System of Networked Sensors for Detection and Characterization of Underground Activity Final Product Report

Augustus Standeven, Tyler McKean, Vulindsky Fanfan, Yohannes Kidane
Teaching Mentor: Professor Tomas Materdey, Customer Mentor: Captain William Shepherd

Abstract: For this project we were tasked with designing a ground vibration-based security system that can capture and classify vibration signals and display information to a GUI for user operation. A subsystem for vibration data acquisition was designed using SM-24 geophone sensors, a DC Offset PCB, and Arduino Due microcontroller. The ground vibrations would be captured and converted to digital value via the ADC of the Arduino and then serially imported into the Raspberry Pi subsystem. The Raspberry Pi handled data preprocessing of the digital signal information, feature extraction using statistical methods and functions on the data and compiled the features into a table that was fed into a Kth-Nearest Neighbor machine learning classification algorithm. The entire process was designed to operate with a real-time functionality with the GUI on the Raspberry Pi 4" display updating predicted signals and keeping a history tab of all signals classified along with the date and time of incidence. The goal of the project was to have a future team continue the progress after the initial team has graduated and optimize and/or increase the system's functionality. The Technical Manager for this project was Dr. Tomas Materdey and the Customer Mentor for this project was Captain William Shepherd from the Steven's Institute for Technology.

I. Problem Definition

A. Motivations

For our senior design project, we are tasked with designing a system of networked sensors for detection and characterization of unauthorized underground activity. The main purpose of this networked sensor array is to provide an additional security system for government properties like military bases home or abroad, government facilities, and national border security. The detection of unauthorized underground activity is costly in terms of consumer market products, often complex, time consuming to set up/calibrate, and difficult to interpret as they can output ambiguous information. Hence, this presented our team with an opportunity to design a cheaper and more accurate sensor array that can categorize, detect, and transmit these different types of underground activities.

B. Customer Requirements

We received this research topic from the Capstone Marketplace through our Customer Mentor at the Stevens Institute for technology. Since this research topic comes from a government contract, the main purpose of this project is to provide security for government property. The Capstone Marketplace prompt for this project provided a list of features that our customer would like the system to have. These system requirements include that the security system must be:

- Portable, durable, and unobtrusive
- Easy to set up and take down
- Able to cover a 100-meter radius
- Cheap
- Wireless Network of sensors (at least 2)
- Self-contained power source
- Able to process data and categorize types of signals
    - i.e. determine the likelihood of whether a signal received was from a passing truck or from a tunnel being dug.
- Able to send processed signal data to a laptop for user to interpret data

Over time and through many meetings among the team members, we have concluded that the project will be able to meet some of these requirements, but not all. We plan to leave this project, as well as all our research and work, available to future engineering students so they can continue this project with the hopes of improving it where we could not.

C. Engineering Requirements

For the engineering requirements for this project, our team followed the coursework requirements created by our Technical Manager, Dr. Tomas Materdey, who was also the professor in charge of our senior design course.

For our senior design class, the team was tasked with tracking their progress through a variety of assignments and presentations. Throughout the year, our team was tasked with creating four presentations: a Formal Design Presentation, a Project Readiness Presentation, a Design Validation Presentation, and a Final Product Presentation. Each of these presentations were designated as critical milestones in our team's project timeline. As well as these four presentations, our team communicated in weekly and bi-weekly Zoom meetings with our CM and TM, creating weekly agendas to track our progression, and built a team webpage on the UMass Boston servers to showcase our presentations.

II. System Diagram

A. Sub-systems and Interconnections

This section displays the subsystems and interconnections between the components throughout the system. For more details on each component see section IV: Components and Final Budget.

1) Flow Chart Diagram

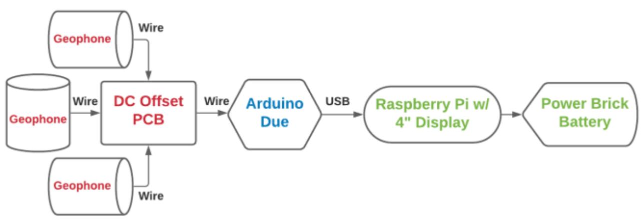


Figure 1 – Flow Chart Diagram of Subsystems and their interconnections.

