A DAS-Based Approach to UNE Monitoring

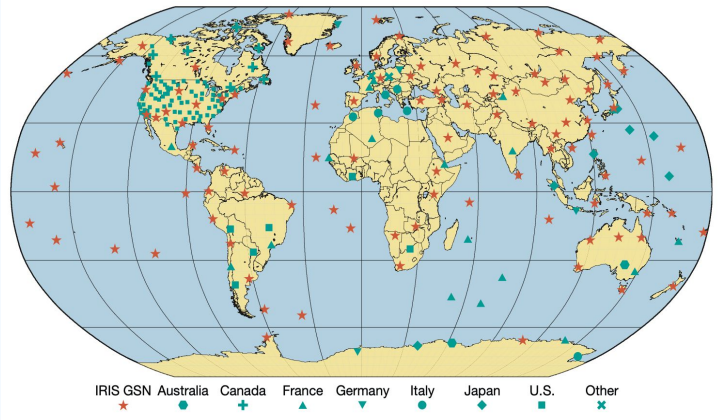


- Sofian Belghiti, Osman Farooq, Soweata Emma Himi, Susamy Liberato, Ahnaf Maahi, Mahmoud Zaghloul

ME 47300 | Professor Richard La Grotta

Overview

PROBLEM	Nuclear test monitoring relies on land-based seismometers → a large portion of Earth's surface remains unmonitored.
CHALLENGE	DAS turns existing fiber cables into thousands of seismic sensors, but no nuclear explosion has ever been recorded in DAS format.
OUR APPROACH	Convert archived nuclear test recordings into DAS-compatible signals → train ML classifier to distinguish UNEs from earthquakes and noise.
SEMESTER 1	Data processing and detection system built and tested on two real DAS earthquake datasets in two different file formats.



Global Seismographic Network (GSN) station locations. Source: USGS, 2024

01

Background and Approach

Semester Overview

Technical Challenges

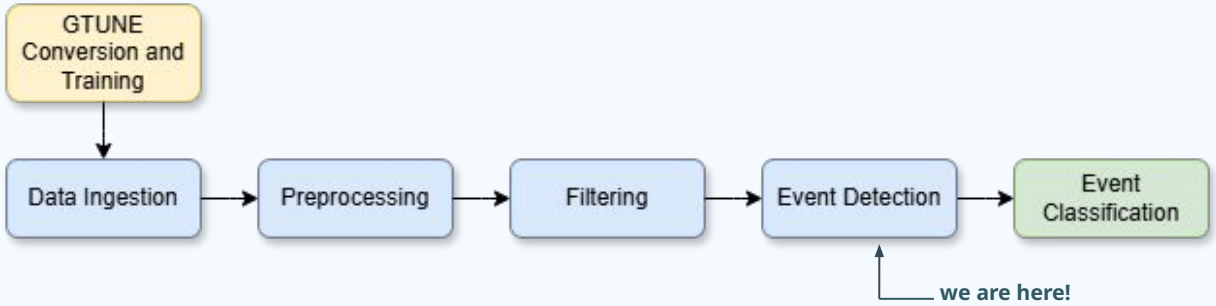
- DAS is ultra-sensitive: noisy environment
- No native DAS-format UNE data exists
- DAS responds differently to wave direction than seismometers
- Cable-to-ground coupling varies channel to channel

Requirements

- Detect candidate seismic events in raw DAS data
- De-noise via bandpass filtering tuned to seismic frequencies
- Classify each event: UNE, earthquake, or noise
- Estimate back-azimuth, slowness, and origin time

Interfaces

- Inputs: HDF5, SGY, SAC, CSS waveform files
- Processing: Python in VS Code + Jupyter (primary), Google Colab Pro (backup)
- Data: GTUNE, WFNE, Pawnee, Fairbanks
- Outputs: classification label, source estimate, plots



Datasets We Worked With

	Pawnee M5.8	Cook Inlet M5.0
Use in Project	Initial testing + validation	Large-scale DAS example
Location	Oklahoma, USA	Alaska, USA
Year	2016	2023
Format	SGY	HDF5
Channels	626	8,531
Cable Length	5 km	81.65 km
Sampling rate	50 Hz	25 Hz
Processing tool	ObsPy	DASCore / DASPy

Pawnee was our smaller test case, so it was easier to use for early testing.

Cook Inlet was our larger dataset, so it helped show that the system can handle large-scale real-world DAS data.

Together, they confirm that our approach works with different file formats, channel counts, and cable lengths.

