Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa

Internet das Coisas
MEI/MI/MSI/MCD
2021/22

Project

An IoT battery monitoring application for mobile devices

1. Introduction

Internet of Things (IoT) is being incorporated in all enterprise and personal monitoring systems. Thus, we need efficient and dependable algorithms and approaches to analyze the IoT sensor data to find useful patterns and insights.
The frequency of the upcoming data may be very fast. Sensors may read hundreds to millions of data per second. There are many application areas in the real world, like leisure, health, weather monitoring and forecasting, agriculture, enterprise maintenance, data centers, and others. Therefore, wrangling, analyzing, and grasping insights from data is important for multiple application sectors.
In this project we want to create an IoT application to monitor the battery of mobile devices. Several mobile IoT devices have been used, from which the data related to battery charge/discharge cycles have been collected to generate data sets.
In this work we want to analyze and display over time the performance of the various mobile device's batteries. All the output information must be displayed in intuitive dashboards, that allow users to easily view their device battery performance/data.
The project can be split in two stages. In the first an offline data set is used, whereas in the second stage and on-line data set will be employed. In the first stage the groups must:

1. Implement the IoT architecture using the Google Cloud Platform [3] [5].
2. Understand the offline data set and create a dashboard to display statistical results about each mobile device battery performance.

In the second stage the groups must:

1. Use the offline data set to train machine learning models able to:
   a. predict the remaining battery available time (when discharging).
   b. predict the remaining time until full charge is attained (when charging).
2. Understand the online data set and create a dashboard to continuously display the evolution of the battery charge/discharge level. This dashboard must also show the predicted remaining battery availability time and the time until fully charged as applicable (depending on whether the battery is charging or discharging).

The course planning that can be found in moodle suggests indicative timings for the development of each stage.

2. Datasets description

The work proposed in this project is supported by two datasets: the offline and the online datasets. Each dataset is represented in JSON format and comprises several battery charge/discharge cycles.

Each cycle in the offline data set is represented by a device ID (battery_ID) that allows identifying the battery, an identifier for the operation cycle (cycle_number), which represents the number of charges and discharges, a label representing the type of operation (charge or discharge), the ambient temperature measured during the operation cycle, a timestamp representing the beginning of the cycle, and arrays of measurements related with the battery cycle. Then, there is an array that represents the acquisition instant of each measurement related to the elapsed time from the beginning of the operation cycle. Finally, an elapsed time field represents the duration of the operation cycle. Figure 2 shows the representation of one operation cycle presented in the offline dataset.


```
" cycle_0_1 ": {
    "battery_ID": "2",
    "cycle_number": 1,
    "type": "charge",
    "amb_temp": "24",
    "date_time": "28 May 2021, 11:09:42",
    "voltage_battery": [
        0.23635618415267867,
        …
    ],
    "current_battery": [
        -0.003484426276184136,
        …
    ],
    "temp_battery": [
        23.372047504812105,
        …
    ],
    "current_load": [
        0.0,
        …
    ],
    "voltage_load": [
        0.003,
        …
    ],
    "time": [
        0.0,
        …
    ],
    "elapsed_time": 12.656000000000002
  }
```


Figure 1: Battery operation cycle representation in the dataset

The representation of the online dataset is similar to the offline dataset but does not include any time reference. Each group must simulate the time according to the data transmission period. This period must be configurable through the dashboard.

Both datasets are available in moodle. The offline.json file represents the offline dataset, while the online_1.json represents the online dataset for battery 1 (battery_ID 1)and the online_2.json corresponds to the second battery (battery_ID 2).

3. Project development

To answer the questions raised before, the groups must create an IoT solution according to the architecture shown in Figure 2.

