations, this project proposes a Smart Flood Detection and Alert System using Edge Machine Learning (Edge ML).The proposed system integrates multiple environmental sensors, including an ultrasonic sensor for water level measurement, a rain sensor for rainfall intensity detection, a moisture sensor for assessing ground saturation, and temperature and light intensity sensors to assist in rain prediction.

An ESP32 microcontroller serves as the central processing unit, acquiring sensor data and executing a lightweight machine learning model (TinyML) locally at the edge. The embedded model classifies the environmental conditions into three categories—Safe, Warning, and Danger—enabling rapid and intelligent decision-making directly on the device without relying on cloud servers. This edge-based processing significantly reduces response time and ensures system operation even in low-connectivity or offline scenarios. For immediate action, the system activates local alerts through LEDs and a buzzer, ensuring timely warnings to nearby communities and relevant authorities. By leveraging Edge ML, the system minimizes latency, reduces bandwidth usage, and enhances overall reliability. The proposed solution is low-cost, energy-efficient, and scalable, making it highly suitable for deployment in flood-prone regions. This intelligent approach to disaster management contributes to improved early warning capabilities and helps protect human lives, property, and agricultural resources.

I. Introduction

Floods are one of the most devastating natural disasters worldwide, leading to significant loss of human life, extensive damage to infrastructure, and severe impacts on agriculture and the economy. The increasing frequency and intensity of floods, driven by climate change and unplanned urbanization, highlight the urgent need for efficient and reliable flood monitoring and early warning systems. Traditional flood detection methods often depend on fixed threshold values and centralized, cloud-based data processing, which can result in delayed responses and reduced reliability, especially in regions with poor or unstable internet connectivity.

To overcome these challenges, this project presents a Smart Flood Detection and Alert System using Edge Machine Learning (Edge ML). The proposed system employs a network of environmental sensors to continuously monitor critical parameters such as water level, rainfall intensity, soil moisture, temperature, and light intensity. An ultrasonic sensor measures the rise in water levels, while rain and moisture sensors assess precipitation and ground saturation conditions. Temperature and light intensity data further support accurate prediction of rainfall patterns and flood risk.

At the core of the system is an ESP32 microcontroller, which performs real-time data acquisition and executes a lightweight TinyML model directly on the device. By processing data at the edge, the system intelligently classifies environmental conditions into three levels—Safe, Warning, and Danger—without relying on cloud servers. This edge-based intelligence significantly reduces latency, ensures rapid decision-making, and enables uninterrupted operation even in low-connectivity or offline scenarios.

The system provides immediate local alerts through LEDs and a buzzer, enabling timely warnings to nearby communities and authorities. By integrating Edge ML with low-cost hardware, the proposed solution offers a scalable, energy-efficient, and reliable approach to flood detection. This intelligent early warning system enhances disaster preparedness and plays a vital role in safeguarding human lives, infrastructure, and agricultural resources in flood-prone regions.

II. Literature Survey

Flood monitoring and early warning systems have evolved significantly with advancements in sensing technologies, communication networks, and intelligent data processing techniques. Early research emphasized the role of the Internet of Things (IoT) in enabling continuous environmental monitoring through interconnected sensor networks. Gubbi et al. [2] presented a comprehensive vision of IoT, describing its architectural components such as sensing layers, communication infrastructure, and data processing mechanisms. Their work established the foundation for applying IoT in large-scale applications including environmental monitoring and disaster management, highlighting scalability, real-time data acquisition, and system interoperability.

Building upon IoT architectures, Mishra, Pandey, and Singh [4] proposed an IoT-based flood monitoring and early warning system that utilized sensors to measure water levels and rainfall intensity. Their system demonstrated effective real-time monitoring and alert generation by transmitting sensor data to centralized servers. While the approach improved flood awareness, it relied heavily on cloud-based processing and continuous internet connectivity, which could result in increased latency and reduced reliability during extreme weather conditions or in remote regions.

