An IoT battery monitoring application for mobile devices

João Marques
52769

Jóni Pereira
57855

Tomás Dias
57582

***Abstract*—Este documento é o relato completo de todas as etapas do trabalho prático da unidade curricular "Internet das Coisas" para o desenvolvimento da aplicação para monitorização de baterias de telemóveis. É feita uma breve introdução do projeto, e é explicado ao detalhe o desenvolvimento do projeto e os seus principais desafios. Por fim, são referidas as principais conclusões do projeto.**

***Index Terms*—iot, battery dataset, machine learning, node**

I. Introdução

Foi proposto na unidade curricular "Internet das Coisas" o desenvolvimento de uma aplicação para monitorização de baterias de telemóveis, focando-se principalmente no seu desempenho ao longo do tempo.

Para tal, fez-se o estudo rigoroso do enunciado do trabalho com atenção especial para o conteúdo dos datasets, para a arquitetura do sistema e para informação a ser apresentada ao utilizador.

Após isto, definiu-se as tecnologias a serem utilizadas, o ambiente de desenvolvimento e os métodos de trabalho.

Todo o processo de desenvolvimento encontra-se descrito de seguida.

II. Desenvolvimento do projeto

A. Preparação e Configuração

Primeiro que tudo, é aconselhado a visualização do conteúdo dos scripts shell (.sh) visto que contêm comandos importantes para a configuração do projeto.

Seguiu-se a arquitetura presente no enunciado, começando-se por criar as máquinas virtuais na Google Cloud Platform. A primeira máquina é a responsável por executar o Mosquitto broker e a segunda por executar o Node-RED Server bem como a base de dados.

Optou-se por utilizar uma base de dados MongoDB, dada a facilidade que esta apresenta em persistir documentos com vários campos e em realizar as operações necessárias (inserção, atualização, remoção, recolha).

É também de realçar que foi feita uma pequena alteração na formatação do dataset offline, de modo a que o processo de inserção na base de dados fosse mais simples.

B. Primeira Parte

Nesta etapa, o objetivo foi que a dashboard do Node-RED apresenta-se a informação relacionada com a bateria selecionada pelo utilizador. Todo este trabalho encontra-se no flow "Stage 1".

Primeiramente, definiu-se que o utilizador consegue selecionar uma bateria através de um HTTP GET ao endpoint: /battery/:[id da bateria].

A informação é apresentada com o auxílio dos nodes "text" do node set "dashboard". De modo a recolher esta informação, são feitas as queries necessárias à base de dados, através de um node "function". Utilizou-se o node set "node-mongodb" para comunicar com a instância da base de dados MongoDB.

Para apresentar o gráfico que relaciona os ciclos escolhidos pelo utilizador num respetivo intervalo de tempo, foi utilizado o node "chart". Os ciclos bem como intervalo de tempo são passados como queries ao endpoint descrito anteriormente (ver também o node ao qual foi denominado "HTTP - Params & Query"). Por fim, ainda foram utilizados dois nodes "function" aos quais foram atribuídos os nomes "Build Data - Cycle 1" e "Build Data - Cycle 2", para cada um dos dois ciclos, responsáveis por estruturar a informação passada ao node "chart".

C. Segunda Parte

No fluxo do Node-RED denominado "Flow 2", encontra-se o trabalho realizado para a segunda etapa do projeto.

Inicialmente, são criados e treinados os modelos de previsão tanto para o tempo de carregamento como para o de descarregamento da bateria. Este processo é feito pelo script em Python "models.py" que por sua vez gera dois ficheiros com os respetivos modelos guardados. Contudo, apenas o modelo de previsão de carregamento foi utilizado no Node-RED (isto deveu-se a questões relacionadas com o tempo, visto que a introdução dos dois modelos no Node-RED é feita de forma semelhante).

De modo a testar o modelo, criou-se o o script em Python "mqtt-pub.py" que implementa um cliente com função de publisher que envia os dados persistidos localmente para a máquina virtual com o broker instalado, utilizando o protocolo MQTT.