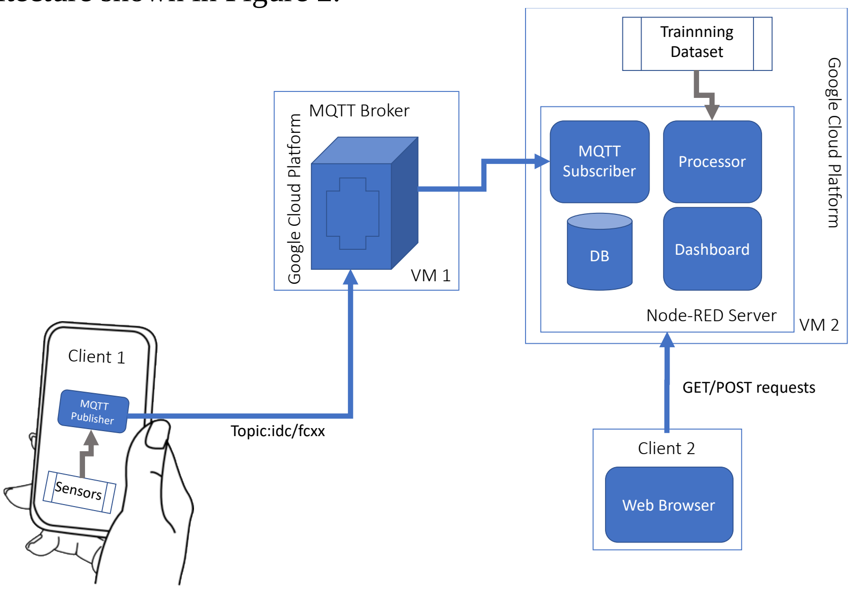

Figure 2: System architecture

This architecture is divided into four parts: an MQTT Broker, a Node-RED Server, and clients 1 and 2.

The MQTT Broker is a cloud machine that must be created by students, where the Mosquitto broker [1] must be installed, configured and executed. It should be configured to listen at port 1883 (default MQTT port) and to be accessible remotely.

The Node-RED [2] [4] server will be used by students to implement their solutions. It must be installed in a second virtual machine in the Google Cloud Platform.

For the first stage (offline part), the Node-RED application must read the offline dataset (offline.json), and process it to obtain and display the following information for the selected battery (user must be able to select one battery):

- Both the number of charge and discharge cycles.
- Both the average and standard deviation of the charge and discharge time.
- Range, average and standard deviation of the battery temperature, separately when charging and discharging.
- A chart displaying two lines simultaneously on the same axis, each for a complete charge/discharge cycle selected by the user. The two cycles will be selected with reference to two distinct timestamps, hence allowing the visualization of the battery performance decrease between the two selected timestamps.

For the second stage, based on the offline data set, the groups must select an appropriate machine learning technique to implement the two predictive models mentioned in the introduction, to:

a. predict the remaining battery available time (when discharging).
b. predict the remaining time until full charge is attained (when charging).

Additionally, the groups must use the online data set (online_1.json and online_2.json) to continuously send the battery sensors data from their own local computer to the cloud-based MQTT broker, similarly as the mobile devices would on a realistic scenario.

Finally, as the data arrives and is stored complementing the offline data, an online dashboard must be continuously updated to show a line plot of the current and previous battery cycle, as well as the output of the two predictive models implemented.

4. Report

Students are requested to prepare a report named IdC-2022-Project-FC-XX.pdf (where XX is your group number), in pdf format, to be submitted in the course Moodle page.
The report must:

- be elaborated according to the presentation rules of scientific articles, using the template that will be made available in Moodle;
- respect the minimum and maximum number of pages, 4 and 6, respectively;
- provide a description of the solutions implemented, describing the setup, communication and logic details adopted in the implementation. Additionally, describe the main challenges and difficulties encountered.

Furthermore, a compressed file named IdC-2022-Project-FC-XX.zip, containing the source code of the project as well as other documents that might have been used as references to carry out the project, must also be submitted through Moodle.

5. Discussion

In addition to the report, students will be required to demonstrate the solutions developed. As suggested, the dashboards must be intuitive to allow for a fast and accurate assessment of the work developed.

6. Delivery date

May 20th, 2022, until 23h59.
Instructions will be given on moodle.

7. References

[1] https://mosquitto.org/
[2] https://nodered.org/
[3] https://cloud.google.com/gcp/
[4] J. Cecílio, “Create an Internet of Things Application Based on Node-RED Framework”, FCUL, 2022, available at Moodle.
[5] J. Cecílio, “Create and Configure a Virtual Machine for Internet of Things applications on Google Cloud”, FCUL, 2022, available at Moodle.