

ML Lab Research Proposal - Earthquake Predictor

Team B - Arya Addagarla & Anubhav Chatterjee (AI for Scientific & Medical Applications)

Date: 4-24-26

1. Problem Statement

What are you solving and why does it matter? (2–4 sentences)

We are addressing the unpredictability of earthquakes and how incorporating sensor data and analyzing it with neural network-based predictive models can yield high-reliability predictions. This matters because living in California, a place with relatively high seismic activity, keeps us civilians, at greater risk of the next deadly earthquake.

2. Proposed Approach

What model/method are you using? Why this approach over alternatives? (3–6 sentences)

We are using a Multimodal Graph Neural Network. This treats every seismic station, GPS point, and fiber-optic cable as a connected node in a massive web, similar to how the brain connects different senses. We chose this over traditional methods because it doesn't just see a line from data; instead, it tracks how that vibration moves across a map. By requiring the signal to travel at the actual speed of a tectonic wave, we can instantly tell the difference between a real earthquake and a truck driving past a single sensor.

3. Dataset

What data are you using? Where is it from? How much of it? Any preprocessing needed?

We are using: raw vibrations from the Stanford Earthquake Dataset, satellite movement logs from GPS stations, and high-speed fiber-optic data. We will start by using the R programming language to explore and clean this information, making sure every data point is perfectly synchronized. This step is like tuning all the instruments in an orchestra so they are playing on the same beat, which allows us to find the hidden patterns that connect them.

4. Evaluation Metrics

We will know the project is a success if the AI reaches a predictive accuracy of over 90 percent. We are using a metric called an F1-Score to make sure the system is both sensitive enough to catch tiny quakes and smart enough to avoid "crying wolf" with false alarms. We will also measure how close our predicted epicenter is to the real location and, most importantly, how much extra warning time we can give people before the shaking actually starts.

5. Compute Requirements

Estimated training time, GPU memory needs, and any special dependencies.

To handle this massive amount of data, we need a high-end NVIDIA GPU with at least 24GB of memory. This specialized hardware allows the AI to "think" about multiple data streams at the same time rather than one by one. We expect the full training process to take about 48 hours using a combination of R for the initial math and PyTorch for the heavy AI lifting.

6. Expected Deliverables

What will you have at the end of the two weeks? (model, paper, demo, etc.)

At the end of the two weeks, we shall have produced the model. We plan on performing data analysis on the sensor data using R and then performing an initial analysis. Then we will train an AI model to make predictions on this data.

7. Risk & Mitigation

What's most likely to go wrong, and what's your backup plan?

By trying to find correlations in the patterns of the data to predict earthquakes, many times it's possible to find illusory correlations, so we will run the AI to find real predictors and then compare them to real data to check if the correlation is real. Additionally, regular tremors cannot be an indicator (vibrations from regular sources such as traffic), and our data needs to reflect that, and