ELEC4848 Senior Design Project 2022-2023

Thapa Kushal (3035477082)

# Machine Learning and Data Analytics for Smart Water Auditing

Supervisor: Dr. E.C.H. Ngai
Second Examiner: Dr. X. Qi

# Machine Learning and Data Analytics for Smart Water Auditing

Kushal Thapa (3035477082)
Department of Electrical and Electronic Engineering
The University Hong Kong, Hong Kong

***Abstract -*** *Traditional methods of conducting water auditing involve the use of intrusive and bulky water flow rate sensors on water outlets, they are often costly and difficult to install which may discourage participants from joining water auditing projects. With the goal of solving this issue, this project aims to introduce a non-intrusive water use data collection method for smart water auditing. Time series data sets containing the water flow of different water outlets are used to train machine learning models to categorize the water flow events. The project has demonstrated the capacity to detect water flow events using a non-intrusive vibration-based IoT water collection system and streams the data to a cloud-based database through Wi-Fi. Feature extraction was then conducted on the water outlet dataset and used to train an SVM model and a Decision Tree Classifier. The results have shown a classification accuracy of around 73%. Differentiation between some outlets such as the washroom basin and kitchen basin have a lower classification accuracy, this suggests that some water outlets share similar patterns and features and are more difficult to distinguish than others. Overall, this project has shown that the non-invasive approach to water data collection has the potential to be implemented in future large-scale smart water auditing projects.*

## 1. Introduction

During the past two decades, domestic water use per capita in Hong Kong has increased significantly [7]. This trend goes against the international norm for developed cities. Currently, there is a lack of data regarding the water consumption usage patterns of residents throughout Hong Kong's households. Acquiring such water consumption data is the first step towards any effort for water use analysis and water conservation efforts [5]. To combat this issue, efforts to launch a water auditing project have already been initiated and are in the early development stage [2]. In a similar vein, the goal of this research is to build and deploy an Internet of Things (IoT) system to carry out water auditing using water use information gathered from a household and to train a generalized machine-learning model capable of categorizing the end-use type of the water expenditure event. The main distinction of this project is the emphasis on a non-invasive data collection method. Traditionally, water usage data have been collected with a water flow rate sensor which often has the disadvantage of being bulky and intrusive, while this project's approach is to use small-form microcontrollers to monitor the vibration of the water flow from water outlets.

### 1.1 Objectives and Scope of Investigation

The first objective of this project is to build a non-intrusive IoT water collection system capable of accurately logging the water use activity of a household. The goal of the IoT system is to satisfy the following conditions:

- The IoT system should be capable of detecting vibrations.
- The IoT system should be capable of determining if the water flow of a water outlet is running or not based on the vibration levels.
- The IoT system should be capable of storing the water status data and timestamp data in a retrievable database.
- Installation of the IoT system at a water outlet should not take more than 10 minutes.
- The IoT system should be capable of withstanding hot, wet, and humid environments.
- The IoT system should be stable for at least 24 hours.

The second objective is to build a machine-learning model capable of distinguishing end-use water activity based on the household's water meter readings. Ultimately, the goal of this project is to determine if our non-invasive approach to water auditing is truly a viable option and whether it can yield results with a degree of accuracy comparable to those of conventional water auditing techniques.

Due to limited time and resources, the scope of our investigation will primarily be on a single household, and the data collection period will be designated for five to eight weeks. Despite these limitations, we believe the findings of this investigation will serve as proof of concept for a non-intrusive approach to water auditing.

### 1.2 Report Outline

This paper is organized into several sections. Section 2 covers the theoretical principles of our approach to data collection, the design of the system architecture of the IoT system, and a description of the machine learning process. Section 3 covers the detailed implementation of the non-intrusive IoT water data collection system. Section 4 covers the detailed process of training the machine learning model. And Section 5 summarizes the project's work.

## 2. Methodology

The collection of water data using IoT technology requires the integration of microcontrollers, sensors, and software to function harmoniously to collect, process, exchange, and transmit data to a cloud-based platform over the Internet.

### 2.1 Theoretical Principles Behind Data Collection

The adoption of a non-intrusive technique although uncommon is not entirely new, [1] has shown that a building's metal pipe network can provide a suitable environment for precise water consumption tracking by using a set of sensors and signal processing techniques called a "nonintrusive load monitor for water" (WaterNILM). In this project's approach, sensors will be deployed to measure the vibration levels at various water outlet locations, such as at the neck of the flexible piping of a water outlet. These sensors will mark down the timestamp, amplitude of the vibration, and whether the water flow is on or off. However, vibration measurement alone is not sufficient enough to determine the water flow rate at these outlets. Therefore, in combination with these sensors, another data logger system needs to be set up at the water bill meter, which will record the instantaneous water flow rate of the household as a whole. Such a water bill meter logger was initially planned to be provided by the project supervisor. Then, by comparing the timing of the timestamps of these two sets of measurements, the water flow rate of a given water outlet can be acquired at a given time. Once sufficient data has been acquired, machine-learning models will be trained to identify patterns in water use and classify them according to their end-use types. There are two key phases to this project: the construction of the non-invasive IoT water data collection system and the training of machine learning models to classify the water flow events.

The IoT water collection system will be installed in a household with 5 types of water outlets: The kitchen basin, the washroom basin, the washing machine, the showerhead nozzle.