Figure 1 shows the final flow chart diagram for our system. In the Data Acquisition subsystem (shown in red), there are three SM-24 geophones that are fed into a DC Offset PCB. After this, the offset vibration signals are fed into the Arduino Due for A/D conversion (shown in blue). The digitally converted signals are then fed to the Raspberry Pi subsystem (shown in green) where they are preprocessed, fed through a feature extraction script, and then compiled into a table of features that are then inputted into a KNN machine learning classification algorithm. The result of the classification from the KNN model, displays to a custom GUI on the 4" touchscreen display attached to the Raspberry Pi. Finally, the entire system is powered by a 12V 6000mAh Power Brick battery.

2) Computer Graphics

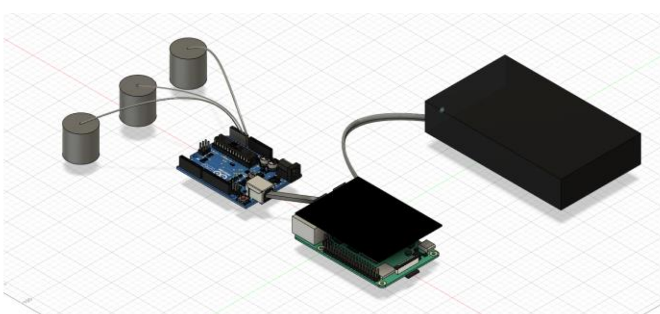


Figure 2 - CAD model of our system design created in Fusion360.

Figure 2 above shows a 3D model of our system design with three SM-24 geophones tied to an Arduino Due, which is then fed to a Raspberry Pi 4 and displayed on the 4" LCD screen. The entire system is powered by the battery on the right side.

III. Team Dynamics

A. Team Roles

This section addresses the roles each team member played throughout the course of the project.

1) Augustus Standeven

Augustus helped with many different sections of the project. As one of the Senior EE students in the project, he helped conduct the initial research and design of the project from the starting phases. Augustus worked extensively on the vibration data acquisition, system design and implementation, design validation tests, project budgeting, and Python coding. For the Python coding, he specifically helped create and implement the data acquisition code, the machine learning algorithm, and the master script. Augustus was also in charge of the project's CAD model, writing the final paper, and scheduling weekly zoom meetings with the team and CM/TM.

2) Tyler McKean

Tyler assisted in several sections of the project as a Senior EE student member of the project. He participated in the initial researching and designing into the area of underground vibration detection. In terms of hardware, Tyler led the team in the data capturing of vibration signals using the Arduino subsystem and designing a DC Offset PCB required to fully capture oscillation data from the ground vibrations. For software, Tyler primarily wrote the preprocessing and feature extraction algorithms in MATLAB that later were converted to Python scripts, which would run on our Raspberry Pi subsystem. Tyler also handled the project management by utilizing the Microsoft Project software to create a Gantt chart that timelined the team's progress and deadlines.

3) Vulindsky Fanfan

Vulindsky, the third Senior EE student, joined the team at the halfway mark of the project. He was responsible for the Python implementation of the graphical user interface (GUI), which enabled the user to see when an vibration activity was detected by one or all three of the geophone sensors. The user will be able to see the prediction of trigger events. Within the GUI, he was tasked with also creating a history tab the user could access in case a trigger event was missed by the user or occurred while the system was unattended. Vulindsky also helped with the classification algorithm and design validation tests, along with assisting with the data acquisition from the Arduino and data storage management on the Raspberry Pi.

4) Yohannes Kidane

Yohannes was the Junior EE member that contributed in the fields of research, Power Management and Filter/Amplification design. Yohannes participated in the initial research along with his teammates to gather information regarding potential systems out in the field. He led the team in the calculations and design for the power needed for systems, along with ideas to improve the efficiency. Yohannes also demonstrated how the use of analog Filter/Amplifier could further improve our system in the future, which Yohannes could potentially continue and lead in the expansion of this system for the future teams in the Fall.

B. Team Communication

The team communicated via text and email daily in order to further the progress of the project. Weekly zoom meetings were scheduled for just the student team members as well as meetings with Dr. Materdey and Cpt Shepherd. All the team's research, presentations, and papers were stored on a shared Google Drive so that the entire team had access to all our collective and individual work.

