

# IoT WITH AI-DRIVEN DISASTER FORECASTING AND RESPONSE SYSTEM USING NEURAL NETWORKS

**Mrs. K. Bhagya Rani<sup>1</sup>, Mrs. B. Maha Lakshmi<sup>2</sup>, Mrs. P. Pratyusha<sup>3</sup>, L. Kalyani<sup>4</sup>,  
K. Reshma Sri<sup>5</sup>, N. Sowmya<sup>6</sup>**

MTEch, Electronics & Communication Engineering, Bapatla Women's Engineering College, Bapatla, AP, India<sup>1</sup>

Ph.D, MTEch, Electronics & Communication Engineering, Bapatla Women's Engineering College, Bapatla, AP, India<sup>2</sup>

MTEch, Electronics & Communication Engineering, Bapatla Women's Engineering College, Bapatla, AP, India<sup>3</sup>

BTEch, Electronics & Communication Engineering, Bapatla Women's Engineering College, Bapatla, AP, India<sup>4-6</sup>

**Abstract:** The proposed Disaster Response System is a hybrid IoT and Artificial Intelligence-based solution designed using an Arduino/ESP32 microcontroller and a laptop for advanced ML/DL processing. The system collects real-time environmental data such as soil moisture, rainfall, vibration, and water level using sensors connected to the microcontroller.

This sensor data (text input) is transmitted to a laptop, where Machine Learning algorithms analyze patterns to predict disaster conditions. Additionally, Deep Learning models process image and video inputs to detect disasters such as floods, landslides, or structural damage.

Based on predictions, the system generates alerts via Telegram cloud, and LCD display. This approach ensures faster response, improved accuracy, and reduced disaster impact.

**Keywords:** Arduino/ESP32 microcontroller, soil moisture sensor, rainfall detection sensor, Ultrasonic Sensor (Water Level), MPU6050 (Vibration Sensor), Machine Learning, Deep Learning, Telegram cloud, LCD display

## I. INTRODUCTION

Natural disasters such as floods, earthquakes, and landslides have increasingly impacted human lives and infrastructure worldwide. Despite advancements in monitoring technologies, existing systems often suffer from limited real-time capabilities, poor integration between components, and delayed alert mechanisms. With the emergence of IoT and Artificial Intelligence (AI), there is an opportunity to design intelligent systems capable of continuous monitoring, data analysis, and early warning generation. This research presents a smart automated disaster prediction system that combines IoT-based data collection, ML/DL-based analysis, and cloud-based alert systems.

## II. LITERATURE SURVEY

### A. Evolution of AI-Based Disaster Prediction Models

Recent research indicates a strong shift toward intelligent systems for early disaster prediction. The work by Dauren Zhhexabay et al. (2025) highlights the effectiveness of deep learning in identifying seismic signals at early stages, significantly improving warning times. Similarly, A. Mustafa (2024) explores explainable deep learning approaches, emphasizing transparency in prediction models, which is critical for decision-making authorities.

Further contributions by Akhyar (2024) demonstrate how artificial intelligence can be applied across various disaster domains, including floods, earthquakes, and storms. Earlier foundational work by S. Kreakie et al. (2017) introduced convolutional neural networks for seismic event detection, setting the stage for modern AI-driven disaster monitoring systems. These developments collectively show a progression from basic signal processing methods to highly adaptive learning models.

### B. Limitations of Traditional Disaster Detection Systems

Conventional disaster detection frameworks face several critical challenges that limit their effectiveness in real-world scenarios.

- **High Computational Complexity:** Early models required significant processing power, making real-time deployment difficult.

- **Limited Interpretability:** Traditional machine learning models often lack transparency, which reduces trust in critical applications.
- **Data Dependency:** Many systems rely heavily on large, labeled datasets, which are not always available during emergencies.
- **Low Adaptability:** Static algorithms struggle to adjust to changing environmental conditions and unexpected disaster patterns.

Research by Ghaffarian (2023) stresses the need for explainable AI to overcome interpretability issues, while earlier multimodal studies highlighted the inefficiency of single-source data processing.

### **C. Emergence of Multimodal and Hybrid AI Frameworks**

To address the above limitations, recent studies focus on combining multiple data sources and advanced architectures. The work of Rohit Saha et al. (2022) introduces multimodal learning techniques that integrate satellite imagery, sensor data, and textual information for improved disaster prediction and response.

