1. Background research summary

For phase 2 of the project, I needed to develop embedded application running on IOT device that will be reading from different sensors, establishing communication with Pelion device management platform and develop a classifier for recognising Human activity using the sensor values.

1.1 Sensor choices and understanding the readings

According to the user manual of the given board, there are 8 sensors available:

1. Microphone
2. Temperature sensor
3. Humidity sensor
4. Magnetometer
5. Accelerometer
6. Gyroscope
7. Barometer
8. Time-of-flight/gesture detection sensor

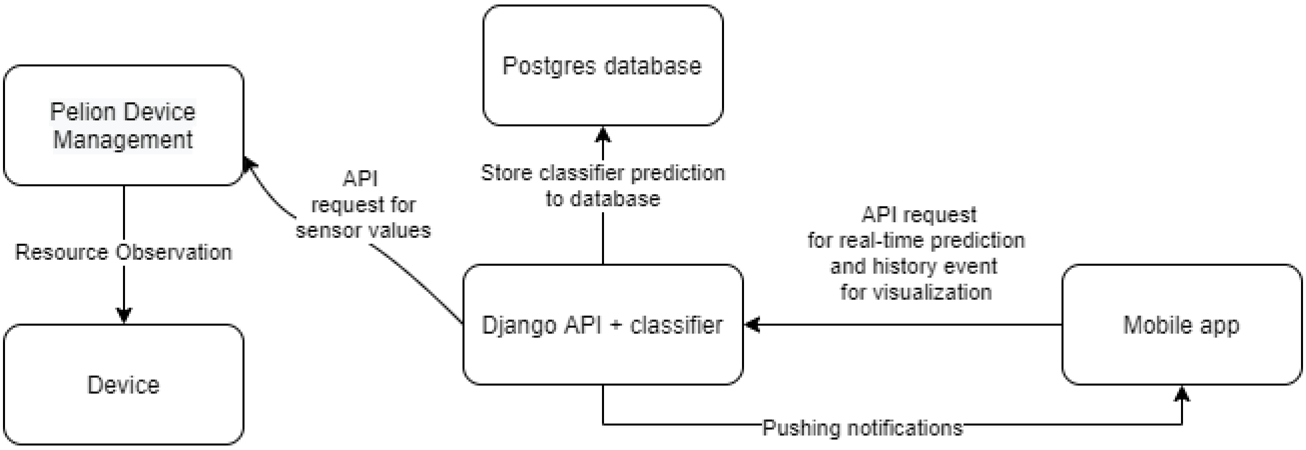
In one of the previous research[1] on various sensors for human activity recognition (HAR), it was shown that many of the researches on HAR where done using Accelerometer, Gyroscope and Magnetometer. On top of these three, temperature and humidity sensor reading may contribute toward HAR if we are measuring the human body temperature and humidity cause by sweat.

However, it is very clear that microphone is not needed as we are not recording any sound. Barometer only measure air pressure, which does not change much if we are at sea level. The time-of-flight sensor is more commonly used for obstacle avoidance on robot.

For the remaining 5 sensors, to process the sensor values correctly for our classifier, we need to first understand the unit/format of which the sensor values are presented on Pelion platform. This information was found on the user manual and sensor datasheets.[2][3][4]

Sensors	Reading unit
temperature	Temperature measure in Degree Celsius from 15 to +40 °C ± 0.5
Humidity	Humidity measured in relative humidity from 20 to +80% rH ± 3.5
Accelerometer	acceleration measured on each axis in milli-gs with format (x, y, z)
Gyroscope	angular speed measured in millidegrees per second with format (x, y, z)
Magnetometer	Measurement of magnetic field in mill-gauss with format (x, y, z)

1.2 Communication design



This full-stack communication design is derived from the requirement of the coursework and the tutorials available on Pelion Device management platform [5].

1.3 Machine learning model choices

Common machine learning model used for classifying task such as human activity recognition (HAR) includes, support vector machine (SVM), decision tree, k-nearest neighbours (KNN), Naïve bayes classifier and Multilayer perceptron (MLP), all of which has pros and cons depending on the usage scenario. In one of the studies[1], comparison on HAR classification accuracy were made between various models. It was found that performance of most classification models is similar when the model is trained and tested on using the sensor data collected from a single subject. However, when the model is trained using 9 different subjects, and tested on one other subject, NB, Random Forest (RF) and J48 decision tree gave the best classification accuracy. The latter scenario would be more realistic in normal usage as we will not be able to retrain a model for every single new subject.