Conclusion:

We used real DAS earthquake/noise data to validate the pipeline, and GTUNE/WFNE converted data to train the UNE classifier.

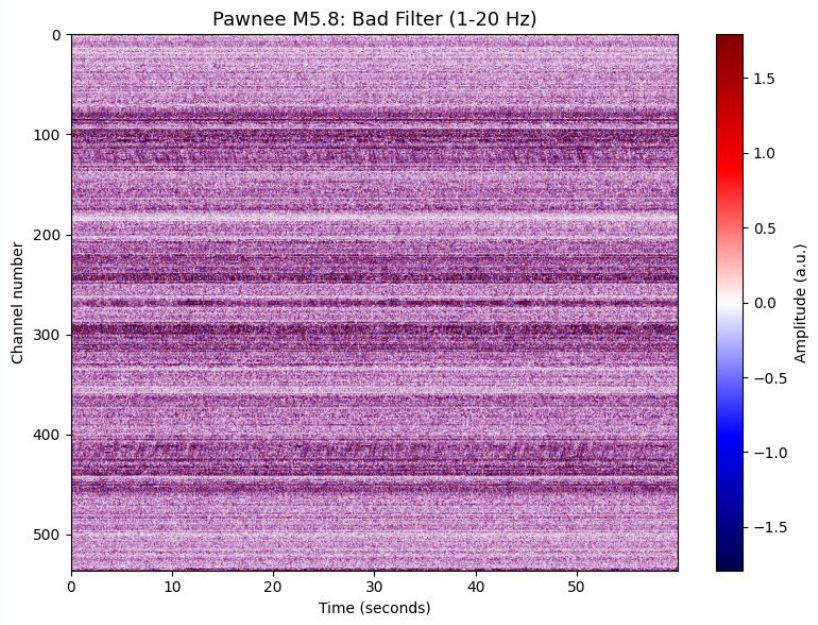
02

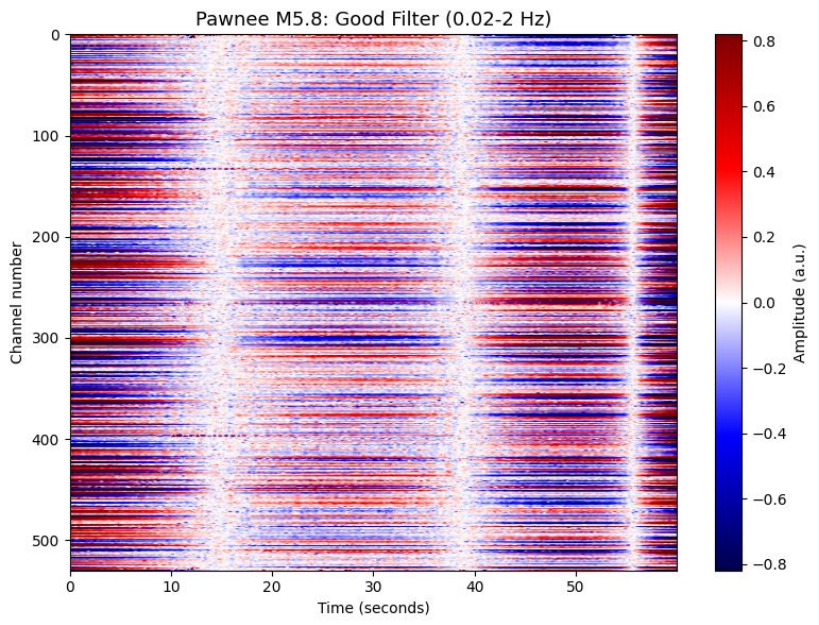
Summary of Solution

Making the Data Usable

Raw Data (SGY/HDF5) → **Detrending** — **Differentiating** — **Denoising** — **Filtering** — **Faulty Sensor Removal** → **CLEAN DATA!**