Similarly, V. Gupta et al. (2021) propose transformer-based models for disaster event classification, achieving higher accuracy compared to traditional neural networks. Ensemble techniques presented by L. Zhao et al. (2020) further enhance reliability by combining multiple deep learning models for post-disaster damage assessment.

Earlier work by X. Li et al. (2018) and Nguyen et al. (2019) demonstrates the importance of sensor fusion and multimodal data integration, particularly in applications like wildfire detection and infrastructure damage analysis.

### **D. Technological Integration and Practical Implementation**

Modern disaster management systems increasingly rely on integrated technological ecosystems. AI models are often deployed using programming environments like Python, enabling flexibility and scalability. Real-time data processing, supported by edge computing, ensures faster response during critical events.

Additionally, visualization tools and interactive dashboards allow authorities to interpret predictions effectively. Explainable AI frameworks, as discussed in recent literature, play a crucial role in improving user trust and system transparency. These integrated approaches ensure that disaster detection systems are not only accurate but also practical for real-world deployment.

## **III. EXISTING SYSTEM**

Early detection of earthquakes plays a crucial role in reducing potential damage and enhancing public safety. Traditional seismic detection techniques, such as the short-term average/long-term average (STA/LTA) algorithm and the Akaike Information Criterion (AIC), often face challenges in accurately identifying primary (P) waves, particularly under high-noise conditions caused by industrial and human activities. To address these limitations, recent advancements in deep learning—especially Convolutional Neural Networks (CNNs)—offer an effective alternative for seismic signal analysis. In this study, a CNN-based automatic P-wave detection model is proposed, specifically designed for the Almaty region. The model is trained and evaluated using seismic data obtained from the IRIS database, collected from seven stations within a 333 km radius of Almaty, Kazakhstan.

The proposed approach demonstrates strong performance, achieving an accuracy of 94.1% and a recall of 89.1%, outperforming several existing deep learning models. These results indicate that the CNN-based method significantly improves the precision and reliability of P-wave detection. Furthermore, the study highlights the uniqueness of region-specific seismic modeling, emphasizing importance in capturing local geological characteristics. The application of CNNs in seismic monitoring supports the development of efficient and automated earthquake early warning systems, ultimately minimizing risks and enhancing disaster response capabilities.

## **IV. PROPOSED SYSTEM**

The proposed system is a smart disaster detection framework that integrates sensor networks, machine learning, and deep learning for real-time monitoring and alert generation. Environmental data from sensors such as soil moisture, rainfall, water level, and vibration is collected using an Arduino or ESP32 and transmitted to a laptop for processing. Machine learning algorithms including Random Forest, SVM, and KNN analyze the sensor data, while deep learning models such as CNN, YOLO, and LSTM process image, video, and time-series inputs. Based on this analysis, the system predicts potential disaster conditions accurately. Alerts are immediately generated through an LCD display, buzzer, and Telegram notifications, ensuring rapid response. This hybrid approach improves detection accuracy, reduces false alarms, and enables efficient, automated disaster management.

## A. Technologies Used

Technologies and Tools	Purpose
Python	Core logic and ML and DL processing
Arduino	Sensor Data Collection And Transmission
Pandas & Numpy	Data Preprocessing
OpenCV	Image And Video Processing
TensorFlow	Deep Learning Model Implementation
Scikit-Learn	Machine Learning Algorithms
Telegram API	Alert Notification System
LCD	Displays Real Time Sensor Data

## Technologies and Tools Purpose

## B. PROPOSED SYSTEM BLOCK DIAGRAM

### Power Supply:

Provides electrical power to the entire system so all components can work.

### Soil Moisture:

Measures the water content in soil. Helps to know whether the land is dry or wet.

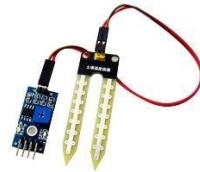


Fig 1: Soil Moisture Sensor

### Rain Sensor:

Detects rainfall. It tells whether it is raining or not.

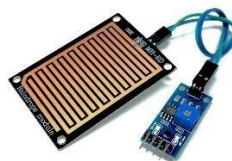


Fig 2: Rain Sensor

### Vibration / MPU6050 Sensor:

Detects vibrations or movements. Useful for identifying ground movement or disturbances.



Fig 3: Vibration Sensor

**Ultrasonic Sensor:**

Measures distance using sound waves. Used to detect object level or water level.