### 2.2 System Architecture of the Vibration-based Data Collection Method

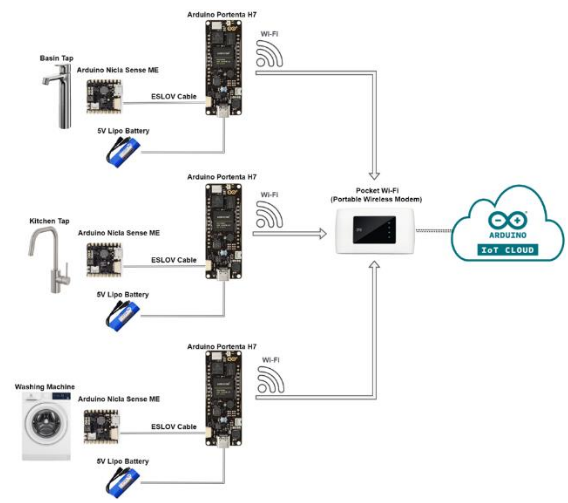



*Figure 1: System Architecture of the Vibration-based IoT Water data collection system.*

The vibration-based data collection method illustrated in Figure 1 will be utilized for all the water outlet. This setup mainly involves the implementation of two microcontrollers, the Arduino Nicla Sense Me and the Arduino Portenta H7. The Arduino Nicla Sense Me acts as the hardware that houses the vibration sensor, it will simply relay the "vibration amplitude" data to the next microcontroller. The Arduino Portenta H7 is the main component that will analyze the "vibration amplitude" data gathered from the Nicla Sense Me by running algorithms to determine the water flow status, the microcontroller will then transmit the relevant data to a cloud-based data storage system using Wi-Fi.

The Nicla Sense Me sensors are mounted to suitable locations for each of the outlets. For the kitchen basin and washroom basin, the sensor will be attached to the neck of the flexible pipe underneath the sinks, this is to ensure minimum unwanted vibrations are caused to the sensors by the users. These sensors sense vibrations caused by water flowing through the pipes. For the washing machine, the Nicla Sense Me sensor is fitted to its main body frame to detect vibrations during wash cycles. For the showerhead, the sensor will be installed to the pipe of the cold-water provider that is attached to the boiler, since cold water is often used in conjunction with hot water.

The Nicla Sense Me is connected to the Portenta H7 through an ESLOV cable. This cable serves two purposes, it transmits the data between the two boards, and it also supplies power to the Nicla Sense Me board from the Portenta H7 board. Thus, only a single power source is required for both the microcontrollers. The Portenta H7 has two input ports for the power source, an ACH port, and a USB Type-C port. It was decided that the USB Type-C port be used to provide power to the board since there had been difficulties in procuring a charger for an ACH head battery due to its rarity in the market. The board will be powered by a 5V Li-po battery with 20000maH, calculations estimate that optimally the battery will last for around 50 hours. The Portenta H7 would then be connected to a portable wireless modem device through Wi-Fi to send the sensor data to the Arduino IoT could service through Message Queuing Telemetry Transport (MQTT), where it can be retrieved as a CSV file.

### 2.2 Supervised Machine Learning

The data from the water outlets are considered as labeled data, the technique of training machine learning models based on labeled data to make predictions or classifications is commonly known as supervised learning. The goal of supervised learning is to use the labeled data to train a model to recognize patterns and relationships between input and output data, such that it can then classify new, unseen data into the correct category. In this project's case, the goal would be to categorize a water flow event into one of the following water outlets: kitchen basin, washroom basin, shower nozzle, or washing machine. There are various types of supervised machine learning models, this project will mainly focus on using the Support vector machines (SVMs), and Decision Tree Classifiers.

## 3. Non-Intrusive IoT Water Data Collection System

The Arduino Nicla Sense Me device is a microcontroller capable of measuring process parameters such as temperature, humidity, and movement. Its small form factor makes it suitable for our non-invasive approach to data collection. Additionally, due to its capacity to function in humid conditions with humidity levels ranging from 0% to 100% and its ability to withstand temperatures between -40°C and +85°C, it is appropriate for data collecting in hot and/or wet environments like the kitchen sink and bathroom sink. The accelerometer sensor measures the acceleration of the microcontroller in the x, y, and z planes. The project's goal is to use these acceleration readings to detect the vibrations that would occur when the water tap is on, and then log the timestamp when these water on/off events occur.

After flashing a relay code into the Nicla Sense Me, the Portenta H7 would be able to get the acceleration measurements from the Nicla Sense Me using I2C communication through the ESLOV cable. To set up the Wi-Fi transmission function of the Portenta H7, an Arduino IoT Cloud account must be created. Then, a web-based server will serve as the "Host" for some preset variables for the Portenta H7. The Portenta H7 is then capable of sending its request to the web server through MQTT using its authentication key to set up a "Host" and "Client" relationship. Through this connection, the microcontroller is capable of updating the values of the preset cloud variables. The latest result of the cloud variables can be viewed using a dashboard, as illustrated in the following Figure 2:

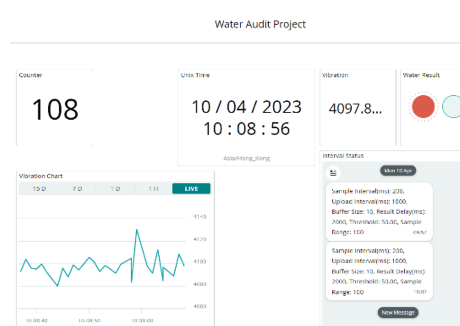



*Figure 2: Dashboard of Cloud variables from the IoT water collection system*

### 3.1 Design of Main Software Program

The programming code responsible for most of the vital processes of the IoT water collection system is within the Portenta H7 module. The following Figure 3 shows the stages of the program, which will also be indicated by the RGB LED status of the device. After being powered up, the program will first try to connect to the Arduino IoT Cloud System. Once the connection is established, it will wait for a period of ten seconds to give the users enough time to walk away and not disturb the vibration sensor. This is because the next stage will start reading the vibration measurements to calculate the base acceleration of the water outlet. In the next stage, the user is expected to turn on the water flow of the water outlet. This is because, during this stage, the program samples the vibration measurement to adjust the threshold level. Then, in the last stage, the program will continuously run the code to determine the status of the water flow outlet and will update the cloud variables accordingly.

