POLITECNICO DI TORINO
Collegio di Ingegneria Informatica, del Cinema e Meccatronica (ICM)
Master's Degree in Data Science and Engineering

Master's Thesis Summary Session: March/April 2025

"An on-edge Machine Learning model to estimate State-of-Health in Electric Vehicle Batteries"

Candidate: Giuseppe Cavaleri
Supervisor: prof. Edoardo Patti
Co-supervisor: prof. Alessandro Aliberti, dott. Raimondo Gallo
Company: Edison S.p.a. – Company Supervisor: Dott. Paolo Tosco

Introduction

This study explores the potential of a data-driven solution implemented through an on-edge machine learning model for real-time State-of-Health (SOH) estimation in electric vehicle (EV) batteries. In recent times the EV market has been experiencing a growth trend. The fundamental core of an electric vehicle is represented by its high-voltage Lithium-ion battery pack, which represents the primary source of energy for the entire system. These batteries are subject to degradation over time, and it is important to be aware of their condition. The main parameter that describes the battery's status is the SOH, a percentage value that indicates the effective capacity of the battery. The principal target of the project is to initialize the prototyping of a system capable of operating onboard an electric vehicle while providing real-time SOH estimation. Currently, the estimation of the SOH relies on complex and indirect methods, the possibility of having a device capable of providing an estimation of this value by reading data from the vehicle's BMS opens up various scenarios, such as the immediate evaluation of the battery pack in the used electric vehicle market. The first objective was to assess the suitability of the selected embedded system, the Raspberry Pi 4 Model B, by performing hardware performance tests during the model execution, making evaluations in terms of accuracy and errors metrics. Additionally, we developed an on-board data acquisition system by using the Raspberry Pi to extract battery parameters from the Battery Management System (BMS) of a real electric vehicle.

Data preprocessing

The real data were acquired from a private company which is specialized in the collection and analysis of EV's battery parameters. The data extraction process starts from raw CSV files containing data about real drive cycles conducted by EVs. Only medium-long drive cycles were selected, during on average 1 hour, with state of charge discharging profiles and speed variation. From each drive cycle, the time series data related to four EV's battery pack parameters were extracted, which are: Voltage, Current, State of Charge (SOC), and Internal Temperature. The time series data are organized and divided in SOH labeled time windows. Each time window contains 100 data point for each one of the four parameters. From the real dataset were extracted 150 127 time windows. Since the real dataset does not cover the entire range of SOH values of interest (from 80% to 100%), with a not equal distribution in terms of numerosity between the SOH values, a data balancing algorithm was developed. To tackle the lack of real data, another synthetic dataset generated with an EV simulator was integrated, to have a more complete data availability, covering the entire range of interest. The synthetic dataset is significantly larger, including 1 364 994 time windows.

Model implementation

The developed machine learning model is based on Long-Short Term Memory (LSTM) units, which are particular structures widely used for time series analysis and forecasting. The model takes as input the labeled time windows, which are analyzed from the LSTM model during the training phase, Figure 1. Initially, the network was pre-trained with the synthetic dataset. Subsequently, by applying the transfer learning technique, a fine tuning is performed on the network by using the real dataset. This technique allows to have a high-performance network, even with a smaller amount of real data available.

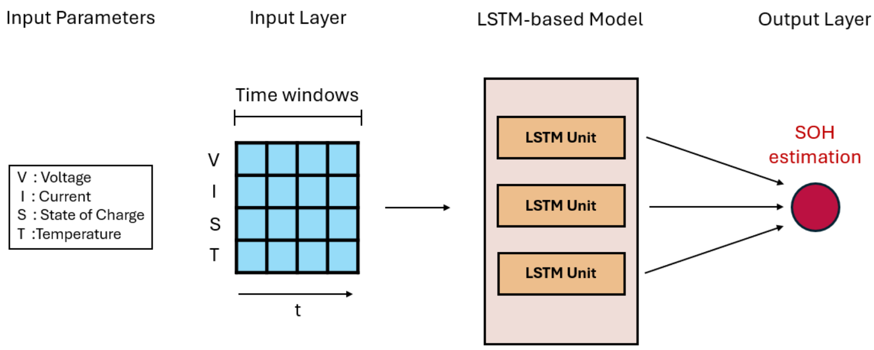

Figure 1. Input data and model definition

After the training phase, the model is exported in TensorFlow Lite format and deployed on the embedded system, the Raspberry Pi 4 Model B. Operating within a resource-constrained environment, it is important to ensure that the system performs efficiently and reliably during the algorithm execution. For this reason, we conducted hardware performance tests to evaluate the system's capabilities. The primary points of analysis included the CPU status, by assessing usage percentage, frequency, temperature, and the RAM occupation. To finalize our validation, we evaluated the model by measuring mean latency on the single time window prediction, model size, and prediction accuracy, along with percentage errors tested on a batch of real drive sessions data, as reported in Table 1.

