Vehicle-Based Disaster Alerting System Using LoRa Communication

Abhiram.RS, Arjun.RM, Pranav.J, Muhammad Shafi.MS, Ms. Athira Sarath
Rajadhani Institute of Engineering & Technology

Abstract—Natural disasters such as floods, landslides, and wildfires pose significant risks to road safety, often leading to severe accidents and fatalities. This paper presents a novel Vehicle-Based Disaster Alerting System (VDAS) that leverages LoRa communication to provide real-time warnings to drivers about hazardous road conditions. The system integrates environmental sensors with a LoRa-based communication network to detect disasters and alert vehicles within a 200-meter range. The proposed approach enhances road safety by providing early notifications, enabling preventive actions, and minimizing disaster-related accidents. The paper details the system architecture, implementation, and benefits of this innovative approach.

***Index Terms*—LoRa, disaster alert system, vehicle safety, IoT, wireless communication.**

I. INTRODUCTION

The demand for reliable and energy-efficient communication systems has increased significantly with the rise of Internet of Things (IoT) applications. The proliferation of smart cities, industrial automation, and remote environmental monitoring has necessitated the development of robust wireless technologies capable of transmitting data over long distances while consuming minimal power. Traditional wireless communication technologies such as Wi-Fi and GSM often suffer from limitations related to coverage, power efficiency, and high operational costs, making them less viable for large-scale IoT deployments.

LoRa (Long Range) technology has emerged as a promising alternative, offering a low-power, long-range, and cost- effective communication solution for various applications. LoRa operates on unlicensed frequency bands and employs spread spectrum modulation, enabling reliable data transmission over several kilometers with minimal energy consumption. These features make LoRa particularly suitable for remote sensing, industrial monitoring, precision agriculture, smart grid applications, and disaster alert systems. The ability to establish robust wireless networks in rural or infrastructure- limited areas further enhances its appeal for IoT-based applications.

This paper explores the design and implementation of an Arduino-based LoRa-controlled embedded system. The primary objective is to enable real-time data acquisition, wireless transmission, and remote accessibility of sensor data for smart applications. The system consists of multiple sensor nodes interfaced with a Node-2 Arduino processing unit that collects, processes, and transmits data using LoRa modules.

Unlike conventional wireless solutions, the proposed system ensures reliable connectivity even in challenging environments, such as disaster-prone areas and remote locations. Additionally, the implementation focuses on minimizing power consumption, extending device lifespan, and supporting scalability for various IoT use cases.
The remainder of this paper is structured as follows: Section II discusses the system architecture, including hardware and software components. Section III details the implementation process, covering hardware setup, firmware development, and network configuration. Section IV presents experimental results and performance evaluation, highlighting key findings such as communication range, power efficiency, and data accuracy. Finally, Section V concludes the paper and outlines potential future enhancements, including AI integration

II. SYSTEM ARCHITECTURE

The proposed system follows a layered architecture consisting of sensing, processing, communication, and power management units, each contributing to seamless data acquisition, transmission, and control. The sensing unit captures environmental or physical data through appropriate sensors and converts it into digital signals. The processing unit, typically a microcontroller like Arduino or ESP8266, handles data analysis, filtering, and local decision-making. The communication unit enables wireless data transmission using LoRa, Wi-Fi, or GSM, supporting both short- and long-range communication. The power management unit ensures a stable power supply through batteries or solar panels, incorporating low-power modes for energy efficiency. Overall, the system is designed to be modular, scalable, and energy-efficient, making it adaptable to a wide range of IoT applications.

