Occupancy Tracking for Smart Facility Management

12-778: Sensors, Circuits, and Data Interpretation & Management for Infrastructure
12-741: Data Management

Prof. Mario Berges
Fall 2023

Team:
Adem Taiyr
Jiaqi Meng
William Qiu
Natasha Susan Fernandez
Bhanu Rayudu

---

1 Introduction

1.1 Motivation

Heating, ventilation and air conditioning (HVAC) systems consume approximately 40% of a building's electricity, and is one of the most significant factors in the operating expenses of a building [1]. Especially in commercial buildings, where the spaces are large, varyingly occupied, and most commonly centrally conditioned, the operating costs shoot up even though people may use the facilities for only about 8-9 (working) hours each day. Therefore, more efficient design and operation of HVAC gives promise to significant energy savings and reduction in maintenance costs and frequency.

1.2 Aims

We aim to create a self-reliant and automated HVAC control system that uses smart sensors to dynamically adjust to the changing occupancy, ensuring the efficient use of the HVAC in spaces that are large enough to have more than one HVAC unit. We also aim to use this system to collect occupancy data for the sake of a more informed and efficient design and placement of HVAC units across the facility.

1.3 Goals

- Efficient use of energy by appliances such as Air Conditioning, Heating, and Ventilation
- To maintain appropriate and efficient levels of ambient temperature in buildings based on occupancy patterns
- To make more informed decisions about and ascertain locations for HVAC installation at the design stage

2 Proposed Solution and Use Cases

2.1 Solution

Our system is designed to collect sound, vibration, and temperature readings using sensors that are strategically placed across the space. This system is able to plot occupancy in real-time, and, based on occupancy patterns, signal HVAC to adjust to the new environment, by deactivating or reducing use of HVAC units that are far from human congregations, and activating or increasing use of those that are close. The system also stores the data received from sensors for plotting time-series graphs and other uses, such as design or validation of the HVAC system. Compared to many of the existing solutions that involve more sophisticated sensors such RFID or even video cameras [2], our solution involves simple and cheap to produce sensors that are easy to install, operate and maintain.

2.2 Use Cases

2.2.1 Case 1: Design

Based on historical data which depicts time-based occupancy patterns as well as the local temperature patterns within the space, HVAC system designers or architects will be able to determine potential location of the HVAC units and subsequently the ductwork, enabling efficient and more well-informed design.

2.2.2 Case 2: Operation

Based on real-time occupancy localization data, HVAC systems can operate much more efficiently by adjusting to changing occupancy patterns dynamically so as to save energy and reduce maintenance.

3 Review of Physical Phenomena Being Measured

Two primary physical phenomena used for occupancy tracking are sound and vibrations, and to a lesser extent, temperature. Sound is generated from footsteps, speech, physical activity, etc., that cause vibrations in a medium, usually air, which propagates outwards from the source in the form of sound waves. The frequency and amplitude of these waves correspond to pitch and volume respectively. As the waves travel, they lose energy to the medium and spread out in a wider extent, causing the sound to diminish in volume as it travels further from the source. When sound is produced, the waves it is associated with will also remain in that medium for some duration.

Vibrations are generated from human interactions with physical objects, such as chairs, which result in mechanical disturbances (waves) that propagate through the chair frame. Some examples of physical interactions include sitting down, standing up, moving, shifting weight, etc.

Human activity also results in slight temperature changes, which can be used to either correlate them with other readings for validation, or to extract insights about temperature changes/trends in particular locations of a room.

4 Sensing System Architecture and Design

The following flowchart demonstrates the high-level sensing system architecture, from physical stimuli all the way up to occupancy map and data analytics:

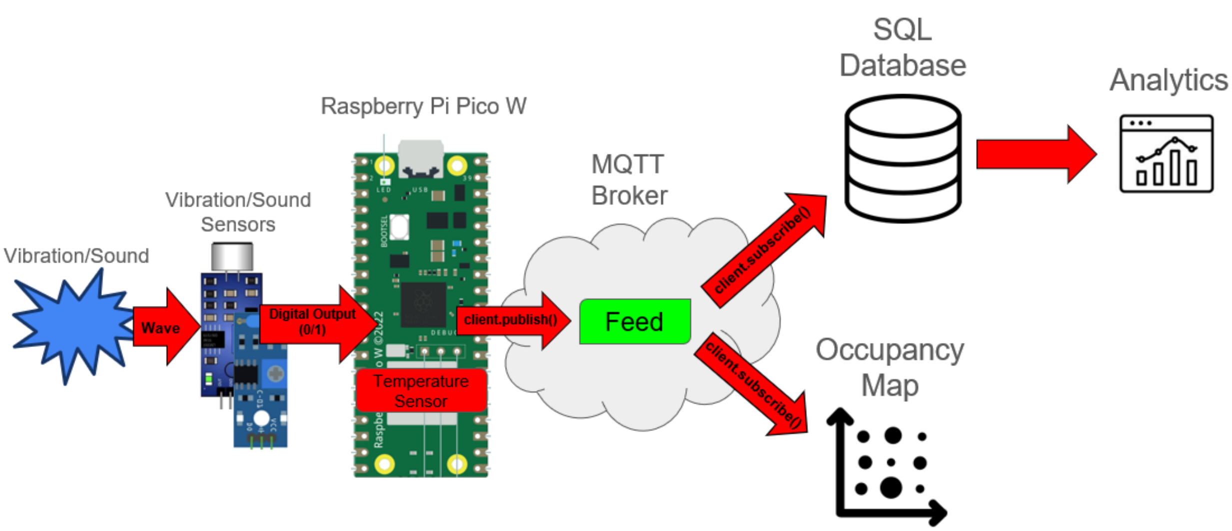


Figure 1. System architecture flowchart [3]–[7]