No servidor Node-RED, é utilizado um node "mqtt subscriber" que recebe os dados enviados pelo cliente. À medida que vai recebendo os dados, encaminha-os para o node "exec" que executa o script em Python "charge-predictor.py". Este script utiliza o modelo de previsão gerado anteriormente (neste caso, apenas o modelo de previsão do tempo de carregamento) e gera a previsão para os dados recebidos.

Por fim, esta previsão em conjunto com ID do ciclo recebido são apresentados na dashboard através de "text" nodes.

III. Principais Desafios

Aqui estão descritos alguns dos maiores obstáculos encontrados durante a realização deste trabalho prático:

- Sendo o Node-RED o principal ambiente de desenvolvimento do trabalho, a pouca documentação existente em relação a alguns nodes nativos, nomeadamente de exemplos de uso, dificultou a aprendizagem de uma tecnologia anteriormente desconhecida para nós. Também é de destacar o tempo de atenção dado ao funcionamento geral dos fluxos.
- Ainda no Node-RED, o uso necessário do node set de machine learning dificultou de forma considerável o trabalho a ser realizado dado ao seu mau funcionamento, destacando-se os nodes de criação dos modelos. Isto obrigou a que toda esta parte do trabalho fosse feita externamente ao Node-RED.
- A forte componente de aprendizagem automática (machine learning) existente neste projeto dispensou bastante tempo devido à complexidade de alguns dos seus conceitos chave. Um exemplo específico disto, é todo o processo em volta do ajuste dos parâmetros do modelo de previsão a ser treinado.
- Existiram no decorrer do trabalho alguns problemas de comunicação entre algumas das tecnologias utilizadas como a máquina virtual do broker, a base de dados e o Node-RED,
- O processo de identificação e resolução de problemas bem como a simples execução de todo o sistema foi algo complicado dada a natureza do trabalho.
- Alguns problemas na apresentação dos gráficos.

IV. Conclusão

Foram atingidos a maioria dos objetivos deste trabalho, ficando em falta:

- A apresentação do grafismo online que compara dois ciclos em atualização constante.
- Replicar o modelo de previsão para o descarregamento de bateria no Node-Red (apenas foi implementado o modelo, disponível no ficheiro models.py)

Também o ajuste dos parâmetros dos modelos de previsão poderiam ter sido mais explorado.

De um modo geral, este trabalho permitiu fortalecer o nosso contacto com as tecnologias IoT, sendo que adquirimos algumas das competências necessárias para possíveis implementações futuras neste ramo.

References

[1] Node-RED Documentation: https://nodered.org/docs
[2] MongoDB: https://www.mongodb.com/docs
[3] Node-RED - MongoDB: https://flows.nodered.org/node/node-red-node-mongodb
[4] Node-RED - Python Function: https://flows.nodered.org/node/node-red-contrib-python-function
[5] Mosquitto: https://mosquitto.org/documentation
[6] Kaggle: https://www.kaggle.com
[7] Scikit-Learn: https://scikit-learn.org/stable/modules/classes.html
[8] Google Cloud Documentation: https://cloud.google.com/compute/docs
[9] MQTT Essentials: https://www.hivemq.com/mqtt-essentials/
[10] Conteúdo disponibilizado no Moodle da UC

