

The City College of New York

The City College of New York
Department of Mechanical Engineering

Distributed Acoustic Sensing System For Underground Nuclear Explosion Detection

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

ME 47300 - Senior Design I
Professor Richard La Grotta
May 19th, 2026

Executive Summary **3**

1. Description of Problem **3**

1.1 Problem Background 3

1.2 Technical Challenges 5

1.3 Requirements 6

1.4 Interfaces 6

2. Solution Summary **7**

2.1 Solution Overview 7

2.2 System Design 8

2.3 Trade Study and Conclusions 9

2.4 Detailed Design Analysis 11

2.5 Tool Selection 12

2.6 As-Designed vs. As-Built 13

2.7 Cost Analysis 14

3. Plans for Manufacturing **15**

3.1 Development Approach 15

3.2 Implementation Stages and Techniques 15

3.3 Estimated Timeline and Cost 16

4. Plans for Testing **16**

4.1 Testing Strategy Overview 16

4.2 Tests to be Run 17

4.3 Comparison to Analysis 17

4.4 Estimated Costs and Time 18

References **19**

Appendix A **21**

Executive Summary

Nuclear weapons testing remains a significant threat to global security, and the ability to monitor and verify compliance with international test-ban treaties is crucial. The 1996 Comprehensive Nuclear-Test-Ban Treaty (CTBT) bans all nuclear explosion testing, yet non-compliant countries continue to conduct underground tests. Current monitoring relies on the CTBTO's International Monitoring System (IMS), which depends on land-based seismometer networks that leave approximately 71% of the Earth's surface without coverage.

Distributed Acoustic Sensing (DAS) offers a cost-effective way to address this issue by repurposing existing fiber cables. A single DAS interrogator unit can generate thousands of sensing points along an existing cable without new installations. However, applying DAS to UNE monitoring introduces a data challenge, since no underground nuclear explosion has ever been recorded in DAS format. All historical nuclear test recordings exist as seismometer velocity data, not as the strain-rate measurements produced by DAS systems. This project addresses that by developing an algorithm to convert existing seismometer-format nuclear explosion recordings into synthetic DAS strain-rate signals, which will serve as training data for a machine learning classifier able to distinguish UNEs from earthquakes and background noise.

This report covers the work completed during Semester 1 of the project. The research foundation has been established through reviewing relevant literature spanning seismic wave physics, nuclear monitoring systems, DAS signal processing, and machine learning-based event classification. A basic data processing system has been built and validated on two real DAS earthquake datasets: the 2016 M5.8 Pawnee, Oklahoma earthquake (SGY format), and the 2023 M5.0 Cook Inlet, Alaska earthquake (HDF5 format), demonstrating that the system correctly handles different file formats, removes faulty sensors, isolates seismic frequencies through filtering and automatically detects seismic events using the STA/LTA algorithm. Semester 2 will focus on implementing the GTUNE conversion algorithm, training and validating the ML classifier and integrating source localization into the completed system.

1. Description of Problem

1.1 Problem Background

Preserving international security against the threat of nuclear weapons relies almost entirely on the ability to monitor and verify compliance with test-ban treaties. The 1996 Comprehensive Nuclear-Test-Ban Treaty (CTBT) has been signed by 187 countries and prohibits all nuclear explosive testing with the goal of reducing nuclear weapon proliferation [1]. Despite this, the treaty has not been universally ratified, leaving global security vulnerable to the underground tests still conducted by non-complying nations. For instance, North Korea has conducted six confirmed underground nuclear explosions between 2006 and 2017, with the final test in September 2017 registering a seismic magnitude of 6.3 and a yield of 120 kilotons [2]. Seismic monitoring has proven to be the fastest and most reliable way of detecting and identifying Underground Nuclear Explosions (UNEs) from natural seismic events, forming the foundation of the CTBTO’s International Monitoring System (IMS) [3].

Despite seismic monitoring being effective, the current IMS network has significant coverage limitations. The IMS relies on land-based seismometer stations that cannot monitor the deep ocean, which accounts for approximately 71% of the Earth's surface [4]. Seismic waves from underground tests at remote sites have to travel thousands of kilometers to reach the nearest station on land, resulting in weaker signals from low-yield events that fall below the detectable range by the time they are picked up. This creates a demand for a cost-effective and scalable solution that places sensors closer to where those waves travel.

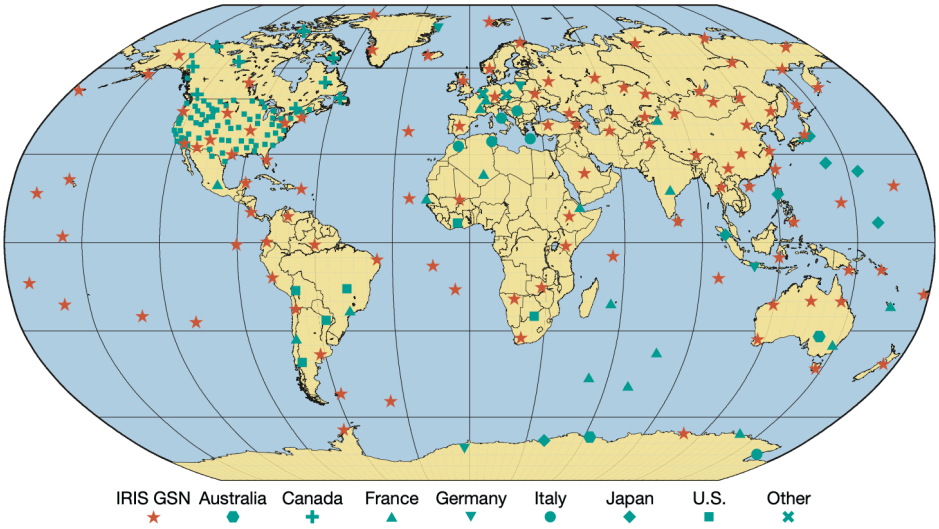


***Figure 1**: Global Seismographic Network (GSN) station locations [19].*

Distributed Acoustic Sensing (DAS) technology offers a way to address this issue. DAS converts existing fiber-optic telecommunication cables into a string of seismic sensors, which allow a single interrogator unit to capture measurements over tens of kilometers [5]. While Lindsey et al. (2017) proved that DAS sensitivity matches seismometers in the 0.8-1.6 Hz range, the research by Mazur et al. (2024) showed its spatial potential by detecting P-, S- and surface earthquake waves using subsea cables between Iceland and Ireland [5][4].

