POLITECNICO DI TORINO

Master's Degree in Ingegneria matematica

Master's Degree Thesis

Long Short Term Memory neural network for battery State of Charge estimation

Supervisors

Prof. Tania CERQUITELLI

Federica CALDARA

Candidate

Irene CAPODICASA

March 2023

This research paper ("Thesis") has been produced is the context of an internship program that Irene Capodicasa (the "Intern") has attended within Accenture S.p.a. ("Accenture"). The Thesis is confidential and cannot be used other than by the Intern for the purpose of presenting her work. It must not be disclosed other than to Politecnico di Torino's professors on a confidential basis and to the extent required for Politecnico di Torino to carry out an evaluation and assess the Intern's final exam. In preparing this document, the Intern acknowledges that he/she has relied on confidential information provided by Accenture only for the purposes of producing the Thesis. The Intern shall defend, indemnify and hold harmless Accenture from and against any claim, demands, actions, judgments, awards, settlements, fees, liabilities, losses, damages, costs and expenses (including and without limitation attorneys' fees and court costs) ("Obligations") arising out of or relating to any allegation or claim that the Thesis or any use thereof infringes, misappropriates or otherwise misuses or violates the Intellectual Property or other rights of any Person. Accenture shall have the right, in its sole discretion, to participate in the defense of any such allegations or claims at its expense with a counsel of its choosing. The Intern shall not compromise or settle any such allegation or claim, or agree to binding arbitration thereof, in any manner without Accenture's prior written consent, unless such settlement is solely monetary in nature, and releases all Accenture Indemnified Parties from all Obligations with no admission of liability and has no adverse effect on any of them. This paper was written for my university thesis and is my personal work. The views expressed are my own and may not necessarily reflect those of Accenture.

Summary

Estimating the state of charge (SOC) of lithium-ion batteries (LIB) is a critical task that has also become highly desirable as electrified vehicles become more widely used. It is necessary to improve the accuracy of the battery SOC estimation, to develop more efficient, reliable and affordable electrified vehicles. Due to the non-linear behavior of these batteries, an accurate estimation of SOC is still challenging. For this reason, thanks to the greater availability of battery data and advances in artificial intelligence (AI), traditional theory-based methods are often replaced by data-driven approaches. In particular, recurrent neural networks (RNN) should be a promising method to be exploited since they can capture dependencies in time and predict SOC without a battery model. Therefore, this thesis project shows how a particular type of RNN, called long short term memory (LSTM), can accurately predict SOC values in real time and forecast future values of the battery SOC within different time horizons.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
artificial intelligence

BEES
battery energy stationary storage

BMS
battery management system

BP
battery passport

CC
Coulomb counting

EMS
energy management system

EOL
end of life

EV
electric vehicle

EVB
electric vehicle battery

HEV
hybrid electric vehicle

LIB

lithium ion battery

LSTM

long short term memory

NMC

nickel-manganese-cobalt

NN

neural network

OCV

open circuit voltage

RNN

recurrent neural network

RUL

remaining useful life

SOC

state of charge

SOH

state of health

SOP

state of power

Chapter 1

Introduction

World Health Organization (WHO) studies show that more than 99% of people are living in areas where air pollution is above the WHO limits and more than 4 million deaths are due to air pollution each year. Pollutants of major public health concern came from household combustion devices, motor vehicles, industrial facilities and forest fires. This is the reason why a lot of countries are reducing petrol and diesel powered vehicles. The climate change and the energy transition of the globe for the fulfillment of net-zero target are nowadays nevralgic and challenging topics in our society. For this reason, governments push for new research development and changes in everyday activities.
Since electric vehicles are responsible for the 1/4 of the global CO_2 emission, transport electrification is becoming crucial to perform the green deal requirements. Transports are at the first step of their journey towards electrification and all the e-mobility panorama is quite immature. Nevertheless, the scenario evolves fast trying to accomplish user and energy needs.

Currently, lithium-ion batteries (LIBs) are the most commonly utilized in the industry, particularly in the electric vehicle sector, and have gained significant popularity in recent years. The main advantage of LIBs is their efficiency: for the same volume and mass, they can deliver a greater amount of energy compared to conventional batteries, such as lead-acid batteries. Moreover, they have longer life (number of cycles) and lower self-discharge rate, compared with traditional cells. This is the reason why nowadays LIB batteries are the key technologies in the European energy transition, not only for the electric mobility but also as BESS (Battery Energy Stationary Storage) to stabilize the grid power by fluctuations enabling the inclusion of renewable sources in the energy taxonomy.
The battery market will grow exponentially in the next decades. The amount of end of life (EOL) batteries is estimated about 3,5 million tons in the year 2030. This represents both a major issue, since the used batteries have to be managed but also a business opportunity, since waste batteries are valuable wastes, that contains

precious raw materials that can be restored or used in second-life applications. To reduce the environmental impact of used batteries, a European Regulation Proposal will be adopted. Digital battery passports (DBPs) can enable greater sustainability and circularity in battery value chain. DBP for electric vehicles will be regulated by *Article 65* of EU regulation. The goal of this initiative is to update the EU's regulatory framework for batteries, targeting three closely connected issues:

- the absence of favorable framework conditions encouraging investment in sustainable battery production capacity;
- the inadequate performance of recycling markets;
- the social and environmental hazards not currently addressed by EU environmental regulations.

Accurate state of charge (SOC) estimation comes into play especially when addressing the second bullet point: estimating the battery's state of charge is a critical issue for making the right decisions at all stages of the battery's life. Moreover, it helps minimizing the costs related to battery oversizing, as the battery pack is still one of the most expensive and crucial vehicle components. SOC good estimation also helps to improve vehicle performance, safety, and passenger comfort: it reflects the battery performance because it prevents the overcharging and overdischarging, improving its lifespan and saving energy. Unfortunately SOC cannot be directly measured, but must be estimated.

The remainder of this work is organized as follows: next paragraphs contain an introduction to electric vehicle batteries, the battery management system and the related European regulation. In Chapter 2 there is an overview of most common battery SOC estimation methods. Chapter 3 describes the setting of the experiment and the used approach to estimate SOC. Chapter 4 shows the achieved results and, in Chapter 5 there are some conclusions and future perspectives.

1.1 Electric vehicle battery

Between 2010 and 2018, battery demand grew by 30% annually and reached a volume of 180 GWh in 2018. According to the base case scenario in Figure 1.1, the market is expected to continue expanding at an annual rate of approximately 25%, and it is estimated to reach a volume of 2,600 GWh by 2030. By that year, passenger cars are expected to account for the largest portion (60%) of global battery demand, followed by the commercial vehicle segment at 23%.

An electric vehicle battery (EVB, also known as a traction battery) is a rechargeable battery used to power the electric motors of a battery electric vehicle (BEV)

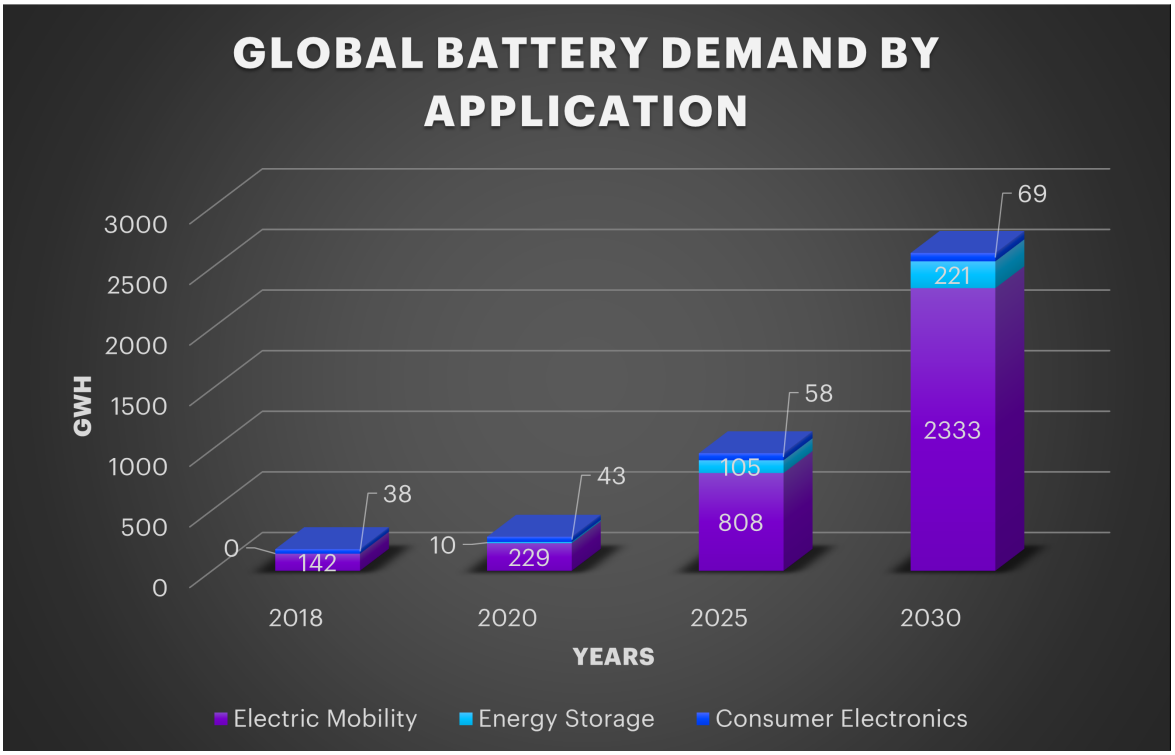


Figure 1.1: Global battery demand expected in next years, partitioned by application.

or hybrid electric vehicle (HEV). Lithium-Nickel-Manganese-Cobalt-Oxide (LiNiMnCoO2), abbreviated as NMC, is now widely used as a cathode powder in the development of batteries for power tools, e-bikes, and electric powertrains. It offers high overall performance, excellent specific energy, and the lowest self-heating rate compared to other mainstream cathode powders, making it the preferred choice for automotive batteries. This helps to increase the range of electric vehicles and reduce their overall weight. The disadvantage of NMC is the lower safety compared to LFP[1] technologies that have lower energy density and are more suitable for energy stationary storage. Lithium-ion batteries also have some limitations, including their high cost and the need for careful management to avoid overcharging or overheating, which can lead to safety issues.

The designs of battery packs for electric vehicles (EVs) are very complex and differ significantly depending on the manufacturer and the intended use. Nevertheless, they all include a mixture of various fundamental mechanical and electrical

[1]Lithium ferrophosphate is a type of lithium-ion battery composed of lithium iron phosphate $LiFePO_4$ as the cathode material, and a graphitic carbon electrode as the anode.

components that carry out the necessary functions of the pack. As represented in Figure 1.2, a *battery pack* is a series of individual modules and protection systems that are organized in a shape that can be easily installed in a vehicle. Each *module* is composed of cells that are connected in series or in parallel. The basic unit of the electric vehicle battery is the *cell*, which converts the chemical energy into electrical one. They come in many shapes and forms but the three most common ones are prismatic, pouch and cylindrical.

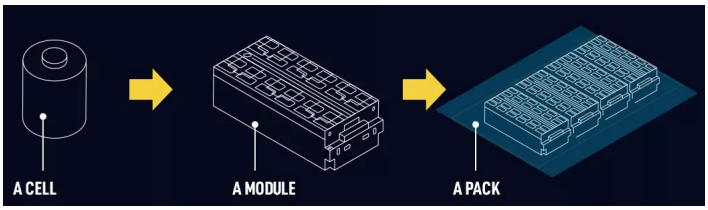


Figure 1.2: Different components of electric vehicle battery.

The production of battery cells primarily involves chemical processes, while module and pack production typically involves mechanical assembly processes. Once produced, the battery cells are then arranged in modules to form serviceable units. The cells are connected in series and in parallel, into battery packs, which are the complete enclosure that deliver power to the electric vehicle, to achieve the desired voltage and energy capacity. For example, an EV battery requires 400-800 volts and one single battery cell typically features 3-4 volts.

Some of the main measures for electric vehicle batteries include:

- voltage: this is the measure of the electrical potential difference between the positive and negative terminals of the battery. Higher voltage batteries can provide more power to the motor.

- Current: refers to the flow of electrical charge (in the form of electrons) through a circuit, which is generated by a battery. It is a measure of the rate at which electric charge is moving through the circuit.

- Internal resistance: The opposition to current flow within an element or battery, i.e. the sum of electron and ion resistance as a contribution to the total effective resistance including inductive-capacitive properties.

- Impedance: refers to the opposition of a battery to the flow of an alternating current (AC) or direct current (DC). It is a measure of the resistance of the battery to the flow of current and is often used as an indicator of the battery's health and performance. Higher impedance can result in a lower battery capacity and reduced performance, as it can lead to voltage drops and reduced

current flow. Battery impedance can be affected by a variety of factors such as the age of the battery, the temperature, and the rate of discharge.

- Temperature: EV batteries can be sensitive to extreme temperatures: high temperatures can accelerate battery degradation and reduce the battery's capacity, while low temperatures can reduce the battery's ability to provide power. so effective temperature management is important to maintain optimal performance and extend battery life.
- Capacity: this refers to the amount of energy a battery can store and release before needing to be recharged. The higher the capacity, the longer is the operational range of the EV: the capacity of a battery depends on various factors, including its chemistry, size, temperature, and discharge rate. As batteries age and are cycled (charged and discharged), their capacity may degrade over time.
- Power density: This measures the amount of power that can be delivered per unit of battery weight or volume. A high power density is important for providing quick acceleration and maintaining high speeds.
- Charge time: is the amount of time it takes to charge the battery from empty to full. Fast charging systems can recharge an EV battery in a matter of minutes.
- Cycle life: is the number of charges and discharges a battery can have before it loses significant capacity. EV batteries with longer cycle life can last for numerous years without demanding to be replaced.

These measures are important for ensuring that EV batteries provide reliable and efficient performance while also meeting safety and environmental standards.

1.2 Battery Management System

The battery management system (BMS) is an electronic control unit responsible for communicating information between the vehicle energy management system (EMS) and the battery, ensuring safe operation by interrupting the battery current in the event of a safety concern. It continuously monitors and protects the battery during operation, in order to avoid any electrical or thermal abuse.

The BMS is make up of different sections monitoring each module of the battery pack, which includes cell balancing circuits, and a BMS Master that processes the data sent by the various sensors, and controls cells and modules to coordinate the measurements, and executes the balancing algorithm. The BMS is responsible for linking the battery to the traction system and transmitting important

battery-related information. In the event of any safety issues, the BMS is required to halt the flow of current through the battery. Furthermore, the BMS contains, also, all the algorithms and control policies necessary to guarantee the safety and performance of the battery, including algorithms for estimating battery states, such as state of charge (SOC) to determine the remaining capacity of the battery, state of health (SOH) to estimate the capacity fade from the beginning of battery's life, and state of power (SOP) for a power capability estimation.
So, the BMS performs the current sensing, measures the total voltage and the impedance, controls power switches for battery protection and isolation. Those information are then used by advanced estimation algorithms to accurately determine SOC, SOH and SOP. The algorithms are generally stored in an external unit that is the energy management system.