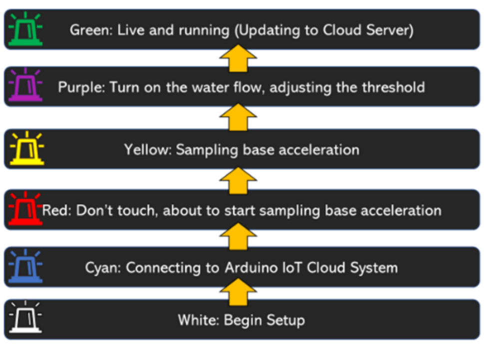



*Figure 3: RGB LED Indicator of Current Stage of Process*

The Nicla Sense Me relays the accelerometer readings to the Portenta H7 when prompted, it returns the acceleration of the device on the x-axis, y-axis, and z-axis. Tests have shown that the acceleration value of these three dimensions all rest on different base values even when the sensor is stationary. It would add an extra layer of unnecessary complexity if the program were to treat the acceleration values from each of these dimensions separately. Therefore, to simplify future calculations, the program would calculate the vector value from the three scaler values. The magnitude of this vector value would represent the "vibration amplitude" of the device. This value is calculated with the following equation:

$$\mathrm{MV} = \sqrt{[(xAxis)^2 + (yAxis)^2 + (zAxis)^2]}$$

Experiments have shown that the base value of the "vibration amplitude" (MV) when stationary would be different for different water outlet locations and Nicla Sense Me devices. In the program, it is necessary to know the base value of MV as this would be the value that would represent when the water flow event is off. Let us denote this resting base value of MV as B. The B value would be used in the calculation process of detecting a water flow event. For instance, let there be a threshold value Th, such that if the current MV is greater than (B +Th) or if the current MV is less than (B-Th) then the water flow would be considered as on. In the setup() function where the initialization stage begins, the following pseudo-code is implemented to record the B value:

```
Void setup():
        delay(10 seconds)
        for i in range 100:
                VM_sum += getMV()
                delay(1 second)
        B = MV_sum/100
```

*Figure 4: Pseudo code to record base value.*

The program collects and sums up 100 samples of the magnitude of the accelerometer's vector with an interval of one second between each iteration. Then the B value is calculated by getting the average of the summed-up samples.

The next objective of the program is to find the threshold (Th) value, experimental results have found that the Th value is unique to each water tap outlet and should be tuned accordingly. The device will start sampling the current MV value to adjust the default Th. During this stage, the user is expected to turn on the water flow of the water outlet. The program will sample the MV for 50 seconds, and record the largest MV value during this period. Then, if this maximum value is larger than the maximum value recorded during the base value collection stage by 10 (adjustable), then it would update the Th value. The Th value would be updated to be one-third of the maximum MV value. In scenarios wherein the user is not able to turn on the water flow of the outlet, the program will use the default Th value which is tuned to satisfy most water outlets. This also ensures that if the device were to unexpectedly reset when a user is not present, it would still be able to function to some degree.

After calculating the Th value, the next stage of the program is to determine the status of the water flow. The program's initial approach to determining whether the water flow outlet is on or off was using the following pseudo code:

```
Void loop():
        MV = getMV()
        if (MV > B+Th or MV < B-Th):
                water = On
        else:
                water = Off
        delay(1 second)
```

*Figure 5:Pseudo Code for Water Flow Status (Version 1).*

The issue with this method is that the MV value is a measurement of vibration, so it is an oscillating value and the MV value may fall into an "off" case even when the water flow is on. So, a new algorithm was written, with the idea of having an "off" counter, where the program would only consider a water event as "off" if the MV value has been under the boundary of the Th consecutively for n number of times. The following is a pseudo-code for the new algorithm:

```
Void loop():
        MV = getMV()
        if (MV > B+Th or MV < B-Th):
                water = On
                offcounter = 0
        else:
                offcounter++
                if (offcounter > n):
                        water = Off
        delay(1 second)
```

*Figure 6: Pseudo Code for Water Flow Status (Version 2).*

The following Figure 7 is the resulting graph from the implementation of the new algorithm:

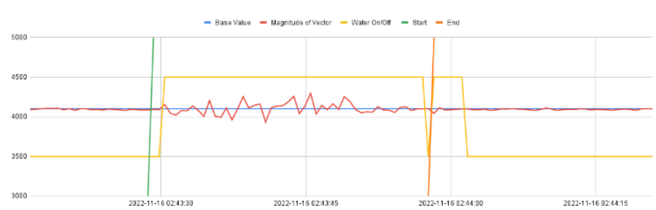



*Figure 7: Water Result Status during a single Water Event.*

The Start and End vertical lines represent the real start and end time of the water flow event measured by hand. It can be noticed that after the End event, there is a delay for the output result to indicate an "off" event. This delay is due to the implementation of the "off" counter, the program currently would have to wait for n "off" case samples before it truly outputs the result as "off", resulting in a delay of (n × sampling period). Our solution to this conundrum is to introduce a buffer queue of n size such that it stores the output value of the past n iterations. Then, when the "off" counter indicates that the water flow is "off", each of the output values in the buffer will be assigned the "off" result. In other words, our program waits for n iterations before determining if the output result is "on" or "off", so in real time users will get the result of the current water output after (n × sampling period). The following Figure 8 is the pseudo code implementing this solution:

```
Initialize: buffer[n] = {off, off, … ,off}
Void Loop():
        After every sample Interval:
                MV = getMV()
                buffer.enqueue(On)
                if (MV > B +Th or MV < B – Th):
                        offcounter = 0
                else:
                        offcounter++
                        if(offcounter > n):
                                resetOffcounterToOff()
                waterResult = buffer.pop()
```

