Progress Report 1: Designing a Low-Cost, Machine Learning-Based Early Earthquake Detection System for Developing Countries

Nikhil Patil

Mahib Rahman

Eliab Melaku

November 20, 2025

Objective: Finalize the list of materials, approach to machine learning model, work on design of prototype/apparatus, and finalize schematics/CAD.

Materials and Methods

In the past weeks we have accomplished the following: We are collaborating with a George Mason Professor and have acquired access to more advanced testing facilities. For testing, we will be using an APS 400, ELECTRO-SEIS® Long-Stroke vibration exciter. This is an electrodynamic force generator, the output of which is directly proportional to the instantaneous value of the current applied. We will create 2 data sets; the first will be derived from the APS400 being set to a constant vibration frequency. Once a data analysis is complete and put into the machine learning model embedded in a Raspberry Pi model 4 or the existing ESP dev bored. The apparatus will then be tested again, this time with APS400 being set to arbitrary vibration frequencies. This will test if it is effective in utilizing machine learning and simulating field testing.

Additional safety materials for soldering:

- Masks to prevent fumes from entering our lungs.
- A fire extinguisher present for emergencies.
- Heat-resistant gloves.
- Safety glasses.
- High quality power source and cable

Frame materials (WIP): Figure 1 depicts a preliminary model.

- Solid Rubber Cylinder 2" Diameter, 2-3/16" High - RS 12. Resonant and vibration resistant base.
- Solid Aluminum Cylinder 2" Diameter, ~12-16" High. Weatherproof material.

- Metal Box (~4"x6" aluminum plates)
- Silicone Adhesive Sealant
- TBD in the following weeks

The materials list excluding these include

- PZT piezo discs (20–27 mm)
- ADXL345 accelerometer module
- OPA2132 op-amp IC ADS1115
- Analog to digital converter
- Resistor/Capacitor kit
- PCB
- ESP32 dev board
- Raspberry Pi 4
- microSD module/card
- WOR/Radio LoRa SX1276 modules
- Power 18650 Li-ion battery (3000 mAh)
- Battery holder
- 3.3V buck converter
- Internal clock RTC DS3231

Here is a rough schematic of the PCB:

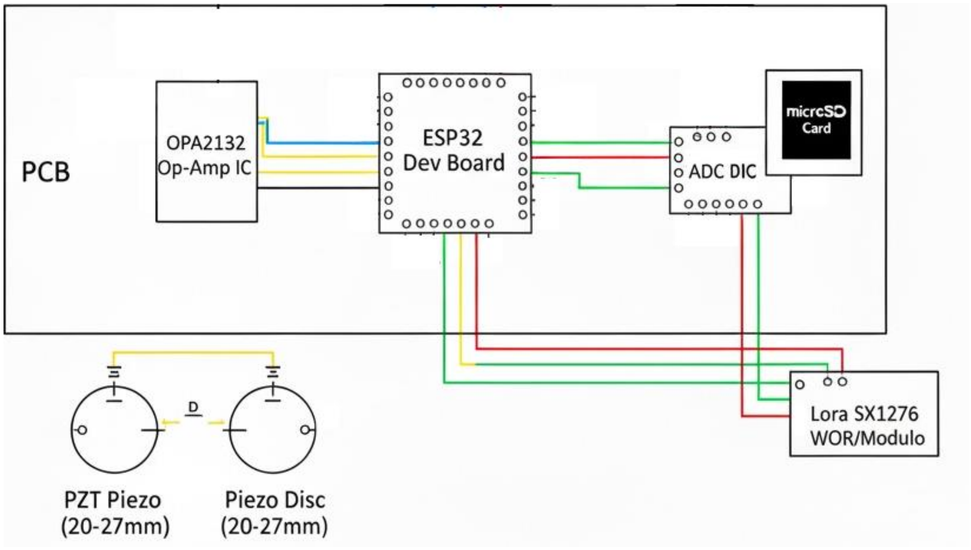


Figure 1: This block shows the necessary components of the PCB: It includes 2 PZT piezoelectric transducers that feed data into the OPA2132 Op-Amp for signal conditioning. The data is then transformed into legible data by the ADC analog converter and sent to the ESP32 to process.

In the next couple of weeks, we plan to finish schematic and evaluate parts such as whether the Raspberry Pi 4 is necessary or whether we can keep the existing ESP32 for computing machine learning. Additionally, we will finalize our materials list including the frame of the project and order, and we will start a CAD model of our testing apparatus. Most importantly, we will create a testing success criteria.