1.3 EU Regulation

On December 10, 2020, a proposal for a regulation on batteries and waste batteries was presented by the European Commission. The primary goal of the proposal is to enhance battery development and production, which are crucial components for Europe's transition to clean energy. A strong increase in the electrification of passenger cars is expected to take place between 2020 and 2030 and this will help reducing greenhouse gas (GHG) emissions and noxious emissions from road transport.

The goal of the proposal is to achieve the following objectives:

- enhancing the internal market's effectiveness, which includes products, processes, waste batteries, and recycling, by ensuring a level playing field through a common set of rules;
- encouraging a circular economy;
- minimizing environmental and social impacts at all stages of the battery's life cycle.

Given the strategic significance of batteries and the need to prevent discrimination, trade barriers, and market distortions among all stakeholders, there is a need to establish regulations on sustainability criteria, safety, performance, recycling, and second-life applications of batteries, as well as providing information to consumers and economic operators. To achieve this, a unified regulatory framework is essential for managing the entire life cycle of batteries marketed in the European Union.

According to the Regulation, the BMS is responsible for controlling and overseeing the electrical and thermal functions of the battery, as well as storing and managing data related to the battery's state of health and expected lifetime, so

that those aspects may be determined at any time by end user or any other third party acting on his behalf, by communicating with the vehicle or appliance in which the battery is incorporated. The battery management system of stationary energy storage systems (which are batteries without attached external devices to store energy) and electric vehicle batteries equipped with a BMS must include real-time data on parameters that determine the state of health, safety, and expected lifetime of batteries. Those parameters are listed in Annex VII of the Regulation:

- Parameters for determining SOH of the batteries
  - remaining capacity;
  - overall capacity attenuation[2];
  - residual power capacity and power attenuation[3];
  - residual charge/discharge efficiency[4];
  - actual cooling demand[5];
  - evolution of spontaneous discharge rates;
  - ohmic resistance and/or electrochemical impedance.
- Parameters for determining expected battery life:
  - dates of manufacture and commissioning of the battery;
  - energy efficiency;
  - capacity yield.

At any time, the legal or natural person who has legally purchased the battery or any third party acting on their behalf shall be provided with read-only access to the data from the BMS, in order to:

- assess the remaining value and potential for continued use of the battery;

[2]The decrease, as a function of time and use, of the charge that a battery is capable of delivering at nominal voltage, with respect to the original nominal capacity declared by the manufacturer.

[3]The decrease, over time and usage, in the amount of energy a battery is capable of delivering at rated voltage.

[4]The ratio between the net energy delivered by a battery during the discharge test and the total energy required to restore the initial state of charge by means of a normal charge.

[5]Amount of cooling required to maintain the temperature of the battery within a safe and optimal range during charging and discharging. When a battery is charged or discharged, it generates heat due to the flow of current through its internal components.

- facilitating the preparation for reuse[6], reuse, the preparation for repurposing[7], repurposing or remanufacturing of the battery;
- allow independent aggregators or market participants to access and use the battery's energy storage capacity for various purposes, such as providing grid services or participating in energy markets.

Manufacturers are required to provide real-time data on the state of health, state of charge, power set point, and capacity of electric vehicle batteries that include a battery management system. Furthermore, by January 1, 2024, the battery management system must be designed to be compatible with smart charging systems, including by having vehicle-to-grid, vehicle-to-load, vehicle-to-vehicle, vehicle-to-power bank and vehicle-to-building charging functions.

The supervisor committee is the Committee on the Environment, Public Health and Food Safety (ENVI). ENVI members held an exchange of views with the Commission on the proposed regulation on 15 June 2021, and adopted a report on 10 February 2022, elevating the level of intention of the Commission proposal. On 10 March 2022, Parliament's plenary adopted the report and interinstitutional negotiations started with a first trilogue on 20 April 2022. Sustainability requirements will be introduced gradually from 2024, while,on the other hand, liability on extended producer responsibility will start applying in mid-2025. The regulation provides a broad regulatory framework, meaning that some technical details will require additional legislation (such as delegated and implementing acts) to be fully operational. The agreement reached is provisional and still needs to be formally approved by both the Council and Parliament. On January 18, 2023, ambassadors of the Member States to the EU endorsed the agreement. The ENVI Committee vote took place on 24 January, with an approval of the text adopted in interinstitutional negotiations at first reading.

1.4 Battery Passport

The journey of a battery begins with the extraction of raw materials and continues through to its use as stationary energy storage or as a recycling source. The production process of a rechargeable battery, from mining the raw materials to use in an electric vehicle, involves numerous social and environmental risks. These

[6]Means the complete or partial direct re-use of the battery for the original purpose the battery was designed for.

[7]It refers to the process of reusing in parts or complete batteries that are no longer suitable for their original intended purpose, but can still be used for other applications such as stationary energy storage or as backup power sources.

include concerns around mineral extraction such as child labor, unsafe working conditions, and the violation of indigenous rights. Additionally, the production processes involved in making batteries have significant impacts on sustainability, including contributing to CO_2 emissions, consuming significant amounts of water, causing biodiversity loss, and pollution. All of these factors can impact the overall sustainability of the end product, making it crucial to consider the entire life cycle of batteries and their impacts on both people and the environment.

The *battery passport* (BP) is an electronic record of a battery that provides details about its environmental, social, and governance (ESG) aspects, as well as its life-cycle requirements. A sustainable battery is one that is produced in a manner that minimizes its negative impact on the environment, while also being socially responsible and economically viable. This includes considering the entire life cycle of the battery, from the extraction of raw materials to end-of-life recycling or disposal, and ensuring that its production and use do not contribute to significant carbon emissions, pollution, or other harmful impacts on people or the environment. A sustainable battery also takes into account the social and economic impacts of its production, such as labor conditions and economic viability. Additionally, a sustainable battery may prioritize the use of recycled or sustainably sourced materials and the implementation of circular economy principles to reduce waste and improve resource efficiency. Overall, a sustainable battery seeks to balance environmental, social, and economic concerns to provide a responsible and viable energy storage solution for a low-carbon future. The battery passport is essentially a digital copy of the physical battery and is made possible through the digital BP platform. This platform provides a secure way of sharing information and data on batteries on a global scale.

By 1 January 2026, every industrial battery and electric-vehicle battery will have its unique identifier, that the economic operator placing the battery on the market shall attribute to it, which will contain all the information about the basic characteristics of each battery type and model, that will be stored in the data sources of the system established according to Article 64 of the Legislation and shall be accessible online to ensure availability over time; that information should also be made available by means of QR codes printed or engraved on all batteries. Article 64 sets up the electronic exchange system for battery information: the system will contain all information about internal production and supervised verification, which all be sortable and searchable, respecting open standards for third party use. The battery passport shall be linked to the information contained in the system, and shall be accessed online through electronic systems.

The battery passport will enable access to information regarding the performance and durability parameters outlined in Article 10(1) at the time of the battery's initial placement on the market, as well as during any subsequent changes in its status. The Article contains information about requirements on the electrochemical

performance and durability parameters for rechargeable industrial batteries, electric vehicle batteries and light means of transport batteries[8]: starting from 12 months after the Regulation enters into force, all industrial batteries must be accompanied by technical documentation that includes the values for electrochemical performance and durability parameters, as specified in Part A of Annex IV:

- Nominal capacity (in Ah) and capacity attenuation (in %);
- Power (in W) and power attenuation (in %);
- Internal resistance (in Ω) and increase in internal resistance (in %);
- Charge/discharge efficiency and relative attenuation (in %);
- Indication of the expected life of the batteries under the conditions for which they were designed.

The technical documentation mentioned in the previous statement must also provide an explanation of the technical specifications, standards, and conditions that were utilized to measure, calculate, or estimate the values for the electrochemical performance and durability parameters. This explanation must include, at minimum, the components outlined in Part B of Annex IV:

- Discharge rate and charge rate applied;
- Ratio of maximum permitted battery power (W) to battery energy (Wh);
- Depth of discharge measured during the cycle life test;
- State-of-charge power capacity of 80% and 20%;
- Any calculations performed with the measured parameters, if any.

The anticipated widespread adoption of batteries in areas such as transportation and energy storage has the potential to reduce carbon emissions, but to fully realize this potential, it is crucial that the entire life cycle of these batteries has a low carbon footprint[9]. Therefore, the technical documentation accompanying all

[8]refers to vehicles that have an electric motor with a maximum power output of 750 watts, designed to carry passengers while in motion, and can be powered entirely by the electric motor or a combination of electric and human power.

[9]It is the sum of greenhouse gas (GHG) emissions and GHG removals in a product system, expressed as carbon dioxide (CO2) equivalents and based on a Product Environmental Footprint (PEF) study using the single impact category of climate change.

industrial batteries that are placed on the market in the European Union should also include a declaration of their carbon footprint.
By 1 January 2026, all these information shall be available via the publicly available part of the electronic exchange system (Article 64) and shall be available to consumers prior to purchase.

According to Article 9 and Annex III, starting from 1st January 2026, portable batteries of general use can only be introduced to the market if they meet the electrochemical performance and durability requirements. The Commission is authorized to adopt delegated acts to update the minimum values and introduce additional electrochemical performance and durability parameters based on technical and scientific advancements. The minimum values for electrochemical performance and durability listed in the regulation must not result in a decrease in the level of performance and durability for electric vehicle batteries.

In addition to the parameters already specified, the battery passport must also provide access to information regarding the state of health of the battery, as outlined in Article 14. This information must be available when the battery is first placed on the market, as well as when it undergoes any changes in its status. At the end of the first life, used batteries are considered waste (except for reuse), and repurposing is considered a waste treatment operation. When batteries are repurposed or given a second life, they are considered as new products and must meet all product requirements when they are introduced to the market. If the change in the battery's status is due to repurposing or remanufacturing activities, the responsibility for maintaining the battery record in the battery passport must be transferred to the economic operator who is responsible for introducing the repurposed battery into the market or putting it into service. The record for repurposed or remanufactured batteries must be linked to the record of the original battery: if a battery undergoes repurposing or remanufacturing, a new label must be affixed to reflect its updated status. As per Article 59, independent operators that carry out repurposing or remanufacturing operations must be granted equal and adequate access to all relevant information pertaining to the handling and testing of electric vehicle batteries, components of such batteries, appliances, or vehicles, including safety aspects.

Operators are responsible for ensuring that the examination, performance testing, packing, and shipment of batteries and their components are conducted with adequate quality control and safety instructions. Moreover, in case of repurposed or remanufactured batteries, they must ensure that the battery complies with the requirements specified in the Regulation.
Upon request, the battery holder must demonstrate the following in order to document that a waste battery, which has undergone repurposing or remanufacturing, is no longer considered waste:

- provide evidence of a state of health evaluation or testing confirming that the

battery can perform as required for its intended use after the repurposing or remanufacturing process. This evidence must be made available to end-users and third parties acting on their behalf;

- document further use of the battery after repurposing or remanufacturing through an invoice or a contract for the sale or transfer of ownership of the battery;
- provide evidence of appropriate protection against damage during transportation, loading, and unloading. This includes sufficient packaging and appropriate stacking of the load.

Chapter 2

State of art about State of Charge

State of charge (SOC) refers to the amount of electrical energy stored in a battery, expressed as a percentage of the battery's total capacity. It is the equivalent of a fuel gauge used in traditional gasoline vehicles: it is a value from "Empty" (0 %) to "Full" (100 %) that indicates the percentage of useful energy left inside the battery. A good estimation of battery states is crucial to reduce over-design costs and increase the overall vehicle efficiency and performance: for example if a battery cell is subjected to over-charging or over-discharging it may has the potential to cause irreversible harm and decrease its overall lifespan. Also accurate battery state estimates allow battery pack to be used aggressively within design limits, so pack does not need to be over-engineered allowing smaller battery packs and smaller costs.
Thus, SOC is an important parameter to monitor in battery-powered devices and systems, for several reasons:

- Battery life: SOC is a critical parameter to monitor for maximizing the life of a battery. Operating a battery at a high SOC or a low SOC for prolonged periods can reduce its lifespan. For example, lithium-ion batteries tend to have a longer lifespan if they are not charged to 100% or discharged to 0% too often.

- Performance: The SOC of a battery can affect its performance. As the SOC decreases, the voltage of the battery also decreases, which can lead to a drop in performance. In some applications, such as electric vehicles, maintaining a certain SOC is crucial for optimal performance.

- Safety: Overcharging or over-discharging a battery can cause damage or

even lead to a fire or explosion. Monitoring the SOC can help prevent such situations by alerting the user to the need to charge or discharge the battery.

- User experience: In devices like smartphones, tablets, and laptops, the SOC is often displayed to the user. Knowing the SOC can help users plan their usage and ensure that their device has enough charge to last the required amount of time.

Overall, monitoring the SOC of a battery is crucial for optimizing its lifespan, performance, and safety, as well as ensuring a good user experience As a result, significant design effort is placed on the battery management system software design to perform a SOC and state of health estimation accurately.

Unfortunately no sensor is available to measure SOC so it has to be estimated. With greater availability of battery data and advancements in computing power, methods that rely on data, including machine learning (ML), are gaining popularity for making estimations of state of charge and state of health, but the task still remains challenging due to battery's non-linear behaviour. The estimation is done thanks to a great variety of methods which trade on measurable signals such as the battery terminal voltage, current, and temperature.

The main requirements for SOC estimation of electric vehicle batteries are high accuracy to achieve maximum efficiency and safety, robustness because the battery's internal states change as the vehicle's states change, and because there always exist sensor measurement errors. Finally the SOC estimate must be reasonably smooth so that it does not indicate that the battery is charging while it is actually discharging.

2.1 SOC algorithms

Key methods for lithium-ion EV battery state of charge estimation can be categorized under five groups:

- conventional methods such as ampere-hour counting (further details in Section 2.1.1) and open circuit voltage (explained in Section 2.1.2), that use the physical properties of the battery, which includes voltage, discharge current, resistance and impedance;
- adaptive filter algorithm such as Kalman Filter presented in Section 2.1.3;
- ML algorithms like artificial neural network (ANN) (explained in Section 2.1.4) and support vector machine (SVM) that require a large amount of training data and heavy computation;
- non-linear observer that is designed to handle with the highly nonlinear system;

- others and hybrid algorithm method which combines two or three SOC algorithms to estimate SOC.