First, human activity generates physical stimuli: sound or vibrations. Second, in the case of sound, a sound sensor picks up variations in air pressure caused by sound waves propagating through air. These variations cause air molecules to vibrate, which in turn cause oscillations in the diaphragm within a sound sensor, which then generates electric current with a magnitude proportional to the magnitude of oscillations. The electric current is then converted to a binary digital output signal using a system consisting of a potentiometer, which is used to set a threshold voltage that can be fine-tuned, and an op-amp, which compares the incoming current from diaphragm with this threshold voltage and outputs 1 if the current exceeds threshold, or 0 otherwise. This digital output is then sent to Raspberry Pi Pico W board, which samples the output at its full frequency. At the same time, Pico also samples temperature from an in-built temperature sensor. The vibration sensor follows the same process as the sound sensor, with only difference being the source of the current in the form of a suspended mass within the encasing that oscillates due to vibrations, deforming piezoelectric crystal, which in turn generates current due to piezoelectric effect.

Before running the sensors, Pico W connects to MQTT Broker. Following that, each time a positive signal from sensors is sampled, it is immediately published to the Feed within MQTT Broker. SQL Database and the machine plotting the occupancy map also connect to the same MQTT Broker and continuously listen to the same Feed, retrieving the messages immediately as they come. SQL Database then stores the data and runs analytics on it (currently it is limited to plotting time series), while another machine accumulates messages, plots the occupancy map, and clears cache, repeating this cycle every two minutes. Based on the occupancy map, the HVAC system adjusts HVAC units to the new updated occupancy environment. This design allows a continuous and uninterrupted communication between sensors and an HVAC system, while a distributed manner of the data use (database vs occupancy plotting running on different machines) allows computationally efficient handling of data inflow and also acts as a safety layer in case one of the systems goes down.

The rationale behind the placement of sensors is as follows. The first principle that we followed is the installation of sensors across the room in a way that they cover the entire area leaving no blind spots. The second principle is to install one vibration sensor on every physical object that can be used as part of human activity, such as tables, chairs, etc. The third principle is to install sound sensors to cover areas where there are no suitable objects for placement of vibration sensors, such as hallways or empty spaces, or if the number of vibration sensors to cover the entirety of the room is insufficient. Vibration sensors provide more granular occupancy tracking since they track vibrations on specific objects, hence they should be prioritized. On the other hand, sound sensors provide more general information on if there is occupancy in its entire range without considering the number or specific location of occupants.

One key assumption for our system was that each vibration sensor would detect the motion of one occupant. To make this assumption as realistic as possible, we put vibration sensors on chairs where the general use case was for 1 person to sit in each chair. Because the vibration sensors detect the vibration of the chairs they are attached to, it is assumed that when an occupant is sitting in that chair, they are moving around on the chair enough to be constantly sending vibrations. This is assumed because if an occupant sits in a chair and does not move around enough to cause vibrations, our system will inform the user that there is no one in that chair when there actually is (a false negative result). The system reduces the probability of this displaying on the user interface by using readings over a 2 minute time period to determine real-time occupancy, assuming that there is a low chance that an occupant will sit in a chair unmoving for over 2 minutes.

Our system also assumes that the location of each sensor is fixed within the room. The system uses this assumption to identify the sensor that readings are coming from based on their location. It also assumes that each of the sensors are completely independent of each other. When multiple sensors are triggered due to the detection of vibration, our system assumes that the cause/source of the vibration for each sensor is different from each other. This is especially important for sound sensors, which are more susceptible to picking up readings from the same source than vibration sensors because they share the same medium (air). To reduce the likelihood of this occurring, sound sensors were generally placed on opposite sides of the room outside of each other's range.

5 Data Processing Pipeline

The data processing pipeline in our project is engineered to manage and analyze data efficiently, from its initial collection to the final analysis. The primary component of this system is a MySQL database, tailored to handle complex and varied data obtained from our sensor network. This database functions as more than a storage entity; it plays a crucial role in organizing and processing the data. Within this database, key tables such as SoundReadings, VibrationReadings, and TemperatureReadings are strategically designed to store sensor data along with vital metadata, including timestamps and sensor coordinates. This design is critical for conducting in-depth analyses and extracting meaningful insights by identifying patterns within the data.

The SoundReadings table is a prime example, specifically configured to record sound levels accurately, an essential aspect of room occupancy assessment. This level of specialization extends to other tables like VibrationReadings and TemperatureReadings, each catering to different data types. Segregating data into distinct tables addresses the challenges of inefficient and cluttered data storage systems. This approach ensures operational efficiency and data integrity by minimizing unnecessary null entries. The database's structure aids in streamlining data retrieval and analysis processes, thereby enhancing the overall effectiveness of the data management strategy.

In concert with the database, the Graphical User Interface (GUI) developed for this system is integral to the data management framework. This dynamic interface, consistently updated with the latest database inputs, is designed to be accessible for users of varying technical expertise. It includes functionalities for filtering and querying data based on diverse parameters, such as time, date, location, or sensor type, which significantly enhances user interaction and engagement with the data.