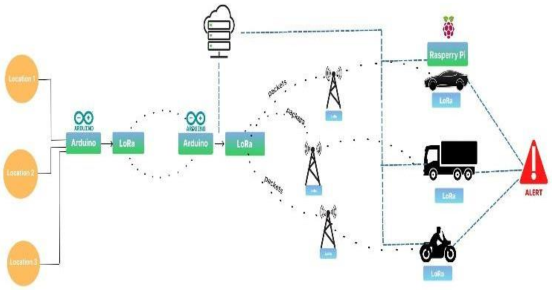


Figure 1: System Architecture

1. Sensing Layer: This layer consists of various environmental and industrial sensors that monitor parameters such as temperature, humidity, pressure, motion, and air quality. The sensors used in this system include:
   - Temperature & Humidity Sensor (DHT11/DHT22): Measures environmental temperature and humidity levels.
   - Motion Sensor (Accelerometer): Detects acceleration in X, Y, Z axes.
   - It can measure tilt, free fall, motion, and tap/double tap.
   - Low power consumption (ideal for battery- powered devices).
   - Soil Moisture Sensor: To measure the water content of the soil.
   - Water lever sensor : detects and measures the level (height) of water

2. Processing Layer: The Processing Layer of the disaster detection system is built around the NodeMCU microcontroller, which serves as the central computational and control unit. Unlike the earlier use of Arduino, the NodeMCU offers enhanced functionality with built-in Wi-Fi and better support for IoT applications. In this setup, the NodeMCU does not directly interface with the physical sensors. Instead, it receives sensor data via LoRa communication from remote nodes deployed in the field. These nodes continuously monitor environmental parameters such as vibrations, water levels, or soil movement in vulnerable areas. The NodeMCU remains in a continuous listening state, processing the incoming data and checking for any anomalies or threshold breaches that might indicate a potential disaster.

When the NodeMCU detects a disaster event, it initiates two simultaneous actions. First, it broadcasts an emergency alert via LoRa, ensuring that nearby devices—such as roadside display units or in-vehicle receivers—are immediately notified. This broadcast includes important details like the latitude, longitude, and type of disaster detected. Second, the NodeMCU uses its Wi-Fi capability to upload the disaster data to a Firebase cloud server, enabling centralized monitoring, alert logging, and integration with mobile or web applications for remote access. The system is designed for low power consumption, making it suitable for long-term deployment in remote areas. Overall, this layer ensures quick, reliable communication and real-time decision-making for effective disaster response.

3. Communication Layer: The system employs LoRa technology to enable long-range wireless communication between sensor nodes and the central gateway. The key features of this layer include:

- LoRa Transceiver Module (SX1276/SX1278):

Facilitates long-range data transmission, typically up to 5 to 10 kilometers in open environments, making it ideal for wide-area monitoring in disaster-prone regions.

- LoRa Gateway:

Acts as a central hub that collects data from multiple sensor nodes and forwards it to a cloud-based server or local database for further processing and alert generation.

4. Power Management Layer: To ensure long-term operation in remote or battery-powered environments, the system integrates an optimized power management module, which includes:
   - Rechargeable Lithium-Ion Battery: Provides a sustainable power source with sufficient capacity for prolonged usage.
   - Solar Panel (Optional): Supports renewable energy harvesting for self-sustained deployments.
   - Low-Power Sleep Mode: Extends battery life by reducing energy consumption during idle periods.

- Voltage Regulators & Power Optimization Circuits: Ensure stable power distribution across components.

The modular nature of this architecture allows easy customization and expansion by integrating additional sensors, communication modules, or processing units based on specific application needs.

III SENSOR OVERVIEW

1. Flood Detection: Water Level Sensors:
Monitors rising water levels in flood-prone areas. Triggers alerts when a threshold is exceeded. Helps in predicting flood risks based on seasonal patterns.

Figure 2 : NodeMCU microcontroller

5. Earthquake Detection:
Seismic Sensors (Accelerometers like ADXL345): The ADXL345 accelerometer is used in earthquake detection systems to monitor ground vibrations and acceleration across three axes (X, Y, Z). It continuously measures the intensity and frequency of vibrations, allowing the system to distinguish between normal ground movement and hazardous seismic activity. When the detected acceleration exceeds a predefined threshold, it indicates a possible earthquake.