2.1.1 Coulomb counting

The *Coulomb counting* method is one of the easiest to implement: it is based on the integration of battery current with respect to time while the battery is charging or discharging. The mathematical expression is the following:

$$SOC(k) = SOC(0) - \frac{T}{C_n} \int_0^k (\eta I(t) - S_d)$$

where $SOC(k)$ is the battery's SOC at time k, $SOC(0)$ is the initial value of SOC, $I(t)$ is the current at time t, T is the sampling period, C_n is the battery's nominal capacity, S_d is the self-discharging rate and η is the coulombic efficiency, defined as the ratio of energy delivered by a battery during discharge compared to the amount of energy it accepted during charging. It is expressed as a percentage and is a measure of how efficiently a battery is able to convert electrical energy into chemical energy during charging and vice versa during discharging.

The biggest advantage of the CC method is its low power computation cost, but the main disadvantage is that it cannot remain accurate for a long time: this is due to the difficulties in determining the initial value of SOC, which causes a cumulative error effect. Moreover, current and temperature measurement errors and noise, may also result in cumulative effects: for example degradation of the battery results in a loss of battery capacity, which causes an estimation error; also self discharge (the consumption of charge inside the battery) cannot be observed by current measurement. In order to improve the accuracy of the method, the initial capacity and SOC value of the battery, and the current sensor drift, caused by integration, need to be corrected and adjusted regularly by complete discharging of the cell and periodic capacity calibration to obtain maximum capacity, because there are no mechanisms of feedback compensation in this method. Unfortunately those frequent adjustment require a lot of time and can shorten the battery lifespan.

2.1.2 Open circuit voltage

The OCV of a battery cell is defined as the potential difference between the positive and negative electrodes, when the battery is at equilibrium, thus it is subjected to no load. The OCV method approximates the relationship between SOC and OCV and this is possible because SOC in a lithium-ion battery is related to embedding quantity in the active material. The method uses a table look-up technique to estimate the SOC from voltage measurements. The relationship between SOC and OCV can differ from different batteries, because it depends on both the capacity

and the internal electrode material of the battery. Some traditional batteries have an approximate linear relationship, but for example, lithium-ion batteries does not have that linear relationship between SOC and OCV. Moreover, this relationship can be affected by external factors such as the environmental temperature and cycle life of the battery, thus it is unstable and changes continuously.

Unfortunately the amount of time needed to reach the steady equilibrium condition of the battery depends on SOC, temperature and so on, and it can take hours. Thus, this method is only applicable when the vehicles are stationary and parked, rather than being operated while in driving mode. This method is relatively simple to implement because it has a very low power computation and a relatively high accuracy and these are the reasons why it is generally used as a calibration auxiliary technology.

2.1.3 Kalman filter

The adaptive filter algorithm is designed to reduce the noise influence on the battery model in order to improve accuracy and robustness of SOC estimation. This method can be identified by three main blocks:

- a battery model to properly estimate a predetermined SOC as an input of the model;
- a battery voltage model to calculate the battery voltage as an output of the model;
- an appropriate algorithm to evaluate the gain to update the SOC by comparing the measured voltage and model voltage.

The Kalman filter is a well-designed method, which filters parameters from uncertain, inaccurate observations: its self-correcting nature helps to tolerate a high variation of current and this is the main reason why KF has gained immense popularity in battery state estimation in recent years, regardless of its high computational cost. KF's mathematical equations provide a recursive solution through a linear optimal filtering for estimation of state space variables: the first equation predicts the current state x_t from the earlier state x_{t-1}; the second one predicts a measurement and updates the current state measurement to converge to the real value.

$$\begin{cases} x_{t+1} &= A_t x_t + B_t u_t + \omega_t \\ y_t &= C_t x_t + D_t u_t + \nu_t \end{cases} \tag{2.1}$$

In the above Equation 2.1, x is the system state, u is the control input, y is the

measurement input, ν is the sensor measurement noise, ω is the process noise, and A,B,C and D are the covariance matrices that describe the dynamics of the system. The process noise and measurement noise are assumed to be independent of each other and to be zero Gaussian, while the state-space form equations consider a discrete-time version of the cell dynamics. Unfortunately KF cannot be used to prediction state of a non-linear system (it necessitates of predetermined SOC) and it requires highly complex mathematical calculations.

2.1.4 Artificial neural networks

A biological neuron is composed of a cell body which contains the nucleus and most of cell's components, a lot of branching extensions called dendrites and a longer one called axon. Each biological neuron receives short electrical impulses called signals from other neurons via synapses which are located at the tip of each branch. When a neuron receives a sufficient number of signals it fires its own: this means that biological neurons are organized as a very complex network, and recent studies show that they are distributed in consecutive layers.
One of the simplest artificial neural network is called *perceptron* and was invented in 1957 by Frank Rosenblatt. Each neuron is a threshold logic unit (TLU) that has as numbers as input. Each input connection is associated with its own weight: as represented in Figure 2.1, the TLU performs a computation by multiplying each of its inputs by a weight, and then adding these weighted inputs together. After this, a step function is applied to the sum, which produces an output. Summarizing a

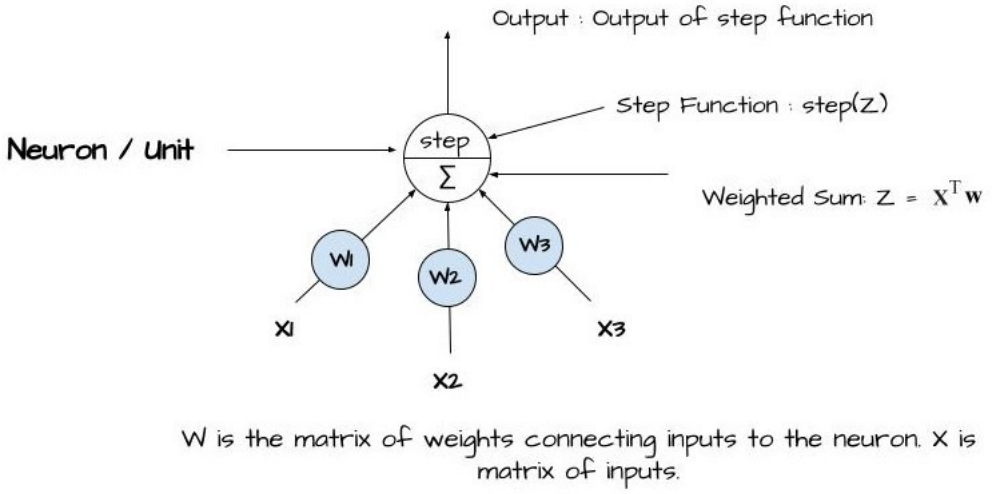


Figure 2.1: Representation of a single layer perceptron.

perceptron is composed of a single layer of TLUs, each one connected to all the inputs. A layer in which each neuron is connected to every neuron in the preceding layer, that is, to its input neurons, is referred to as a *fully connected layer* or a *dense layer*. All the input neurons form the input layer; moreover, an extra bias neuron is typically added to these, representing an extra bias feature $x_0 = 1$, which just outputs 1 all the time. By adjusting not only the weights but also the biases of the network, the model can better adapt to the data, enabling it to more effectively meet the needs of the dataset. The following equation can summarize the output of a perceptron:

$$Y = \phi\left(X^T W + b\right) \tag{2.2}$$

where ϕ is an activation function, X is the matrix of inputs, W is the associated weight matrix and b is the vector of input biases. If the model has an input layer, multiple hidden layers and an output layer, it is called *Multi-layer perceptron*. It works similarly to the perceptron: first of all the inputs are multiplied by their weights, then are added up to pass to a step function. The output of each neuron's step function is multiplied by its corresponding weight, and the resulting values are added together. This sum is then fed into another step function to generate the final output of the network. When neurons are combined to form a deep neuron layer and more layers are stacked up into a model then this model is known as *deep neural network* (DNN).

ANN are trained with an algorithm called *backpropagation*: it computes gradient of a loss function with regards to every single model parameter and then weights and biases are updated in order to minimize a function loss. The detailed steps of the algorithm are the following:

- input data is grouped into *mini-batches*;
- forward pass: each mini-batch is passed into the input layer and then to the first hidden layer. The output of all neurons is computed and then passed to the next layer, and this step is repeated until the last layer is reached;
- backward pass: a *loss function* is computed to measure the network's error, and also how much each output connection contributed to the error, thanks to the *chain rule*, and this step is repeated until the input layer is reached;
- in order to improve the connection wieghts and biases of the network, the algorithm performs a *Gradient descent*: it measures the local gradient of the loss function with regards to the parameter vector, and it goes in the direction of descending gradient.

A single forward and backward pass of all the training instances passed through the network is called *epoch* and is denoted by ϵ. A DNN requires lots of epochs to get good prediction results.

To speed up the process the step function used in perceptron is replaced by a different *activation function* ϕ. Some popular activation functions are:

- logistic[1] function: $\sigma(z) = \frac{1}{1+e^{-z}}$
- hyperbolic tangent[1] function: $\tanh(z) = 2\sigma(2z) - 1$
- rectified linear unit[2] function: $ReLU(z) = \max(0, x)$
- softmax[2] function: $\sigma(z)_i = \frac{e^{z_i}}{\sum_{i=0}^{K} e^{z_i}}$ for $i = 1,2,\ldots,K$
- SeLu[2] function: $f(z) = \begin{cases} \lambda z & \text{if } z \geq 0 \\ \lambda\alpha(\exp(z) - 1) & \text{if } z < 0 \end{cases}$ with $\alpha \approx 1.6733$ and $\lambda \approx 1.0507$.

where z is the vector of inputs and K is the number of output classes. In ANN, when applied to regression problems, it is common to avoid using activation functions to prevent output values from being restricted within a certain range determined by the activation function.

To evaluate the performance of the system in regression tasks, the *mean squared error* (MSE) is usually chosen as the loss function. This is because it provides an estimate of the average error in the system's predictions, with a greater emphasis on larger errors. The *mean absolute error* (MAE) is the arithmetic mean of the absolute deviations between the predicted values and actual observations in the test sample, where each deviation is given equal weight. Thus, it is used to reduce the effect of outliers. The *root mean squared error* (RMSE) is a scoring rule that measures the magnitude of the error between predicted and actual observations, by taking the square root of the average of the squared differences between the predicted and actual values: since the errors are squared before they are averaged, the RMSE as the MSE, gives a relatively high weight to large errors. Therefore, RMSE should be more useful when large errors are particularly undesirable. When it comes to interpreting the results, MAE is considered to be more straightforward. On the other hand, RMSE not only measures the average error but also considers the magnitude of the error through the squared differences between prediction and observation, which can be harder to interpret. Moreover, RMSE does not necessarily increase with the variance of the errors, but increases with the variance

[1]For single-label/multi-label binary classification.

[2]footnoteFor multi-class single label classification.

of the frequency distribution of error magnitudes. Another aspect to take into account is that as the size of the test sample increases, the difference between RMSE and MAE tends to become greater. This can create issues when comparing RMSE results calculated on differently sized test samples, which is a common occurrence in real-world modeling. However, a benefit of using RMSE over MAE is that it eliminates the need for absolute values, which can be problematic in certain mathematical calculations. To combine the effects of MAE and MSE, the *Huber loss* can be used: it gives quadratic weights for small values of errors and linear weights for larger values.
The mathematical expressions for the above errors measures are given below:

$$MAE = \frac{1}{N}\sum_{k=0}^{N}|SOC_k - SOC_k^*|$$

$$MSE = \frac{1}{N}\sum_{k=0}^{N}(SOC_k - SOC_k^*)^2$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=0}^{N}(SOC_k - SOC_k^*)^2}$$

$$L_\delta = \begin{cases} \frac{1}{2}(SOC_k - SOC_k^*)^2 & if\ |(SOC_k - SOC_k^*)| < \delta \\ \delta((SOC_k - SOC_k^*) - \frac{1}{2}\delta) & otherwise \end{cases}$$

where N is the length of the sequence, δ is a threshold and SOC_k and SOC_k^* are the estimated and true values of the battery's state of charge.

Recurrent Neural Networks (RNNs) are a type of artificial neural networks designed to process sequential data such as time-series data or natural language processing. Unlike traditional feedforward neural networks, RNNs have a feedback loop that allows information to persist across time steps, making them particularly effective for analyzing time-dependent data.
The basic building block of an RNN is a cell, which takes an input vector and a hidden state vector from the previous time step as inputs, and produces an output vector and a new hidden state vector as outputs. The new hidden state vector is then fed back into the cell at the next time step.
RNNs have been successfully applied to a wide range of applications, including speech recognition, machine translation, text generation, image captioning, and music generation, among others. They continue to be an active area of research, with ongoing efforts to improve their performance and scalability.

Long short term memory (LSTM) is a type of RNN which introduces an intermediate type of storage via the memory cell: simple recurrent neural networks possess two types of memory. The first type is long-term memory, which is encoded in the weights that change gradually during training and reflects the general knowledge

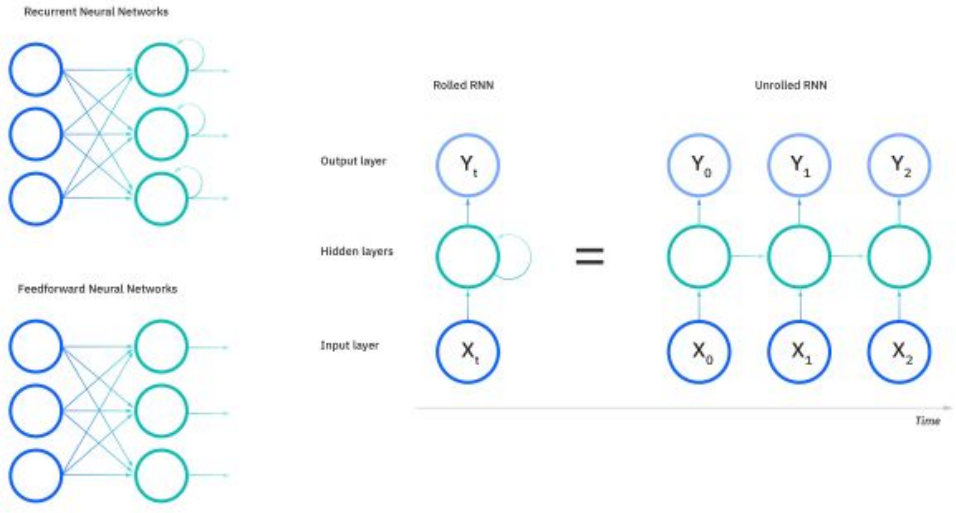


Figure 2.2: Comparison between recurrent and non recurrent neural network.

about the data. The second type is short-term memory, which is facilitated by the activation functions that pass information from one node to the next nodes and has a transient nature. A memory cell is a complex building block constructed from simpler nodes following a particular pattern of connections. It includes multiplicative nodes and comprises an *internal state* represented by H_k, along with three gates that control the flow of information into and out of the cell. In particular those gates determine whether (1) a given input should impact the internal state (the input gate), (2) the internal state should be flushed to 0 (the forget gate), and (3) the internal state of a given neuron should be allowed to impact the cell's output (the output gate). In this way, LSTM will learn to skip irrelevant temporary observations, and to reset the latent state whenever needed.