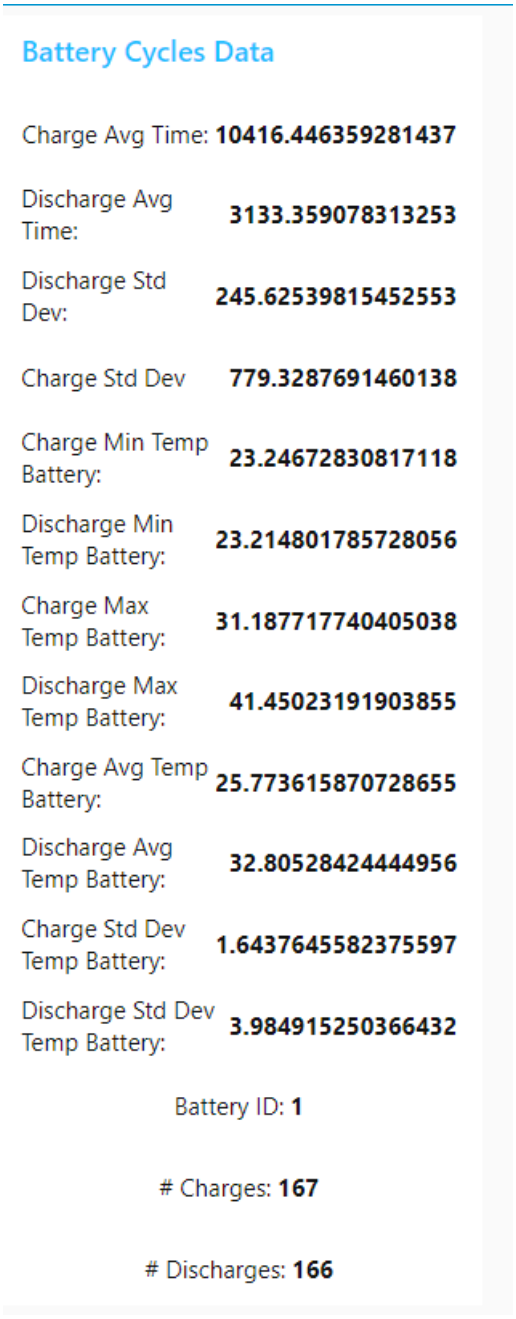

Fig. 1. Exemplo de apresentação da informação de uma bateria.