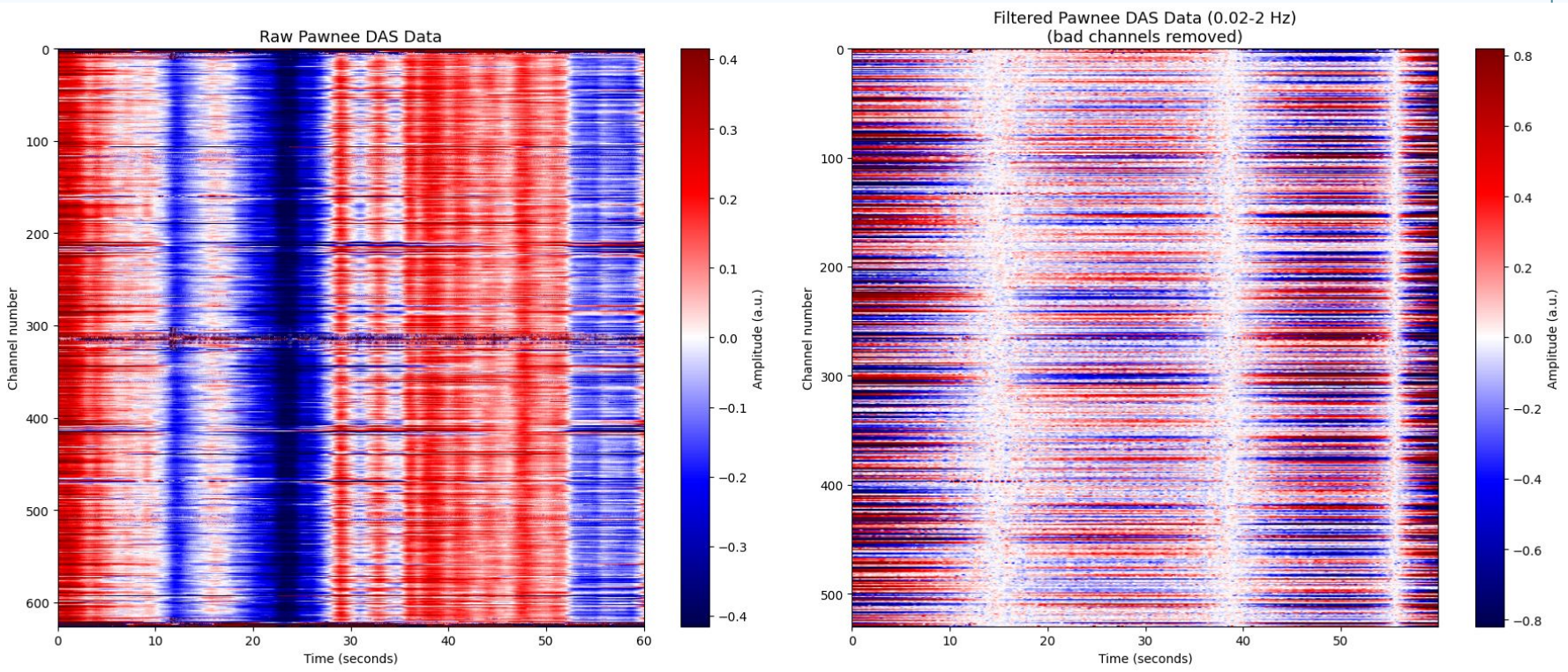
Key finding: filter frequency has to match the wave type in the data; otherwise, the signal is buried.