A key feature of the GUI is the integration of Matplotlib for advanced data visualization. This tool is pivotal in presenting data in varied graphical formats, like time-series plots, to effectively highlight trends and anomalies in sensor data. These visual representations make complex data sets more accessible and intuitive, facilitating a clearer understanding than raw numerical data alone.

In conclusion, the data processing pipeline in our project represents a harmonious integration of a well-structured database and a user-centric GUI. This synergy is crucial in managing the entire data lifecycle, from collection and storage to analysis and visualization, aiming to provide clear and comprehensive insights from the sensor data.

6 Data Management

The data management aspect of our project is centered around a MySQL database, designed to organize and analyze data collected from various sensors. The database's architecture is meticulously structured to accommodate a wide range of data types. Tables such as SoundReadings, VibrationReadings, and TemperatureReadings are developed to store primary sensor data and essential metadata, including timestamps and geographical coordinates. This structure plays a pivotal role in enabling thorough data analysis and facilitating advanced querying capabilities.

Each table is tailored to specific data types, as exemplified by the SoundReadings table, which is dedicated to recording instances when sound on a particular sound sensor exceeds the threshold. This specialization is consistent across other tables, including VibrationReadings and TemperatureReadings, facilitating detailed analyses and the extraction of relevant insights, particularly in the context of room occupancy trends. Segregating data into different tables enhances the system's efficiency and maintains data integrity by preventing the accumulation of irrelevant or null values, thus ensuring the database's relevance and functionality.

The development of a Graphical User Interface (GUI) is a significant element of the data management system. Regularly updated with the most recent data, the GUI is designed to be user-friendly, catering to a broad range of technical skills among users. It features customizable data views based on various parameters, enhancing user interaction and experience with the data.

The inclusion of Matplotlib in the GUI for advanced data visualization is notable. This feature allows for the representation of data in various graphical forms, particularly time-series plots, which are effective in depicting temporal trends in the data. These visualizations significantly enhance the data analysis process, making it more interactive and insightful for users.

In summary, the data management strategy in this project combines a carefully structured database with a user-oriented GUI. This combination ensures a comprehensive management of the data lifecycle, including collection, storage, analysis, and visualization. The goal is to provide deep insights into room occupancy patterns, fully utilizing the sensor data to support informed decision-making and enrich our understanding of environmental dynamics.

The following figure shows E-R diagram adopted for the database design. Notably, it has three one-to-many relationships from Sensors table to SoundReadings, VibrationReadings, and TemperatureReadings tables, since each sensor may produce many readings, whereas each reading is produced once and by one sensor only.

```
+-------------------+     +---------------------+     +---------------------+
| SoundReadings     |     | VibrationReadings   |     | TemperatureReadings |
+-------------------+     +---------------------+     +---------------------+
| - id: int         |     | - id: int           |     | - id: int           |
| - timestamp: datetime | | - timestamp: datetime | | - timestamp: datetime |
| - location: varchar   | | - location: varchar   | | - location: varchar   |
| - signal_type: varchar| | - signal_type: varchar | | - signal_type: varchar|
+-------------------+     +---------------------+     +---------------------+
        ^                           ^                            ^
        |                           |                            |
        +---------------------------+----------------------------+
                                    |
                                    |
                           +---------------------+
                           | Sensors             |
                           +---------------------+
                           | - sensor_id: int    |
                           | - type: varchar     |
                           | - location: varchar |
                           | - installation_date: datetime |
                           +---------------------+
primary key for Readings table: id
primary key for Sensors: sensor_id
```

Figure 2. E-R database diagram

7 Validation and Experiments

7.1 Validation of Individual Sensors

A few trial runs were made prior to evaluation to fine tune the sensitivity of sensors. The following metrics were used to calculate the performance of sensors:

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$; $Precision = \frac{TP}{TP+FP}$

$Recall = \frac{TP}{TP+FN}$; $F1 - score = \frac{2TP}{2TP+FP+FN}$

Explanation: TP (True Positive) is when a signal is produced and the sensor correctly picks it up; TN (True Negative) is when no signal is produced and the sensor also does not pick anything up (the entire interval with no signal counts as one instance of TN); FP (False Positive) is when there is no signal but sensor erroneously shows that there is a signal; FN (False Negative) is when there is signal produced but sensor does not detect it.

7.1.1 Vibration Sensor

The vibration sensor was run on different chairs (classroom chairs, stools, sofas, etc.) while a test subject was producing different types of vibrations (standing up, sitting down, moving around, tapping on the chair, moving the chair, etc.). The following metrics have been obtained after evaluating the sensor cumulatively for approximately 2 minutes:

Table 1. Vibration Sensor Validation Metrics

TP	TN	FP	FN	Accuracy	Precision	Recall	F1-score
45	35	0	10	0.889	1	0.818	0.9

Generally the vibration sensor performed well on all types of chairs, and on most types of activities. It had some difficulty detecting a person standing up or moving the chair.

7.1.2 Sound Sensor

While the vibration sensors were only located on chairs, sound sensors were placed on both tables and chairs. The goal of the sound sensors was to detect the sound of people doing activities, such as talking, but they also detected sound from other sources. Testing was done in a room with somewhat noisy air vents and the results of 5 minutes of testing are shown in Table 2.