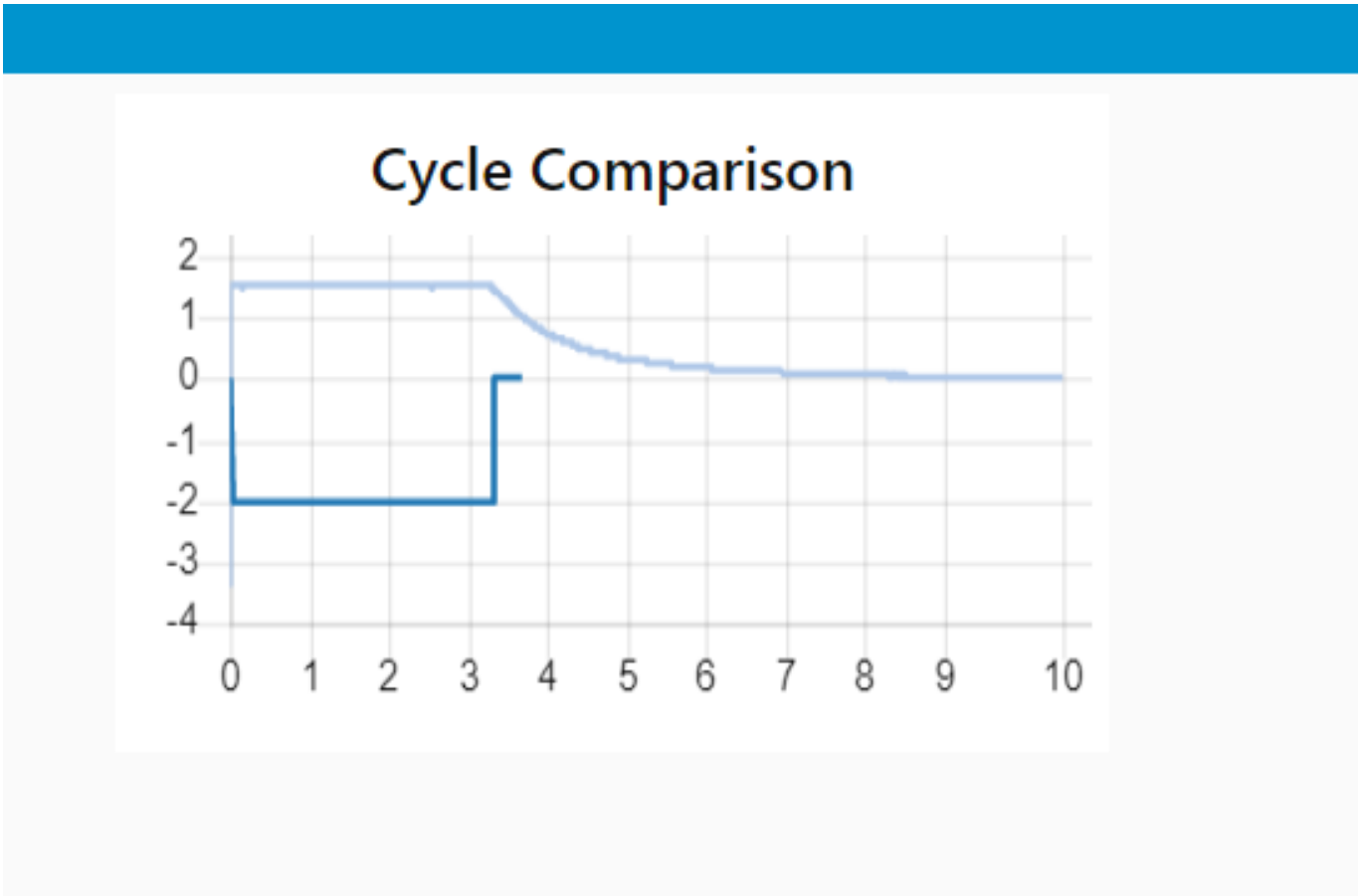

Fig. 2. Exemplo de um gráfico que compara dois ciclos de uma bateria.

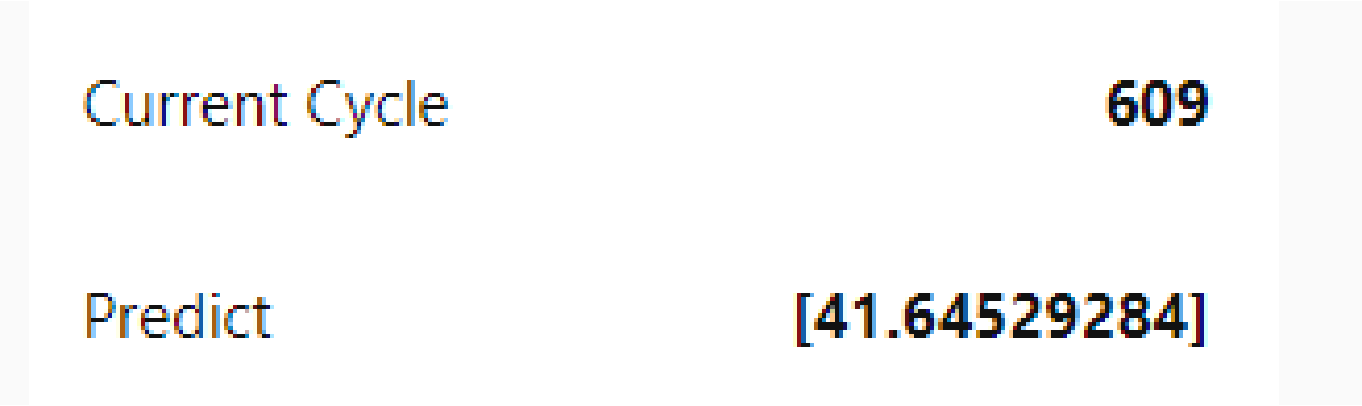

Fig. 3. Exemplo de uma previsão feita para um ciclo de bateria.