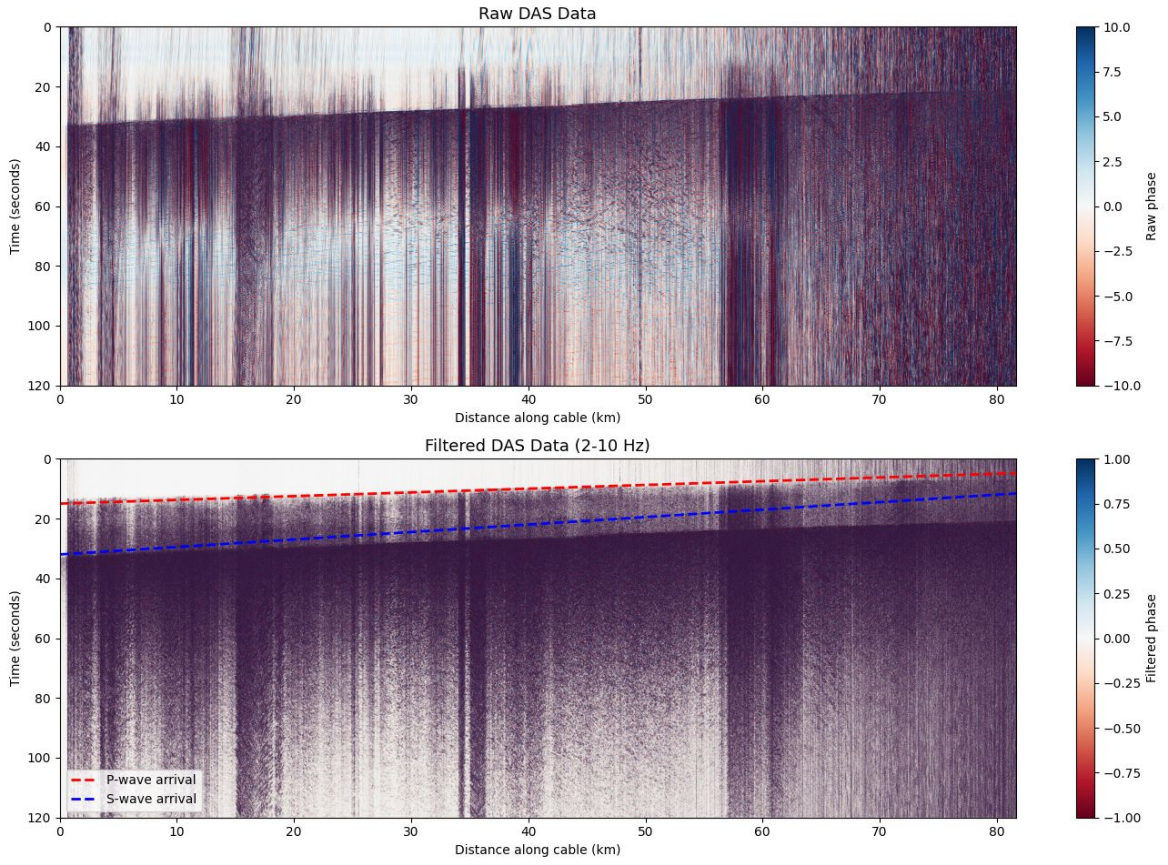
Pawnee M5.8 (Before and After Processing)