However, integrating DAS technology into global UNE monitoring introduces some data constraints. Although datasets like GTUNE [6] and WFNE [7] offer historical nuclear test recordings spanning from 1945 to 2017, these cannot be directly applied to a DAS system. The data contains seismometer measurements of velocity and acceleration, rather than the strain-rate measurements generated and required by DAS systems. Since no underground nuclear explosions have ever been recorded using DAS directly, this project aims to develop an algorithm that converts existing seismometer nuclear explosion recordings into simulated DAS strain-rate signals, forming a foundation for training a machine learning classifier capable of separating UNE signals from other seismic noise.

1.2 Technical Challenges

Integrating DAS technology into a global monitoring system for underground nuclear explosions introduces some technical challenges. These challenges stem from the design of fiber-optic arrays, the fundamental physics of the sensing technology, and the specific format of historical training data currently available.

The primary obstacle involves the high noise floor within typical sensing environments. In a standard deployment, a DAS interrogator unit generates thousands of high-density, co-located channels by measuring the strain-rate at each gauge interval along the fiber length [5]. While this spatial density is the technology's greatest advantage, it also results in the recording of all local ground motion sources, including human-caused vibration, traffic, industrial activity, wind, and ocean microseisms. All of this gets recorded alongside relevant seismic signals, which means heavy filtering is needed to isolate target events before automated classification can be attempted.

The lack of nuclear explosion data in DAS format is the second problem. Inertial seismometers, which measure ground velocity at a single spot, recorded every known declassified underground nuclear test [7]. Axial strain along a cable is measured by DAS and is roughly proportional to the spatial derivative of particle velocity along the fiber direction [5]. A classifier trained on seismometer data simply won't work on DAS strain-rate data. In order to close this gap, historical velocity data must be transformed into artificial strain-rate signals. This procedure relies on assumed apparent wave velocity values. Apparent velocity fluctuates along the cable as the fiber passes through various subsurface conditions and varies with the seismic phase under consideration (P-waves travel faster than S-waves, which travel faster than surface waves) [9]. Instead of committing to a single conversion factor, a convincing synthetic dataset must sample throughout a reasonable range of these velocities.

The directional sensitivity of fiber laid horizontally presents the third difficulty. DAS is most sensitive to motion along the cable directions ans picks up very little from motion perpendicular to it [5]. This means signal amplitude drops significantly on a single fiber segment for almost vertically incident P-waves, which contain a large portion of the discriminative information required in earthquake-explosion classification. Surface waves and S-waves couple into horizontal fibers more effectively because they have larger horizontal motion components. This must be taken into consideration by any classification system based on DAS data, either by training on synthetic data that covers the relevant range of incidence angles or by using array geometry.

The fourth challenge is how inconsistently the fiber couples to the ground. DAS relies on whatever contact exists between the fiber and the surrounding ground or conduit, in contrast to a seismometer, which is connected to a predetermined mounting medium with a defined response. Although it has a lower signal-to-noise ratio than a directly buried cable, Lindsey et al. showed that fiber placed loosely inside a conduit may nonetheless detect teleseismic phases [5]. Point-sensor data does not exhibit the consequent channel-to-channel variability in signal

amplitude and frequency response, which must be taken into account during preprocessing instead of being ignored.

1.3 Requirements

Upon detecting a seismic event, the system must generate a classification label and source-parameter estimate derived from the raw data. The fundamental performance of this monitoring framework is established through four core functional requirements. To identify candidate events within continuous DAS streams, the system must isolate arrivals that reflect actual ground motion from ambient background noise. This detection phase produces an arrival-time flag, utilizing filtered strain-rate waveforms to facilitate downstream classification.

The system must implement bandpass filtering calibrated to target seismic frequencies to denoise raw input. This passband is optimized through validation against established earthquake signals within the Pawnee dataset [5]. While suppressing out-of-band energy, the filtering process must preserve the integrity of P-wave and S-wave arrivals as primary discriminative features.

Upon the identification of a seismic arrival, the classification framework must categorize the event as either background noise, a natural earthquake, or an underground nuclear explosion. The system is required to achieve a minimum classification accuracy of 90% on a held-out validation dataset while maintaining a constrained false-alarm rate. This performance metric is intentionally conservative when compared to established machine-learning benchmarks for explosion discrimination; specifically, Barama et al. (2023) demonstrated class-specific accuracies of 98.4% for UNEs, 95.6% for earthquakes, and 99.6% for noise within the GTUNE repository [8]. The 90% threshold accounts for the increased technical complexity of training on synthetic DAS strain-rate waveforms rather than native seismometer recordings. Because rare-event identification necessitates robust performance beyond raw accuracy, the false-alarm rate is established as a discrete pass-fail criterion within the testing methodology detailed in Section 4.

In the case of a potential explosion, the system must estimate the wave direction, apparent speed and origin time for each candidate event using array processing across DAS channels [5]. Full 3D location coordinates are out of scope for a single linear cable; this technical limitation is further detailed in Section 1.4.

1.4 Interfaces

The system is designed to ingest raw seismic data across multiple formats associated with both inertial seismometers and DAS interrogator units. Specifically, the system processes SGY files from the Pawnee earthquake repository [5], historical SAC and CSS seismograms sourced from the WFNE archive [7], and high-density HDF5 or TDMS recordings typical of modern fiber-optic sensing deployments.

Computational analysis is conducted mainly within a local Python environment using VS Code and Jupyter notebooks, with the specialized Python libraries detailed in Section 2.2. Data management and DAS-specific signal operations are facilitated through DASCore [14] and

DASPy, while ObsPy [15] provides the necessary infrastructure for traditional seismic processing. Numerical computing and array-based processing tasks are further optimized using the NumPy and SciPy ecosystems.

The monitoring framework leverages several external data repositories to facilitate training and validation. These include the GTUNE dataset, which provides 28,123 vertical-component velocity seismograms from 774 declassified underground nuclear tests [6], and the WFNE repository containing 82,000 waveforms from 802 historical nuclear explosions [7]. Real-world DAS performance is validated using recordings of the 2016 magnitude 5.8 Pawnee earthquake sequence [5] and the Fairbanks permafrost DAS dataset for ambient noise characterization [12]. Integration of supplemental datasets may occur as directed by the project adviser.