To address the limitations of cloud-dependent systems, edge computing has emerged as a promising solution. Bhattacharya, De, and Pal [1] explored the application of edge computing in real-time flood monitoring and early warning systems. Their study showed that processing data at the edge significantly reduces response time and enhances system reliability by minimizing dependency on network connectivity. The edge-based approach ensured faster decision-making and uninterrupted operation even during network failures, making it highly suitable for time-critical flood detection scenarios.

The effectiveness of intelligent flood monitoring systems can be further improved through the integration of machine learning techniques. Ray [5] presented a review of machine learning algorithms, emphasizing their applicability in real-time and resource-constrained environments. The study highlighted that lightweight machine learning models are capable of efficient pattern recognition and classification, making them suitable for deployment on embedded and edge devices.

In addition, LeCun, Bengio, and Hinton [3] introduced deep learning as a powerful paradigm for learning complex data representations. Their work demonstrated how learning-based models outperform traditional approaches in extracting meaningful patterns from large datasets. Although deep learning models are computationally intensive, their foundational concepts have influenced the development of optimized and lightweight models, such as TinyML, which can be deployed on low-power embedded systems for real-time analysis.

From the surveyed literature, it is evident that the integration of IoT sensing [2,4], edge computing [1], and machine learning techniques [3,5] offers an effective solution for flood monitoring and early warning. However, there remains a research gap in developing low-cost, energy-efficient, and fully edge-based intelligent systems that can operate reliably without cloud dependency. This project aims to address this gap by proposing a Smart Flood Detection and Alert System using Edge Machine Learning.

6. Nidhi Shashikumar (2026) proposed a data-driven framework integrating predictive maintenance and inventory optimization in medical device supply chains. The study employs hybrid machine learning models such as Random Forest, LSTM, and XGBoost to predict equipment failures, while ARIMA is used for demand forecasting. The integration of these models enables proactive maintenance scheduling and optimized inventory control, reducing equipment downtime, minimizing overstocking and shortages, and improving overall supply chain efficiency. The findings highlight that combining predictive analytics with inventory management enhances operational reliability, reduces costs, and supports better decision-making in healthcare logistics[6].

III. PROPOSED SYSTEM

The proposed system is a Smart Flood Detection and Alert System using Edge Machine Learning (Edge ML) designed to provide fast, reliable, and autonomous flood risk assessment. Unlike conventional flood detection systems that depend on fixed thresholds or cloud-based processing, the proposed approach performs intelligent data analysis locally at the edge, reducing latency and ensuring reliable operation even in low-connectivity or offline conditions.

The system integrates multiple environmental sensors to monitor critical flood-related parameters. An ultrasonic sensor measures real-time water level changes, a rain sensor detects rainfall intensity, and a soil moisture sensor evaluates ground saturation levels. In addition, temperature and light intensity sensors are used to support rainfall prediction and environmental condition analysis. These sensors collectively provide comprehensive data required for accurate flood risk evaluation.

An ESP32 microcontroller serves as the central processing unit of the system. It continuously acquires sensor data and executes a lightweight TinyML model deployed directly on the device. The embedded machine learning model processes the multi-sensor inputs and classifies the flood condition into three categories: Safe, Warning, and Danger. This edge-based classification enables rapid and intelligent decision-making without reliance on cloud servers.

Based on the predicted flood risk level, the system activates local alert mechanisms using LEDs and a buzzer to provide immediate visual and audible warnings. This ensures timely alerts to nearby communities and authorities, allowing preventive measures to be taken promptly. The proposed system is low-cost, energy-efficient, and scalable, making it suitable for deployment in flood-prone regions. By leveraging Edge ML, the system minimizes response time, reduces bandwidth usage, and enhances overall

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

reliability, contributing to effective disaster management and early flood warning.