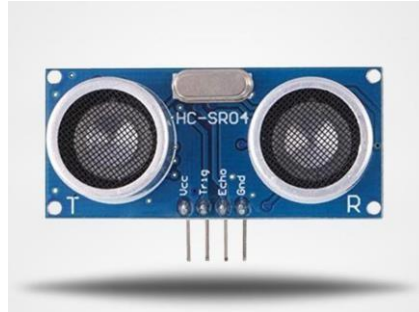


Fig 4: Ultra Sonic Sensor

**Arduino Microcontroller:**

Acts as the brain of the system.  
 It collects data from all sensors and processes it.

**16×2 LCD Display:**

Displays sensor values and system status in real time.

**Input to Camera Module:**

Captures images or videos for monitoring the environment.

**ML and DL Dataset:**

Stores trained machine learning and deep learning data used for prediction.

**Code in Laptop:**

Processes sensor + camera data using ML/DL algorithms to analyze the situation.

**Predicted Output:**

Final result after processing (e.g., flood risk, landslide warning, etc.).

**Telegram Cloud:**

Sends the result to users via Telegram.

**Alert System:**

Gives warning messages/notifications when danger is detected.

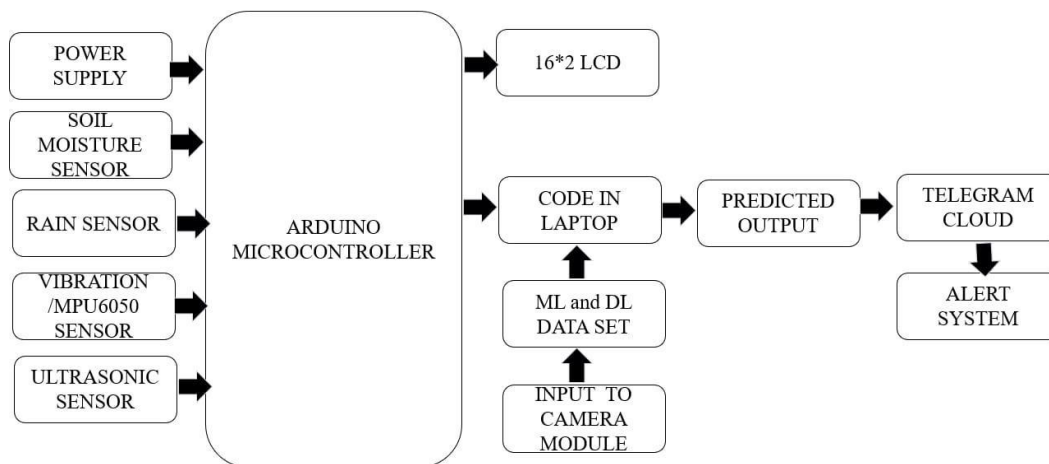


Figure 4.: Proposed System Block Diagram

V. IMPLEMENTATION DETAILS

The proposed Disaster Response System utilizes a hybrid modular architecture that integrates IoT-based sensor data collection with advanced AI-driven processing. This design ensures a seamless transition from real-time environmental sensing to intelligent disaster prediction.

**[2]. Data Acquisition and Environment**

Environmental data is collected using an Arduino/ESP32 microcontroller connected to a multi-sensor array. This setup monitors real-time parameters such as soil moisture, rainfall intensity, vibration levels, and water height to mimic dynamic field conditions.

**[3]. Data Handling and Preprocessing**

Incoming text data from sensors is transmitted via serial communication to a laptop for processing. The system handles multi-modal inputs, ensuring that both structured sensor data and unstructured image or video data are formatted correctly for model analysis.

**[4]. AI and Machine Learning Model Implementation** The system implements a dual-layered analysis approach:

**Machine Learning:** Algorithms like Random Forest, SVM, and KNN are used to analyze sensor patterns and predict disaster types.

**Deep Learning:** CNN and YOLO models process image and video feeds to detect structural damage or flooding, while LSTMs analyze time-series trends.

**[5]. Alert and Response Mechanism**

Based on the AI model output, the system triggers a multi-channel alert protocol:

**Local Alerts:** Real-time status is displayed on a 16x2 LCD and immediate danger is signaled via a buzzer.

**Cloud Integration:** Instant notifications are dispatched through the Telegram cloud API to ensure remote users are alerted immediately.

**[6]. System Integration and Scalability**

The integration of hardware (Arduino/ESP32) and software (Python, OpenCV, TensorFlow) creates a low-cost, scalable solution. This implementation is designed for diverse applications, including landslide detection, flood monitoring, and industrial safety.