IV. Components and Final Budget

A. Components

The components of our project design can be broken down into their respective subsystems. The Arduino subsystem consisted of an Arduino Due microcontroller, DC Offset PCB, and three SM-24 geophone sensors attached to the Arduino by 2m long wires wrapped in a black wire insulator. The DC Offset PCB contained six 1kΩ resistors and three 1µF capacitors. The Raspberry Pi subsystem included a Raspberry Pi 4 with a case and 4" LED screen attachment. In addition to the Raspberry Pi was a 12V 6000mAh Power Brick battery that would supply the power for the entire system. To interconnect all the subsystems to the battery, two micro-USB cables were involved as well.

B. Final Budget

Figure 3 displays the list of components used for the project and estimates the total cost for the design to be about $740. The team was given an initial budget of $1000 from the UMass Engineering department and additional funds were proved through the Stevens Institute of Technology thanks to our Customer Mentor, William Shepherd.

Component	**Price Per**	**Volume to Purchase**	**Supplier**
SM-24 Geophones	$60.00	9	Sparkfun.com
PCBs	$20.00	1	JLCPCB
Resistors	$5.00	24 ($5 total)	Andrew Davis
Capacitors	$5.00	12 ($5 total)	Andrew Davis
Arduino Due	$40.00	1	Amazon Link
Raspberry Pi	$30.00	1	Andrew Davis
Raspi Case & LED screen	$28.00	1	Amazon Link
256 GB micro SD	$30	1	Amazon Link
Lithium Ion Batteries	$35	1	Amazon Link

Total Cost: ~$740

Figure 3 - Final Budget Estimation for project.

V. Design Improvements

The initial prototype of our system involved an Analog Discovery 2 oscilloscope and digital signal generator to capture the ground vibrations from the geophone sensors.

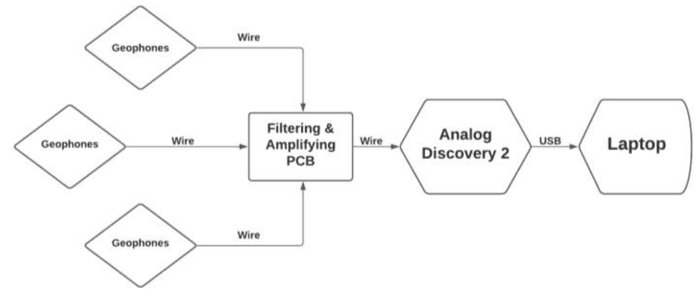


Figure 4 - Initial Phase Prototype

Figure 4 above shows a similar flowchart to the current model of our design; however the early design incorporated a Filtering and Amplifying PCB, Analog Discovery 2, and a Laptop. The team performed several outdoor tests during the first months of the project and noticed the AD2 did not have enough analog inputs to observe the required number of geophones suggested in our initial flowchart design. This required a design change to our system, of which we replaced and added several components to improve our design.

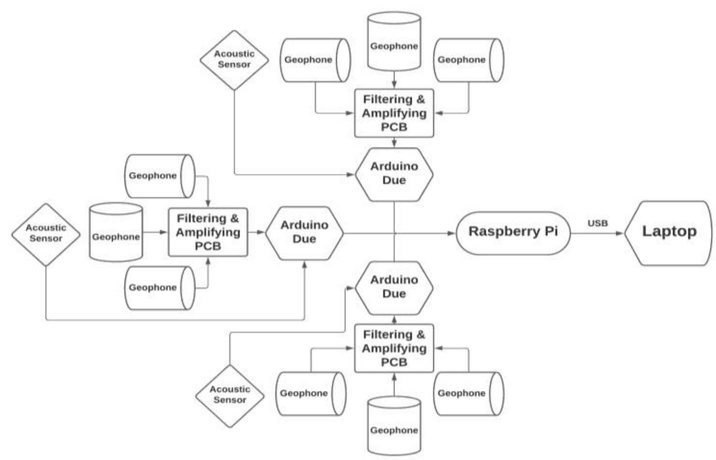


Figure 5 - Second Phase Prototype

Figure 5 above displays the changes the team made to the design, which included the replacement of the AD2 with the Arduino Due microcontroller, the addition of a Raspberry Pi, acoustic sensor, and expanded nodes for signal capturing. The AD2 was replaced for the Arduino Due because its 12-bit resolution ADC and low-power operation. A Raspberry Pi was first introduced into our design at this phase of the prototype and was intended for data logging and processing of the vibration information that was captured by the Arduino Due. An acoustic sensor was also added to observe acoustic information for surface level disturbances. Having acoustic sensors capture atmospheric disturbances, whilst the geophones observe subterranean disturbances would provide the system with more information when classifying vibrations.