A. BLOCK DIAGRAM

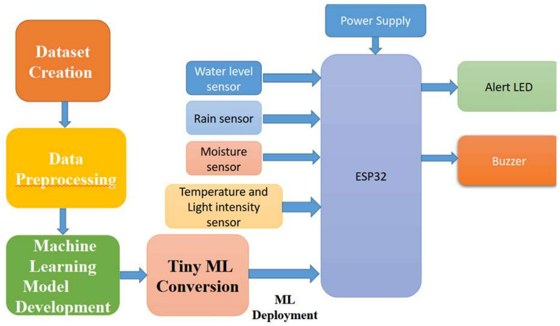

Fig.1 Block Diagram

The figure shows the overall workflow and architecture of the Smart Flood Detection and Alert System through the assistance of Edge Machine Learning. Two major phases are involved in the system; real time edge deployment and model development.

The initial step that is dataset creation and it entails collection of environmental data on the state of floods that includes the level of water, level of rain, moisture content of soil, temperature and light intensity. Such a dataset depicts the different conditions in the environment, which are the normal and extreme conditions of flooding. The data is then processed through a data preprocessing stage where the data is first filtered by removing noise, selecting features and data normalization are performed in order to feed the data into the machine learning process.

An algorithmic machine learning model is then developed on the prepared and pre-processed dataset. The sensor data is utilized to train the model in order to get the patterns and the degree of threat of floods. The trained model is then converted into a miniature format TinyML conversion, in order to be executed on small embedded devices. During the ML deployment, the converted model is loaded to the ESP32 microcontroller.

The real time set up has several sensors which feed the input data on the continuous flow to the ESP32 during the operational stage. They are water level sensor, rain sensor, moisture sensor and temperature and light intensity sensor. An electricity source is required to provide the necessary energy to the ESP32 and other parts that are attached to it. The ESP32 receives and processes sensor data and runs the deployed TinyML model on the edge to understand the current environmental conditions.

The ESP32, as per the forecast of the model, detects the condition of flood and activates appropriate alert mechanisms. Should warning or danger condition be identified, the process is done by the use of an LED to give a visual cue (enabling the alert LED) and an audible signal (regulating the usage of buzzers). These neighborhood alerts will ensure real time notification even in the absence of internet connection.

Overall, the diagram is representative of the whole edge-based intelligent flood detection system, which incorporates sensor data gathering, machine learning-iterated decision-making, and real-time alert generation and offers prompt response and low-latency and reliable work in flood-prone areas..

IV. PROJECT IMPLEMENTATION

A. Dataset Creation and Labeling:

The synthetic dataset is generated based on the key environmental parameters measured by the system sensors:

Water level (ultrasonic sensor)
Rainfall intensity (rain sensor)
Soil moisture level
Ambient temperature
Light intensity (LDR)

These features are selected as they strongly influence flood occurrence and progression.

Using predefined logical rules and thresholds, synthetic sensor values are generated to simulate different environmental conditions. Combinations of sensor readings are systematically created to reflect:

Normal weather conditions
Increasing rainfall and rising water levels
Extreme flood scenarios

Each data sample is automatically assigned a class label (Safe, Warning, or Danger) based on these rules.

B. Data Preprocessing and Model Training

The collected dataset is preprocessed offline using a computer system. This includes noise removal and splitting the data into training and testing sets. A lightweight machine learning model suitable for embedded systems is then trained using these features to classify flood risk levels. The trained model is evaluated to ensure acceptable accuracy.

c. TinyML Conversion and Optimization

To enable execution on resource-constrained hardware, the trained model is optimized using TinyML techniques such as quantization and memory reduction.

The optimized model is converted into a C compatible format (TensorFlow Lite for Microcontrollers), allowing integration with the ESP32 microcontroller.

D.Edge Deployment on ESP32

The optimized TinyML model is embedded into the ESP32 firmware using the Arduino IDE.