- input gate $I_k = \mu\left(W_{XI}X_k + W_{HI}H_{k-1} + b_I\right)$ which controls which value of the input should be used to modify the memory. The sigmoid function μ determines which values to allow through 0 or 1;

- memory cell $C_k = F_kC_{k-1} + I_k\tilde{C}_k$ where $\tilde{C}_k = \tanh\left(W_{XC}X_C + W_{HC}H_{k-1} + b_C\right)$ is the input node whose tanh function gives weights to the values which are passed, deciding their level of importance ranging from -1 to 1; this means that the memory cell input gate governs how much of new information we take into account, and the forget gate addresses how much of the old cell internal state C_{k-1} we retain;

- forget gate $F_k = \mu\left(W_{XF}X_k + W_{HF}H_{k-1} + b_F\right)$ that regulates the details to be discarded from the block using the sigmoid function μ;
- output gate $O_k = \mu\left(W_{XO}X_k + W_{H_O}H_{k-1} + b_O\right)$ that is the result of the input and the memory cell gate. Again here, the sigmoid function can be zero-valued, so it can inhibit the flow of information to the next computational node;
- hidden state of the cell $H_k = O_k \tanh C_k$ that computes the output of the cell as it will be seen by other layers. The tanh function assigns weights to the input values, determining their degree of significance ranging from -1 to 1. If the output gate is close to 1, the memory cell's internal state can influence the subsequent layers without any hindrance. On the other hand, if the output gate has a value close to 0, the current memory is prevented from affecting other layers of the network at the current time step.

Each gate has its set of network weights denoted by W and a bias b is added at each matrix multiplication to increase the flexibility of the network to the data. X denotes the vector of inputs to the network, $X_k = [V(k), I(k), T(k)]$ which are the battery voltage, current, and temperature, measured at time step k.

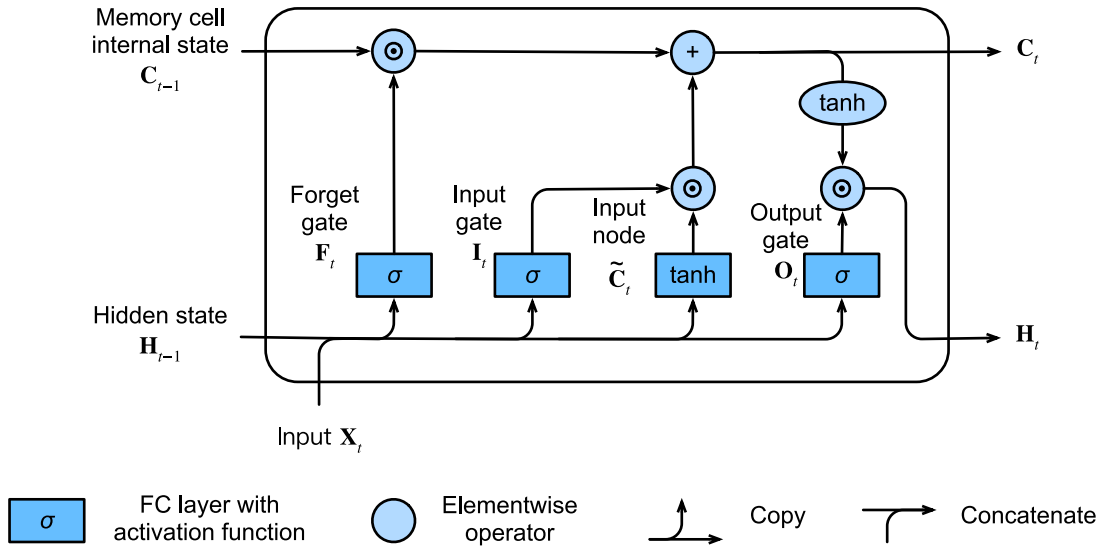


Figure 2.3: Data flow of information through the hidden state of an LSTM model.

The detailed process is showed in Figure 2.3: data fed into the LSTM gates are the input X_k and the hidden state of the previous time step H_{k-1}; then the forget gate controls how much of the information coming from previous memory cell internal state C_{k-1} the model should retrieve; the input gate controls how much of the inputs (input state X_k and previous hidden state) the model should consider

and then those inputs are combined with the output of the forget gate; finally the outputs (the current memory cell internal state C_k and the current hidden state H_k are computed and passed to the next memory cell.

Neural networks are highly adaptable, but this adaptability can also be a disadvantage. There are several hyperparameters to fine-tune, including the number of layers, the number of neurons per layer, the type of activation function, and the weight initialization logic. Despite active research in hyperparameter tuning, it is still helpful to have a rough idea of reasonable values for each hyperparameter to build a quick prototype and limit the search space. Fortunately, some tools are available to aid in this task.

In many cases, beginning with a single hidden layer can yield satisfactory results. In fact, it has been demonstrated that a multilayer perceptron (MLP) with just one hidden layer can effectively model even the most complex functions, provided that there are enough neurons present. Conversely, deep neural networks exhibit much higher parameter efficiency compared to shallow networks: with exponentially fewer neurons, they can model complex functions and achieve significantly better performance with the same amount of training data. The lower hidden layers of a deep network model low-level structures, while intermediate hidden layers combine these low-level structures to model intermediate-level structures. The highest hidden layers and the output layer combine these intermediate structures to model high-level structures. This hierarchical architecture not only helps deep neural networks converge faster to an optimal solution, but also improves their ability to generalize to new datasets. To summarize, for many problems, starting with just one or two hidden layers will suffice. For more complex problems, gradually increasing the number of hidden layers until overfitting the training set occurs, is a viable approach.

For what concerns the number of neurons, in the input and output layers it is determined by the type of input and output of the problem. In the past, it was common practice to arrange the hidden layers in a pyramid shape with progressively fewer neurons at each layer. The earlier idea that merging many low-level features into fewer high-level features was effective has lost popularity. It seems that using the same number of neurons in all hidden layers performs just as well, if not better, in most cases. This approach also has the added benefit of reducing the number of hyperparameters that need to be tuned since there is only one hyperparameter for all layers. Like with the number of layers, the number of neurons can gradually be increased until the network begins to overfit, or a simpler approach is to choose a model with more layers and neurons than necessary and then use *early stopping*[3]

[3]Is a callback (an argument of the *fit* method that let the user specify a list of objects that Tensorflow will call before and after the training phase, epoch and batch processing) that interrupt the training phase if no pregress is measured on the validation set, on a specified number of

to prevent overfitting.
The learning rate is a critical parameter in online learning systems as it determines how quickly the system adapts to new data. When the learning rate is set up too high, the system may quickly adapt to new data but forget old data. On the other hand, if the learning rate is too low, the system may have more inertia and learn slowly, but it will also be less sensitive to noise or non representative data points. To find the optimal learning rate, a common approach is to start with a large value and gradually reduce it until the training algorithm stabilizes. Typically, the optimal learning rate is about half of the maximum learning rate, and dividing the initial learning rate by three at each step is a good heuristic for finding it. Another common technique is to use a non-constant learning rate: the algorithm starts with a large learning rate and then reduces it, according to some rules, until it stops making fast progresses. There are many different approaches to reduce learning rate in training phase, and are called *learning schedules*. The most used are the following:

- power scheduling $\nu(t) = \nu_0/(1 + t/s)^c$: the learning rate drops at each step. After s steps it is down to $\nu_0/2$, then after s more steps it is $\nu_0/3$, and so on, thus this learning schedule reduces learning rate quickly at the beginning and then more and more slowly. The power c, steps s and the initial learning rate ν_0 are hyperparameters to be tuned;

- exponential scheduling $\nu(t) = \nu_0 f^{(t/s)}$: the learning rate drops by a specified factor f every number of s steps. It is similar to power scheduling, but the dropping is faster;

- piece-wise constant scheduling: it starts by setting a constant learning rate for a fixed number of epochs and then it is dropped to a lower constant rate for a fixed number of epochs, and so on;

- performance scheduling: it measures the validation error every n steps and when the error stops dropping, the learning rate is reduced by a user specified factor.

As mentioned at the beginning of Section 2.1, training a NN require a large amount of training data. The main problems that can arise during the training phase are the two following:

- *vanishing gradients problem*: the backpropagation algorithm operates by moving from the output layer to the input layer, while carrying the error gradient along the way. The gradient decreases continuously as the algorithm

epochs.

proceeds through the lower layers. So the lower layer connection weights are left unchanged by gradient descent and training never converges (gradients can also become bigger and bigger and this usually happens in RNNs and is known as exploding gradients problem);

- *training* can be very *slow* because of large amount of input data and also for the high number of parameters whose estimation can easily lead to overfitting;

- *overfitting*: NNs usually have a large number of parameters, which can range from tens of thousands to millions. This contributes to give them a lot of flexibility, and enables them to fit diverse and complex datasets. However, this high flexibility also makes them vulnerable to overfitting on the training set. To address this issue, regularization techniques are needed.

One way to deal with those problems is to choose the correct weight initialization combined with a proper activation function. Researchers demonstrated that the combination showed in Table 2.1 can be useful to help the flow of information in both forward and backward directions, avoiding the vanishing or exploding gradients.

Initialization	Activation function	weights distribution $\sim \mathcal{N}(0,\, \sigma^2)$
Glorot	None, Tanh, Logistic, Softmax	$\sigma^2 = \frac{1}{n_{avg}}$
He	ReLu and its variants[4]	$\sigma^2 = \frac{2}{n_{in}}$
LeCun	SeLu	$\sigma^2 = \frac{1}{n_{in}}$

Table 2.1: Initialization weights for different activation functions.

where n_{in} is the number of inputs and $n_{avg} = (n_{in} + n_{out})/2$ with n_{out} is number of neurons of the layer considered.
Another way to address the vanishing/exploding gradients problem is a technique called *batch normalization*: this technique involves performing zero-centering and normalization on each input, followed by scaling and shifting the normalized values using two additional parameter vectors per layer, one for scaling and the other for shifting. Essentially, this approach enables the model to determine the optimal mean and scale for each input in the layer. This operation is performed just before or after the activation function of each hidden layer. To normalize and zero-center the inputs, the algorithm requires to estimate the mean and standard deviation of each input over the present mini-batch. During training, the batches should not be too small, if possible more than 30 instances, and all instances should be

IID[5]. The vanishing gradients problem is strongly reduced with this technique and convergence is much faster, but there is a runtime penalty: the neural network makes slower predictions due to the extra computations required at each layer.

The above mentioned batch normalization technique, acts also as a regularizer. Moreover one of the most popular regularization techniques is the *dropout*. During each training step of NNs, a dropout technique is applied by setting the output of some neurons to zero. This means that, for a given training step, a neuron may be temporarily "dropped out", and it will be ignored during this step, but it may be active during the next step. The probability of a neuron being dropped out is defined by a hyperparameter p, which is commonly set to 50% and is called *dropout rate*. This technique is applied to every neuron in the network, except for the output neurons. Once the training process is completed, the neurons are no longer dropped out. The basic idea of this tecnique is that neurons trained with the dropout cannot update with their neighbors, but have to be as useful as possible for the network on their own. This results in the network becoming less responsive to minor variations in the input data and, thus, a more resilient and adaptable network is achieved, leading to better generalization performances. Therefore, although dropout can slow down convergence considerably, it often leads to a superior model when the hyperparameters are tuned appropriately. Thus, the additional time and effort required for dropout are usually justified.

The data modeling for the battery's behaviour and its adaptive control technology requires expert system theories and artificial intelligence in the modeling process. Given the amount of data generated by energy storage systems, it is natural to consider machine learning algorithms for state and parameter estimation.

[5]Independent and identically distributed.

Chapter 3

Proposed data driven approach

In this chapter there is a detailed description of the approach used to obtain SOC estimation: it starts from Section 3.1 which describes the collected data; then Section 3.2 illustrates the data preprocessing steps and the network optimization techniques adopted.

3.1 Opensource dataset overview

Data are at the heart of disruptive developments in battery development, modeling, and management. Starting from design and sale, to deployment and management, and across all the value chain, data plays a key role to make up decisions at all stages of a battery's life. Data-informed methods were employed in the design stage to expedite slower discovery procedures like component creation and production enhancement (for electrodes, electrolytes, additives and formation). At the point of sale, batteries can be categorized based on their anticipated lifespan and performance capabilities. During deployment, utilizing data on the anticipated lifespan and efficiency of batteries across different chemistries, shapes, capacities, and manufacturers can aid in identifying the most suitable battery for a specific application, under different ageing stresses such as various charge/discharge currents, operating temperatures, depth of discharges (DODs), and periods of disuse.

The battery management system (BMS) used to control battery operation, as already mentioned in Section 1.2, requires data both for its development, and for training and calibrating the models it uses to estimate battery states such as state of health, state of charge, remaining useful life (RUL).

Batteries are subjected to a wide range of operating conditions influencing their performance, and thus, data covering these conditions is crucial to the design and

validation of accurate models. EV battery requirements differ from those for laptops, cell phones, stationary energy storage, and other devices. Thus, application-specific data are needed for different analysis:

- cycle aging data: generating cycling data that covers the entire lifespan of a battery demands a substantial investment of time and resources, which could span several months or even years. For this reason specific experiments to investigate the influence of in-cycle factors, such as charging and discharging current, temperature, depth of discharge (DOD), on the capacity retention are needed. Typically, input data include in-cycle measurements of current, voltage, and temperature and per-cycle measurements of capacity and internal resistance or impedance;

- drive cycle data: standardised testing procedures capturing the dynamic power demands of driving are indispensable for GHG monitoring and to reduce carbon emissions. These are referred to as driving cycles: they require standardised dynamic vehicle drive schedule encoded by a velocity-time table/profile. The predetermined time steps set the velocity and acceleration, making the necessary mechanical power a time-dependent variable. The total energy required for a given driving cycle can be calculated by integrating the mechanical power over the duration of the driving schedule. BMS for electric vehicles is the one that requires that mechanical energy. Then, input data is gathered by cycling batteries based on the driving schedules;

- chemistry cell modeling: it is used to analyse performance of different lithium battery chemistries under different operating conditions of temperature, current and age. To assess the suitability of lithium batteries for a particular application, it is essential to consider various aspects of battery performance, including the OCV-SOC table, impedance, and internal resistance. The cycling characteristics obtained from battery cycling can then be utilized to construct models that describe the electrical dynamics and cycling performance of the battery. The experimental data collected primarily involves short-term responses of current and voltage, with a focus on changes in impedance at various battery state of charge levels and temperatures;

- calendar aging: comprises all ageing processes that lead to the degradation of a battery cell independent of charge-discharge cycling. This aging effect is more relevant in applications where batteries have periods of idleness longer than periods of operation. Datasets dedicated to calendar aging include information related to battery cycler such as voltage, current, capacity, and energy from periodic characterization tests.