Table 2. Sound Sensor Validation Metrics

TP	TN	FP	FN	Accuracy	Precision	Recall	F1-score
128	20	0	4	0.974	1	0.970	0.985

Sounds produced for the test were primarily speech from conversations between people at about a normal speaking volume. Conversation was also conducted at various distances from the sensor (from about 1 meter to 20 centimeters). Some of the false negatives could be attributed to conversation that took place out of the sensor range of about 1 meter. True Negative readings were counted if the sensor correctly detected no readings during the period between conversations. Although the room was producing noise through the air vents, the sensors produced no false positive readings during the 5 minute period.

The performance of individual sensors is satisfactory for our system given that the resulting metrics are relatively high at around 90%. Notably, the sensors did not produce any False Positives, meaning that as long as the interval for data collection is sufficient, the system should be able to find all “true” occupants without producing any “false” occupants.

7.2 Validation of Sensing System

3 different rooms of varying sizes have been used for testing the system and 5 total scenarios with different arrangements of occupants have been considered. The scenarios have been devised in a way to test all the sensors and to consider edge cases, such as two people sitting near each other, or multiple people within a range of only a sound sensor. Photos of arrangements of people and detailed results are shown in the Appendix.

Table 3. Validation Results

Location	Scenario	Number of congregations (ground truth)	Number of congregations (prediction)	Number of occupants (ground truth)	Number of occupants (prediction)
PH A2 Lab	Scenario 1	6	6	6	6
PH A1	Scenario 1	3	3	3	3
	Scenario 2	2	2	2	2
CEE Study Lounge	Scenario 1	4	4	4	4
	Scenario 2	1	1	2	1
	Total	**16**	**16**	**17**	**16**
	Accuracy		**100%**		**94%**

The system was able to correctly localize non-contiguous congregations of people 100% of time using a 2 min data collection interval, producing no false positives. However, when it came to exact estimation of the number of people in a room, the system did not give a correct result for scenario 2 in CEE Study Lounge whereby two people were sitting close to each other and they were within range of only 1 sound sensor. From this we can conclude that the system can be used to accurately determine the number of congregations while the number of individual occupants may not be entirely accurate.

8 Conclusions and Future Work

Our system used vibration and sound sensors to determine the occupancy of a room for facilities managers to design HVAC systems around. The system's performance was experimentally evaluated and yielded 100% average accuracy in detecting separate congregations of people when sampled for 2 minutes. Tests were done in 3 different rooms with different sensor layouts and varying numbers of occupants and occupant patterns. Hence, it can be concluded that the proposed system can be used to generate real-time occupancy map for smart and efficient HVAC management by tracking congregations of people. Historical time series data can also be used for more informed design and planning of HVAC systems.

With the current system design, only one sensor can send messages to the MQTT broker at a time due to technical limitations imposed by the MQTT server. This causes delays between receiving of a signal from sensor by Pico to sending of messages to the server since Picos get disconnected from the server when multiple Picos are trying to send messages simultaneously, and this necessitates the addition of an error handling block on Picos for automatic reconnection to the server. This can be addressed as part of future work by, for example, adding more brokers/servers to the system or switching to a more robust MQTT platform.

Another limitation of the system is the inability to estimate the exact number of occupants. This mainly pertains to sound sensors as has been observed during evaluation, which currently do not have a system to separate out signals coming from different occupants, and instead they combine all incoming signals together and treat them as one single congregation. There are edge cases that should be considered for vibration sensors too, such as when two people are sharing one chair.

Based on the findings of this study, the system should not be used in conditions too different from those used as part of evaluation, such as extremely large rooms (libraries, etc.), which require more sensors and necessitate separate testing, or rooms with very high congregations of people, in which case the system results may not be reliable.

In addition to improving the system by adding robust MQTT servers, the system could allow for a wider range of applications by adding ultrasonic distance sensors to each Pico. These sensors send out sound waves and calculate the time it takes for those waves to travel back to the sensor, which they then convert into a distance based on the speed of sound [8]. These sensors would provide the location of each set of sensors within a room. For example, if these sensors were added to chairs with the vibration sensors, if the chair is moved to a different location in the room our system would still be able to track occupancy and provide locations for those occupants.

The sensors themselves could be improved. Using more sensors to cover all of the chairs in a room would allow for more complete occupancy data. Both vibration and sound sensors had dials that controlled their sensitivities. If the system used sensors that could change their sensitivities to specific values, each sensor could be calibrated through testing to improve the accuracy of the data and reduce the number of incorrect readings. The sound sensors could be replaced with sound sensors that measure the volume/magnitude of sounds instead of giving simple binary output. These sensors could be calibrated through testing to determine the number of occupants within the sensor range based on the volume of sound that it records.

Our system can also be improved in the future by using machine learning to predict the trend of occupancy in certain locations or rooms. The system may then be able to automatically adjust temperature for those locations in advance, so that the ambient temperature is set at an optimal level even before the occupants are present, thereby increasing comfort while being efficient.

Finally, our system can be adapted for a use beyond HVAC, such as smart and automated lighting, shading, sunlight control, automatic control of appliances (TVs, audio systems, CCTV), etc.