For every detected seismic event, the system should output a classification label, confidence score, estimated wave direction, apparent speed and origin time. Visualization plots of raw and filtered waveforms will also be generated for detailed analysis.

2. Solution Summary

2.1 Solution Overview

The main challenge of applying machine learning to UNE detection using DAS comes from the lack of training data in the needed format. While Barama et al. (2023) showed that ML-based classification of UNEs from seismometer data can achieve more than 95% accuracy [8], that approach cannot be directly applied to DAS systems due to different signal characteristics. This project expands the approach done by Barama et al. to DAS strain-rate data, addressing the format issue that prevents the application of existing classifiers to fiber-optic monitoring systems.

This project aims to use the GTUNE dataset, which contains over 28,000 vertical seismograms from 774 confirmed underground nuclear tests that were conducted between 1961 and 2017, and additional earthquake and noise recordings [6]. Since these events are in seismometer velocity format, a conversion algorithm needs to be developed to transform them into DAS strain-rate format. This conversion will be done through a range of assumed velocities and angles of incidence, since a seismic wave's speed and direction when being picked up by a DAS cable vary depending on geology and cable orientation [9]. Instead of generating a single waveform for each recording, the conversion will be automated through predefined velocity and angle values, which will turn every GTUNE recording into multiple DAS signals and provide a more diverse dataset that best represents the range of signals a real DAS system might experience. This type of approach has been supported by Mellors et al. (2021), who showed that DAS recordings from chemical explosions can be modeled efficiently from known parameters, and Viens and Delbridge (2025), who successfully recorded surface explosion signals in DAS format [10][11].

The obtained UNE data from this method, combined with real DAS earthquake recordings from the 2016 Pawnee earthquake [5] and ambient noise from the Fairbanks

permafrost study [12] done by Cheng et al. (2022), will form a training dataset able to cover the three target events: UNE, earthquakes, and background noise. A machine learning classifier will then be trained on features such as P-wave dominance, onset sharpness, S/P amplitude ratio, and spectral shape, rather than raw data. When applied to incoming DAS data, the model aims to label detected seismic events with over 90% accuracy [13].

2.2 System Design

Following the training phase, the detection and classification algorithm will process the incoming data through five stages: data ingestion, preprocessing, filtering, event detection, and event classification. Each stage is designed to be developed, tested, and validated independently before being added to the full system. The simplified system architecture is illustrated in Figure 2 below.

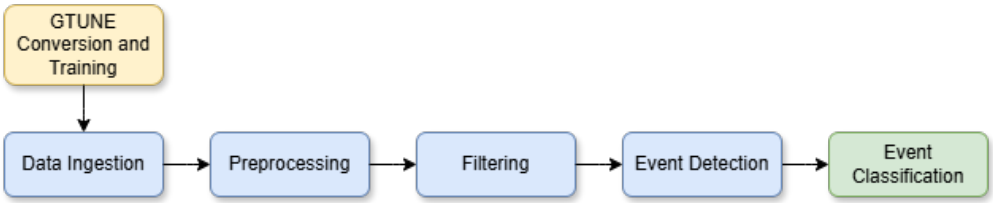


***Figure 2**: Five-stage DAS detection and classification system.*

One of the main challenges is that the data it needs to handle comes in multiple file formats depending on the source. For example, nuclear explosions from GTUNE are stored in SAC and CSS formats, which are older forms of seismological recordings done at single points. Real DAS earthquake recordings, however, come in more modern formats, like SGY for the Pawnee dataset [5], TDMS for the Fairbanks dataset [12] and HDF5 for more recent DAS applications. The data ingestion stage handles this by using two libraries: ObsPy [15], which supports SAC, CSS and SGY, and DASCore [14], which can handle HDF5 and TDMS DAS files. This ensures the system can ingest data from both nuclear test archive files and modern DAS systems.

Once loaded, the data will go through a preprocessing stage where noisy channels are removed by comparing each channel's signal variance against the average across all channels. The ones that fall outside this range will be flagged and excluded. A baseline for the signal will be corrected through detrending. Since DAS is extremely sensitive and picks up everything, cleaning the signal is essential before any analysis can be done [5]. The preprocessing and filtering results on the Pawnee dataset are shown in Figure A1 (Appendix A).

A bandpass filter will then be applied in the 1-20 Hz range to isolate any seismic frequencies relevant to local and teleseismic events. This is consistent with the methodology used by Lindsey et al. in their 2017 paper [5] and previous UNE classification research [8]. After filtering, a short-term average to long-term average (STA/LTA) algorithm [15] will scan the cleaned signal for any amplitude spikes that are above background level, and any event that stands out will be flagged as a possible seismic event. To confirm that a flagged event is a real

seismic signal, semblance analysis will be applied to multiple DAS channels to check whether the same signal shows up consistently along the cable, which is expected from an actual passing wave. STA/LTA detection results on the Cook Inlet dataset are shown in Figure A4 (Appendix A).

Once an event is detected, features will be extracted and fed into the trained classifier. These features include P-wave dominance, onset sharpness, S/P amplitude ratio, and spectral shape, which all demonstrate the differences between UNE and earthquake sources. Underground nuclear explosions produce a more outward force that results in strong compressional P-waves and a weak S-wave, while earthquakes produce strong S-waves [16]. The classifier will output one of three labels: UNE, earthquake, or background noise, and in the case of an actual UNE detection, the system will pinpoint the origin and time of the event by analyzing when waves arrive across DAS channels.

2.3 Trade Study and Conclusions

Several possible design approaches were considered for the detection and classification portion of the DAS-based underground nuclear explosion monitoring system. The main goal of the trade study was to select a solution that could detect a seismic event, reduce false alarms, classify the event type, and still work within the limitations of the available data. Since no real underground nuclear explosion recordings exist in DAS strain rate format, the selected approach had to account for the data gap while still producing a realistic testing path.

The three main alternatives considered were threshold-based detection, rule-based signal classification, and a converted data machine learning classifier. A threshold-based detector would be the simplest option because it only flags signals that exceed a selected amplitude level. This approach is easy to implement and computationally efficient, but it is not strong enough for this project because it can not reliably distinguish between an underground nuclear explosion, an earthquake, and background noise. DAS systems are very sensitive and can record many non-seismic disturbances, such as traffic, wind, and other local vibrations. As a result, a simple amplitude threshold would likely produce too many false alarms.