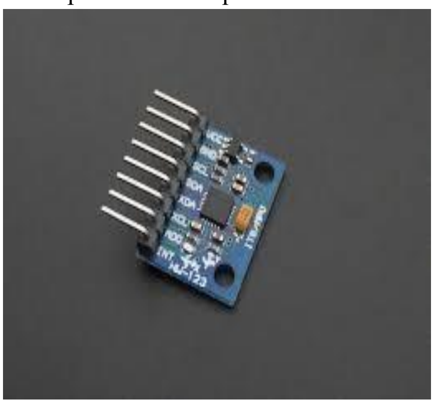


Figure 3: Seismic Sensors

2. Landslide Detection: Slope Stability Sensors:
Measures the angle of slopes to detect unstable terrain.

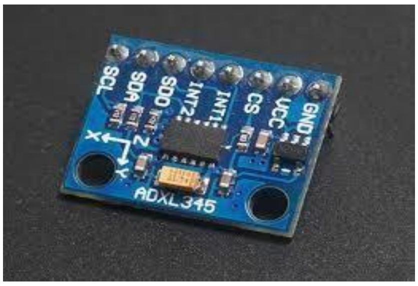
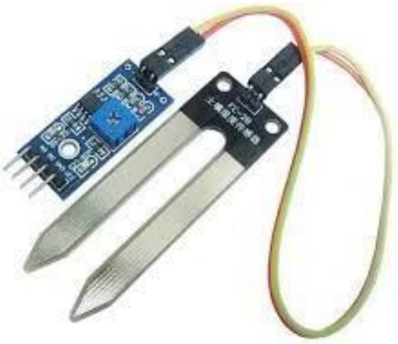


Figure 4: Slope Stability Sensors

3. Soil Moisture Sensors:
Monitors soil moisture content. Helps predict landslides based on water saturation and slope movement.



4. Forest Fire Detect Sensor(DH11): The fire detection system uses the DHT11 sensor to monitor both temperature and humidity levels in the surrounding environment. By analyzing these readings, the system can detect abnormal heat patterns that may indicate the presence of fire. The combination of high temperature and low humidity helps determine the intensity and severity of a potential fire. This data is crucial for assessing the impact of the fire, especially in sensitive areas like forests, and enables the system to send timely alerts for rapid response and mitigation.

Figure 5: Forest Fire Detect Sensor

How These Sensors Work in the System:

- Disaster Alert Sources (Locations 1, 2, 3):

Multiple field locations act as disaster detection points, each equipped with various environmental sensors (such as vibration, temperature, and water level sensors), along with a GPS module to determine the exact location of the detected disaster.

- Node 1 (Sensor Node) + LoRa Transmitter Units

At each disaster detection location (Node 1), the sensors are connected to a LoRa transmitter module through a microcontroller (such as Arduino) Node 1 collects sensor data and sends it via LoRa to the central Node 2 for processing.

- Node 2 (Central Node) + LoRa Receiver and IoT Integration

Node 2 acts as the central hub of the system. It receives the data from Node 1 (via LoRa), processes it, and determines the disaster type, severity, and location (latitude and longitude) of the affected area.

- Disaster Alert Notification System

Upon receiving the alert, the infotainment system triggers a visual warning (e.g., red exclamation mark
⚠ ALRT), displaying the real-time location of the disaster on a map integrated within the vehicle's system. This provides drivers with immediate awareness of the disaster's location and helps them make informed decisions to avoid danger.

Figure 6: MICROCONTROLLER

Arduino Uno is a highly versatile microcontroller that plays a crucial role in collecting and processing data from various sensors, such as water level, seismic (ADXL345), soil moisture, and temperature sensors, to detect potential disasters. It is widely used due to its simplicity, reliability, and flexibility in handling real- time inputs. By integrating with a LoRa module, the Arduino Uno can communicate disaster alerts to nearby vehicles or a cloud-based monitoring system, ensuring quick responses to emergency situations.

Advantages of Using Arduino Uno:

Ease of Use: Simple to program, with a large community and abundant resources.