*Figure 8: Pseudo Code for Water Flow Status (Version 3).*

The following Figure 9 shows the result of implementing this algorithm. Notice that there is no delay in responding to either the rise or fall of the water flow event. This is because the water output result is uploaded with its corresponding correct timestamp to the cloud database.

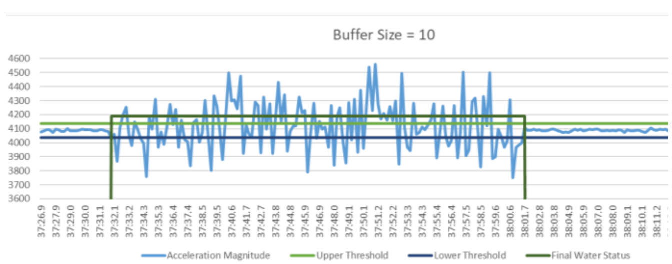



*Figure 9: Water flow status output during a water on event.*

### 3.2 Design and Construction of the Container for the IoT Devices

The microcontrollers of the IoT devices are required to operate in wet, hot, and humid environments and are also required to be able to be quickly deployed in a convenient manner. Without proper care, the equipment may suffer damage and be prone to error.

This necessitates the construction of a robust series of cases/containers for the microcontrollers such that they are waterproof, capable of being installed efficiently, and sturdy enough to be protected from physical damage, such as impact or exposure to moisture or dust. Furthermore, cases can provide a convenient way to organize and arrange the microcontroller and its components. A well-designed case can make it easy to access ports and connectors, and can help to keep cables and wires organized and tidy.

The use of laser cutters was adopted to create the necessary cases for our IoT devices. A laser cutter is a highly accurate and adaptable device used to cut or engrave materials using a powerful laser. It uses a computer-controlled device that can produce highly accurate complicated designs and precise cuts to direct the laser beam. It is able to cut materials such as acrylic, wood and metal. Acrylic was chosen for the design of the IoT device cases. The following figures show the designs used to create the cases. These designs were produced using the Corel software and are in the STL file format.

The design for the Portenta H7 Module consists of four layers of acrylic, the following Figure 10 shows the final form of the case with the Portanta H7 inside.

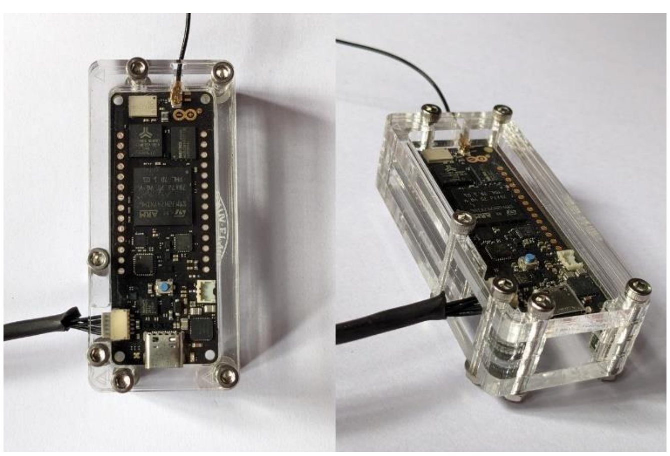


*Figure 10: Portenta H7 with case.*

Compared to the Portenta H7 case, the design for the Nicla Sense Me case had an extra layer of complexity. The Nicla Sense Me is at the frontal point of contact with any water outlet as it houses the sensor for the vibration reading, therefore it must be thoroughly waterproof. It also has to be designed in a manner where it can be easily mounted to the contact point of the water outlet. With this in mind, its STL design has an additional layer of acrylic meant as an attachment mount to a metal clamp. This layer would be bolted to a metal clamp and then the metal clamp would serve as a convenient way for the user to attach the sensor to the contact point of a water outlet. The vibrations from the water outlet would then travel through the clamp to the sensor, experimental results have shown that even minor vibrations of water flow can be detected using the clamp. The following Figure 11 shows the final form of this Nicla Sense Me case.

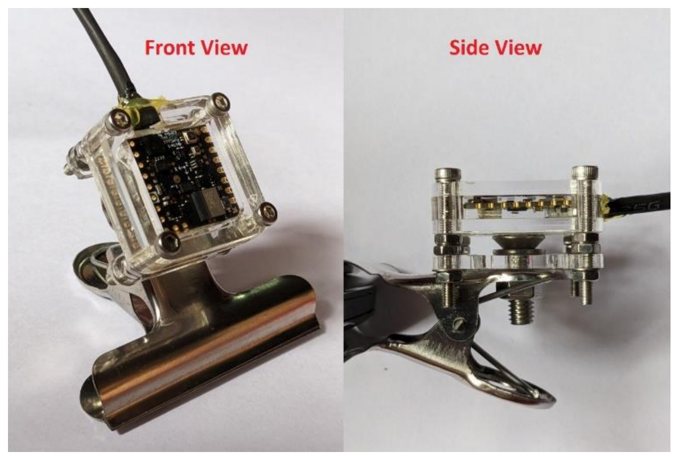



*Figure 11: Nicla Sense Me with case.*

At the top corner of the Nicla Sense Me in Figure 11, the ESLOV cable is attached via the port gap. This is the only gap in the acrylic case, and to ensure that the case is waterproof, extra measures have been put in place. The ESLOV cable itself is wrapped with a heat shrink tubing, this insulates the electrical components from external factors such as moisture, dust, abrasion, and sharp objects that might otherwise damage the wire. Next, the gap between the cable and microcontroller at the port is filled with a waterproof adhesive using a hot glue gun. This has ensured an airtight seal of the microcontroller within the case and made it suitable for use in wet environments such as the shower room.