The second alternative was a rule-based classification system. This method would classify events using manually selected seismic features such as P-wave strength, S-wave strength, onset sharpness, and frequency range; this approach is more useful than a simple threshold detector because it considers physical differences between explosions and earthquakes. Underground nuclear explosions generally produce stronger P wave energy, while earthquakes usually generate stronger shear wave energy [8]. However, a fixed rule-based system is still too rigid for this project. DAS measurements depend strongly on cable direction, ground coupling, geology, apparent wave velocity, and local noise conditions [9]. A rule that works well for one dataset may fail when the cable geometry or the surrounding medium changes.

The third alternative, and the selected design, is a converted data machine learning classifier. In this approach, historical nuclear explosion seismometer recordings from GTUNE and WFNE are converted into synthetic DAS strain rate signals. These converted UNE signals

can then be combined with real DAS earthquake and noise data for training and validation. GTUNE provides a large labeled dataset of underground nuclear explosion waveforms, while WFNE provides additional historical nuclear explosion waveform data in SAC and CSS formats [6] [7]. This approach is more complex than the other alternatives, but it directly addresses the main project problem: the lack of native DAS format underground nuclear explosion data.

Design Alternative	Advantages	Disadvantages	Decision
Threshold-based detection	Simple, fast, and easy to implement	Cannot classify event type and may create false alarms in noisy DAS data	Rejected
Rule-based classification	Uses physical seismic features and is easier to explain	Too rigid when noise, cable coupling, or geometry changes	Rejected
Converted dataset + ML classifier	Can classify UNE, earthquake, and noise; uses historical labeled data; better handles complex signal patterns	Requires a conversion algorithm and careful validation	Selected

Table 2.3.1*: Trade Study Comparison*

The selected design is the most practical and technically complete option for this project. It allows the system to use verified historical nuclear explosion data while adapting it to the DAS measurement format. Previous work by Barama et al. showed that machine learning can classify underground nuclear explosions, earthquakes, and noise with high accuracy using P wave seismograms [8]. However, that method cannot be applied directly to DAS because DAS measures strain or strain rate along a fiber, while traditional seismometers measure velocity or acceleration at a single point [9]. Therefore, the selected solution builds on previous machine learning work while adding a conversion step to make the data compatible with DAS-based monitoring.

2.4 Detailed Design Analysis

The detailed design uses a five-stage process: data ingestion, preprocessing, filtering, event detection, and event classification. Each stage is needed because DAS data is dense, noisy, and different from conventional seismometer recordings. The first stage is data ingestion. The system must accept different file types because the project uses several data sources. GTUNE and WFNE are stored in traditional seismic formats such as SAC and CSS, while DAS earthquake recordings may use formats such as SGY, HDF5, or TDMS [6][7]. The goal is to load all data into one consistent Python-based environment.

The second stage is preprocessing. DAS arrays contain many closely spaced channels, and some channels may have weak coupling, abnormal noise, or signal dropouts. During preprocessing, channels with poor quality are removed or flagged, and detrending is applied to remove baseline drift. This helps prevent bad channels from affecting the detection and classification results.

The third stage is filtering. A bandpass filter is applied to preserve useful P-wave and S-wave arrivals while reducing unrelated background noise. The current design uses a 1-20 Hz bandpass range, which supports seismic event detection and improves the signal to noise ratio before classification [5][8].

The fourth stage is event detection. After filtering, the system uses an STA/LTA method to compare short-term signal energy against long-term background energy. If the short-term average rises above the background level, the signal is flagged as a possible event. To reduce false detections, the system also checks whether the signal appears coherently across multiple DAS channels.

The fifth stage is classification. Once an event is detected, the system extracts features such as P-wave dominance, onset sharpness, S/P amplitude ratio, and spectral shape. These features are passed into the machine learning model, which labels the event as an underground nuclear explosion, earthquake, or background noise. The target accuracy is at least 90%, which is reasonable because the UNE training data is synthetic DAS data rather than real DAS nuclear explosion recordings [8].

Stage	Purpose	Main Output
Data Ingestion	Load seismic and DAS-compatible files	Standardized waveform input
Preprocessing	Remove bad channels and detrend data	Cleaned waveform data
Filtering	Reduce noise and preserve arrivals	Bandpass-filtered signal
Event detection	Identify possible seismic events	Event time flag
Classification	Label the detected event	UNE, earthquake, or noise

Table 2.4.1*: Detailed Design Process*

The conversion process is the key part of the design. Traditional seismometers measure velocity or acceleration at one location, while DAS measures strain or strain rate along the fiber direction [9]. Therefore, GTUNE and WFNE waveforms cannot be used directly as DAS training data. The conversion algorithm approximates what a DAS fiber would measure by transforming the seismometer waveform into synthetic strain-rate data.

The conversion should use a range of apparent wave velocities instead of one fixed value because velocity changes with wave type, geology, angle of incidence, and fiber orientation. This makes the training data more realistic and improves model robustness. Prior studies show that DAS strain-rate waveforms from subsurface explosions can be modeled and compared with conventional sensors [10] and that DAS can record explosion-generated signals in field deployments [11].

The final system output will include the event label, confidence score, estimated arrival time, and source-parameter estimates such as back-azimuth and apparent slowness when enough DAS channel information is available.

2.5 Tool Selection

The tools were selected based on compatibility with seismic data, DAS data, machine learning, and available computing resources. Since the first prototype is software-based, the tools must support waveform processing, filtering, feature extraction, classification, and visualization.

Python was selected as the main programming language because it is widely used for scientific computing, seismic processing, and machine learning. Google Colab Pro was selected because it provides cloud computing access without requiring dedicated local hardware.

ObsPy is used to process traditional seismic formats such as SAC and CSS [6][7]. DASCore and DASPy are used for DAS-style channel-based waveform data. NumPy and SciPy support filtering, normalization, conversion, and feature extraction. Matplotlib is used for raw waveform plots, filtered plots, STA/LTA detection plots, and DAS waterfall plots.

The main datasets are GTUNE, WFNE, Pawnee DAS earthquake data, and Fairbanks DAS data. GTUNE and WFNE provide labeled historical nuclear explosion data for conversion and training [6][7]. Pawnee provides real DAS earthquake data for validation, while Fairbanks supports testing with real background and environmental noise [5][12].