Multiple GPIO Pins: Allows for the connection of multiple sensors for comprehensive environmental monitoring.

Built-in USB Interface: Makes programming and power management straightforward.

Compact and Power-Efficient: Ideal for vehicle-mounted or remote installations where space and power efficiency are crucial.

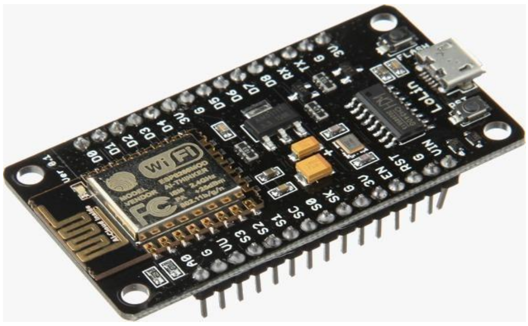


Figure 7: NODEMCU

The NodeMCU is a powerful and compact microcontroller based on the ESP8266, offering Wi-Fi capabilities. It plays a key role in receiving disaster data from sensors and processing it in real-time. NodeMCU efficiently broadcasts alerts through LoRa and can upload data to a cloud server.

Figure 8: RASPBERRY PI 2

The Raspberry Pi 2 is a credit-card-sized computer equipped with a 900 MHz quad-core ARM Cortex-A7 processor, 1GB of RAM, and multiple connectivity options, including Ethernet and USB ports. It is ideal for handling data processing tasks in disaster detection systems. The Pi 2 can receive alerts from other nodes (such as NodeMCU), process disaster-related data, and send notifications to cloud services or display real- time information on vehicle screens.

Figure 9: DISPLAY

The *15-inch Screen* is used in the demo setup for real-time data visualization and status monitoring. It displays sensor readings, disaster alerts, system logs, and network status, allowing users to observe the functionality and behavior of the system. While this screen serves as a demonstration tool, in real-world applications, the results would be shown on a vehicle's *infotainment system*, providing disaster alerts and information directly to the driver during emergencies.

IV REAL-WORLD IMPLEMENTATION

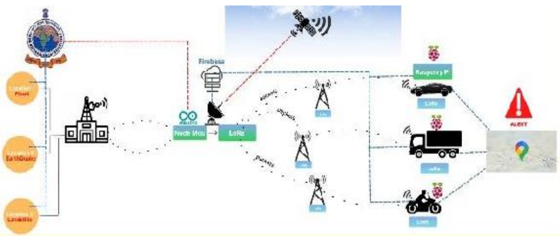


Figure 10: Alerting System Architecture

The Vehicle-Based Disaster Alerting System is a robust and innovative solution designed to deliver timely, location-specific alerts to drivers during natural disasters. It integrates official disaster monitoring systems, Internet of Things (IoT) devices, long-range LoRa communication technology, and cloud-based platforms to create a highly reliable and scalable warning mechanism. The primary goal of the system is to enhance safety on the roads by ensuring that drivers are informed in real-time about potential hazards such as floods, earthquakes, and landslides, even in areas with poor internet connectivity.

From the central node, the alert will be sent using LoRa, uploaded to Firebase, and also transmitted via satellite communication. This multi-channel alert mechanism enhances the durability and accuracy of the system. If a vehicle has internet access, it can receive the alert from the Firebase cloud server. If there's no internet, it receives alerts via LoRa. In cases where both LoRa and internet fail due to severe disasters like floods or earthquakes, the alert will still be received through satellite communication.

At the core of the system are disaster detection centers operated by official monitoring agencies. These agencies are equipped with advanced sensors and technologies capable of detecting natural disasters as they occur. When such an event is detected, an alert is generated and transmitted to a local control unit. This control unit consists of a NodeMCU (ESP8266) microcontroller paired with a LoRa module. The NodeMCU serves as the main communication hub, responsible for uploading the alert information to the Firebase cloud platform, while simultaneously transmitting the same data via the LoRa module to a distributed network of LoRa relay towers, and also sending it through satellite communication for maximum coverage and reliability.