9 References

[1] "Where's the waste? The inside story on your HVAC system." Accessed: Oct. 12, 2023. [Online]. Available: https://www.pge.com/en_US/small-medium-business/business-resource-center/energy-management-articles/energy-management-articles/past-articles/wheres-the-waste.page

[2] H. Saha, A. R. Florita, G. P. Henze, and S. Sarkar, "Occupancy sensing in buildings: A review of data analytics approaches," *Energy Build.*, vol. 188–189, pp. 278–285, Apr. 2019, doi: 10.1016/j.enbuild.2019.02.030.

[3] "Download Website Analytics Icon for free," Vecteezy. Accessed: Dec. 09, 2023. [Online]. Available: https://www.vecteezy.com/vector-art/2205990-website-analytics-icon

[4] "Raspberry Pi Documentation - Raspberry Pi Pico and Pico W." Accessed: Sep. 16, 2023. [Online]. Available: https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html

[5] Admin, "How to use Vibration Sensor with Raspberry Pi Tutorial," DIY Projects Lab. Accessed: Sep. 16, 2023. [Online]. Available: https://diyprojectslab.com/vibration-sensor-with-raspberry-pi/

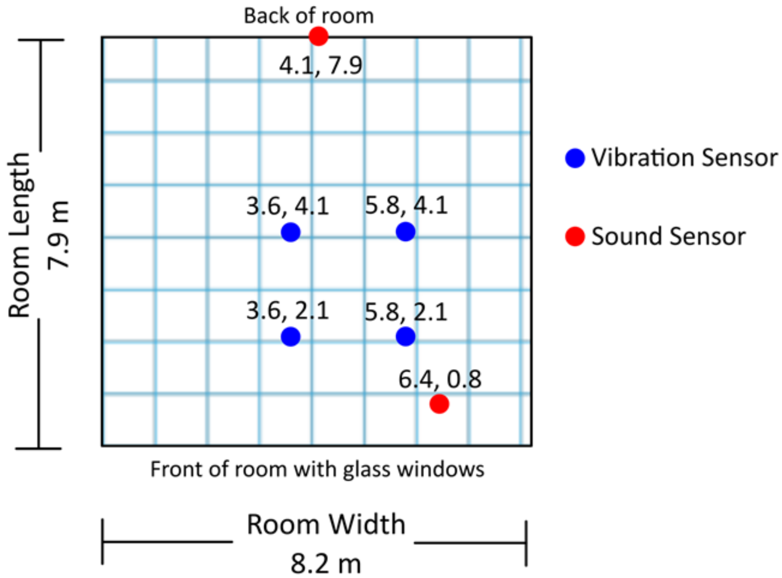
[6] "Heatmap Icon - Free PNG & SVG 107041 - Noun Project," The Noun Project. Accessed: Dec. 09, 2023. [Online]. Available: https://thenounproject.com/icon/heatmap-107041/

[7] "Database free icons designed by Smashicons," Flaticon. Accessed: Dec. 09, 2023. [Online]. Available: https://www.flaticon.com/free-icon/database_149206

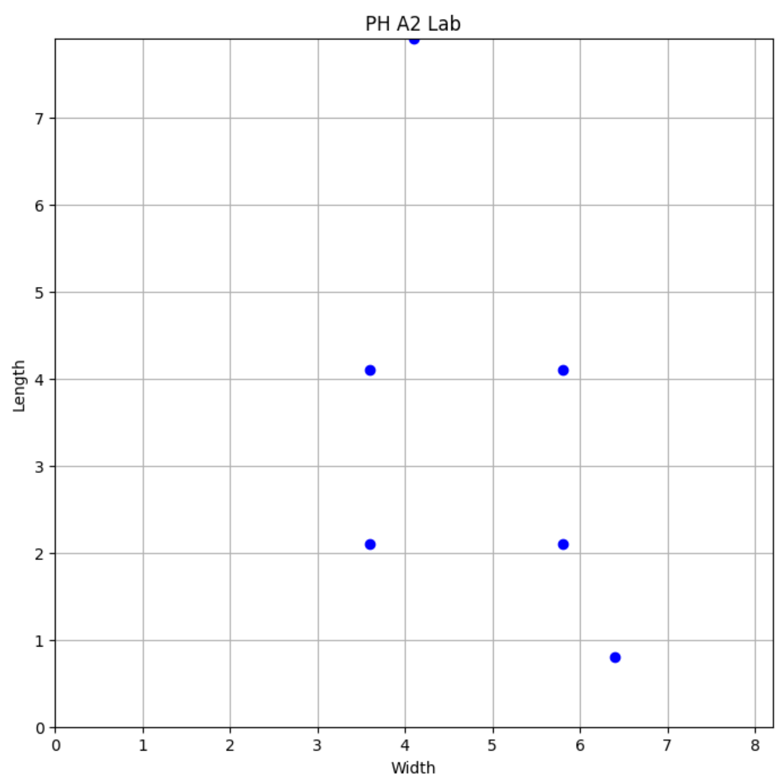
[8] "HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi," The Pi Hut. Accessed: Dec. 10, 2023. [Online]. Available: https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