Various countries and organizations create driving cycles, which are employed to evaluate vehicle performance in terms of factors like pollutant emissions, fuel

consumption, and traffic impact. According to the driving schedules, a variety of data collected by cycling batteries are used for SOC estimation algorithms under realistic conditions. The globally accepted driving cycle tables can be divided into European, American, and Asian driving cycles.

Most used in literature are the following American (US) driving cycles:

- Highway fuel economy test (HWFET) is a chassis dynamometer driving schedule representing highway driving conditions under 60 mph, for the determination of fuel economy of light-duty vehicles over highway driving cycle;

- Federal test procedure (FTP-75) has been created by US EPA (Environmental Protection Agency) to represents a commuting cycle that includes both urban driving with frequent stops and highway driving;

- US06 is a supplemental federal test procedure cycle that represents aggressive, high speed and/or high acceleration driving behavior;

- LA92 unified dynamometer driving schedule was developed as an emission inventory improvement tool, and is for Class 3 heavy-duty vehicles (power-to-mass ratio is greater than 34);

- Urban dynamometer driving schedule (UDDS) is also known as "the city test" and represents city driving conditions. It is used for light-duty vehicle testing.

The dataset used for training, testing and validation of the network was released by the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, and is freely available online [1].

Brand	LG
Cell dimensions	18mm and 65mm
Chemistry catode	Li[NiMnCo]O_2
Chemistry anode	Graphite + SiO
Nominal capacity	3.0 Ah
Energy density	240 Wh/kg
Charge method	Nominal: 1.5A 4.2V, 50mA End-current (CC^1-CV^2) Fast: 4A 4.2V, 100mA End-current (CC-CV)
Discharge	End voltage: 2V Max current: 20A (continued discharge current)

Table 3.1: Battery specifications.

The 3Ah LG 18650HG2 cell, whose technical specifications are summarized in Table 3.1, was tested with the equipment showed in Figure 3.1: the control computer holds the driving schedules databases, and once the thermal chamber is set to the desired ambient temperature and the cell is completely charged, the system commences recording the current drive cycle. The thermal chamber used is a 75A, 5V Digatron Firing Circuits Universal Battery Tester channel, with a voltage and current accuracy of 0.1% of full scale. Various tests were performed at six different temperatures (one of those is 25°C, the only one used in this work). After each test the cell was charged at 1C rate until 4.2V was reached, with a cut-off of 50mA, and maintaining the battery temperature at 22 degrees Celsius or higher. The tests were performed as follows. Series of four drive cycles performed, in following order: UDDS, HWFET, LA92, US06. A series of eight drive cycles (mix 1-8) consist of random mix of UDDS, HWFET, LA92 and US06. The drive cycle power profiles are repeated until 95% of the 1C discharge capacity at the respective temperature has been discharged from the cell. The drive cycle power profile was calculated for a single LG HG2 cell in a compact electric vehicle.

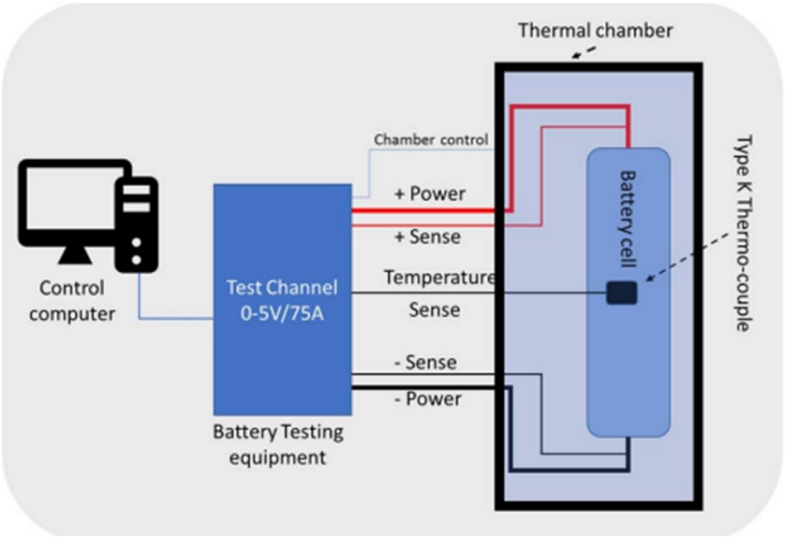


Figure 3.1: Battery test equipment.

The cycler collects the following measurements:

- Time (time in seconds)

[1]constant current charging

[2]constant voltage charging

- TimeStamp (timestamp in MM/DD/YYYY HH:MM:SS AM format)
- Voltage (measured cell terminal voltage, sense leads welded directly to battery terminal)
- Current (measured current in amps)
- Capacity (measured amp-hours, with Ah counter, typically reset after each charge, test, or drive cycle)
- Energy (measured watt-hours, with Wh counter, reset after each charge, test, or drive cycle)
- Battery_Temp_degC (battery case temperature, at the middle of battery, in degrees Celsius)

3.2 NN structure

The data modeling for the battery's behaviour and its adaptive control technology requires expert system theories and artificial intelligence in the modeling process. Given the amount of data generated by energy storage systems, it is natural to consider machine learning algorithms for state and parameter estimation.

The model is implemented in Python 3.10.6 version with Tensorflow 2.9.1 package: it is built sequentially, starting with the input layer, then an LSTM layer, and finally, a dense layer that computes the output. The parameters of the network, summarized in Figure 3.2, were kept constant in all experiments in order not to influence the results.

Because parameters optimization is very complex and require lots of effort, network structure and parameters were chosen accordingly to the guide lines contained in [2]. This is the reason why those parameters are maintained constant and are not further improved in this work. The learning rate function scheduling is defined according to the factors in Figure 3.2 as follows: every 100 epochs the initial learning rate is dropped by 0.5, as showed in Figure 3.3.

Data pre-processing

The train and test set are chosen as follows:

- train: is composed of the eight drive cycle mix and their respective charges;
- test: is composed of three drive cycles in the following order UDDS, LA92, US06 and their respective charges.

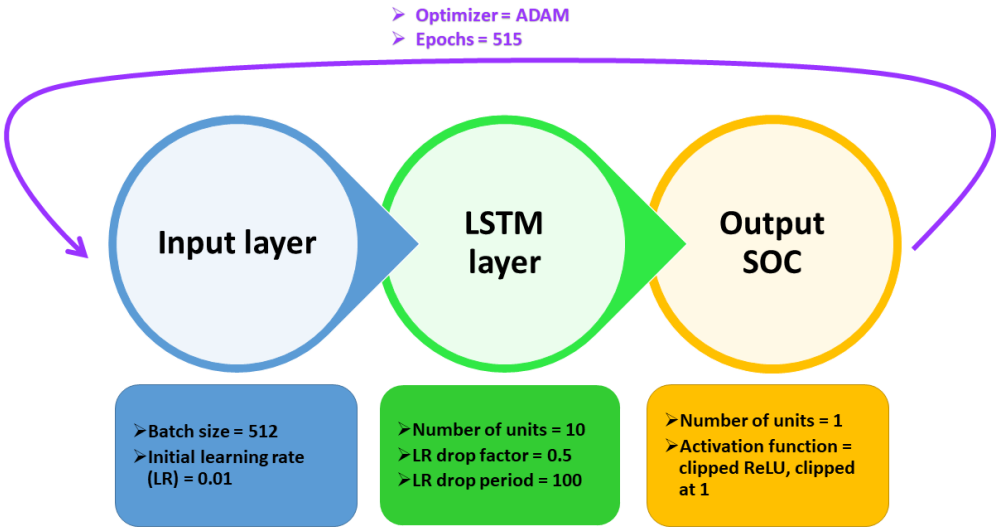


Figure 3.2: Network layers with corresponding parameters.

```
# This function keeps the initial learning rate for the first 100 epochs
# and decreases it of 0.5 factor after every 100 epochs.
def scheduler(epoch, lr):
  if epoch%100 == 0 and epoch != 0:
    return lr * 0.5
  else:
    return lr
```

Figure 3.3: Function for learning rate scheduling.

All those measurements are collected during tests with a thermal chamber temperature of 25°C.
The first step performed on data pre-processing is the feature selection: only battery related data are selected, while cycler data are discarded. Then relevant features are selected with the correlation analysis.
In figure 3.4 voltage, current, capacity, energy, and temperature correlation with the response variable SOC is showed. The computed correlation coefficient is the Pearson's one: it measures how much the relationship between variables is close to a linear function. The more the absolute value of the coefficient is higher, the more the correlation is stronger and the feature strongly affects the prediction. It ranges from -1 to 1. If it is close to 1, it indicates that there is a strong positive correlation: for example the voltage goes up when SOC goes up. On the other hand, when it is close to -1 there is a strong negative correlation: for example when SOC goes down battery temperature tends to go up. When coefficient is close to zero, it means

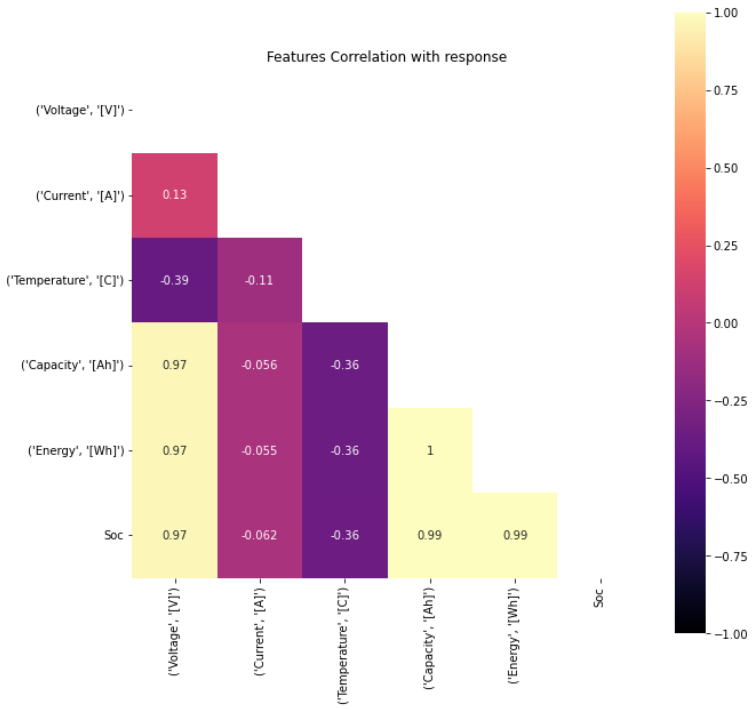


Figure 3.4: Correlation between features and response for data collected at 25°C.

that there is no relevance of linear correlation. The response variable, the state of charge of the battery, is obtained from the capacity (Ah) of the battery calculated by the cycler. This is the reason why both capacity and energy have been excluded as predictors.
By looking at the correlation matrix in Figure 3.4 it seems that there is no correlation between the response variable and the battery temperature and current. Moreover, by looking at the correlation matrix computed on data collected at a lower temperature of -10°C, showed in Figure 3.5, it can be immediately noticed that correlation between current and voltage increases a lot, and also correlation between current and SOC increases. This effect is due to the fact that LIB batteries have a non-linear nature: researchers have observed that, at lower temperatures, battery's internal resistance increases and this produces a voltage drop, so the battery charge/discharge cycle will be shorter. Moreover, the temperature increases significantly due to the loss related to the increasing resistance.

The US06 driving schedule is more aggressive compared to the UDDS which represent city driving conditions. This is the reason why current distribution represented in Figure 3.6 is different: the current distribution for the US06 driving schedule is more widespread because of the high accelerations and high speed driving behaviour. The effect is better represented by the violin plot in Figure 3.7.

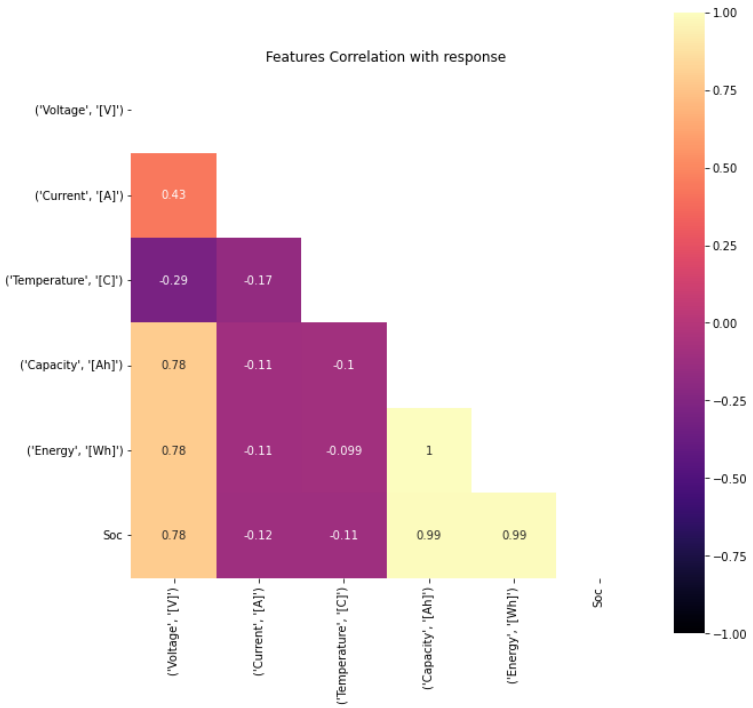
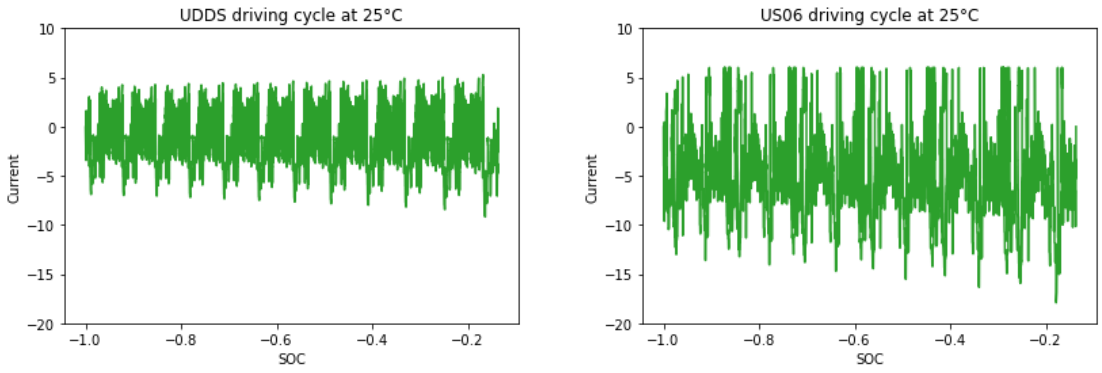


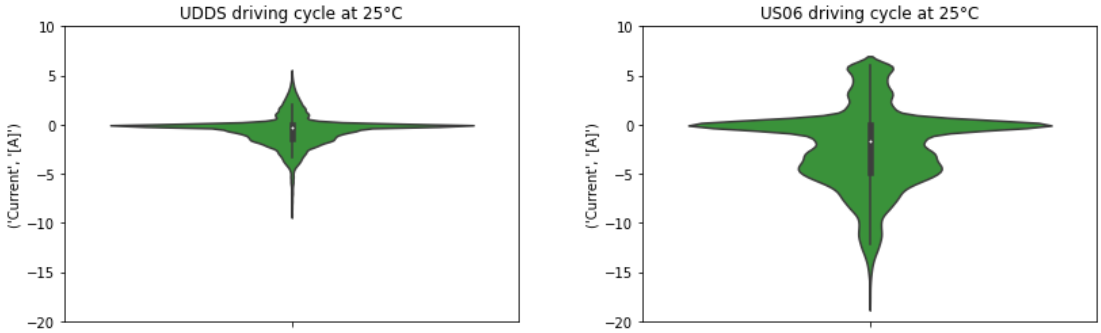
Figure 3.5: Correlation between features and response for data collected at -10°C.