As previously shown in Figure 1, the system was reduced to only contain three SM-24 geophones, a DC Offset PCB, an

Arduino Due, a Raspberry Pi with a 4” LCD screen, and a power brick battery. The team prioritized using design’s functionality with only three geophones to save time and reduce the complexity of the project. Our intention was to prove the system design could work successfully on a small scale and could later become scaled in size and components to better meet the original customer requirements. The Filter/Amplifier PCB was originally intended to be a part of the system, but due to the lack of equipment and lab access for half of the project, instead the team decided to create a DC Offset PCB to meet the PCB requirement from the Senior Design class syllabus, though a Filter/Amplifier circuit could serve a purpose to the project for a future team to design. The role of the Raspberry Pi was redefined as it became the main processing subsystem that would collect the digital data from the Arduino Due, preprocess the signals by converting the digital values back to analog values via some simple arithmetic operations, extract features such as the mean, RMS, peak-to-peak values, etc., and input these features into a KNN machine learning model all in the Python programming language [7].

VI. Ethical Considerations

The list of ethic considerations for the project are listed below:

- ISO/IEC 25064 – Provide useful and necessary data.
- IEC 60478 - Stabilized Power Supplies
- COVID-19 – CDC Health protocol

For the ISO/IEC 25064, we show it fit to strive to design a system that provides the most accurate data possible even if the system is underperforming. This ethical consideration was viewed as incredibly important since we do not want to provide false classifications to our user or lie about the accuracy ratings.

For the IEC 60478 consideration, we made sure to purchase a DC power supply that properly fit the power ratings of all the subsystems in the design and would not damage or harm the components or user. We also soldered two-meter insulated wire extensions for the geophones, so the user did not have to handle small electrical wires and risk an improper assembly of the system.

The last ethical consideration was the COVID-19 protocol us students and faculty have been abiding by over the past year. To gain access to the lab on campus at UMass, each team member was required to get a COVID-19 test on a weekly basis and provide a time sheet for the hours they logged while in the lab on campus. While in the lab, the team remained six feet apart and wore appropriate facial masks in the indoor lab environment. Before departing the lab, each team member sanitized and wiped down their desks and chairs to ensure proper mandated sanitation protocol.

VII. PCB

A. Purpose

The PCB was designed as a DC Offset circuit that could allow the Arduino subsystem’s Analog-to-Digital converter to capture the full peaks and troughs of the oscillation information incident to the geophone sensors. Without the DC Offset, the ADC of the Arduino would not be able to capture any negative voltage values because the specified range for the Arduino’s ADC is voltages from 0-3.3V. Thus, a DC Offset was created using a simple voltage divider circuit that would offset the geophone vibrations to a DC value of 1.65V, exactly in the middle of the ADC’s input range. The full peak to peak oscillations would then be fully captured and converted to digital values without any loss of vibration info.

B. Design

The design of the PCB consists of a voltage divider circuit with two equal valued resistors and a capacitor for each of the three geophones intended to be captured by the Arduino. A schematic diagram for the circuit can be seen in Figure 6 below. The geophone sensor would first run through an electrolytic capacitor that prevents any DC signal from entering the geophone sensor. After the capacitor, two equal valued resistors in parallel offset the geophone signal by a factor of 1.65V. By using the 3.3V pin on the Arduino as the V_{cc} signal to the voltage divider, the offset can add a DC signal of 1.65V to the oscillating signal of the geophone. Thus, the ADC of the Arduino will observe analog voltages at an equilibrium of 1.65V and be able to capture the peak-to-peak information of the ground vibrations. Using Altium Designer, Tyler was able to create the schematic and design the layout of the PCB all within one software application. Figure 7 below shows the layout design of the DC Offset PCB.