Model Performance		Model Characteristics	
Metric	Value	Metric	Value
MAE (%)	0.78	Latency (ms)	21.6
RMSE (%)	1.24	Model size (KB)	204.19
R^2	0.98	N. Params	45 772

Table 1. Model analysis

Data acquisition system

Considering the results obtained from the validation of the embedded system suitability, we decided to begin the development of a first prototype in laboratory, starting from the creation of a data acquisition system mounted on board a real EV. The setup was realized by using three devices which communicates and exchange data through a Wi-Fi connection, as shown in Figure 2. Essentially, the main protagonist is the Raspberry Pi, which gathers data from the OBD-II port of the electric vehicle through the OBD-II Scanner. The exchanging of information is done with a request-response based system by using CAN messages. The Raspberry Pi sends a query CAN message asking for a specific parameter reading, and the

EV's BMS returns a sequence of CAN response messages. The returned data is encoded in hexadecimal format. Inside the acquisition script, decoding functions have been implemented, through which it is possible to transform the data into a physically interpretable domain, providing the measurements of the parameters. After the decoding phase, data are stored in an organized format in exportable Numpy arrays. The laptop PC acts as the controller of the Raspberry Pi, communicating through SSH protocol. The data acquisition system was successfully validated with two tests conducted in real-world driving conditions. The tests were conducted on two different vehicles of the same car model to evaluate the adaptability of the data acquisition system. The collected data are subsequently exported and visualized as illustrated in Figure 3. Currently, the developed architecture enables the collection and decoding of data at a sampling frequency of 0.625 Hz, having that each data sample is acquired every 1.6 seconds.

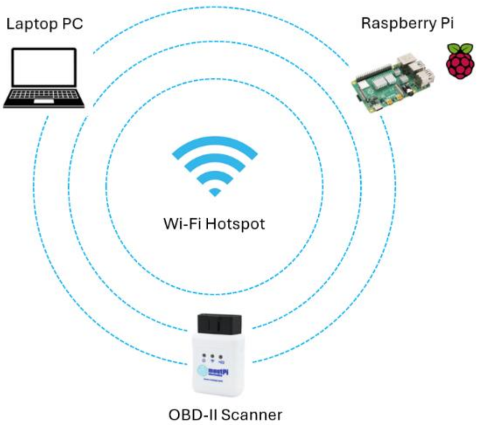

Figure 2. Devices setup

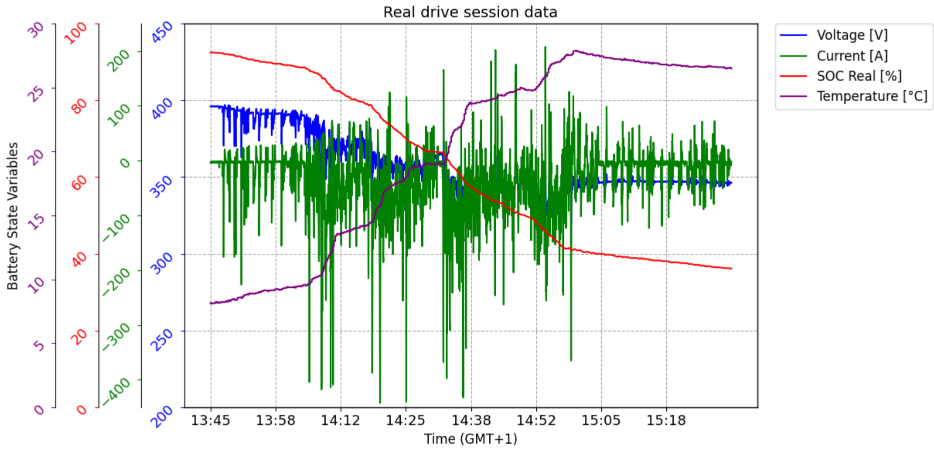

Figure 3. Drive session data visualization

Conclusions

To conclude, the main scope of this project was to verify the suitability of the proposed embedded device for an on-board real-time SOH estimation system. Starting from the data preprocessing, to the model definition and deployment. The main core is centered on the pipeline for compression and exportation of the model, with the subsequent analysis of the hardware performance and model accuracy. The realization of the data acquisition system with the on-board test, contributed to demonstrate the potential of using embedded systems in real-time diagnosis of batteries in EVs, contributing to advancements in EV battery monitoring and predictive maintenance. However, further improvements can be considered in future developments:

- Generation of more accurate synthetic data by exploiting the EV simulator available;
- Expansion of real datasets with specific vehicles following an implementation strategy, aiming to cover the whole range of SOH classes;
- Migration to a cloud-based architecture for model deployment, ensuring security and scalability;
- Improvement of the data acquisition system, leading the system to acquire data at a higher sampling frequency.