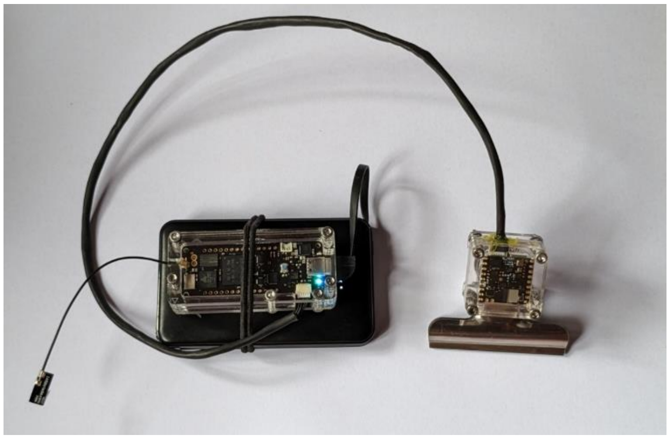


*Figure 12: IoT System with cases installed.*

Figure 12 shows the complete IoT system setup with the cases installed, the Portenta H7 (on the left) is powered by a power bank through the USB Type-C port. Then the Portenta H7 is attached to the Nicla Sense Me (on the right).

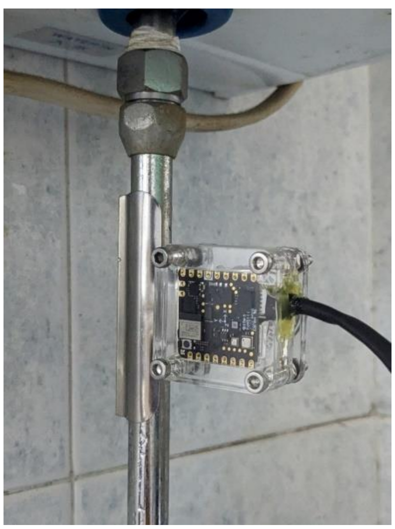


*Figure 13: IoT system attached to a shower pipe.*

Figure 13 showcases the IoT system being attached to the pipe that supplies the cold water to the shower nozzle. Experimental results have shown that the best results are obtained by the sensors when it is clamped onto sections of the pipe where it is either bent or near a junction point since these locations are more prone to turbulence.

### 3.3 Analysis of the Result

The algorithm in Figure 8 can be further optimized by analyzing and tuning the following parameters: Th, sample interval, and the buffer queue size (n).

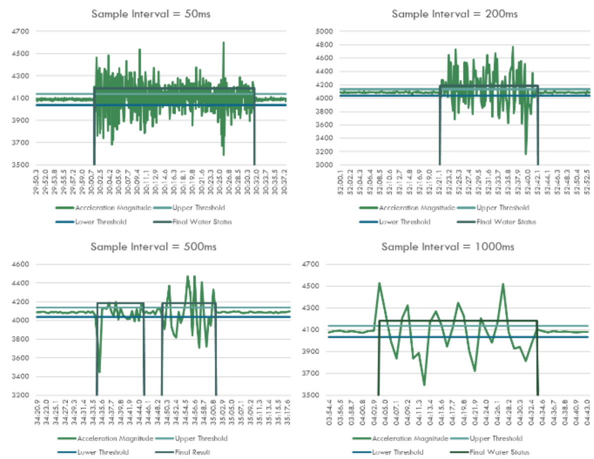



*Figure 14: Water status results with different sample interval values.*

The sample interval value is responsible for controlling the time interval of the data transfer between the Portenta H7 and the Nicla Sense Me. Figure 14 illustrates a series of graphs with different sample interval values. On the top left of the figure, the sample interval is set to 50 milliseconds. Although this would lead to a detailed accumulation of data resulting in a higher accuracy rate and faster response rate, it would also require more processing power and memory. On the other hand, having a high sample interval has been shown to result in gaps of false water-off outputs, as illustrated on the lower left graph. Experimental results have shown that a sample interval of around 200 milliseconds is a suitable value that strikes a balance between accuracy and resource management.


*Figure 15: Water status results with different buffer size values.*

The buffer size is responsible for determining if the current low vibration level is truly caused by a water-off event or if it is simply between oscillations. Its value essentially determines how much leeway to give to a water-off event. Figure 15 shows three graphs each with a different buffer size. The top graph has a buffer size of 2, such a low buffer size results in gaps of false negative outputs. On the other hand, having too large of a buffer size would have the risk of not being able to recognize short-duration water-off events. For instance, assume the sample interval is 200 milliseconds, and the water outlet was to be turned off and on with a 10-second gap. With a buffer size of 50, the algorithm would not be able to detect the water-off event, since it is within the buffer counter value. Therefore, a suitable buffer size is around 10, it would be capable of detecting any water-off events longer than 1 second.