All sensors are interfaced with the ESP32, enabling continuous real-time data acquisition.

The microcontroller processes the sensor inputs locally and performs on-device inference to determine the current flood risk level.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

E.Real-Time Classification and Alert Generation

Based on the model output, the system classifies the situation into Safe, Warning, or Danger. Corresponding visual alerts are provided using LEDs, and an audible buzzer is activated during warning and danger conditions.

This ensures immediate local alerts without dependency on internet connectivity..

F. *System Validation*

The complete system is tested under different simulated environmental conditions to validate real-time performance, response speed.

VI. Experimental Results

The image shows the working prototype of the Smart Flood Detection and Alert System using Edge Machine Learning, assembled on a base platform for testing and demonstration purposes. The setup consists of an ESP32 microcontroller as the central control unit, interfaced with multiple environmental sensors, alert devices, and power connections.

A rain sensor module is placed on the left side of the setup, used to detect rainfall by sensing water droplets on its conductive surface. This sensor provides real-time rainfall intensity data to the ESP32. A soil moisture sensor is also connected, which measures ground saturation levels—an important indicator of potential flood conditions. The water level sensing mechanism is demonstrated using a probe-based arrangement, simulating rising water levels for testing purposes.The ESP32 board is powered through a USB cable and acts as the main processing unit, continuously acquiring data from all connected sensors. The sensor readings are processed locally on the ESP32, where a lightweight TinyML model is deployed to analyze environmental conditions and determine the flood risk level.For alert generation, the system includes LED indicators and a buzzer. The LEDs provide visual feedback corresponding to different flood states (such as safe or warning), while the buzzer generates an audible alert when critical conditions are detected. These alerts ensure immediate local notification without the need for internet connectivity.

All components are interconnected using jumper wires, allowing flexible testing and easy modification. The prototype demonstrates the practical implementation of an edge-based intelligent flood monitoring system, validating real-time sensor integration, on-device machine learning inference, and immediate alert generation. This setup confirms the feasibility of deploying a low-cost, energy-efficient, and reliable flood detection system suitable for real-world applications.

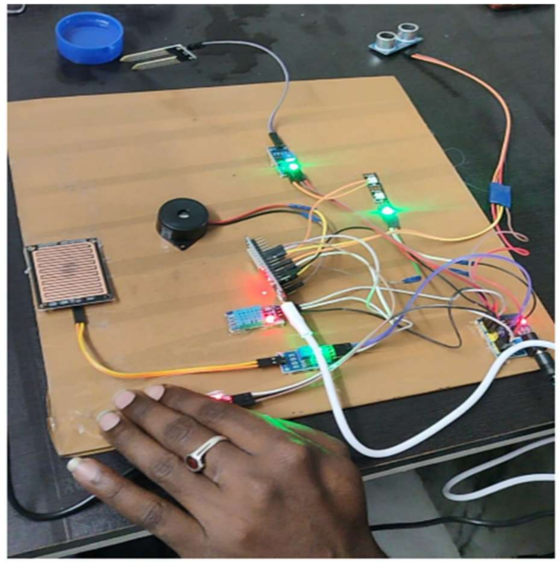

Fig 2:Kit Photo

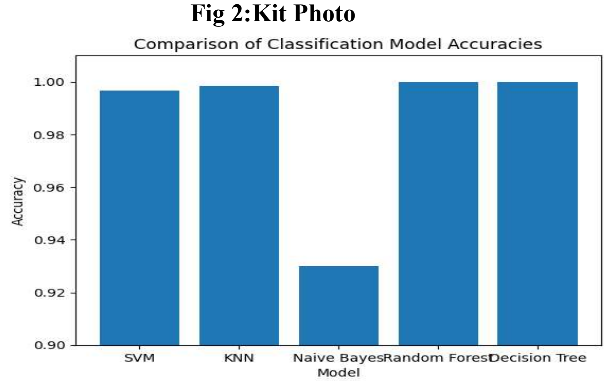

Fig 3:Algorithms Comparion

The figure illustrates a comparison of classification accuracies achieved by different machine learning models used for flood condition classification. The evaluated models include Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes, Random Forest, and Decision Tree. The accuracy metric on the y-axis represents the proportion of correctly classified instances, while the x-axis shows the different classification models.