Tool / Dataset	Purpose	Reason for Selection
Python	Main programming language	Supports scientific computing and ML
Google Colab Pro	Computing environment	Cloud access without dedicated hardware
ObsPy	Seismic file processing	Supports SAC, CSS, and SGY formats
DASCore / DASPy	DAS data processing	Handles DAS-style channel data
NumPy / SciPy	Numerical processing	Used for filtering and conversion
Matplotlib	Visualization	Creates waveform and detection plots
GTUNE	UNE training data	Labeled historical nuclear explosion data [6]
WFNE	Additional UNE data	Historical waveforms and source information [7]
Pawnee DAS dataset	Earthquake validation	Real DAS earthquake recordings [5]
Fairbanks DAS dataset	Noise validation	Real environmental noise testing [12]
Cook Inlet DAS dataset	Earthquake and noise validation	Real DAS recordings (HDF5 format)

		[18]

***Table 2.5.1**: Selected Tools and Purpose*

Overall, the selected tools allow the team to build and test the full processing system without needing a physical DAS interrogator during this stage. The same workflow can also be expanded later for real-time DAS monitoring.

2.6 As-Designed vs. As-Built

This project aims to develop a DAS-based monitoring system that analyzes data from the fiber-optic sensing system to identify underground nuclear test explosions. The desired or as-designed system would consist of a distributed acoustic sensing network that would collect data and then classify the event as a nuclear explosion, earthquake, or background noise by applying a detection algorithm. This method yields the approximate source location and origin time of the event, which is useful for real-time seismic monitoring.

The version of the system being built for testing is more limited than the as-designed system. The system is required to use existing seismic and DAS data due to the lack of real underground nuclear explosion DAS data. The primary test implementation will focus on processing pre-recorded files, such as the Pawnee earthquake data set, and converting nuclear test data from GTUNE. This is consistent with the chosen design strategy for this project, which involves converting existing seismic data to DAS strain rate signals before processing for classification. The as-built system will use pre-recorded offline data instead of live field data. Raw seismic or DAS data will be uploaded to a Python algorithm for cleaning, filtering, and processing. The system will also use signal characteristics like P-wave amplitude, onset sharpness, duration, and frequency characteristics to correctly classify the event as UNE, earthquake, or background noise.

The differences between the as-designed system and the as-built testing system presented are reasonable for the initial prototype of the project. Based on the scope of work for this project, it is impractical to implement a real DAS system because of the required amount of field hardware, fiber access, data collection, and data storage. This version focuses on demonstrating the accuracy of the data processing and classification methods used in this project on known datasets. Offline testing of the system can be used to compare the results with known labels and events before real-time implementation. Figures A1 through A4 (Appendix A) document the preprocessing, filtering and detection results achieved during Semester 1.

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 via Jupyter/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

***Table 2.6.1**: As-Designed vs. As-Built System Comparison*

2.7 Cost Analysis

The proposed system mainly utilizes software, which ensures that the project cost remains low compared to a full DAS deployment. This prototype consists of cloud computing, data storage, and backup equipment instead of DAS hardware, fiber optic installation, and live field data collection, which assists with lowering operational costs.

Table 2.7.1 below breaks down the estimated costs, which include a Google Colab Pro subscription for one shared account, Google Drive storage for seismic and DAS files, and external storage drives for large datasets. The monthly Google subscriptions are slated to run through December 2026. The Python libraries utilized for data processing, filtering, detection, and classification are open source and do not require licensing. External storage drives provide a reliable backup solution for large datasets and would help support a potential transition of the Python development environment to GitHub. DAS hardware is also listed with no cost because the prototype featured in this project specializes in testing offline with existing datasets rather than physical, real-time field deployment.

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**

***Table 2.7.1:** Estimated Project Cost Breakdown*

The total estimated project cost ranges from $204 to $360, which is reasonable for the first stage of development. This covers the expenses needed to test the detection and

classification algorithm without investing in expensive sensing hardware. The use of a real-world DAS monitoring setup would significantly increase the cost of testing due to the need for interrogator units, specific fiber access, installation, and long-term maintenance and management [17].

3. Plans for Manufacturing

3.1 Development Approach

When developing a proper testing model for this project, the methodology used while planning was key to an efficient workflow. The full procedure from inputting datasets to obtaining usable results involves several moving parts, with each phase dependent on the one before it. For example, the raw data from the DAS network must be filtered before any accurate detection can be performed. Even a single error can cascade into larger issues down the line. For this reason, it's critical to ensure a solid foundation during development to minimize such problems.

In the case of DAS analysis, an iterative, modular method is best suited to handle each individual section before consolidating them together. The process involves working on each step in isolation to ensure they function before implementing them together. This allows for more focused development, ease of debugging, and reduced compatibility issues. The platform to be used will be GitHub, as it allows collaboration and simple version control. Local development using VS Code and Jupyter notebooks is ideal as a primary test environment, as it offers built-in virtualization through notebooks, as well as straightforward dependency installation. Google Colab Pro is available as a supplementary environment if needed. In particular, Jupyter allows for local storage and development, while Colab provides high computation power for large-scale testing.

3.2 Implementation Stages and Techniques

The entire process can be abstracted into eight distinct stages, expanding on the five-stage detection system described in Section 2.2 to include the GTUNE conversion, source localization, and visualization components required for full system development. As illustrated in Table 3.2.1, this outlines which steps relate to each other and how. It also shows the software libraries and tools to be used for each phase.

Stage	Description	Tools
1. Data Input	Import DAS data from source files	h5py, npTDMS, ObsPy
2. Preprocessing	Normalize data signals and types	NumPy, ObsPy
3. Filtering	Filter noise and isolate relevant frequencies	scipy.signal, ObsPy
4. GTUNE	Convert measurements into standard seismic	NumPy, SciPy

Conversion	motion readings	
5. Event Detection	Identify and mark potential points of interest	ObsPy MatchFilter
6. Classification	Determine the type of the event with ML	scikit-learn, TensorFlow
7. Localization	Triangulate location and timing of event	ObsPy
8. Visualization	Generate detailed plots and summary	Matplotlib, Seaborn

Table 3.2.1: *Flowchart of DAS analysis system*

3.3 Estimated Timeline and Cost

Given the current plan of action, the projected timeline for the upcoming semester is as follows: Weeks 1-3 will focus on GTUNE conversion; weeks 4-5 on filtering and tuning; weeks 6-8 on classification and training of the machine learning model; weeks 9-10 for localization and validation of results; and weeks 11-12 for full integration and the final report.

