Progress Report 3

Developing a Low-Cost, Machine Learning-Based Early Earthquake Detection System

Eliab Melaku

Nikhil Patil

Mahib Rahman

Ales Psaker

January 21, 2025

Objective: Set up an accurate timeline for designing/building our project, find more data sets to train our model and go as far as possible at the Science Fair.

Material Reception

During the past weeks, all the materials we ordered came in the mail. This includes the breadboard, Raspberry Pi 4, etc. We also spoke with a professor at George Mason University main campus on a request for the use of APS 400 Electro-seis. This is an electrodynamic force generator, the output of which is directly proportional to the instantaneous value of the current applied. We will use APS-400 as an arbitrary waveform input from online data sets to test our detector against real earthquakes. The professor has since stated that he will keep us updated on the availability of the device. The Raspberry Pi model 4 will act as a gateway to implementing machine learning. Now that we have all the materials, we can begin the construction process. Obviously, we need a dedicated plan for this process, as in person data is due February 5. Currently, we have a tentative schedule for how this process will be completed:

January 21 (Today)- Mechanical Assembly

- Mount breadboard + ESP 32 on the aluminum baseplate using standoffs.
- Decide on Sensor Placement; secure everything loosely (No permanent zip-ties yet).

January 22- Power System Bring-Up

- Connect the power bank to the breadboard rails.
- Verify the 5-volt rail and the 3.3-volt rail.
- Use the multimeter to check for stable voltage.
- Power the ESP32 alone (no sensors yet).

January 23- ESP32 Basic Firmware

- Flash ESP32 with a serial output test and an SD card detection test.
- Confirm dummy data can be logged every second, the boot reliability, and the fact that the SD card can create/write files.
- By the end of the day, make sure the ESP32 and SD logging work 100%.

January 24- PZT and Amplifier (Soldering day). Takes time due to this being the most failure-prone step.

- Solder wires to the PZT disc.
- Build OPA2132 charge amplifier on breadboard.
- Connect the amplifier output to the ESP32 ADC pin.
- Test: Tap table, voltage spike.
- No clipping/no constant saturation.

January 25- Accelerometer Integration

- Wire the MEMS accelerometer.
- Verify the static gravity reading, signal clearness during vibration, and the log accelerometer and PZT simultaneously.
- Make sure dual-sensor data appears in SD logs

January 26- Noise and Stability Testing

- Let the system sit idle for 30 minutes.
- Check false triggers, noise floor stability, and secure wiring with zip ties.

- Retest after securing.

January 27- LoRa + Wake-On Radio

- Wire the LoRa module to the ESP32.
- Configure deep sleep mode and wake on vibration threshold.
- Test the system: Vibration is transmitted, system wakes, event is logged.
- We need to measure the wake-up latency, making sure the delay is minimal.

January 28- Full System Test

- Run 30-second trials.
- Log PZT voltage, three-dimensional accelerometer data, and wake time.
- Repeat tests for 5 trials.
- Make sure to label all data clearly.

January 29- Shake Table/Simulated Earthquakes

- Input CESMD waveforms (or simulated equivalents).
- Run multiple magnitudes.
- Record detection success, missed events, and false alarms (try to keep below one percent).

January 30- Hardware Freeze

- No new wiring.

- No component changes.
- Only make bug fixes if the system breaks.
- Take the final photos of the full system, sensors, and full wiring layout.

January 31- Machine Learning and Hardware Integration

- Transfer the SD data to the Raspberry Pi.
- Run the Machine Learning Classifier on real hardware data.
- Confirm that the predictions align with expectations.

February 1- End-to-End Demo

- Demonstrate the whole system working together: Vibration, detection, logging, and lastly, machine learning output.
- Write down the detection latency and energy behavior. Make sure latency is minimized.

February 2- Failure Mode Testing

- Test for weak vibrations and random noise.
- Note these limitations clearly.

February 3- Reliability Day

- Run the system for over an hour.
- Ensure no crashes occur, no SD corruption, and have a final backup of data prepared.

February 4- Final Physical Preparation

- Charge Batteries.
- Pack spare wires, SD cards, and power banks.
- Clean the wiring and baseplate.

February 5- Presentation Day

- Power on the system early.
- Do one quick tap test.
- Leave system untouched during judging.

Risk and Safety

We will be soldering equipment and using electrical equipment so safety measures must be taken. We will stay safe by using electrical gloves and using carefulness when dealing with the system’s construction. Other methods we can use to stay safe are by only using low-voltage systems, and we will avoid coming in contact with exposed circuits. Another risk can be thermal radiation from batteries. We must avoid overheating the batteries by charging them to an appropriate area. Lastly, the testing area must be neat and safe. Tasks that can ensure safety can be wearing safety glasses for the eyes and maintaining a reasonable distance from the site of testing.