The LoRa communication network forms the backbone of the offline alerting system. LoRa, known for its long-range and low-power capabilities, is ideal for reaching remote regions where internet or cellular coverage is limited or nonexistent. Relay towers receive the alert packets from the control unit and rebroadcast them across vast distances, ensuring that the warning signal propagates through multiple layers of the network until it reaches its destination. This offline redundancy ensures that even during network outages caused by the disaster itself, critical alerts continue to reach the people who need them.

Vehicles traveling in disaster-prone zones are equipped with Raspberry Pi boards connected to LoRa receivers. These embedded systems constantly listen for incoming alert signals from the relay towers. When an alert is received, the Raspberry Pi processes the data and activates various onboard warning mechanisms. These can include visual alerts on an in-vehicle display, audible alarms to capture the driver's attention, and map-based notifications highlighting the affected zones. The system will show the exact location of the disaster, allowing the vehicle to reroute or avoid dangerous areas. This immediate feedback allows drivers to respond quickly, minimizing the risk of accidents or entrapment.

Complementing the offline LoRa network is the Firebase cloud system, which serves multiple purposes. It acts as a centralized database for logging and managing alert events and enables remote access to alert data from anywhere with internet connectivity. Firebase can also support mobile and web applications used by emergency responders, transport authorities, or civilians to monitor disaster alerts in real time. This dual-mode approach—combining cloud and LoRa— ensures that the system remains functional and effective under a wide range of conditions.

The system is particularly advantageous due to its reliability, scalability, and minimal energy requirements. It can be deployed in geographically challenging areas such as mountains, forests, and rural roads, and it supports seamless expansion by adding more relay towers or vehicle units. By delivering alerts instantly and efficiently, it minimizes the time gap between disaster detection and driver response. Furthermore, future enhancements such as AI-based prediction models, voice-driven alerts, and integration with navigation services like Google Maps could significantly augment the system's functionality, making it a powerful tool in the effort to save lives during natural disasters.

V RESULT

Figure 11: Smart Vehicle Alert System

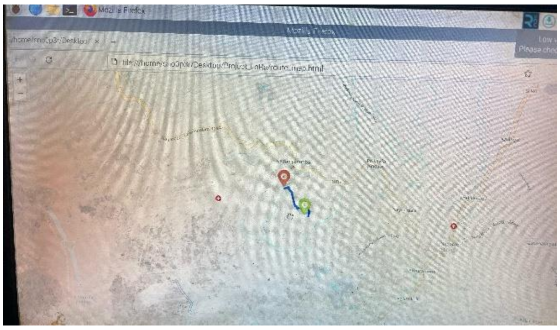


Figure 12: Map-Based Disaster Location Display

The system integrates a map view within the vehicle's infotainment interface to visually indicate the *exact location of the disaster Upon receiving alerts via LoRa, the Raspberry Pi decodes the latitude and longitude and plots the position on the map in real- time. This helps drivers understand the proximity and direction of the hazard. The demo simulates this using a monitor, while actual deployment supports seamless map integration in infotainment systems

Figure 13: Disaster Alert Pop-up in Vehicle Infotainment System

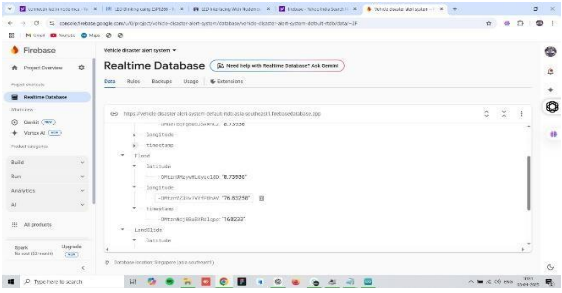
Disaster alerts are displayed as real-time pop-ups on the vehicle's infotainment system. These alerts include the type of disaster,latitude and longitude of the affected area, and optionally the *distance from the vehicle. The system uses a Raspberry Pi to receive LoRa-based alerts and trigger the visual display. In the demo setup, a monitor simulates this interface, replicating how alerts would appear in actual vehicle infotainment systems.