This previous research was done with only accelerometer, gyroscope, and magnetometer sensors. Considering we have also had data for timestamp, temperature and humidity, the best classifier maybe different.

2. Embedded application

The embedded application developed in phase 2 is primarily based on the code we have available from lab3a and 3b. It is now capable of reading from all sensors for a chosen interval, these sensor reading are then sent over WIFI to Pelion device management platform by M2M communication.

To properly establish connection between device and Pelion device management platform, Mbed studio also requires an API key and certificate created on Pelion platform.

After successful client registering, the sensor readings can be view on Pelion device management platform on the chosen resource path e.g., resource path /3333/0/5704 for temperature readings. These resources will later be subscribed by the web app hosted on google cloud to access and classify the activity.

3. Machine learning models

For this phase of the project, I have developed 5 commonly used classifiers using Tensor flow[6] and Scikit-learn[7] including a linear support vector machine, artificial neural network with two hidden layers ([30,10], step=10000), decision tree, k-nearest neighbours (with k=3) and Naïve bayes classifier.

For these models, I used the dataset mentioned in the coursework, and randomly split the dataset into 70% training set and 30% testing set with random seed of 42.

In the dataset we have 'date' and 'time' available. However, it was not possible to train models with Datetime objects, the solution for this is to convert date into ordinal and storing hours, minutes, and seconds individually.

For example, '30/06/2017' would be stored as 736510 and '13:51:15:847724020' would be stored in 4 different columns (hours, minutes, seconds and millisecond) like 13, 51, 15 and 8477240.

The nanosecond component is omitted due to limitation of the datetime library in python and it is very insignificant anyway for our usage.

Testing set accuracy and training time are used to draw comparison between these models.

Model	**Accuracy**	**Training time(seconds)**
SVM	0.858	46
ANN	0.935	14.87
KNN	0.985	0.1
Decision tree	0.985	0.45
NB	0.958	0.01

Table 1. Dataset excluding date and time.

Model	**Accuracy**	**Training time**
SVM	N/A	N/A
ANN	0.515	21.4
KNN	0.875	0.22
Decision tree	0.996	0.26
NB	0.674	0.013

Table 2. Dataset including date and time.

We can see from the two tables, that decision tree performs the best for both including and excluding 'date' and 'time'. Whereas because feature column increased by 5. SVM was not possible to train on a local machine for a reasonable time. However, different SVM kernels may perform differently and that this the next step of the project.

References

1. souza, Warren & Kavitha, R.. (2017). Human Activity Recognition Using Accelerometer and Gyroscope Sensors. International Journal of Engineering and Technology. 9. 1171-1179. 10.21817/ijet/2017/v9i2/170902134.
2. www.st.com. (n.d.). B-L475E-IOT01A - STM32L4 Discovery kit IoT node, low-power wireless, BLE, NFC, SubGHz, Wi-Fi - STMicroelectronics. [online] Available at: https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html#documentation
3. UM2153 User manual Discovery kit for IoT node, multi-channel communication with STM32L4. (2019). [online] . Available at: https://www.st.com/resource/en/user_manual/dm00347848-discovery-kit-for-iot-node-multichannel-communication-with-stm32l4-stmicroelectronics.pdf
4. Digital output magnetic sensor: ultra-low-power, high-performance 3-axis magnetometer Datasheet -production data Features. (2017). [online] . Available at: https://www.st.com/resource/en/datasheet/lis3mdl.pdf
5. Pelion.com. Collecting data from devices - Collecting data from devices | Pelion Device Management Documentation. [online] Available at: https://preview.pelion.com/docs/device-management/current/resources/index.html
6. TensorFlow. tf.estimator.DNNClassifier | TensorFlow Core v2.4.1. [online] Available at: https://www.tensorflow.org/api_docs/python/tf/estimator/DNNClassifier
7. Scikit-learn.org. (2019). scikit-learn 0.22.1 documentation. [online] Available at: https://scikit-learn.org/stable/user_guide.html.