Appendix A. PH A2 Scenario

A.1 Diagram of Sensors in PH A2 Lab

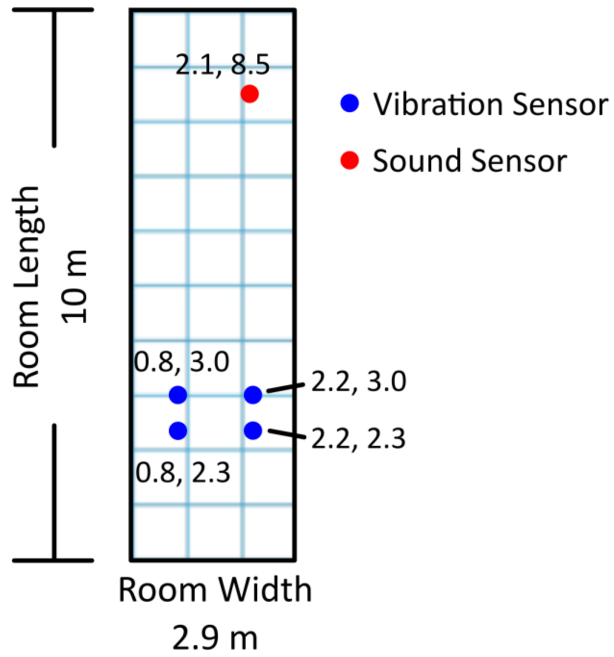


A.2 PH A2 Lab Occupancy Map when all sensors are triggered

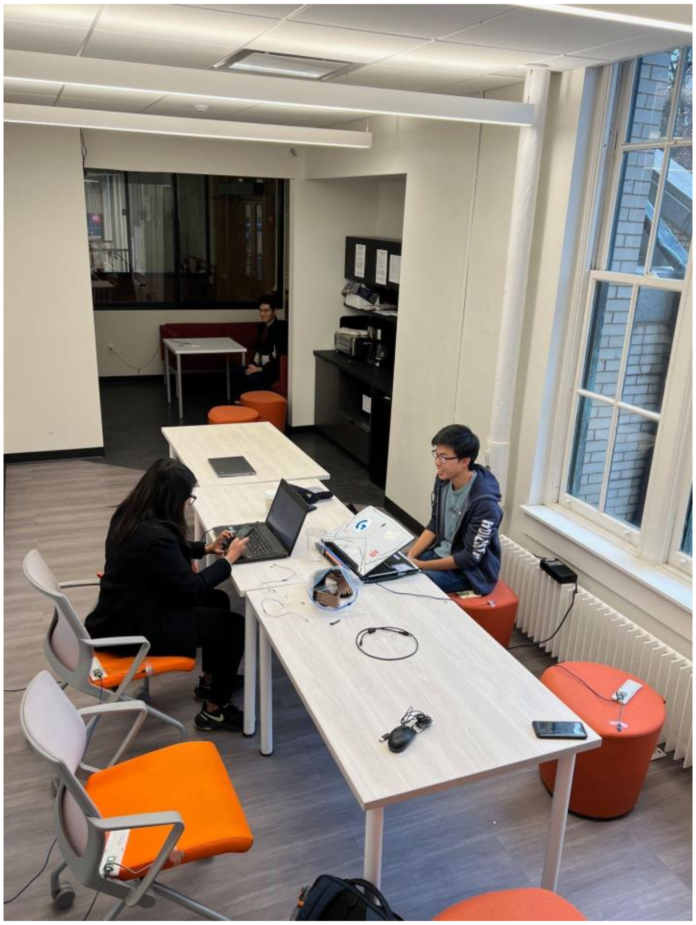


Appendix B. PH A1 Scenarios

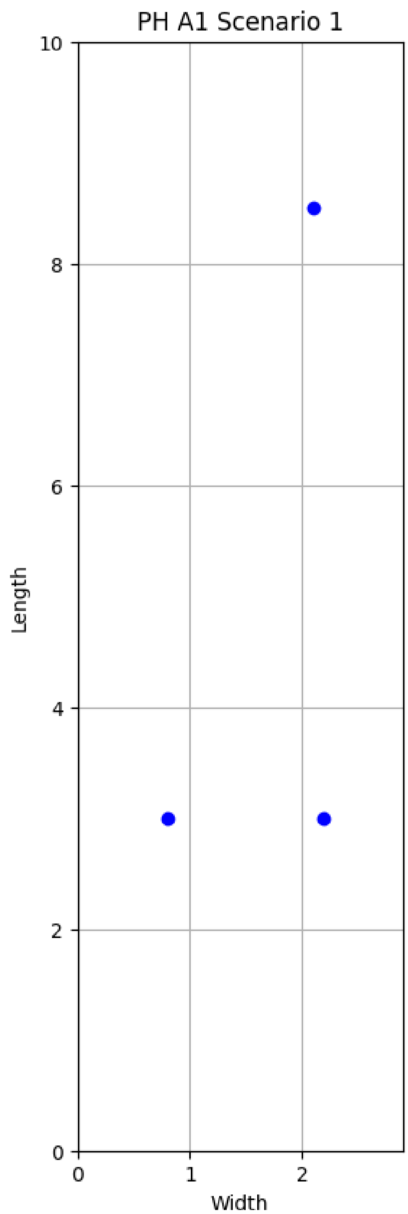
B.1 Diagram of Sensors in PH A1