As previously described in Table 2.7.1, the majority of expenditure will be towards the hardware needed to develop, subscriptions to cloud environments, and storage. With access to open-source tools and historical datasets, the actual development of the program will be of little to no cost.

4. Plans for Testing

4.1 Testing Strategy Overview

All testing for this project is conducted offline on datasets with known ground-truth labels. Since no native DAS-format underground nuclear recordings exist, controlled offline validation is the appropriate method for evaluating algorithm performance before any real-time deployment is considered [3]. The datasets used for testing include the 2016 Pawnee earthquake DAS recordings, synthetic DAS recordings, synthetic DAS strain-rate waveforms produced by running GTUNE seismograms through the project's conversion algorithm, and ambient noise recordings from the Fairbanks permafrost study [5][6][12]. All results are evaluated against the functional requirements established in section 1.3, specifically the minimum classification accuracy of 90% and the constrained false alarm rate for UNE identification [8]. Results from the preprocessing and detection stages completed during Semester 1 are documented in Appendix A (Figures A1 through A4).

4.2 Tests to be Run

Five tests are planned to validate the system across its major functional stages, each targeting a specific stage of the system with a defined pass condition. The first test validates the bandpass filtering stage by applying a 0.02-2 Hz filter to the Pawnee earthquake dataset [5]. This test confirms the background noise is reduced while P-waves and S-waves arrivals remain identifiable in the filtered output. Initial filtering results on the Pawnee dataset are shown in Figure A1 (Appendix A).

The second test evaluates the GTUNE conversion algorithm by comparing converted synthetic DAS strain-rate waveforms against real DAS earthquake recordings from the Pawnee dataset [5]. This is considered successful if the converted waveforms fall within physically plausible amplitude ranges and exhibit coherent wavefront propagation, consistent with Mellors et al (2021) and Viens and Delbridge (2025) [10][11].

The third test evaluates classifier performance on real DAS earthquake recordings from the Pawnee dataset [5]. The test is considered successful if earthquake events are correctly labeled at or above 90% accuracy, with no misclassification as underground nuclear explosions [8].

The fourth test evaluates classifier performance on a held-out subset of GTUNE seismograms converted to synthetic DAS strain-rate format and not used during training [6]. The test is considered successful if UNE events are identified at or above the 90% accuracy threshold and the false alarm rate remains within the limit as defined in section 1.4 [8].

The fifth test is an end-to-end validation in which raw input files are processed through the full system sequence. The test is considered successful if the system produces a correct event label and confidence score, and for detected explosion events, it estimates the wave direction, wave speed and origin time in line with the output requirements as defined in Section 1.4.

4.3 Comparison to Analysis

The algorithm outputs will be compared to the theoretical physical behavior obtained from research papers and the corresponding labels of the datasets utilized. Lindsey et al. demonstrated that DAS recordings can accurately capture P-wave and S-wave arrivals after filtering and that DAS arrays can estimate back-azimuth and slowness through beamforming [5]. This is consistent with the filtered results shown in Figures A2 and A3 (Appendix A), where P and S wave arrivals are clearly identifiable in the Cook Inlet dataset. This comparison will help in verifying the project's detection, filtering, and source parameter results. In addition, L. Barama et al. provide insight into labeled GTUNE underground nuclear test explosion data through precise location coordinates, station metadata, origin time, and P-wave arrival information. The paper also supports the use of P-wave arrival behavior when analyzing explosion waveforms for classification, given its greater visibility relative to S-wave energy [6].

In addition, the final outputs will be verified against known labels, such as earthquake, UNE, or noise, that are provided along with the dataset. The requirements mentioned in Section 1.3 will be referenced to confirm that the results exhibit 90% classification accuracy, preserved seismic arrivals, and source parameter estimates for potential explosions.

4.4 Estimated Costs and Time

For this project's scope of work, testing is not expected to contribute significant additional cost beyond the development and design budget shown in Section 2.7. The local Python environment, Google Drive, external drives, and open-source Python libraries will continue to be used throughout the testing phase, with Google Colab Pro available as a

supplementary option for more intensive tasks. In addition, the testing will occur offline using existing datasets, which means no additional DAS hardware or field equipment is required.

The testing phase is expected to last approximately 3-4 weeks, around weeks 9-12 of Semester 2. The testing period will take place alongside the planned system integration and GTUNE conversion checks because each module must be tested as it's added to the main workflow. The first stage of testing will be centered on enhancing the accuracy of filtering and event detection. This will be followed by classifier testing and final validation using known earthquake, UNE, and background noise labels from datasets.

References

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

[2] D. P. Voytan et al., "Yield Estimates for the Six North Korean Nuclear Tests From Teleseismic P Wave Modeling and Intercorrelation of P and Pn Recordings," *Journal of Geophysical Research: Solid Earth*, 2019. doi: 10.1029/2019JB017418.

[3] CTBTO Preparatory Commission, "Seismic Monitoring." [Online]. Available: https://www.ctbto.org/our-work/monitoring-technologies/seismic-monitoring.

[4] M. Mazur et al., "Global Seismic Monitoring using Operational Subsea Cable," arXiv:2409.19827, 2024. [Online]. Available: https://arxiv.org/pdf/2409.19827.

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

[6] L. Barama et al., "GTUNE: A Dataset of Historical Underground Nuclear Explosions," *Seismological Research Letters*, 2022. [Online]. Available: https://globaltechinc.com/wp-content/uploads/2022/09/SRL-2022036_AUTHOR_1_LB_edited.pdf.

[7] V. Oancea et al., "Waveforms from Nuclear Explosions (WFNE) Repository," *Pure and Applied Geophysics*, 2025. doi: 10.1007/s00024-024-03536-7.

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

[9] B. L. N. Kennett, "The Seismic Wavefield as Seen by Distributed Acoustic Sensing Arrays: Local, Regional and Teleseismic Sources," *Proceedings of the Royal Society A*, 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8826365.