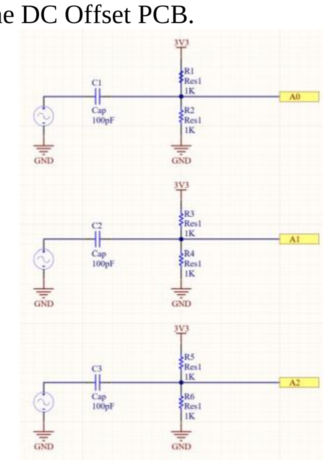


Figure 6 - Schematic of DC Offset

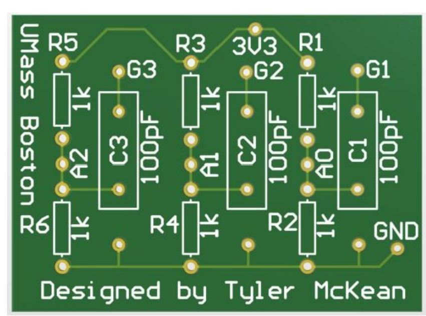


Figure 7 - Altium Designer PCB layout

C. Performance

Using tools such as digital multimeters and DC power supplies, Tyler was able to test the design of the PCB at the on-campus labs at UMass Boston. Each designated output of the PCB that would tie to the ADC inputs of the Arduino were tested, which Figure 5 below displays the results. Each PCB input was well within the marginal range of the expected value of 1.65V. This concludes the design of the PCB was a success.

PCB Input	Measurement (V)	Expected Value (V)	Tolerance (%)
A0	1.64962	1.65	0.02
A1	1.64984	1.65	0.01
A2	1.65023	1.65	0.01

Figure 8 - PCB test results table

VIII. Design Validation

A. Validation Test 1

The first validation test was meant to verify the functionality of our machine learning model. We conducted this test by using the machine learning toolbox in MATLAB. This toolbox has users input a list of features, as well as known classes, and trains a variety of different machine learning models. The toolbox then rates the accuracy of each model, allowing us to determine what type of machine learning model we should use for this project. It is important to note that we are only using MATLAB to determine what type of machine learning algorithm best suits this project. We then implemented the most accurate machine learning algorithm in Python. Figure 9 below displays the user interface of built-in Classification Learner.

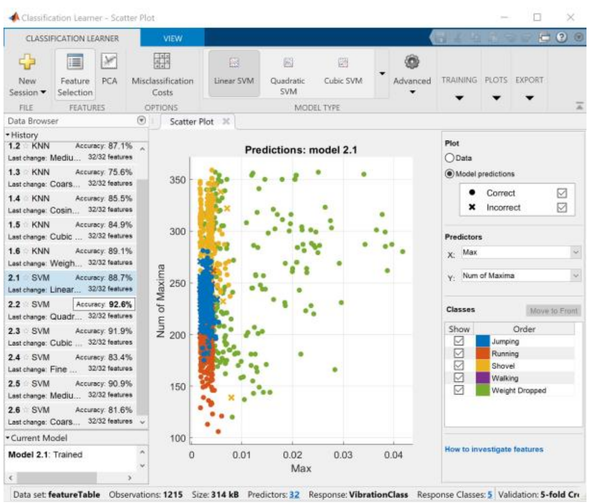


Figure 9 - MATLAB Classification Learner Application part of the Machine Learning Toolbox

The Classification Learner uses Supervised Learning to train different machine learning models from the data imported into the application. The user provides a table of features with known classes and the UI allows us to train multiple models in order to determine which is most accurate for our dataset.

B. Validation Test 2

Validation Test 2 involved both the data acquisition of the Arduino and logging of digital values on the Raspberry Pi. The goal of this test was to make sure that the interconnections between the Arduino and Raspberry Pi could successfully capture raw analog vibration signals and serially transmit the information between subsystems. If done correctly, the data will display the captured vibration signals from the geophones and display them on the Raspberry Pi, after being converted from A/D through the Arduino Due. We also wanted the Raspberry Pi to save these values to a .csv file.

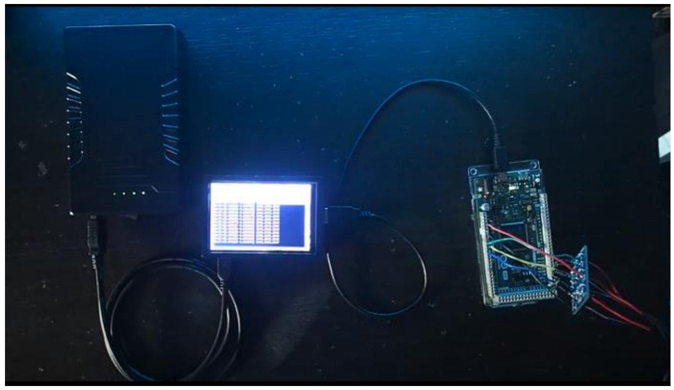


Figure 10 – Validation Test setup with the subsystems interconnected and being powered solely by the Power Brick Battery.