Data Sourcing Methods and Analysis

Regarding preliminary machine learning testing, after training our model on the V2C data which consists of separate acceleration and velocity data, we tested it on V1C data which only contains acceleration data. This means we will have to adapt our dual-sensor input approach. To achieve this, we must duplicate the acceleration data to create a pseudo-dual-sensor input, meaning the same acceleration signal will also go through the velocity channel. This will allow the model to detect earthquakes on a different sensor type. Additionally, we sought to gain an insight into the effectiveness of using CESMD (Center for Engineering Strong Motion Data, n.d.) processed seismic waveform structure and machine learning–based detection performance to test the efficiency on raw unfiltered data. The results from preliminary ML testing were intended to set up a strong data-driven preliminary foundation for our detection system. From CESMD, V2C-level ground acceleration waveform data records were used to train a 1D CNN machine learning model as they derived from high quality strong-motion that were designed to capture near-field shaking alongside being manually processed by seismologists.

Post training the model, we tested against V1C level waveform data. They were not used to training as they come from broadband seismometers that are optimized for capturing ground motion over long distances. Due to this, V1 level data is classified as ‘raw’ data by CESMD and helps simulate the proposed dual sensor system as it relies on sharp vibration sensitivity like that of our MEMS accelerometers and piezoelectric PZT sensors. To add on, V2 and V2C data from unseen locations with varying magnitudes and ground accelerations will be tested to evaluate effectiveness of generalizing a model across differing events and geological regions.

The data we collected and analyzed were used to answer research questions and hypotheses. Our objective is to use this data to evaluate the performance of the piezoelectric transducer network that is integrated with our machine learning classifier. We would then compare the results of our detection system with the performance of traditional continuous-monitoring systems, more specifically those that are in developing countries that are not able to afford advanced seismic detection.   The first step in analysis involves data collection from the dual sensor archetype. This network converts vibration to a proportional voltage whenever there is seismic activity. The OPA2132 circuit amplifies the voltage signal, making it readable data for the ESP32 microcontroller.

All the sensor data is transferred from the ESP32 sensor node to the Raspberry Pi 4 wirelessly, via LoRa communication. Once the data is stored into the Raspberry Pi 4, it processes the digital data and classifies parameters such as the root mean square amplitude, signal energy, zero-crossing rate, and the dominant frequency for each event. These features will then be organized into labeled datasets and evaluated as “false alarm”, “potential seismic activity", and "other environmental disturbances”. Then, model performance is measured using accuracy, precision, recall, and F1-score metrics. Additionally, the responsiveness of the system will be evaluated by measuring the detection-to-alert latency and communication delay of the two LoRa-connected sensor nodes. To analyze the energy efficiency of this system, ESP32’s ability to measure current drawn will allow the system to log the amount of voltage and current drawn to calculate energy usage per event. From there, we will compare the statistical values of energy consumption between our wake-on-radio functionality to that of the continuous monitoring mode.

Lastly, all data is collected and summarized with graphs and statistics. Differences in energy consumption and alert latency between the two modes will be illustrated with bar graphs. Additionally, confusion matrices will be used to evaluate classification performance. From these analyzations, an evaluation will be made on the system as a whole. More specifically, we will determine if our early earthquake detection prototype design is an energy efficient system and capable of providing early earthquake warnings for developing regions of the world that lack such technology and funds to create their own.

References

Center for Engineering Strong Motion Data. (n.d.). *CESMD strong-motion data repository*. University of California, Berkeley. https://www.strongmotioncenter.org

Earthquakes. (n.d.). Www.mcasfutenma.marines.mil. https://www.mcasfutenma.marines.mil/Resources/Installation-Protection/Earthquakes/

Hung, S. L., Ding, J. T., & Lu, Y. C. (2018). Developing an energy-efficient and low-delay wake-up wireless sensor network-based structural health monitoring system using on-site earthquake early warning system and wake-on radio. *Journal of Civil Structural Health Monitoring*, *9*(1), 103–115. https://doi.org/10.1007/s13349-018-0315-2

Jaffe, H., & Berlincourt, D. A. (1965). Piezoelectric transducer materials. *Proceedings of the IEEE*, *53*(10), 1372–1386. https://doi.org/10.1109/proc.1965.4253

Lalithavani, K., Arasan, G., Premsundar, S., Venkatasubash, V., & Vijayaraj, R. (2025). AI-Powered based real time earthquake early warning system using deep learning. *Journal of Engineering Research and Reports, 27*(5), 433–442. https://doi.org/10.9734/jerr/2025/v27i51516

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, *11*(1). https://doi.org/10.1038/s41467-020-17591-w

Sheehan, J., Zhai, Q., Chuang, L. Y., Officer, T., Wang, Y., Zhu, L., & Peng, Z. (2025). Applying EQTransformer to laboratory earthquakes: detecting and picking acoustic emissions with machine learning. *Earth Planets and Space*, *77*(1). https://earth-planets-space.springeropen.com/articles/10.1186/s40623-025-02237-2

Tan, M. L., Becker, J. S., Stock, K., Prasanna, R., Brown, A., Kenney, C., … Chen, Y. (2022). Earthquake early warning systems based on low-cost ground motion sensors: A systematic literature review. *Frontiers in Sensors*, 7. https://doi.org/10.3389/fsens.2022.1020202

Wu, R., Selvadurai, P. A., Chen, C., & Moradian, O. (2021). Revisiting Piezoelectric Sensor Calibration Methods Using Elastodynamic Body Waves. *Journal of Nondestructive Evaluation*, *40*(3). https://doi.org/10.1007/s10921-021-00799-1