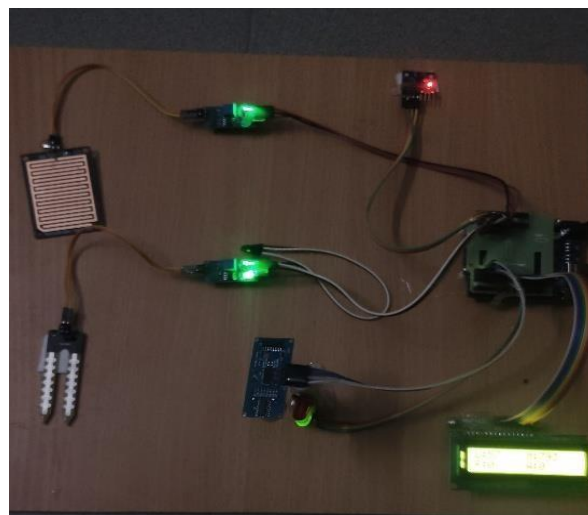
**VI. EXPERIMENTAL RESULTS**

Fig5: Output Kit

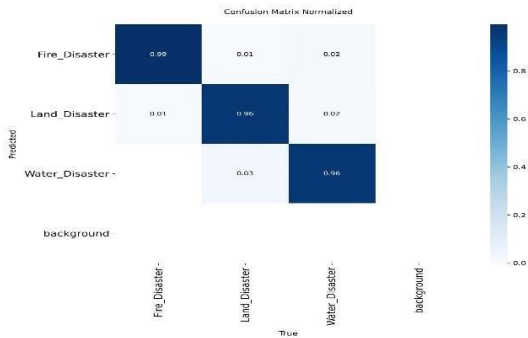


FIG 6: CONFUSION MATRIX



FIG 7: FIRE DISASTER



Fig 8: Water Disaster



Fig 9: Land Slider

## CONCLUSION

The Disaster Response System successfully integrates Arduino with laptop-based ML/DL models to provide a smart and efficient disaster prediction solution.

The system processes multiple data types and delivers accurate real-time alerts, reducing risks and improving disaster management efficiency.

## FUTURE SCOPE

- Integration with Drone Surveillance:** Drones can be used to monitor areas in real time and capture detailed data during
- Cloud-Based AI Deployment:** AI systems can be deployed on the cloud to process data quickly and work from anywhere.
- Mobile Application Development:** A mobile app can provide users with alerts and important disaster-related information.
- Satellite Data Integration:** Satellite data can help monitor large areas and detect early signs of disasters.
- Advanced AI Models (Transformers):** Advanced AI models like transformers can improve prediction accuracy.
- Expansion to Wildfire and Tsunami Detection:** The system can be extended to detect disasters like wildfires and tsunamis early.

## REFERENCES

[1]. D. Zhexebay, A. Skabylov, and M. Ibraimov, "Deep Learning for Early Earthquake Detection," vol. 12, no. 3, pp. 1–6, 2025.



- [2]. A. Mustafa, "Natural Disasters Detection Using Explainable Deep Learning," vol. 11, no. 5, pp. 10–15, 2024.
- [3]. Akhyar, "Deep Artificial Intelligence Applications for Natural Disaster," vol. 11, no. 4, pp. 20–25, 2024.
- [4]. Ghaffarian, "Explainable Artificial Intelligence in Disaster Risk Management," vol. 10, no. 6, pp. 30–35, 2023.
- [5]. R. Saha, M. Fang, and A. Yasodhara, "Multimodal Learning for Disaster Prediction, Management and Relief," vol. 9, no. 7, pp. 40–46, 2022.
- [6]. V. Gupta, S. Roy, and N. Shah, "Multimodal Transformer for Disaster Event Classification," vol. 8, no. 8, pp. 50–55, 2021.
- [7]. L. Zhao, K. Wang, and H. Lin, "Ensemble Deep Learning for Post-Disaster Building Damage Detection," vol. 7, no. 9, pp. 60–65, 2020.
- [8]. Nguyen, P. Lopez, and R. Patel, "Real-Time Wildfire Detection Using Deep Neural Networks and Sensor Fusion," vol. 6, no. 10, pp. 70–75, 2019.
- [9]. X. Li, Y. Zhang, and M. Chen, "Multimodal Fusion for Disaster Damage Assessment," vol. 5, no. 11, pp. 80–85, 2018.
- [10]. S. Kreakie, A. Sattar, and H. Kim, "Seismic Event Detection Using Convolutional Neural Networks," vol. 4, no. 12, pp. 90–95, 2017.