C. Validation Test 3

Validation Test 3 was setup to test the performance of the system's power supply. The power brick has a 5V USB port, that daisy-chains the power to the Raspberry Pi and then the Arduino Due. The Raspberry Pi requires 5V/2.5A to operate properly and the Arduino Due requires 3.3V. We calculated that the battery should be able last for about 15 hours. So, test the battery life, Tyler setup and left the system running over night, starting from a full battery charge, to see how long it could last until the subsystems were no longer receiving the necessary power to maintain their operation.

IX. Validation Data

A. Validation Test 1

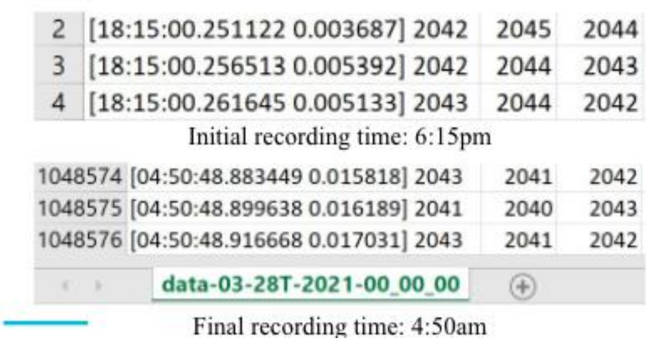
For our first validation test, we were able to acquire 1215 vibration samples across 5 different classes: walking, running, jumping, shoveling, and dropping a weight. We extracted features from this signal set and then used the MATLAB machine learning toolbox to determine which machine learning model best fits our data. We found that the KNN models gave us an accuracy of about 85-89%, which we consider to be in good standing and meeting the project goals.

Figure 11 – Python KNN model results displayed to our custom GUI showing an accuracy of 89.32% for the 1215 sample library.

The KNN model was implemented in Python and setup by parsing the 1215 samples into a ratio of 70% training and 30% testing. The model would use about 850 randomly selected signals from the 1215 set to train the KNN model based on the Supervised Learning approach. The remaining 30% or 365 samples were treated as test samples to verify the accuracy of the trained KNN model. The results shown in Figure 11 above show a total accuracy of 89% when feeding the 30% test signals into the model rendering this as a successful validation test. Our team set out to achieve an accuracy between 70-80% based on our CM's request, so we managed to outperform that metric. The system would then remain in a constant state of testing new signal when setup in the field, and this is how we achieve the vibration classification necessary for the project's design requirements.

B. Validation Test 2

In order to test whether the Raspberry Pi could continuously catalog data from the Arduino Due, Tyler setup the system to run overnight, which allowed us to observe it continuously cataloging sensor data and save the information to a .csv file we could open in Microsoft Excel. The Raspberry Pi was setup in its terminal to record and timestamp data that was being serially transmitted from the Arduino Due. Figure 12 below demonstrates that the test was successful as we managed to save digital values for over 10 hours of continuous recording.



2	[18:15:00.251122 0.003687] 2042	2045	2044	
3	[18:15:00.256513 0.005392] 2042	2044	2043	
4	[18:15:00.261645 0.005133] 2043	2044	2042	

Initial recording time: 6:15pm

1048574	[04:50:48.883449 0.015818] 2043	2041	2042
1048575	[04:50:48.899638 0.016189] 2041	2040	2043
1048576	[04:50:48.916668 0.017031] 2043	2041	2042

data-03-28T-2021-00_00_00

Final recording time: 4:50am

Figure 12 – Excel file displaying timestamp information for the start and end of Validation Tests 2 and 3.