## 4. Training of the Machine Learning Model

In the machine learning phase of the project, a time series data set of the water flow at different water outlets was used to train a supervised machine learning model. Initially, the plan of the project was to use the data gathered from the IoT system for the training of the machine learning model. However, due to time limitations, the water bill meter logger was not available during the data collection phase of the project. Without the water bill meter data, the binary data collected from the IoT system would not be able to produce the time series data for machine learning. The project will alternatively use an open-source data set as a substitute. The data set is obtained from “Water End USE Dataset and TOols (WEUSEDTO)” [8], The data set consists of data collected over a period of one year. The residential apartment being monitored consists of seven water fixtures, with one person residing in the apartment. Data is collected at the fixture level with a resolution of 1 second. In order to simulate the data that would have been produced by the IoT system, only the data set for the washroom basin, kitchen basin, shower, and washing machine have been used.

### 4.1 Data Separation into Events

The time series data set for the various water outlets each span across an entire year, this data set must first be split into individual events. An event can be defined as a section of time between when the water flow is turned on and off. For instance, if an individual were to wash their hands by first wetting their hands, then turning off the faucet to lather soap then washing off the soap, then this would result in two flow events.

Within the code the start of an event would be indicated by comparing the flow value to a flow threshold, if the flow exceeded the threshold, then it would mark the start of an event. Similarly, when the flow dipped below the threshold it would mark the end of an event. When analyzing the data set, there were sections with missing rows, this meant that certain time periods were unavailable. These gaps in time meant that the program could not solely depend on the flow value to detect an end event. The time gap between the current and next rows had been calculated and stored in an array, if this time gap exceeded 5 seconds, then this would also mark the end of an event. The following figures show flow events from each of the water outlets:

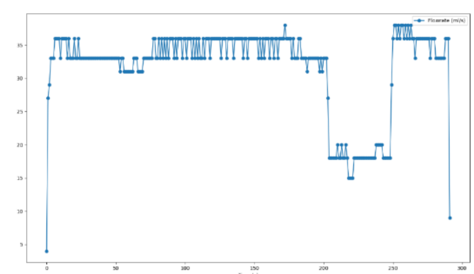


*Figure 16: Water Flow Event of Kitchen Basin.*

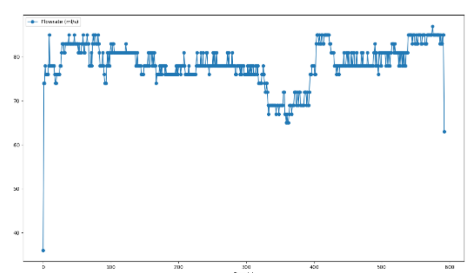


*Figure 17: Water Flow Event of Shower Outlet.*

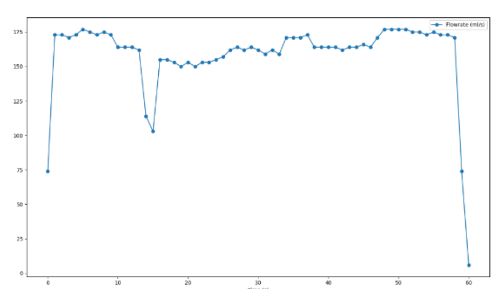


*Figure 18: Water Flow Event of Washing Machine.*

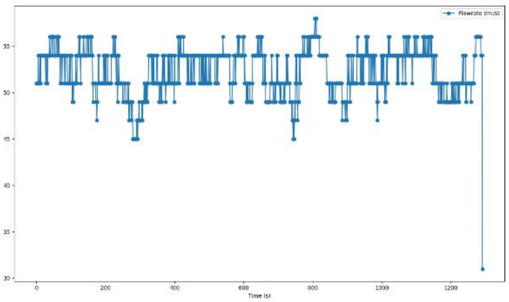


*Figure 19: Water Flow Event of Washroom Basin.*

### 4.2 Feature Extraction from Events

Feature extraction is a common technique in machine learning, it is a process of finding and transforming relevant and useful information from an otherwise messy data set into a more compact representation. Essentially, it is the process of extracting the most important features of a data set. [9] Shows that a common and useful set of features to extract for water end use disaggregation are the following: The average flow rate of an event, the total volume expended in an event, and the duration of an event. The following figures showcase the 3D plots from the extracted features, each dot represents a flow event. Each of the axis in the plot represents one of the features.