REALTIME DATABASE

In the proposed system, disaster alerts are generated based on sensor data and transmitted to a Firebase cloud server for centralized processing. These alerts contain critical information such as the type of disaster, latitude, and longitude of the affected location. Upon receiving this data, the system retrieves and visualizes the alerts on a monitoring interface, displaying the exact coordinates of the event for real- time response.

Alert Type: The type of disaster (e.g., earthquake, flood, fire) is categorized and displayed in the alert.

Latitude and Longitude: The geographic coordinates (latitude and longitude) of the disaster's location are included, allowing for accurate location tracking and response coordination.



VI CONCLUSION

The Vehicle-Based Disaster Alerting System offers a practical and scalable solution for improving road safety during natural disasters, especially in areas with limited communication infrastructure. By leveraging LoRa communication, the system ensures reliable, long-range transmission of disaster alerts even in remote locations where traditional networks like cellular or Wi-Fi may fail. The integration of Raspberry Pi in vehicles enables real-time data processing, allowing for immediate alerts to drivers, which significantly enhances their ability to respond to hazards like floods, landslides, or earthquakes.

This system has the potential to be deployed in real-world scenarios, providing a robust disaster alert mechanism where timely information can save lives. Its adaptability to different environments—whether rural, mountainous, or disaster-prone—demonstrates its resilience in providing continuous communication, even when internet connectivity is unavailable. With additional enhancements like satellite communication for remote areas, AI-based prediction models, and map integration for rerouting vehicles, this system can further improve the effectiveness of disaster management. In the long term, such technologies could be adopted globally, ensuring safer travel and better disaster preparedness.

VII ACKNOWLEDGMENT

We sincerely express our gratitude to everyone who contributed to the development of this Vehicle-Based Disaster Alerting System. Special thanks to our mentors and advisors for their valuable guidance and technical insights, which played a crucial role in shaping this project.
We also appreciate the contributions of our team members for their dedication and collaboration in designing and implementing the LoRa-based disaster communication network. Additionally, we acknowledge the efforts of researchers and developers whose work in LoRa technology, IoT-based alert systems, and disaster management inspired and informed our approach.

Finally, we extend our thanks to the institutions and organizations that provided resources, support, and encouragement for this project. Their support has been instrumental in developing a reliable, real-time disaster alerting system that aims to enhance safety and emergency response for vehicle users.

REFERENCES

[1] M. Rashid, V. Nagarajan, R. Poonia, P. Singh, and A. Jain, “V2X-based highly reliable warning system for emergency vehicles,” *Applied Sciences*, vol. 13, no. 3, p. 1950, 2023, doi: 10.3390/app13031950.

[2] K. Asokan, A. Sundararajan, R. Ramesh, and R. Sujith, “Adaptive vehicle communication for dynamic hazard awareness using IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3954–3962, 2020, doi: 10.1109/JIOT.2020.2969212.

[3] A. Alsharef, R. Hafiz, and M. Hashem, “Integrated V2V communication systems for collision avoidance and hazard alert in urban scenarios,” *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 646–653, 2020, doi: 10.1109/TIV.2020.2992551.

[4] M. Flannery and A. Garcia, “Internet of vehicles: A comprehensive approach to improve vehicle safety and disaster alerting in adverse conditions,” *International Journal of Transportation Science and Technology*, vol. 9, no. 3, pp. 265– 277, 2020, doi: 10.1016/j.ijtst.2019.11.004.

[5] T. U. Jamil, M. El-Khateb, M. M. Ali, and A. R. Al-Ali, “Cloud- based early warning system for disaster management in smart cities,” *Sustainable Cities and Society*, vol. 66, p. 102658, 2021, doi: 10.1016/j.scs