Earthquake Predictor and Alert Provider System

More Aditya Sanjay[1], Khemnar Mayur Rajendra[2], Prof. Dr. Khatal S.S.[3]
[1,2,3]*Sharadchandra Pawar College of Engineering, Otur, Maharashtra, India*

***Abstract*—Earthquakes can cause severe damage within a short time, making timely public warnings important. This paper proposes an Earthquake Predictor and Alert Provider System that collects seismic observations from a vibration sensor or an authorized seismic data feed, evaluates the incoming measurements, and sends alerts when configured thresholds or verified warning messages indicate a possible earthquake-related event. The proposed design combines a sensing unit, a microcontroller such as ESP32, a communication network, a processing module, and a notification interface. The system is intended as an educational prototype for earthquake activity monitoring and rapid alert delivery, not as a device capable of reliably predicting the exact time, location, or magnitude of an earthquake. Actual performance depends on sensor quality, network availability, calibration, and validation against trusted seismic data.**

***Index Terms*—alert system, earthquake monitoring, ESP32, seismic sensor, vibration detection, warning notification.**

I. INTRODUCTION

Earthquakes are sudden movements of the Earth's crust caused by the release of energy along faults. Their effects may include structural damage, disruption of essential services, and risks to human life. Because damaging shaking may begin with little or no local notice, monitoring and fast communication are important parts of disaster preparedness. A low-cost prototype can demonstrate how sensor readings and trusted seismic information can be used to trigger a warning message. The proposed Earthquake Predictor and Alert Provider System is designed to monitor ground vibration, process sensor readings, and notify users when a defined event condition is detected. In a practical deployment, an accelerometer or geophone provides measurements, an ESP32 or similar controller processes the data, and Wi-Fi or an internet connection transfers alerts to a dashboard or mobile notification service. The term 'predictor' in the project title refers to monitoring and event assessment; it does not mean that earthquakes can be predicted precisely from a simple sensor.

II. PROBLEM STATEMENT AND OBJECTIVES

The main problem is to provide a simple and affordable way to monitor vibration-related events and deliver alerts quickly. Stand-alone sensors may generate false alarms due to footsteps, vehicles, construction, or sensor movement. Therefore, the system should use calibration, repeated readings, threshold logic, and where available, confirmation from trusted seismic data sources.

A. Objectives

- Monitor vibration or acceleration readings continuously at a configured sampling rate.
- Identify readings that exceed a calibrated threshold for a specified duration.
- Send a warning to a dashboard, buzzer, LED, or mobile notification channel.
- Record time-stamped readings and alert events for later review.
- Reduce false alarms through filtering, persistence checks, and external data validation.
- Provide a low-cost prototype that can be tested and improved.

III. PROPOSED SYSTEM

The system consists of five main blocks: (1) sensing unit, (2) controller and processing unit,

(3) communication module, (4) alert interface, and (5) data log or dashboard. A vibration sensor or accelerometer measures local motion. The controller reads the sensor, filters noise, calculates a simple signal measure, and compares it with the calibrated threshold. If the condition persists, the controller creates a preliminary alert. If a trusted earthquake early-warning feed is available, its verified message should take priority over a stand-alone vibration threshold.

A. Hardware Components

- ESP32 development board or equivalent microcontroller with Wi-Fi.
- Accelerometer module (for example, an appropriately ranged MEMS sensor) or a suitable vibration sensor.
- Buzzer and LED for local indication.
- Stable power supply, connecting wires, and a protected enclosure.
- Optional display or storage module for showing and recording measurements.

B. Software Components

The software may use Arduino IDE or a similar development environment for the controller, a filtering and threshold routine for signal processing, and a web dashboard or messaging service for notifications. The implementation should store timestamps, sensor values, threshold status, network status, and alert delivery results. Any third-party API must be used according to its access rules and data license.