2016 M5.8 earthquake in Oklahoma | 626 DAS channels |50 Hz Sampling Rate | SGY format

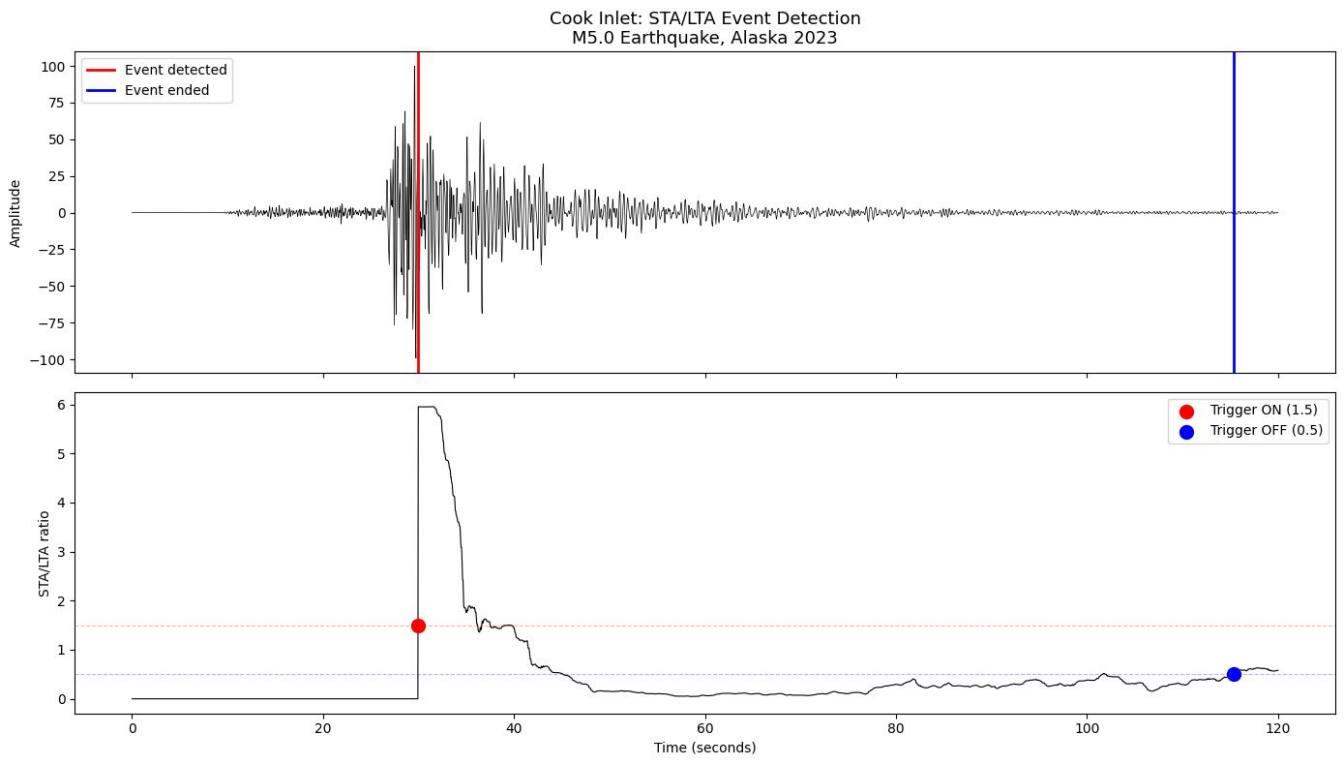


Cook Inlet M5.0

2023 M5.0 earthquake in Alaska | 8,531 channels across 81.65 km of seafloor cable | HDF5 format



Automatic Event Detection (STA/LTA)



$$STA/LTA = \frac{short-term\ average\ energy}{long-term\ average\ energy}$$

→ If this ratio spikes → seismic event detected

As-Designed vs. As-Built System Comparison

Category	As-Designed System	As-Built Testing System
Data Source	Live DAS streams (HDF5/TDMS)	Pre-recorded .sgy files + GTUNE synthetics
Purpose	Real-time UNE monitoring	Offline batch
Processing	Continuous operation	Batch processing in Python/Google Colab
Output	Event label, location, origin time, alert	Classification result, plots, and test metrics
Main Goal	Operational monitoring system	Verify the method before deployment

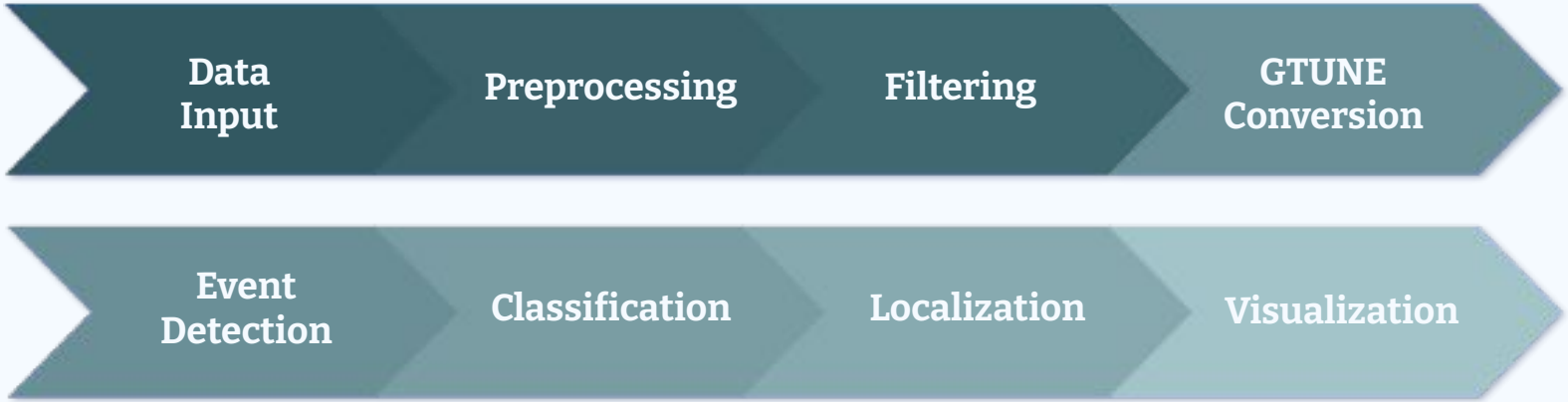
Estimated Project Cost Breakdown

Item	Purpose	Estimated Cost
Google Colab Pro	Cloud notebook environment for processing and testing (through Dec 26')	$80
Google Drive Storage	Cloud Storage for seismic and DAS files (200 GB-2 TB through Dec 26')	$24-$80
External Storage Drives	Backup and transfer of large datasets	$100-$200
Python Libraries	Data processing, filtering, detection, and classification (Open source)	$0
DAS Hardware	Not needed for this prototype	$0
TOTAL		**$204-$360**

03

Plans for Manufacturing

Design Plan



Modular Approach

An iterative, modular method is best suited to handle the various moving parts before consolidating them

Environment

GitHub allows for simple collaboration and version control. Cloud services such as Colab and Jupyter provide virtualization and ease of testing.