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

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

[12] T. Cheng et al., "Watching the Cryosphere Thaw: Seismic Monitoring of Permafrost Degradation Using Distributed Acoustic Sensing," *Geophysical Research Letters*, 2022. doi: 10.1029/2021GL097195.

[13] L. Barama and Q. Kong, "Improving Machine Learning Models for Seismic Explosion Discrimination with Seismology-Driven Features," *Seismological Research Letters*, 2025. doi: 10.1785/0220240177.

[14] D. Chambers et al., "DASCore: A Python Library for Distributed Fiber Optic Sensing," *Seismica*, vol. 3, no. 2, 2024. doi: 10.26443/seismica.v3i2.1184.

[15] L. Krischer et al., "ObsPy: A Bridge for Seismology into the Scientific Python Ecosystem," *Computational Science & Discovery*, 2015. doi: 10.1088/1749-4699/8/1/014003.

[16] D. von Seggern and R. Blandford, "Source Time Functions and Spectra for Underground Nuclear Explosions," *Geophysical Journal of the Royal Astronomical Society*, vol. 31, pp. 83–97, 1972. doi: 10.1111/j.1365-246X.1972.tb02363.x.

[17] Silixa Ltd., "iDAS™ Intelligent Distributed Acoustic Sensor," *Silixa Ltd.*, 2024. [Online]. Available: https://silixa.com/technology/idas-intelligent-distributed-acoustic-sensor/.

[18] UW FiberLab, "Cascadia Initiative DAS Dataset," University of Washington, 2023. [Online]. Available: https://dasway.ess.washington.edu.

[19] Y. Zhu, "Seismic Events: Classification, Prediction, and Detection," Peking University, 2022. [Online]. Available: https://yzhu.io/s/research/cv/earthquake/.

Appendix A

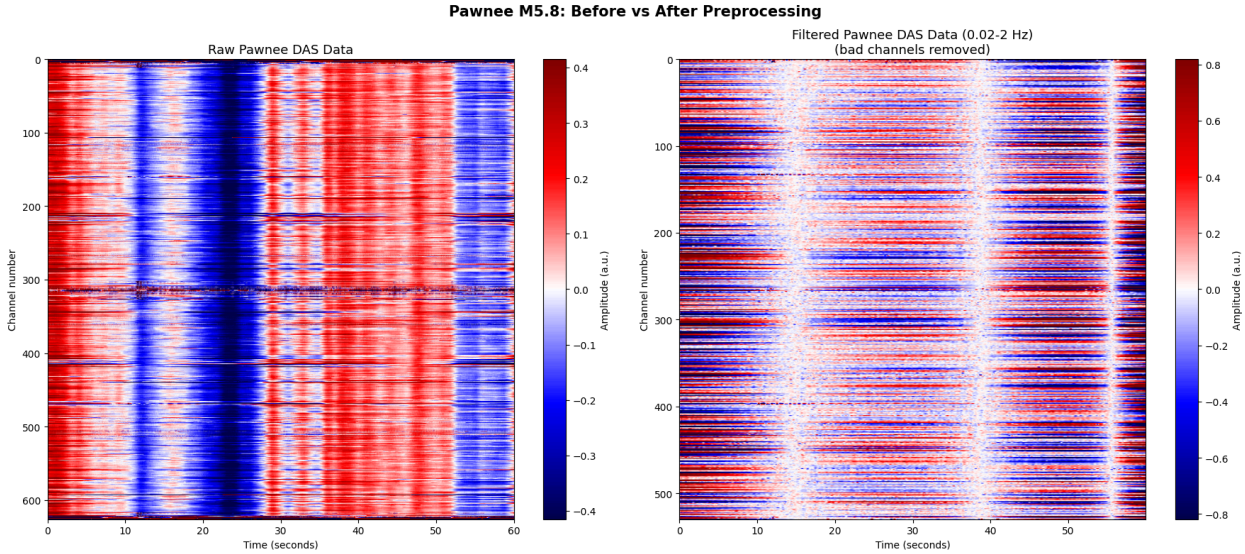


Figure A1: Pawnee M5.8 Earthquake (2016): Raw vs. Preprocessed DAS Data. Left: raw SGY data showing baseline drift and faulty channels. Right: after detrending, strain-rate conversion, laser drift removal, bandpass filtering (0.02–2 Hz) and bad channel removal.