Data Analysis

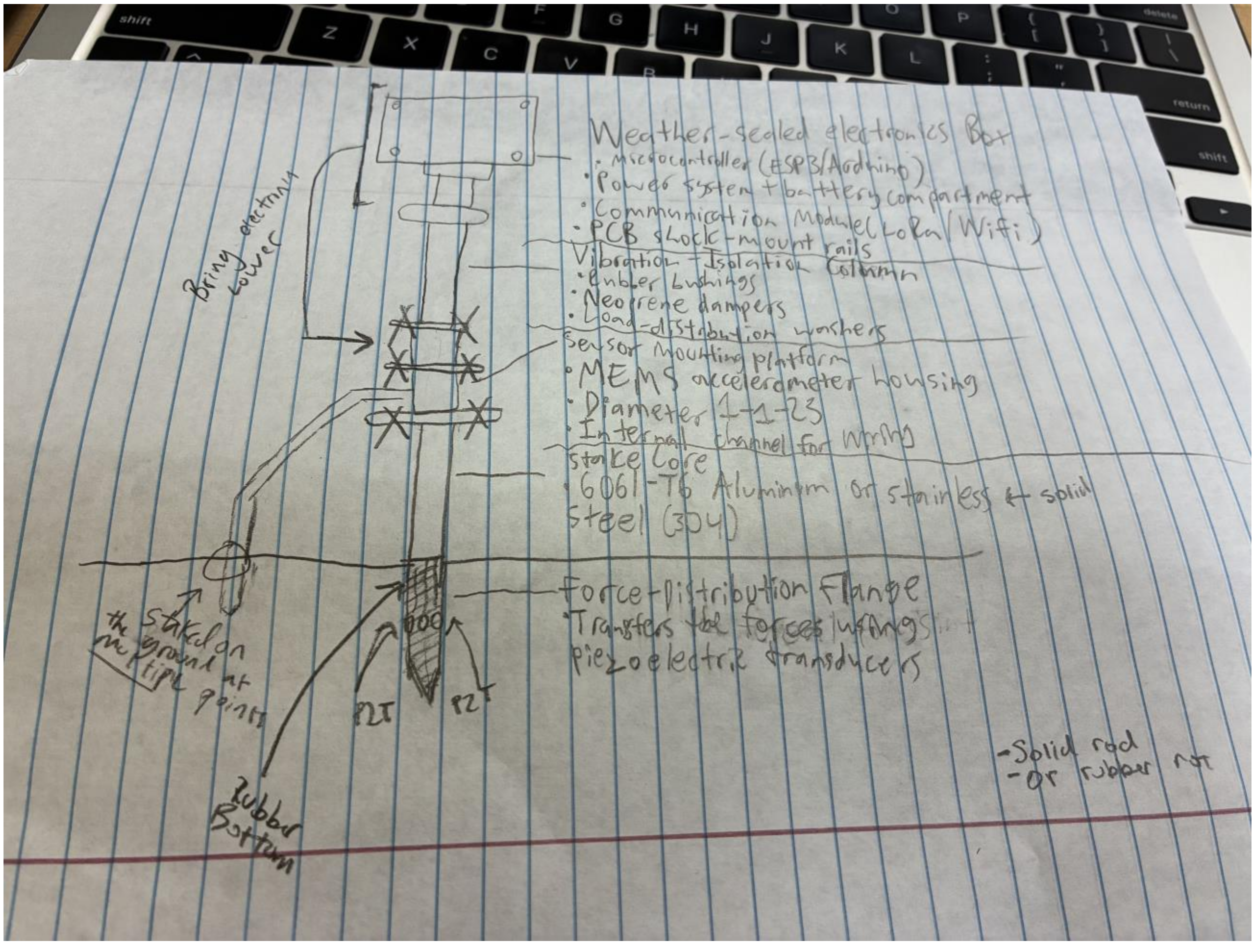

Weather-sealed electronics Box
• Microcontroller (ESP3/Arduino)
• Power system + battery compartment
• Communication Module (LoRa/Wifi)
• PCB shock-mount rails
Vibration-Isolation Column
• Rubber bushings
• Neoprene dampers
• Load-distribution washers
Sensor Mounting platform
• MEMS accelerometer housing
• Diameter 1-1-23
• Internal channel for wiring
Stake Core
• 6061-T6 Aluminum or stainless ← solid
steel (304)
Force-Distribution Flange
• Transfers the forces ~~[illegible]~~ to
piezoelectric transducers
Bring electronics lower
Staked on the ground at multiple points
PZT
PZT
Rubber Bottom
- Solid rod
- or rubber rod