From the graph, it can be observed that Random Forest and Decision Tree classifiers achieved the highest performance, each reaching an accuracy close to 100%, indicating excellent classification capability for the given dataset. The KNN classifier also demonstrated strong performance with an accuracy of approximately 99%, closely followed by SVM, which achieved an accuracy of around 98–99%. In contrast, the Naïve Bayes classifier recorded comparatively lower performance with an accuracy of approximately 93%, suggesting limitations in handling complex relationships among sensor features.

Overall, the results indicate that tree-based models, particularly Random Forest and Decision Tree, are highly effective for flood risk classification due to their ability to handle nonlinear patterns and feature interactions. These findings justify the selection of a lightweight tree-based or optimized model for deployment in the proposed Edge ML–based flood detection system, ensuring high accuracy while maintaining low computational overhead

VII. CONCLUSION

This project successfully presents a Smart Flood Detection and Alert System using Edge Machine Learning (Edge ML) that addresses the limitations of conventional flood monitoring approaches. By integrating multiple environmental sensors with an ESP32 microcontroller, the system enables real-time monitoring of critical parameters such as water level, rainfall intensity, soil moisture, temperature, and light intensity. The deployment of a lightweight TinyML model directly on the edge device allows intelligent classification of flood conditions into Safe, Warning, and Danger states without reliance on cloud-based processing.

Edge-based decision-making significantly reduces system latency and ensures reliable operation even in low-connectivity or offline environments. The inclusion of local alert mechanisms through LEDs and a buzzer ensures immediate warnings, enabling timely preventive actions by nearby communities and authorities. The proposed system is low-cost, energy-efficient, and scalable, making it suitable for large-scale deployment in flood-prone regions. Overall, the system enhances early warning capabilities and contributes to effective disaster management by helping protect human lives, infrastructure, and agricultural resources.

Future Scope

While the proposed system demonstrates effective flood detection and alert generation, several enhancements can be explored in the future. Additional sensors such as water flow rate, humidity, and atmospheric pressure sensors can be integrated to improve prediction accuracy. The system can be extended with wireless communication modules (such as GSM, LoRa, or NB-IoT) to provide remote alerts via SMS or mobile applications.

Advanced machine learning and deep learning models can be investigated to improve predictive capabilities, especially for long-term flood forecasting. Integration with weather forecasting data and satellite imagery can further enhance reliability. The system can also be scaled using distributed sensor networks for river basins and urban drainage systems. Incorporating renewable energy sources, such as solar power, would improve sustainability and enable long-term unattended operation. These enhancements would make the system more robust, intelligent, and suitable for real-world disaster management applications

REFERENCES

[1] Bhattacharya, S., De, D., & Pal, A. (2020). Edge computing for real-time flood monitoring and early warning systems. IEEE Internet of Things Journal, 7(10), 10031–10040.

[2] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660.

[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

[4] Mishra, A. K., Pandey, A., & Singh, P. (2019). Flood monitoring and early warning system using IoT. International Journal of Engineering Research & Technology, 8(6), 1–5.

[5] Ray, S. (2019). A quick review of machine learning algorithms. In Proceedings of the IEEE International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (pp. 35–39). IEEE. https://doi.org/10.1109/MBDC.2019.00012.

[6] N. Shashikumar, "Predictive maintenance and inventory optimization in medical device supply chains: a data-driven approach," Network Modeling & Analysis in Health Informatics and Bioinformatics, vol. 15, no. 1, p. 4, 2026, doi: 10.1007/s13721-025-00673-4.