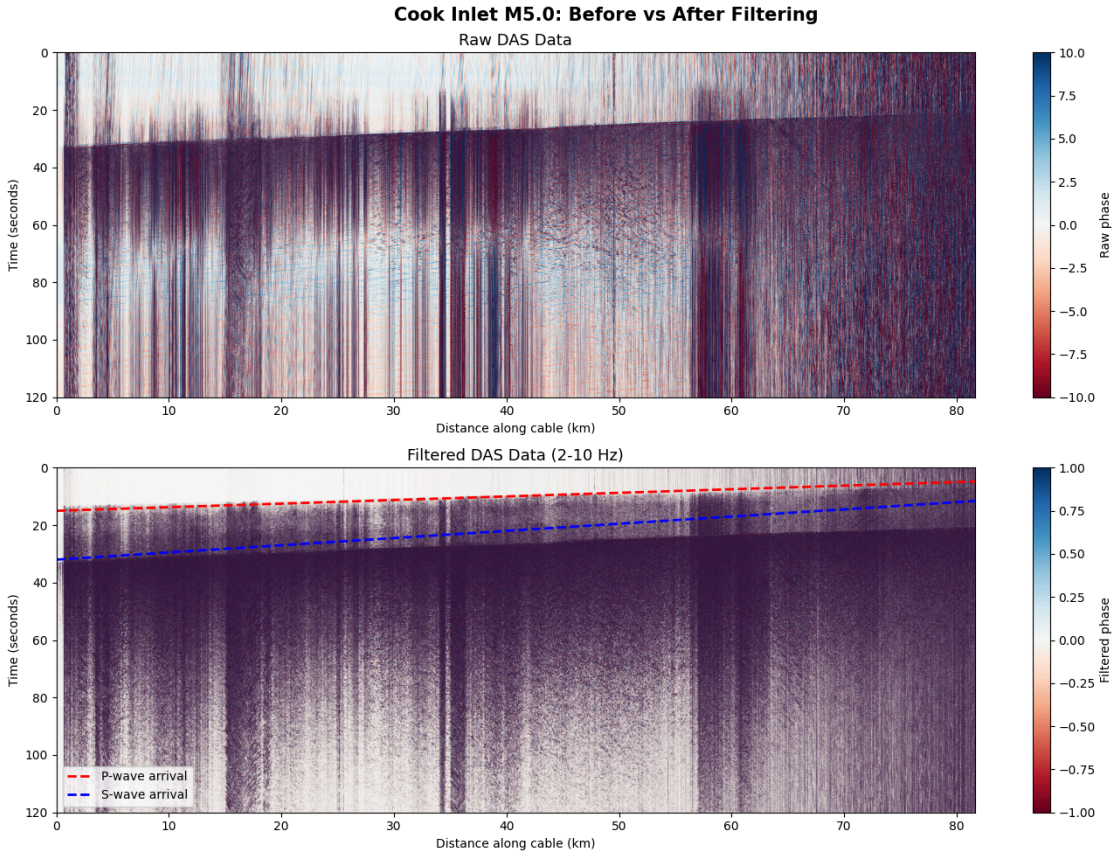


Figure A2: Cook Inlet M5.0 Earthquake (2023): Filtered DAS Waterfall (2–10 Hz). Red line indicates P-wave arrival (~8 km/s apparent velocity), blue line indicates S-wave arrival (~4 km/s). Diagonal wavefront pattern confirms seismic wave propagation across 81.65 km of seafloor cable.

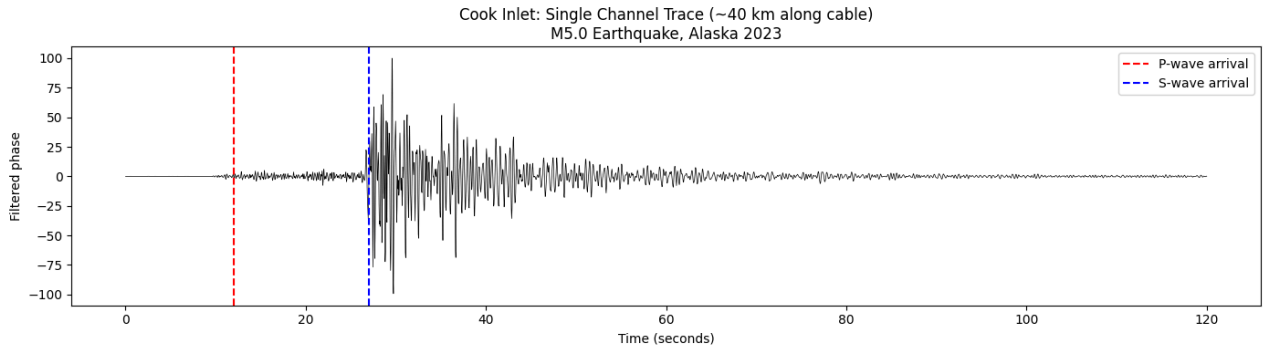


Figure A3: Cook Inlet M5.0: Single Channel Trace (~40 km along cable). Quiet background signal before earthquake arrival, followed by P-wave onset (red line) and larger S-wave arrival (blue), consistent with expected seismic wave behavior.

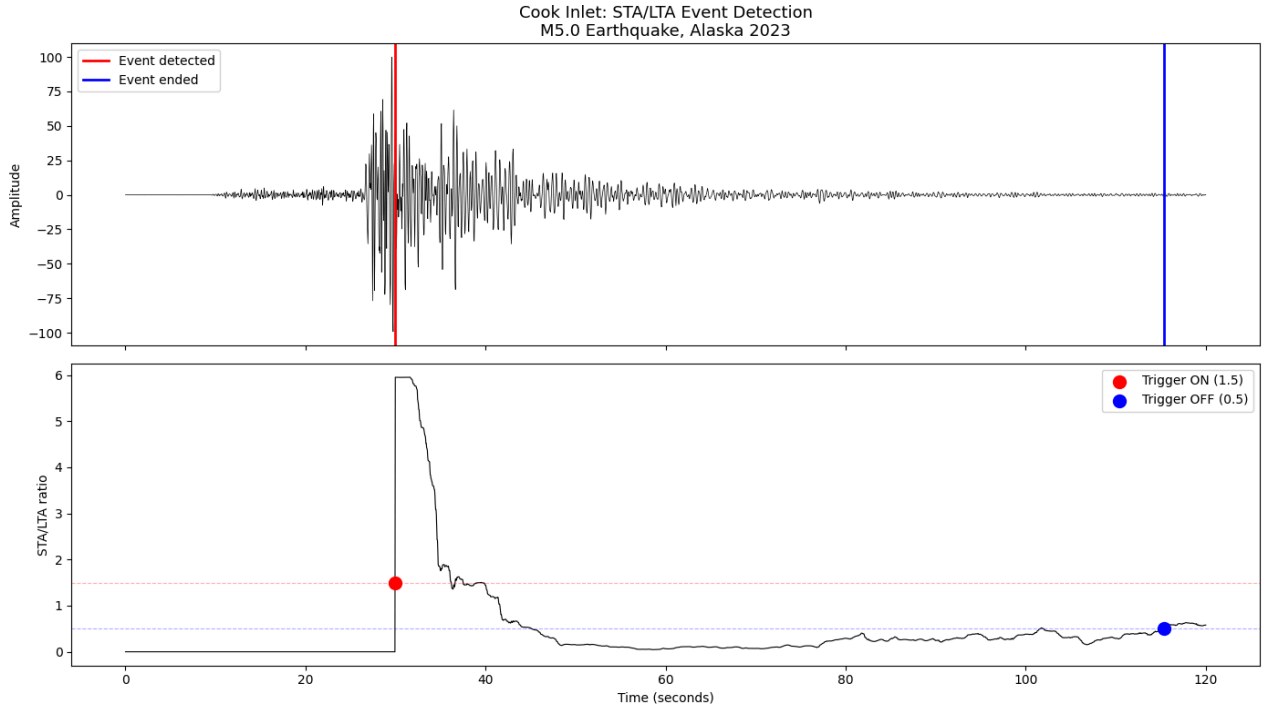


Figure A4: Cook Inlet M5.0: STA/LTA Automatic Event Detection. Top panel: single channel waveform with trigger-on (red) and trigger-off (blue) markers. Bottom panel: STA/LTA characteristic function showing ratio spike at earthquake arrival. Detection threshold set at 1.5.