(a) **(b)**

Figure 3.6: Measured current for UDDS and US06 driving cycles performed at 25°C.

A violin plot is a method for plotting numeric data that is similar to a box plot, but with the addition of a rotated kernel density plot on each side. Essentially, a violin plot conveys the same information as a box plot, but also provides an estimate of the probability density of the data at different values, usually obtained



(a) **(b)**

Figure 3.7: Violin plot of current for UDDS and US06 driving cycles performed at 25°C.

using a kernel density estimator. A violin plot usually contains all the information that can be found in a box plot, such as a median marker, a box or marker that denotes the interquartile range, and, if the number of samples is not excessive, possibly all individual data points. Thus, the driving behaviour clearly influences the performance of the battery: when the driving style is more aggressive with lots of accelerations and high velocity, the current necessary to let the battery traction the vehicle is greater.
This effect produces a voltage drop as showed in Figure 3.8: the duration of the US06 cycle decreases significantly. This is the reason why current is still maintained as predictor for SOC, even if the correlation coefficient is very low, because the battery behaviour is not linear and Pearson correlation coefficient cannot capture this effect.

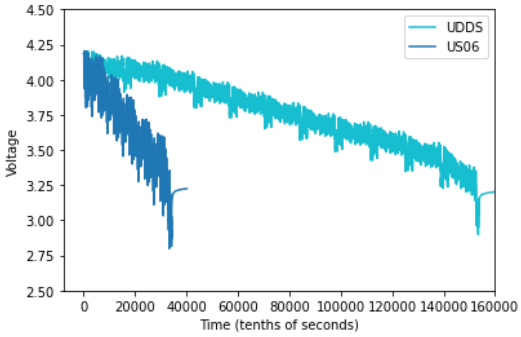
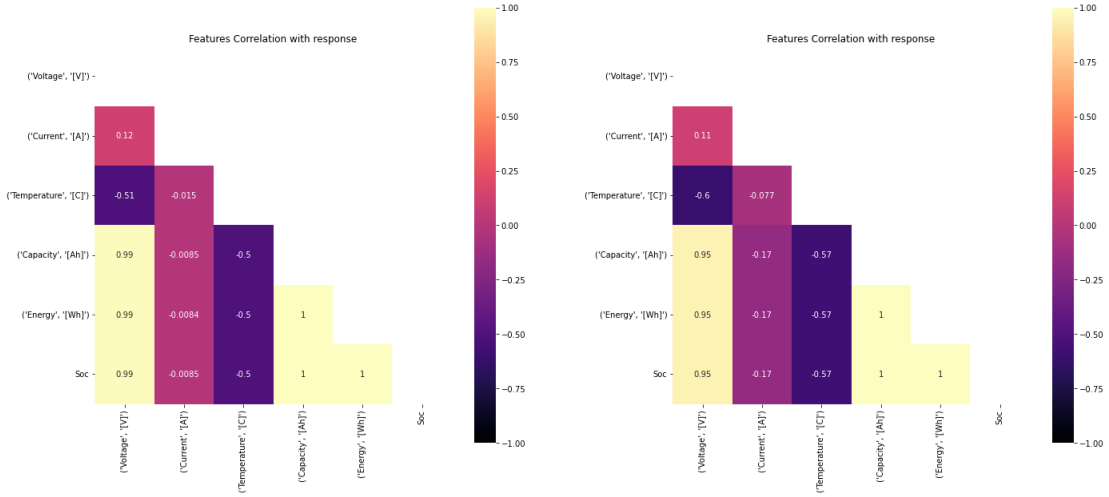


Figure 3.8: Voltage drop for UDDS and US06 driving cycles in time, measured at 25°C.

Figures 3.9 and 3.10 show how the linear correlation between current and SOC and voltage changes, with different ambient temperatures and with different driving cycles. In Figure 3.9 the correlation between current and SOC increases a little bit, because of the aggressive behaviour style. This difference is larger when the ambient temperature is lower (Figure 3.10): the internal resistance of a battery increases with lower temperatures, and as a result, a higher current (i.e., higher amperage) is required to properly operate the battery to maintain a certain speed of a vehicle, and thus, a specific driving behaviour.

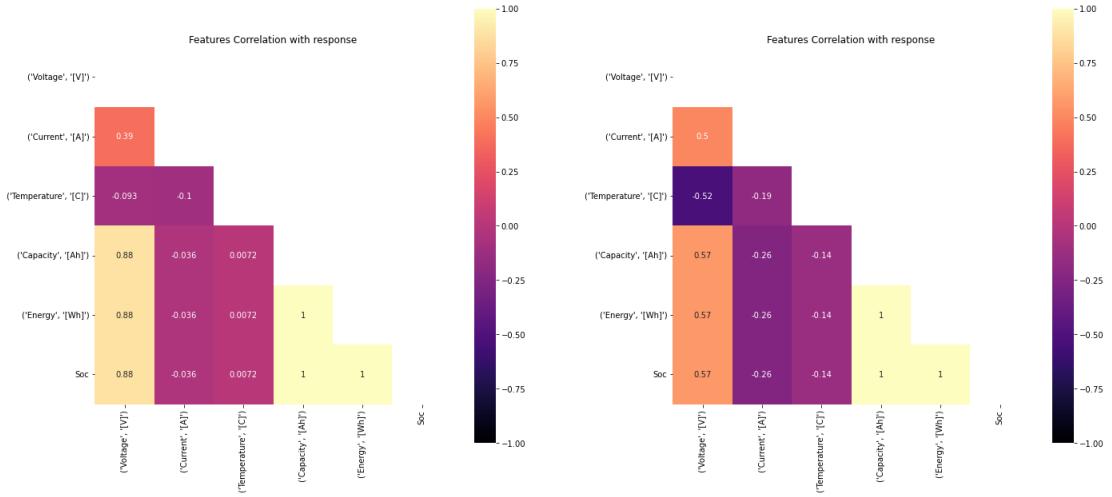


(a) Correlation matrix for UDDS at 25°C. **(b)** Correlation matrix for US06 at 25°C.

Figure 3.9: Comparing correlation matrices for data of UDDS and US06 driving cycles at 25°C.

From the correlation matrices in Figures 3.9 and 3.10 the influence of the battery temperature can be also observed: temperature is negatively correlated with SOC, but this correlation diminishes with lower temperatures, while correlation between voltage and temperature increases. Thus, temperature is also affecting SOC but this effect is not captured by Pearson correlation coefficient: this influence is indirect because lower ambient temperatures causes a voltage drop due to increasing internal resistance and consequently increasing temperature (voltage and battery temperature are negatively correlated) and because voltage and SOC are positively correlated this means state of charge of the battery goes down. When the more aggressive driving cycle US06 is considered, this effect is amplified. In Figure 3.11 can be observed the effect of temperature: at lower ambient temperatures, the driving cycle duration diminishes and the battery temperature rises a lot at -10°C.

Figure 3.12 shows the effect of increasing temperature for the UDDS driving cycle: temperature oscillates and the oscillations are more widespread at lower



(a) Correlation matrix for UDDS at -10°C. **(b)** Correlation matrix for US06 at -10°C.

Figure 3.10: Comparing correlation matrices for UDDS and US06 driving cycles at -10°C.

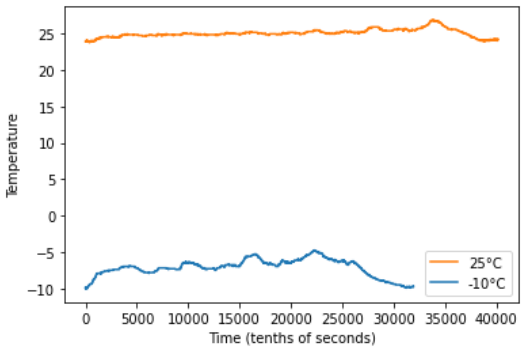
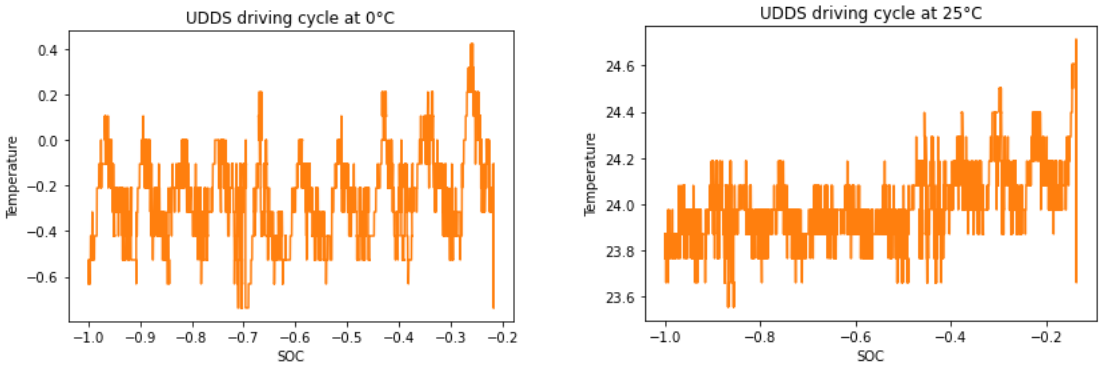


Figure 3.11: Temperature behaviour for LA92 driving cycle.

ambient temperatures. Moreover, a little increase in temperature values can be noticed when SOC approaches zero.

The collected data from the cycler have been saved with different sampling frequencies: data related to driving cycles and mix of driving cycles are collected with a frequency of 10 Hz (the time step is 0.1 seconds). On the other hand, data related to charging phase were considered less important and were collected at a lower rate of one observation every minute. Because both training and test set contains data related to drive cycles and charges, both must be up-sampled to obtain the same frequencies. The two data granularities considered are: one observation per minute and one observation per second. In both cases down/up



(a) UDDS at 0°C. **(b)** UDDS at 25°C.

Figure 3.12: Temperature behaviour for UDDS driving cycle.

sampling are needed, thus some aggregation measures that are mean and standard deviation of voltage and current are computed, to avoid past information loss. When up-sampling a forward fill is used: it propagates last valid observation forward, to fill the missing values.

```
X_train.index =  pd.to_datetime(pd.to_timedelta(X_train.index)
                                .values.astype('datetime64[60s]'))
df2 = X_train.asfreq('1S').ffill()
##aggregation measures
df0 = X_train.rolling('1S', min_periods = 1).mean()
df0 = df0.iloc[:, 0:2]
df0.rename(columns={'Voltage': 'MeanV', 'Current': 'MeanC'}, inplace=True)
df1 = X_train.rolling('1S', min_periods = 1).std()
df1 = df1.iloc[:, 0:2]
df1.rename(columns={'Voltage': 'StdV', 'Current': 'StdC'}, inplace=True)

X_train_final = df2.join([df0, df1])
```

Figure 3.13: Code snippet of sampling and aggregation.

The above Figure 3.13 shows the steps followed to obtain a training set with observations' frequency 1Hz. First of all the index of *X_train*, tha training set with the selected features, is converted to the correct format. Then it is sampled to 1Hz (one observation per second) and eventually missing values are filled with the forward fill. Then aggregated mean and standard deviation are computed with the rolling method, which provides rolling window calculations. Only first two columns (the ones corresponding to voltage and current) are saved. Finally the sampled dataset $df2$ and the two with the aggregated measures computed $df0$ and $df1$ are

joined.

As mentioned in Section 2.1.4, NNs does not work well with unbounded inputs and are more addressed to the exploding gradients problem. This is the reason why a normalization technique in necessary to reduce this effect and to speed up the training phase. The Mix-Max scaling normalization technique is applied to the input features, in order to obtained values bounded between 0 and 1, which are more suitable to NNs.

$$X = \frac{x - min(x)}{min(x) - max(x)},$$

where X is the normalized input feature and x is the original value.

Neural networks require apriori knowledge of the response variable (i.e., SOC in this thesis) to be properly trained. Because SOC cannot be directly measured, it has to be estimated. One way to do so, is to use its definition: SOC is the available capacity in a battery expressed as a percentage of nominal capacity (i.e., the one in Table 3.1). Each computed label (SOC value) is associated with the corresponding features for real-time predictions. When the prediction is extended to future time horizons, each label is shifted with respect to its initial position by the number of rows needed to reach that time horizon. For example if data is sampled with a frequency of 1 observation per minute, and the prediction time horizon is 10 minutes, each label is shifted of 10 rows.