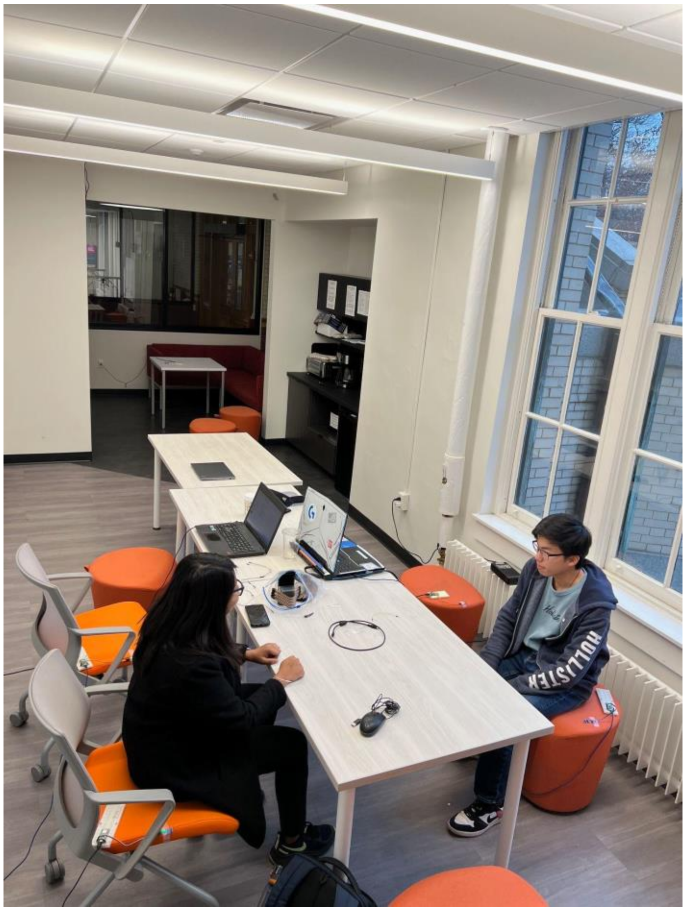
B.2 PH A1 Scenario 1



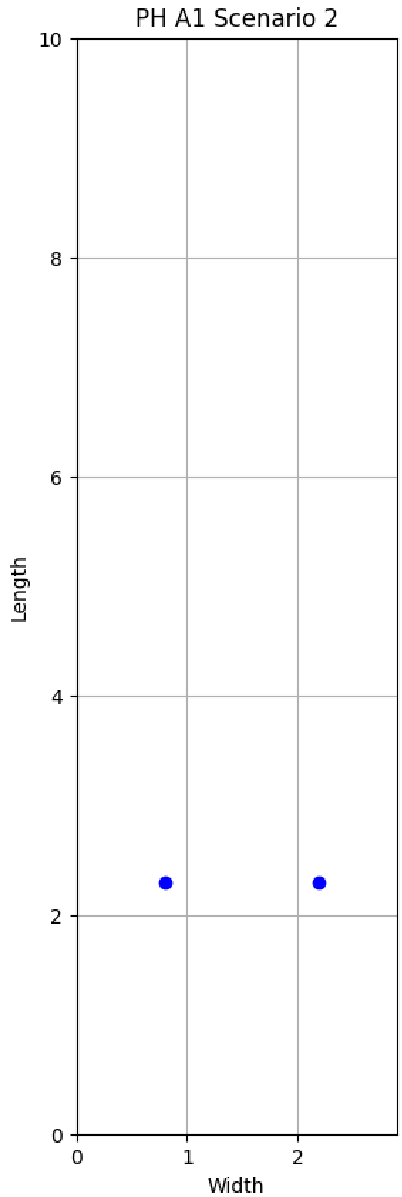
B.3 PH A1 Scenario 1 Occupancy Map



B.4 PH A1 Scenario 2

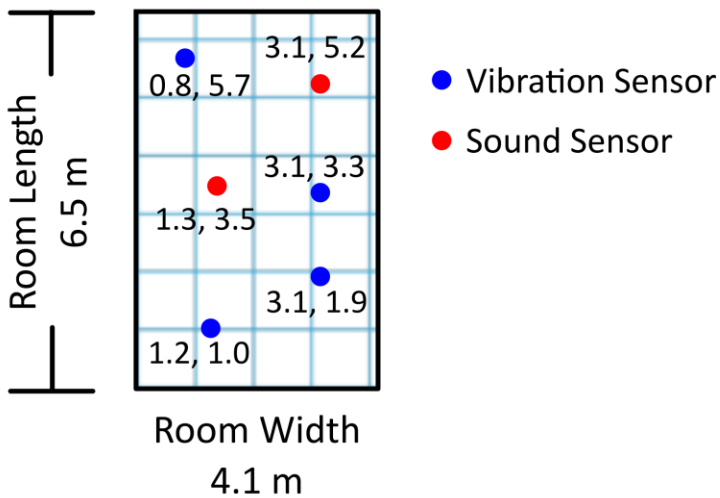


B.5 PH A1 Scenario 2 Occupancy Map



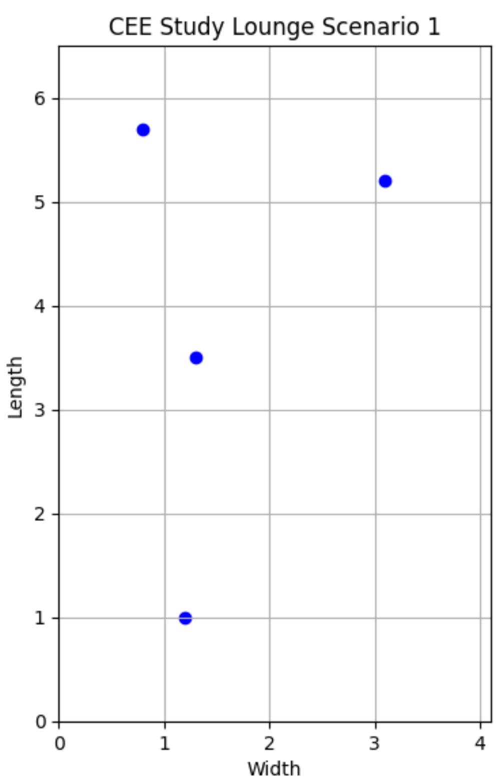
Appendix C. CEE Study Lounge Scenarios