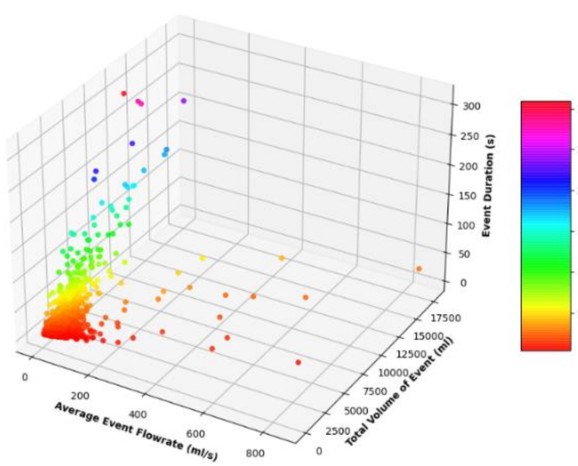



*Figure 20: 3D Plot of Kitchen Basin Feature.*

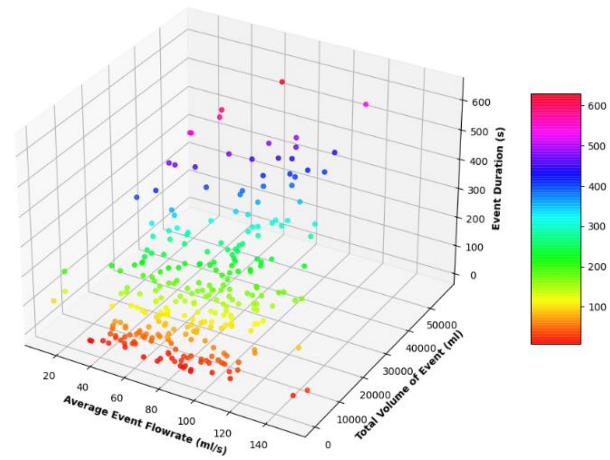



*Figure 21: 3D Plot of Shower Outlet Feature.*

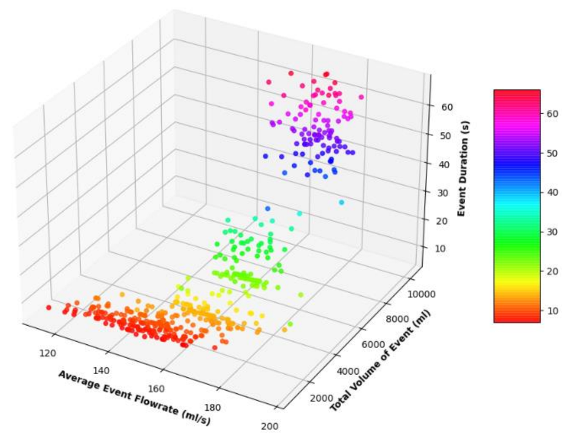



*Figure 22: 3D Plot of Washing Machine Feature.*

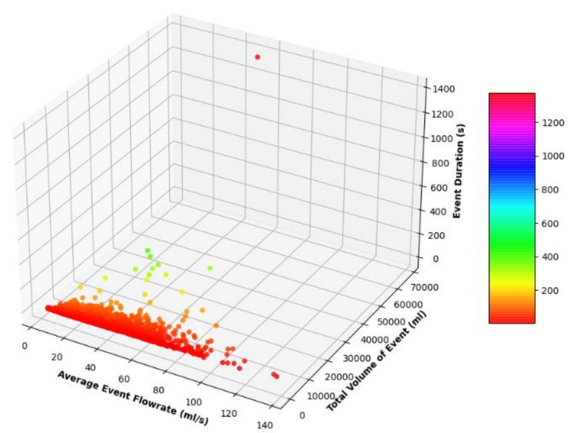



*Figure 23: 3D Plot of Washroom Basin Feature.*

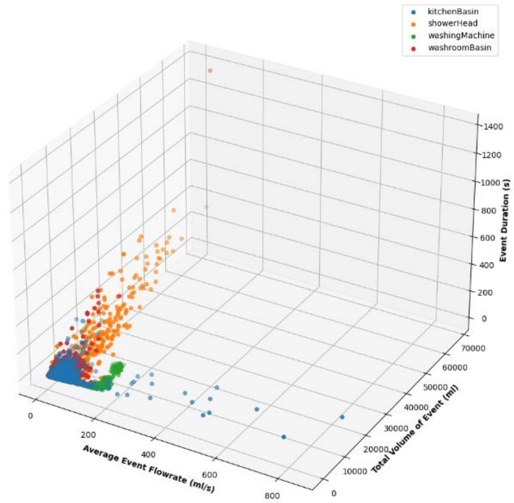



*Figure 24: 3D Plot of the features of all the outlets.*

### 4.3 Data Cleansing

The 3D plots in section 4.4 reveal that there are quite a number of data points that are outliers, training machine learning models with misleading data would result in inaccurate classification. Therefore, it is important to clean the data set of any outliers. In this project, the outliners are detected by calculating the Z-score of each value in the column feature, relative to the column mean and standard deviation. Then if the absolute Z-score value is above a certain threshold, it will remove that particular row from the data set. The following Figure 26 shows the 3D plot of the features after the data set has been cleansed using this method. Then, 20% of the data set was split for the testing data set and 80% of the data was split for the training data set.

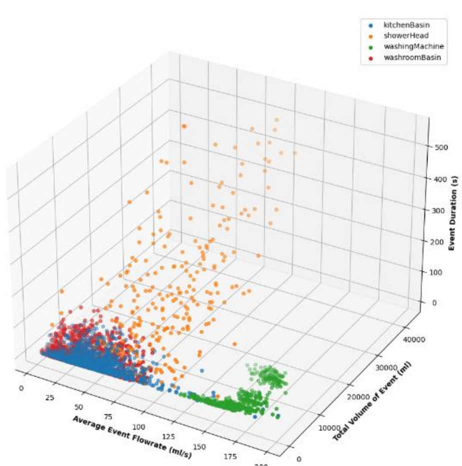


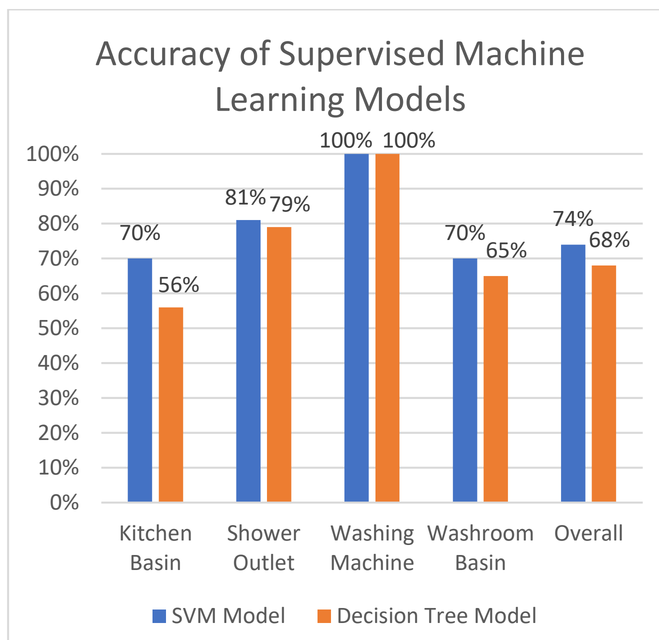

*Figure 25: 3D Plot of the features of all the outlets (After Cleaning the Data Set).*