Network optimization

The algorithm is trained and tested considering the two sampling frequencies: 1 Hz (one observation per second) and $\frac{1}{60}$Hz (one observation per minute). As explained in Section 3.2, aggregated mean and standard deviation of voltage and current were computed to avoid information loss when down-sampling. Thus, the vector of inputs fed into the network is

$$X_k = [V(k), I(k), T(k), MeanV(k), MeanI(k), StdV(k), StdI(k)]\,.$$

The parameters used to train the network were listed at the beginning of Section 3.2, and were kept constant in all experiments. The main advantage of the LSTM layer is the ability to capture dependencies in time, thanks to the memory cell, which controls how much of the information from the previous neuron should be used in the current memory cell. In Tensorflow LSTM implementation, the described information flow can be controlled by the *recurrent dropout*, whose default value is 0. A recurrent dropout equal to 1 means no information is passed to the next memory cell, from the previous one (i.e., a non recurrent NN). The recurrent dropout does not remove any inputs between layers, as done by the dropout of non recurrent layers, but inputs between time steps, as showed in Figure 3.14. Neural Network models frequently encounter overfitting issues, particularly when

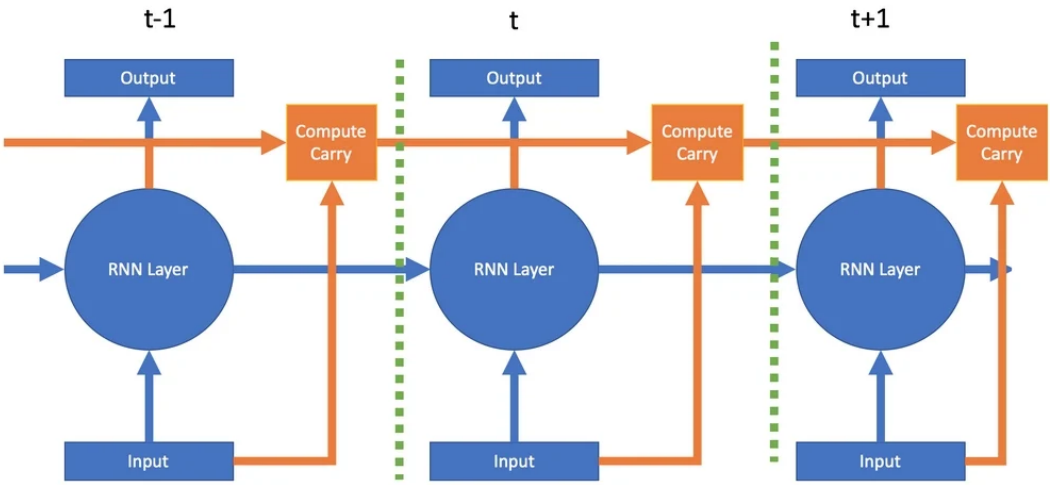


Figure 3.14: Recurrent dropout in RNN structured network.

the number of parameters in the network is large, and the quantity of training data is limited. Recurrent dropout, like standard dropout, has a regularization effect that can prevent overfitting in neural network models. Overfitting can negatively impact model accuracy and decrease its performance on unseen data in real-world applications.

To find the best recurrent dropout value, an optimization tool called *Optuna* has been used: Optuna is a hyperparameter optimization framework that can be applied to machine learning frameworks and black-box optimization solvers. It simplifies the process of hyperparameter tuning by allowing to define the types and ranges of hyperparameters directly within the code using a trial object: an objective function is defined, which generates a numerical value that is used to assess the performance of the hyperparameters and determine where to sample in subsequent trials. This eliminates the need to learn specialized syntax for hyperparameters and allows the user to use standard Python code to loop through or define them. Optuna's default sampler is the Tree-structured Parzen Estimator (TPE), which is a form of Bayesian Optimization. TPE searches more efficiently than a random search by selecting points closer to previous successful results. In contrast to random or grid search, Bayesian approaches use past evaluation results to form a probabilistic model that maps hyperparameters to a probability of achieving a high score on the objective function. The primary objective of Bayesian reasoning is to enhance the accuracy of the model as more data becomes available. This is achieved by updating the probability model after each objective function evaluation. Bayesian methods can evaluate more promising hyperparameters from past results, resulting in better model settings than random search in fewer iterations. This

means that Bayesian model-based methods can find optimal hyperparameters in less time by reasoning about the best combination of hyperparameters based on past trials. Optuna also provides various other features, including visualizations, alternative samplers, optimizers, and pruning algorithms. Users can also create their own customized versions of Optuna.

```
search_space = {"rec_dropout": [0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]}
study = optuna.create_study( directions=['minimize', 'minimize'],
                             study_name="SOC-study",
                             sampler = optuna.samplers.TPESampler(seed = 10) )
```

Figure 3.15: Optuna study settings.

In Figure 3.15 there are the command lines used to create the study: there is a search space for the only hyperparameter to be optimized (i.e., the recurrent dropout), and *optuna.samplers.TPESampler* is used. The study direction is set to let the algorithm minimize the training and validation errors. The goal of this optimization step is to find the best recurrent dropout that minimizes the errors (i.e., MAE and RMSE).

The study is performed for both sampling frequency of 1Hz and $\frac{1}{60}$Hz. For both frequencies two studies were performed: one optimizing training and validation MAE and the other optimizing training and validation RMSE.
When data is down-sampled to $\frac{1}{60}$Hz, as showed in Figures 3.16, 3.18, as expected the training error is always smaller with respect to the validation error, both for MAE and RMSE. The best recurrent dropout that minimizes training errors is clearly 0: the more we have information from the past, the more the algorithm can learn and will be more accurate. Unfortunately lowest training errors does not correspond to lowest validation errors and this is due to overfitting: the model makes accurate predictions for the training data but not for the test data. Figures **??** and **??** show the *Pareto curves*: a Pareto curve is defined as the set of all efficient Pareto solutions. Pareto efficiency refers to a state in which no individual or entity can be made better off without making another individual or entity worse off. The following three concepts are closely related:

- a Pareto improvement is a new situation where at least one individual or entity benefits and no one is worse off compared to the initial situation;
- a situation is said to be Pareto-dominated if there exists a possibility of a Pareto improvement;
- a situation is Pareto-optimal or Pareto-efficient if no change can be made to improve the satisfaction of one individual or entity without reducing the

satisfaction of another individual or entity. In simpler terms, it means that there is no possibility of improving one objective without causing a reduction in the performance of another objective.

From Figures 3.16, 3.18, 3.17 and 3.19 a common value of recurrent dropout that minimizes MAE and RMSE validation errors and correspond to reasonable small training errors is 0.5 (also 0.9 is promising, but after results comparing 0.5 is chosen). Because of the chosen sampler, the algorithms starts with random values for recurrent dropout, sampling from the defined search space, and then uses a "best guess" approach using Bayesian optimization. This is the reason why in the displayed plots, errors measurements are not evaluated for all the values of the search space.

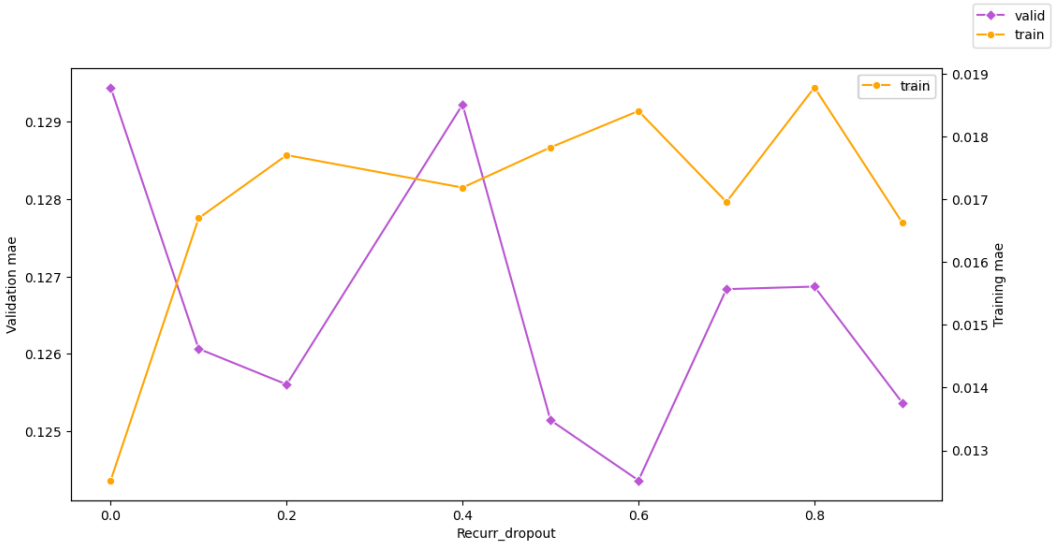


Figure 3.16: Train and validation MAE for sampled data at $\frac{1}{60}$Hz.

When the sampling frequency of data is 1Hz, again training errors are smaller compared to validation errors. From Figures 3.20, 3.21, 3.22 and 3.23 the best recurrent dropout value that minimizes validation MAE and RMSE and correspond to low training errors is 0.7.
It is reasonable that with a fine-grained dataset the dropout value is higher, compared to the one corresponding to the coarse-grained dataset: if measurements are collected too frequently, the values will change slowly and thus can be difficult for the algorithm to derive a proper model and this is the reason why a greater dropout is needed with fine-grained datasets. Moreover, the overfitting risk is higher. Counterwise in a coarse-grained dataset the best recurrent dropout is lower, because if dropout is too high it will cause information loss.

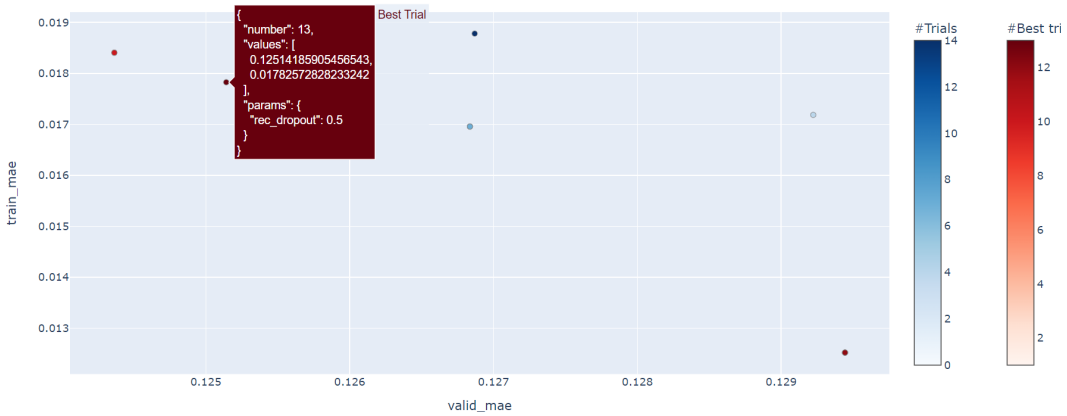


Figure 3.17: Best trial for sampled data at $\frac{1}{60}$Hz.

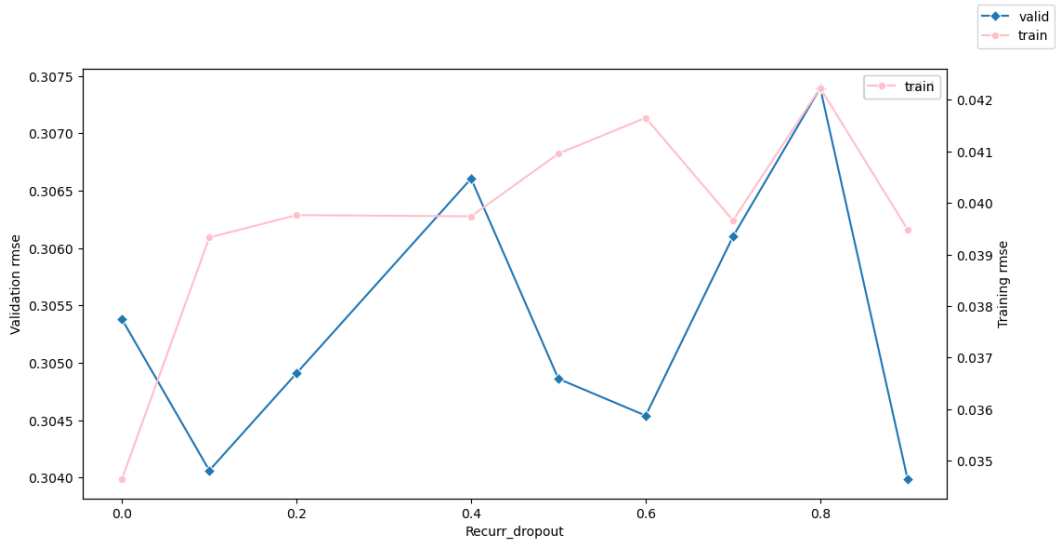


Figure 3.18: Train and validation RMSE for sampled data at $\frac{1}{60}$Hz.

In general, when comparing two models, it's a good idea to use both MAE and RMSE to get a more complete understanding of how the models are performing, and this is the reason why both measures are used to choose the appropriate recurrent dropout. RMSE is calculated by taking the square root of the average of the squared errors between the predicted and actual values. Squaring the errors has the effect of penalizing larger errors more heavily than smaller ones. This means that if there are outliers or large errors in the data, the RMSE will be larger than the MAE. In contrast, MAE is calculated by taking the average of the absolute errors between the predicted and actual values. Absolute errors do not penalize

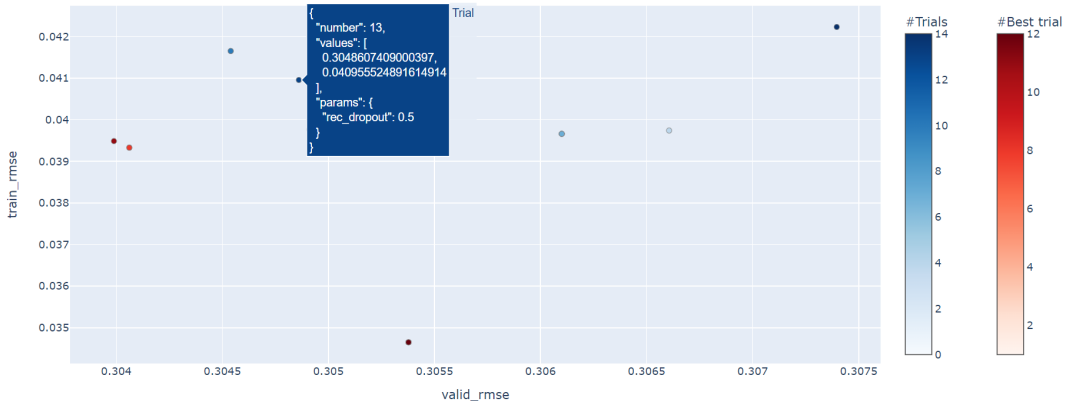


Figure 3.19: Best trial for sampled data at $\frac{1}{60}$Hz.