Development Cost

With open source tools and datasets, the cost of development itself will be little to nothing. Most expenses will be from hardware and subscriptions.

Timeline & Flowchart

- **(Weeks 1-3)** GTUNE Conversion
- **(Weeks 4-5)** Filtering, Tuning
- **(Weeks 6-8)** Classification, ML Training
- **(Weeks 9-10)** Localization, Validation
- **(Weeks 11-12)** Integration, Final Report

Stage	Description	Tools
1. Data Input	Import DAS files from datasets	h5py, npTDMS, ObsPy
2. Preprocessing	Normalize data signals/types	NumPy, ObsPy
3. Filtering	Filter noise, isolate frequencies	scipy.signal, ObsPy
4. GTUNE Conversion	Convert into standard seismic format	NumPy, SciPy
5. Event Detection	Identify/mark points of interest	MatchFilter, ObsPy
6. Classification	Determine type of event with trained model	scikit-learn, TensorFlow
7. Localization	Find location/timing of event	ObsPy
8. Visualization	Generate plots and summary	Matplotlib, Seaborn

04

Plans for Testing

Testing Strategy

Test 1: Filter Validation
A bandpass filter applied to Pawnee databases confirms noise removal while preserving P-wave and S-wave arrivals.

Test 2: GTUNE Conversion Accuracy
It converts GTUNE velocity waveforms to synthetic DAS strain rate and verifies amplitude range and wavefront against real DAS recordings.

Test 3: Classifier on Earthquake Data
It runs on real Pawnee DAS recordings, where "pass" means earthquakes is correctly labeled, with no false UNE detection and >90% accuracy.

Test 4: Classifier on Synthetic UNE Data
It runs on held, converted GTUNE waveforms; "pass" means UNE events correctly identified, > 90% accuracy and false alarm rate within defined limit

Test 5: End-to-End Validation
The raw input processed through full system, where "pass" means correct event label, confidence score, and estimated source parameters for detected explosion

Comparison Framework: verified against literature, known dataset tables, and performance target.

Estimated time: 3 to 4 weeks in semester 2

Estimated cost: no additional costs beyond existing environment

Semester 2 Plan

Milestone 1

GTUNE velocity to synthetic DAS strain-rate conversion algorithm complete and validated

Milestone 2

ML classifier trained on synthetic UNE + real earthquake + noise data

Milestone 3

Validation > 90% classification accuracy on held-out test set

Milestone 4

Source localization module + full end-to-end system integration

By the end of Semester 2:

a system that ingests raw DAS data and outputs UNE / earthquake / noise classification with estimated source parameters

Where We Stand

Semester 1 Accomplishments:

1. **Good research foundation**
   The data gap was identified, the approach was validated against published work and all key datasets were sourced

2. **Two real DAS datasets acquired and processed across two file formats**
   The Pawnee, Oklahoma (SGY), and Fairbanks, Alaska (TDMS), datasets were successfully acquired

3. **Basic preprocessing and filtering built and validated**
   Noisy channels were removed, and the bandpass filter was confirmed on real earthquakes.

4. **Basic Automatic event detection demonstrated**
   Seismic events were successfully detected using the STA/LTA algorithm. So no manual input was required.

References

[1] CTBTO, "The Comprehensive Nuclear-Test-Ban Treaty." Available: https://www.ctbto.org/our-mission/the-treaty.

[2] N. J. Lindsey et al., "Fiber-Optic Network Observations of Earthquake Wavefields," *Geophysical Research Letters*, 2017. doi: 10.1002/2017GL075722.

[3] L. Barama et al., "Global Nuclear Explosion Discrimination Using a Convolutional Neural Network," *Geophysical Research Letters*, 2023. doi: 10.1029/2022GL101528.

[4] M. Mazur et al., "Global Seismic Monitoring using Operational Subsea Cable," arXiv:2409.19827, 2024.

[5] R. Mellors et al., "Modeling Subsurface Explosions Recorded on a Distributed Fiber Optic Sensor," *JGR: Solid Earth*, 2021. doi: 10.1029/2021JB022690.

[6] L. Viens and B. Delbridge, "Rapid and Temporary Deployment of Fiber-Optic Arrays to Measure Surface Chemical Explosion Series with DAS," *SRL*, 2025. doi: 10.1785/0220250039.

THANK YOU!

Any questions?