### 4.4 SVM and Decision Tree Model

SVM is often used for classification and regression analysis, it essentially functions by finding a suitable hyperplane that best separates the data points into different classes. In this project's case, it would separate the feature data points with a hyperplane that best categorizes each cluster. In a linear implementation of SVM, the hyperplane would be a flat plane, however the SVM could also implement the kernel trick to transform the data set to a higher-dimensional space, which essentially creates a non-linear hyperplane. In this project, both linear and non-linear implementations were attempted. The linear implementation had shown to provide better performance.

The decision tree model essentially functions by creating a series of flowchart-like decision trees. Each node would denote a test on an attribute, the branches would represent the output of the test, and then the terminal node would determine the classification label.

The following figures showcase the results of the supervised machine-learning models.




### 4.5 Analysis of Results

Overall, the results of the machine learning model have shown a positive performance in classifying the water flow events. However, the accuracy rate of the shower outlet and washing machine have far outperformed that of the kitchen basin and washroom basin. If only the shower outlet and washing machine is to be considered, the SVM model would have an average accuracy of 90%. This suggests that the features and data pattern of the kitchen and washing basin are quite similar, and differentiating between these two outlets would prove to be difficult. However, this is to be expected as generally these two basins often share similar functions in the household and could be categorized into the same type of outlet.

Another point to consider is that the pattern of water consumption in this household may be unique to the individuals living in the household or to the outlets of this household, and the model trained to recognize such patterns may not be applicable on mass to the general public. To address this issue in the future, more data collection IoT systems could be applied to a larger number of households such that the project would have more water use data to work with, enabling it to find the commonality in water use patterns across different households. Thus, the issue of overfitting the model would be negligible.

## 5. Conclusion

The central goal of this project has been to determine whether the adoption of a non-intrusive approach to water data collection is indeed practical for smart water auditing. In this veil, a vibration based IoT water data collection system has been built. The IoT system has successfully fulfilled all of the following objectives: It is capable of detecting vibrations and correctly determine the water flow status of an outlet, it is capable of transmitting the water activity data to a cloud database, it is able to be installed within a few minutes, and it can perform in hot, wet and humid conditions. Despite the difficulties previously mentioned of collecting the water flow time series data set, the substitute data set has proven to be useful in determining the validity of training a machine learning model by performing feature extraction. The SVM and Decision Tree training models have shown an average accuracy of around 73%, and around 90% if the kitchen basin and washroom basin were to be considered as the same category. In summary, this project has demonstrated that the non-intrusive method of collecting water data has the possibility of being used in future smart water auditing initiatives on a wider scale.

## References


[1] Schantz, J. Donnal, B. Sennett, M. Gillman, S. Muller and S. Leeb, "Water Nonintrusive Load Monitoring," in IEEE Sensors Journal, vol. 15, no. 4, pp. 2177-2185, April 2015, doi: 10.1109/JSEN.2014.2372053.

[2] M. -H. Luk, C. -W. Yau, P. W. T. Pong, A. P. Y. Lee, E. C. H. Ngai and K. -S. Lui, "High-Resolution Tap-Based IoT System for Flow Data Collection and Water End-Use Analysis," in IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22822-22835, 15 Nov.15, 2022, doi: 10.1109/JIOT.2022.3187999.

[3] "Seawater for Flushing." WSD - Water Conservation - Seawater for Flushing. Water Supplies Department, The Government of the HKSAR. Accessed January 15, 2023. https://www.waterconservation.hk/en/at-school/secondary-school/water-resources-in-hk/sea-water-for-flushing/index.html.

[4] Amphiro AG. (n.d.). The smart way to shower - Amphiro AG. Retrieved January 15, 2023, from https://amphiro.com/assets/documents/Amphiro_Product_Flyer_en.pdf

[5] "Water Audits and Water Loss Control for Public Water Systems - US EPA." U.S. Environmental Protection Agency. Accessed January 15, 2023. https://www.epa.gov/sites/default/files/2015-04/documents/epa816f13002.pdf.

[6] Team, The Arduino. "Uploading Sketches over-the-Air (OTA): Arduino Documentation." Arduino Documentation | Arduino Documentation. Accessed January 16, 2023. https://docs.arduino.cc/arduino-cloud/features/ota-getting-started.

[7] Office of the Government Chief Information Officer of the Government of the Hong Kong Special Administrative Region. (n.d.). Per capita domestic fresh water consumption. Per Capita Domestic Fresh Water Consumption | DATA.GOV.HK. Retrieved March 15, 2023, from https://data.gov.hk/en-data/dataset/hk-wsd-wsd5-per-capita-domestic-fw-consumption

[8] Water-End-Use-Dataset-Tools. (n.d.). Water-end-use-dataset-tools/WEUSEDTO. GitHub. Retrieved April 11, 2023, from https://github.com/Water-End-Use-Dataset-Tools/WEUSEDTO

[9] Pastor-Jabaloyes L, Arregui FJ, Cobacho R. Water End Use Disaggregation Based on Soft Computing Techniques. Water. 2018; 10(1):46. https://doi.org/10.3390/w10010046