Figure 2: Rough draft of prototype design illustration

This is a rough sketch of our design. The diagram shows each part of the design. Up to this point, we learned that the tip should be made of rubber in order to mitigate the resonance. We also learned that it would be optimized if we added rocks to the bottom of the design for stability. We will be purchasing most of these materials in the coming weeks. We will work in any way we can to optimize the design for the greatest results. We have organized meetings with experts from universities such as Stanford, Johns Hopkins, and the University of Texas in the coming weeks to consult. We will ask questions about our project and hopefully gain further insight into the matter we are dealing with.

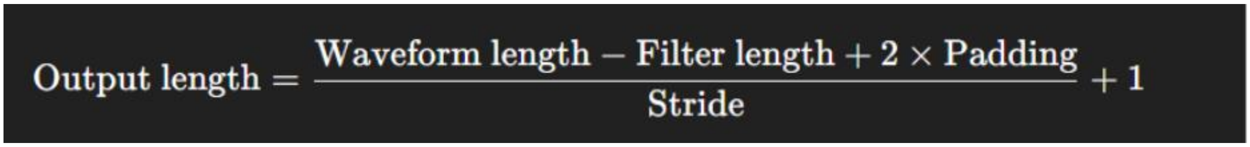
In the next few paragraphs, we discuss our progress made on our approach to analyzing earthquake waveform data. The main objective for the ML model is to automatically identify seismic events from waveform data as fast and accurately as possible. It will then provide us with a final decision on whether the characteristics of the data match those of an earthquake or other background vibrations. The output classification will simply be “earthquake detected” or “non-seismic event”.

To explain how the ML model analyzes the input data, a fixed-length array of vibration samples, it first receives the waveform in numerical data. The model then breaks down the waveform into different segments using convolution filters which is a tiny pattern detector that slides across the vibration data, the filters will then detect sudden changes, spikes, repeating oscillations, p-wave/s-wave patterns, etc.

To gain a better understanding of how this works in practice, we can consider the following example: if the signal were 1024 samples long, the filter would identify 16 samples at that instance. It will then multiply the values in the segment by the set of weights, sum them, and pass the result to the activation function. This is how the filter will detect any useful patterns that were discussed earlier.

More specifically about filters, each one will have its own set of weights which the model learns as it trains by receiving many samples of waveforms. Many filters will be used (8-16 for the first layer), so a variety of features can be detected at the same time. To elaborate, the weight is a number the model will use to decide how impactful any part of the input signal has on the final prediction. This model will include stacked layers where the early layers are for detecting simple features, and later layers will have the job of detecting more complex patterns.

The next step in the process is the creation of the feature map. As stated earlier, the filter will send the output through the activation function. Each of the outputs becomes one value on the feature map. Going more in depth, if the waveform were to have 1024 samples and the filter length is 16, the feature map would have (1024-16+1) = 1009 values. This equation comes from the basic formula for the length of a feature map after a 1D convolution occurs which is:



$$\text{Output length} = \frac{\text{Waveform length} - \text{Filter length} + 2 \times \text{Padding}}{\text{Stride}} + 1$$

Figure 3: Formula for length of feature map (in this case, padding = 0 and stride = 1)

Each value within the map will display how strongly the filter detected its designated pattern at that exact point in the waveform. Feature maps will be crucial to our machine learning model because they will turn the raw numerical waveform data into a structured representation that only highlights the detected patterns.

In the next few weeks, we will work to fully understand the remaining components of the ML model. These involve how the filters obtain their weights, how the model learns which patterns are important, how it reduces data size while keeping the data it cares about, classification, and measurement of metrics (accuracy, precision, recall, and false alarms). We will finish evaluating parts sfor computing machine learning. Next, we will build a machine learning model and will embed it on Raspberry Pi 4 or ESP32.

References

Dauren Zhexebay, et al. “Deep Learning for Early Earthquake Detection: Application of Convolutional Neural Networks for P-Wave Detection.” *Applied Sciences*, vol. 15, no. 7, 1 Apr. 2025, pp. 3864–3864, https://doi.org/10.3390/app15073864.

Kubo, Hisahiko, et al. “Recent Advances in Earthquake Seismology Using Machine Learning.” *Earth, Planets and Space*, vol. 76, no. 1, 28 Feb. 2024, https://doi.org/10.1186/s40623-024-01982-0.