C. Validation Test 3

With a fully charged battery, the system was capable of continuously recording vibration data from the start time of 6:15PM to the last recorded timestamp at 4:50AM the following morning. Given the first and last timestamp values in Figure 12, the battery was able to sustain about 10 hours and 35 mins worth of signal capture. This underperformed the calculated value of 15 hours of battery life, because that calculation represented how long the actual power brick could sustain a change. However, we learned from this validation test that the battery only needs to drop under the necessary 5V threshold to power down the system and disrupt signal capturing. This caused the team to potentially look into a replace battery with a higher mAh rating that could help the system to sustain a longer operation period. Though, our team was advised to let another team resolve the battery life and to work with what we had.

X. Deliverables

For the final project deliverables, we plan to submit a real-time vibration classification system that includes:

- DC Offset PCB
- 9 SM-24 Geophone Sensors
- Arduino Due microcontroller
- Raspberry Pi 4 with case, 4" LED screen attachment, and 256 GB SD card
- GUI that displays vibration classification via LED screen
- MATLAB to Python conversion documentation
- Real-Time Python Data Acquisition and Feature Extraction scripts
- 12V 6000mAh Lithium-Ion Power Brick Battery
- Project Video Summary
- Project Poster

XI. Future Work

The main goal of our project is to not only create a network of sensors that can detect and classify unauthorized underground activity, but to also create a project that has room for expansion in the future. We believe this project could further be improved by adding additional components such as analog filters and amplifiers to increase the signal strength and quality, which could further improve the accuracy of weaker classification signals and increase the range of signal capturing. We also believe there is a potential to eliminate the use of wire and create a network of wireless geophone sensors to increase our range and usability. We hope future teams can use this system as a foundation to their improved and optimized model to improve the project requirements initially given to us in the Capstone project specifications.

XII. Appendix

Troubleshooting Table		
Problem Observed	**Possible Reason**	**Solution**
Drastically abnormal values	The normal range of values (received by the RPI) is centered around 2048, if there are values +- 2000 off from 2048 there may be an issue with the soldering connections	Check the soldering connections on both the PCB and the geophones. These joints are fragile, and our team has had to resolder them multiple times.
Error when running code in real-time	The values streamed into the RPI should be in three columns,	Check that this is indeed the issue and modify the code to delete any rows that have

	sometimes these columns are merged or extended by accident.	more than 4 columns.
Classifications are incorrect	Different ground mediums significantly affect how the vibration signals are received. If the machine learning algorithm is trained on signals from a particular area it may misclassify those same signals if they are received via a different ground medium.	Design a library to train a ML model for a particular location or find a way to remove the transform function of the ground medium via code.

A. List of Materials

- 3 wired geophones
- 1 PCB
- 1 Arduino Due
- 1 Raspberry Pi 3 B+
- 1 12V battery
- Source Code
- Signal Library

The three wired geophones are our data acquisition devices and should be connected directly to the PCB. This PCB adds a DC offset so that both the peaks and the troughs of the signals can be captured. This PCB is then sent to the Arduino Due for analog to digital conversion (note: our team found that it was necessary to use the Arduino Due since it has an excellent A/D converter, and the Raspberry Pi does not have any analog inputs). These digital signals are now sent to the Raspberry Pi where they are run through Python code and assigned a class. This whole system is powered by the 12V battery, and the only component that needs to be directly tied to the battery is the Raspberry Pi. The GUI and resulting classification will display on the Raspberry Pi screen although it is possible to display the Raspberry Pi desktop on an external monitor.

Operation Manual:

Open the "source code" python file on the Raspberry Pi desktop and run it. This should display the GUI and it will continuously run until stopped. This code is thoroughly commented, so refer to the comments for any further questions about the code. The signal library used to train the machine learning algorithm should be in the same folder as the source code file. Below, in figure 13, is a conceptual flowchart for how the code is supposed to run in real time.