IV. WORKING METHODOLOGY

- Initialize the controller, sensor, network connection, and alert interface.
- Read sensor measurements at regular intervals and check for missing or invalid values.
- Apply calibration and basic filtering to reduce noise and unrelated vibration.
- Calculate a signal feature, such as peak acceleration or short-window vibration amplitude.
- Compare the feature with a threshold determined during controlled testing.
- Require the threshold condition to persist across multiple samples before raising a preliminary local alert.
- If available, check a trusted seismic data or early-warning service before issuing a confirmed event message.
- Send the alert with time, sensor location, event status, and a clear instruction to follow official authorities.
- Log the event and return to continuous monitoring after the condition clears.

V. ALGORITHM

Input: sensor readings, calibrated threshold T, persistence count N, and optional trusted seismic feed.

Output: local warning and/or network notification.

1. Start the system and initialize all modules.
2. Acquire a new sensor sample and validate it.
3. Filter the sample and calculate the selected vibration feature A.
4. If A exceeds T, increase the consecutive-event counter; otherwise reset or decrease it.
5. If the counter reaches N, mark the condition as a preliminary vibration alert.
6. When a trusted seismic feed is configured, verify whether an official event or warning supports the alert.
7. Activate the local buzzer/LED and send a notification containing the event status.
8. Save the readings and alert record, then continue monitoring.

Note: T and N must be selected by calibration and validation; they are not universal earthquake thresholds. A vibration threshold alone cannot reliably distinguish an earthquake from every other source of motion.

VI. PERFORMANCE EVALUATION

The prototype should be evaluated in controlled tests using known vibration inputs and, where possible, historical or authorized seismic datasets. Testing should not involve creating dangerous shaking or placing people at risk. Record the number of correctly detected test

events, missed events, false alerts, alert delivery time, packet loss, and system uptime. Precision can be calculated as TP/(TP+FP), and recall as TP/(TP+FN), where TP is a correctly detected event, FP is a false alert, and FN is a missed event. Notification latency is the time between the detection decision and receipt of the message. No experimental measurements are claimed in this report because a physical prototype and test dataset have not been supplied. Results should be added only after tests are performed. The system's performance will depend on sensor placement, sampling rate, calibration, power stability, internet connectivity, and the quality of the reference data.

VII. LIMITATIONS AND SAFETY CONSIDERATIONS

- A low-cost vibration sensor cannot reliably predict the exact time, location, or magnitude of an earthquake.
- Threshold-based detection may trigger false alarms from traffic, machinery, footsteps, or construction.
- A network outage or power failure can delay or prevent remote notifications.
- A single sensor provides limited information about event location and seismic magnitude.
- The prototype must not replace official earthquake early-warning systems, emergency services, or public safety instructions.

VIII. APPLICATIONS AND FUTURE SCOPE

Possible educational applications include laboratory demonstrations, vibration monitoring in buildings, local sensor-network experiments, and dashboards that display seismic data. Future improvements may include multiple synchronized sensors, GPS-based timestamps, edge processing, secure communication, battery backup, data visualization, and integration with an authorized earthquake early-warning service. A larger deployment would require engineering-grade sensors, robust calibration, independent validation, fault-tolerant networking, and review by relevant seismic and public-safety experts.

IX. CONCLUSION

This paper presents a proposed Earthquake Predictor and Alert Provider System for monitoring vibration-related events and distributing timely notifications. The design uses a sensor, a microcontroller such as ESP32, processing logic, a communication link, and local or remote alert outputs. Threshold checks and persistence logic can demonstrate event detection, while trusted seismic data can improve the reliability of event confirmation. The system is suitable as a student prototype for learning about sensors, IoT communication, data logging, and alert delivery. However, a simple prototype cannot provide dependable earthquake prediction or replace official warning services. Calibration, controlled testing, false-alarm analysis, and clear safety messaging are essential before any real-world use.

ACKNOWLEDGMENT

The authors may acknowledge their project guide, department, institution, and any organization that provided equipment or technical support. Replace this text with the actual acknowledgments before submission.

REFERENCES

[1].U.S. Geological Survey, “Earthquake Hazards Program,” accessed Oct. 2026.
[2].U.S. Geological Survey, “Earthquake Early Warning,” USGS Earthquake Hazards Program, accessed Oct. 2026.
[3].Espressif Systems, “ESP32 Series Datasheet.”
[4].Arduino, “Arduino Documentation,” accessed Oct. 2026.