C.1 Diagram of Sensors in CEE Study Lounge

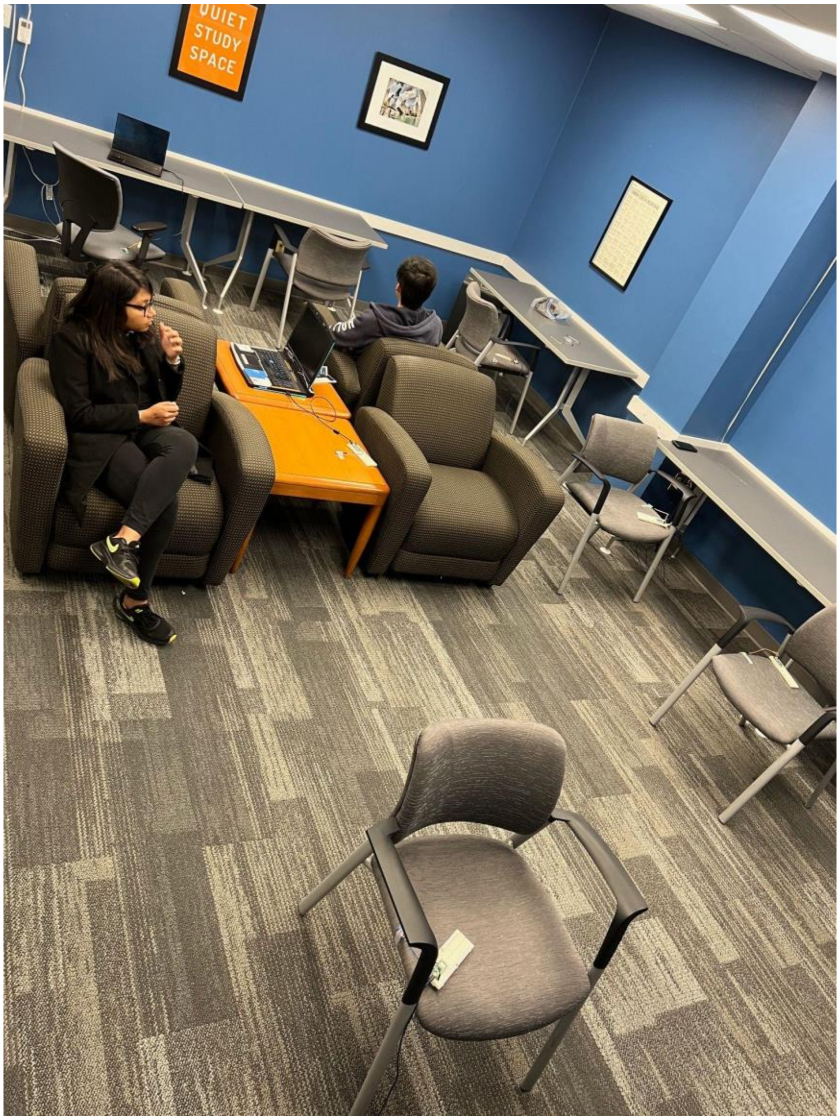


C.2 CEE Study Lounge Scenario 1

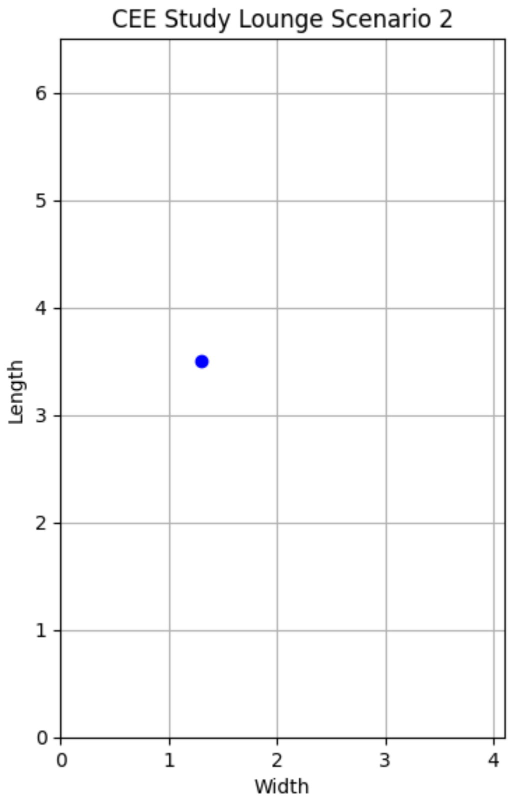
C.3 CEE Study Lounge Scenario 1 Occupancy Map



C.4 CEE Study Lounge Scenario 2



C.5 CEE Study Lounge Scenario 2 Occupancy Map



Appendix D. Historical Data Time Series Graph example