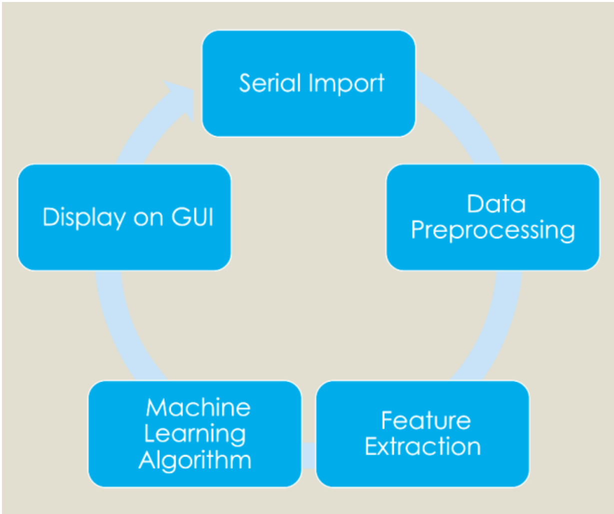


Figure 13 – Conceptual flowchart of the Python code

B. Previously Unsuccessful Approaches

Throughout this project our team tried several different approaches in order to implement our classification algorithm. Our team wanted to originally have the feature extraction and machine learning algorithm all created by MATLAB. Our team was successfully able to create a feature extraction process and a machine learning model in MATLAB, but we were unable to implement it in real time. We investigated converting the MATLAB code to C code with the "MATLAB C Coder" application, but we were unable to convert the code. We also tried to use the "MATLAB Application Compiler" in order to export our pretrained machine learning model as an application that could run independently of MATLAB on a Raspberry Pi. We were also unsuccessful with this approach. Finally, our team tried to run the signal capture and machine learning algorithm on the Arduino by implementing Simulink, but this was also unsuccessful. Our team would encourage any future teams to research if any of these approaches seem viable, since they all seem more accurate than coding the project in Python. All of our MATLAB code can be found in the same folder as this README and the source code file.

C. Possible Design Improvements

Our team has several design improvements that we would like to suggest to any future teams. First of all, we would like to suggest an expansion of both the geophones and the nodes. One improvement would be to implement wireless geophones, as well as adding 3 nodes instead of 1 (as our early prototype diagrams show). We believe this would greatly increase the range of the sensors and create a better security system. As well as this we would like to suggest adding acoustic sensors or visual sensors. This could act as a "verification" method for incoming signals. If both the geophones and the acoustic sensors classify a signal as "RUNNING", then we will be able to classify the observed signal with much more certainty than before.

Our team would also like to suggest that any future teams expand the signal database used to train the machine learning algorithm. A larger signal database would create an even more accurate machine learning algorithm. We would also suggest that any future teams consider the transform functions of differing ground mediums and their effects on observed signals. We have theorized that if a team were to obtain the transfer function of a particular ground medium, then they could filter out this transfer function from all received signals (for more information on this topic research "signal processing", "convolution", and " transform function").

Our team would also suggest that a future team should buy a better battery. The one we have currently implemented runs for about 10 hours, and a longer battery life would greatly improve the usability of the system.

Finally, our team would suggest that future teams make it so the "source code" script runs at the startup of the Raspberry Pi. This would make the system appear more user-friendly to any soldiers or personnel that do not have coding experience. Our team decided not to implement this idea because we would like future teams to work on this project before we consider it "finished".

References

[1] D. H. Cross, "Terrain Considerations And DataBase Development For The Design And Testing Of Devices To Detect Intruder-Induced Ground Motion," May-1978. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA055602.pdf.

[2] Rubin, Marc & Camp, Tracy & van Herwijnen, Alec. (2012). Automatically Finding Avalanches in Geophone DataL A Pattern Recognition Workflow. International Snow Science Workshop (ISSW).

[3] "Signal Processing and Machine Learning Techniques for Sensor Data Analytics - Video," *Video - MATLAB*. [Online]. Available: https://www.mathworks.com/videos/signal-processing-and-machine-learning-techniques-for-sensor-data-analytics-107549.html. [Accessed: 19-Apr-2021].

[4] "Designing and Implementing Real-Time Signal Processing Systems Video," *Video - MATLAB*. [Online]. Available: https://www.mathworks.com/videos/designing-and-implementing-real-time-signal-processing-systems-1502972724722.html. [Accessed: 19-Apr-2021].

[5] KAFADAR, O. A geophone-based and low-cost data acquisition and analysis system designed for microtremor measurements In-text: (Kafadar, 2020) Your Bibliography: Kafadar, O., 2020. A geophone-based and low-cost data acquisition and analysis system designed for microtremor measurements. *Geoscientific Instrumentation, Methods and Data Systems*, 9(2), pp.365-373.

[6] N. Evans and D. Chesmore, "Automated Identification of Vehicles using Acoustic Signal Processing," *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3342–3342, 2008.

[7] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.