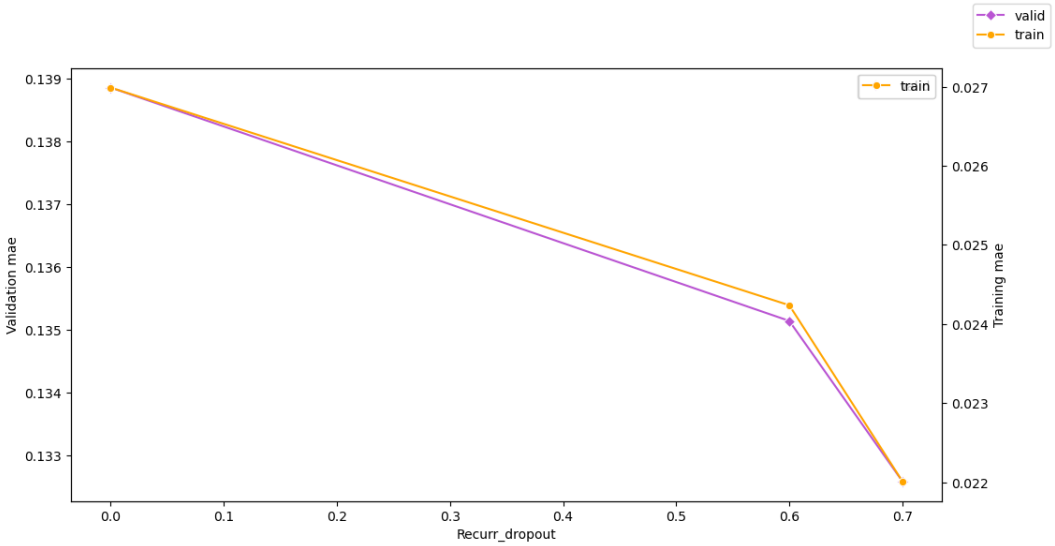


Figure 3.20: Train and validation MAE for sampled data at 1Hz.

larger errors more heavily than smaller ones. Therefore, if there are outliers or large errors in the data, the MAE will be similar to the errors for the other data points and will not be affected as much by the outliers. The RMSE result will always be larger or equal to the MAE. If all of the errors have the same magnitude, then RMSE=MAE, and this is the reason why the first measure minimized in always MAE and then RMSE. As already mentioned minimum training errors does not correspond to minimum validation errors, thus there is not a unique value that minimizes both validation and training errors of MAE and RMSE. This is the reason why the recurrent dropout value chosen does not always correspond to one

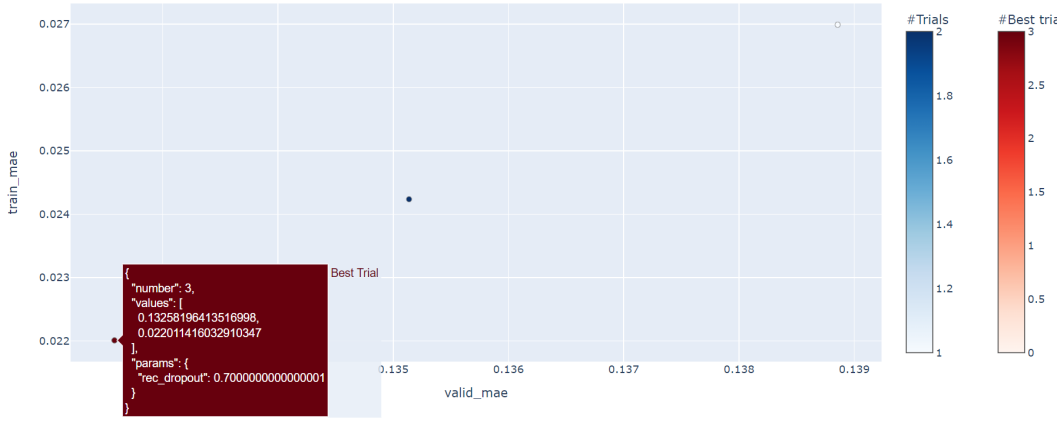


Figure 3.21: Best trial for sampled data at 1Hz.

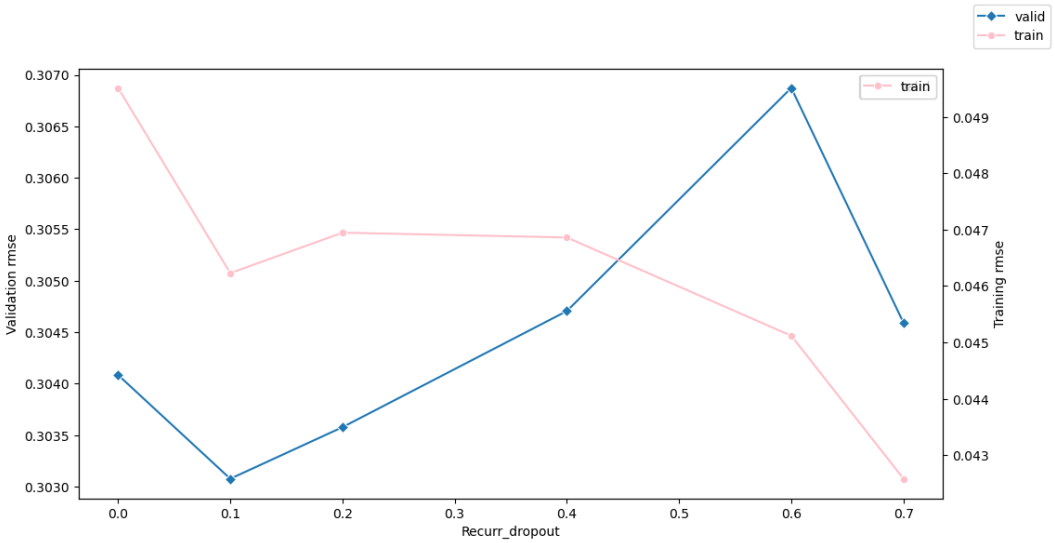


Figure 3.22: Train and validation RMSE for sampled data at 1Hz.

of the best trials, and also the reason why there is more than one best trial, and thus more than one recurrent dropout value that minimizes the corresponding error measure.

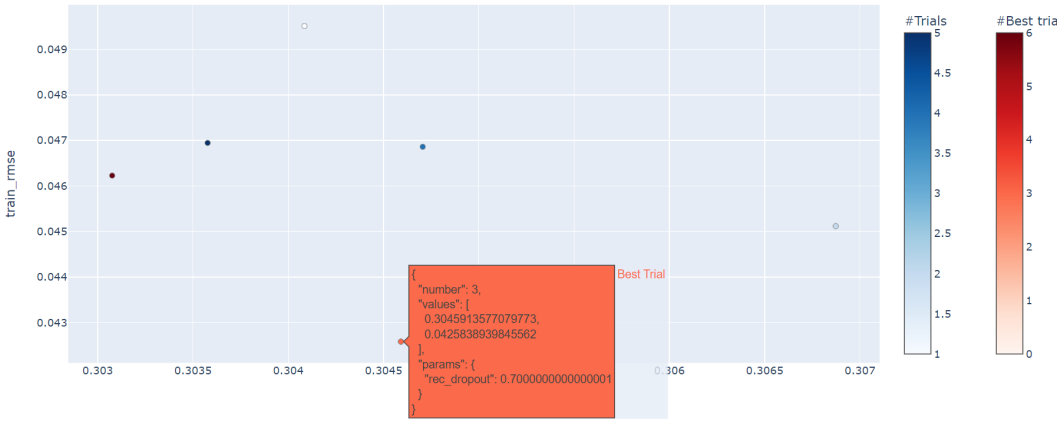


Figure 3.23: Best trial for sampled data at 1Hz.

Chapter 4

Experimental results

After data collection and all the preprocessing steps explained in Section 3.2 are performed, the sequential network is trained. The LSTM layer was set with the optimized dropout explained in previous Chapter 3, with the network parameters listed in Figure 3.2. After training, the network is used to predict SOC on the test set.

4.1 Achieved results

The prediction is performed on the test set, considering for both the sampling frequencies, four distinct time horizons, measured in minutes, denoted by h:

- h = 0 means real-time prediction: using data available at time step k the algorithm tries to predict SOC_k that is the state of charge of the battery at time k;
- h = 10 means that with data available at time step k the algorithm tries to predict SOC_{k+h} that is the state of charge of the battery at time $k + 10min$;
- h = 20 means that with data available at time step k the algorithm tries to predict SOC_{k+h} that is the state of charge of the battery at time $k + 20min$;
- h = 30 means that with data available at time step k the algorithm tries to predict SOC_{k+h} that is the state of charge of the battery at time $k + 30min$.

To evaluate training and prediction performance the same two measures used to find the optimal recurrent dropout are evaluated: MAE and RMSE. The remainder of this chapter presents the achieved results for the two selected sampling frequencies: in Section 4.1.1 there are the errors obtained for data sampled at frequency of 1HZ (one observations per second); Section 4.1.2 shows the errors obtained when data is sampled to $\frac{1}{60}$Hz.

4.1.1 Sampling frequency 1Hz

To account for the different frequencies of stored data for drive cycles (and mix) and charges, training and testing sets are obtained following the steps already explained in Figure 3.13. Drive cycles were down-sampled from 10Hz to 1Hz while charges were up-sampled from $\frac{1}{60}$Hz to 1Hz. The outcomes of the LSTM network are shown in Tables 4.1 and 4.2: the first table reports the training and test errors obtained in each of the four different time horizons. The second one shows the errors obtained applying the optimal recurrent dropout of 0.7.

	h	MAE	RMSE
Training error	0min	0.027	0.050
	10min	0.026	0.049
	20min	0.039	0.070
	30min	0.065	0.117
Test error	0min	0.139	0.304
	10min	0.131	0.296
	20min	0.145	0.292
	30min	0.151	0.267

Table 4.1: Errors obtained with different time horizons (h) with observations frequency of 1Hz.

	h	MAE	RMSE
Training errors	0min	0.022	0.043
	10min	0.026	0.049
	20min	0.045	0.078
	30min	0.075	0.127
Test errors	0min	0.133	0.304
	10min	0.130	0.296
	20min	0.146	0.294
	30min	0.159	0.294

Table 4.2: Errors obtained with different time horizons (h) with observations frequency of 1Hz, with dropout of 0.7.

From Table 4.1, it can be observed that training errors are quite low, especially for time horizons of 0, 10 and 20 minutes. When the prediction is within 30 minutes the training MAE and RMSE doubled with respect to real-time prediction (h=0). The prediction errors on the test set are quite high and the both the errors increase when time horizon becomes far away. Thus, the algorithm is not able to capture SOC values with good accuracy when data is sampled with high frequency and the time horizons are in minutes. When applying recurrent dropout, as showed in Table 4.2, can be observed som improvements only with real-time prediction: training error diminishes, but on the other hand there are not great improvements on prediction. Therefore, even with dropout, if input data is too fine-grained, the algorithm is not able to capture with good accuracy the relation between features and SOC.
This limit is supported by the autocorrelation between rows of the response variable (i.e., SOC). It is computed using the *pandas.Series.autocorr*, as showed in Figure 4.1, which evaluates the Pearson correlation coefficient between a series and its shifted self. It accepts as a parameter the *Lag* that is the number of lags to apply before performing the autocorrelation. The coefficients obtained are the following:

```
#autocorrelation
for i in [1, 10, 20, 30]:
    autocorrelation_lag = Y_train['Soc'].autocorr(lag=i)
    if i == 1:
        print(f"Lag is of {i} minute: ", autocorrelation_lag)
    else:
        print(f"Lag is of {i} minutes: ", autocorrelation_lag)
```

Figure 4.1: Autocorrelation for data sampled at $\frac{1}{60}$Hz.

- Lag is of 1 second: 0.9939
- Lag is of 10 minutes: 0.9263
- Lag is of 20 minutes: 0.7968
- Lag is of 30 minutes: 0.5404.

It can be noticed that when two values of SOC have a difference of half an hour, the correlation coefficient drops significantly. It confirms that this algorithm with fine-grained data cannot correctly predict SOC for 20, 30 minutes or larger time horizons, even if there are some regularization techniques applied, such as the recurrent dropout.

4.1.2 Sampling frequency $\frac{1}{60}$Hz

In order to address the varying frequencies of the stored data drive cycles were down-sampled from 10Hz to $\frac{1}{60}$Hz. The results generated by the LSTM network are presented in Tables 4.3 and 4.4. The former displays the training and testing errors for each of the four different time horizons, while the latter shows the errors achieved by implementing the optimal recurrent dropout of 0.5.

Train error is quite low for all of the four time horizons considered. On the other hand the validation error still increases when time horizon is larger, except for MAE of h = 10min: here there is a big improvement, because validation error prediction is lower than validation error prediction on real-time SOC (h=0). When larger time horizons of 20 and 30 minutes are considered, training error is quite the same for h = 20 and diminishes for h = 30, compared to the values obtained in Table 4.1 and Table 4.2.
When recurrent dropout of 0.5 is applied, training errors observed in Table 4.4 are quite similar to the ones obtained in Table 4.3. On the contrary validation errors have some improvements especially for time horizons of 20 and 30 minutes, in both MAE and RMSE. Thus, this behaviour suggests that the recurrent dropout needs

	h	MAE	RMSE
Training errors	0min	0.012	0.035
	10min	0.028	0.119
	20min	0.040	0.068
	30min	0.042	0.067
Test error	0min	0.128	0.305
	10min	0.052	0.280
	20min	0.149	0.280
	30min	0.168	0.261

Table 4.3: Errors obtained with different time horizons (h) with observations frequency of $\frac{1}{60}$Hz.

	h	MAE	RMSE
Training error	0min	0.016	0.039
	10min	0.033	0.057
	20min	0.040	0.066
	30min	0.041	0.067
Test error	0min	0.126	0.304
	10min	0.117	0.285
	20min	0.137	0.263
	30min	0.160	0.254

Table 4.4: Errors obtained with different time horizons (h) with observations frequency of $\frac{1}{60}$Hz, with dropout of 0.5.

to be adapted to the time horizon considered, in addition to the frequency of the input data.

The ability of the algorithm to correctly predict SOC for h = 0 and 10 minutes, without any dropout, is confirmed by the values of the autocorrelation below:

- Lag is of 1 minute: 0.9965
- Lag is of 10 minutes: 0.9692
- Lag is of 20 minutes: 0.8991
- Lag is of 30 minutes: 0.7938.

The values obtained are higher compared to the ones with input data sampled at 1Hz, but for larger time horizons the correlation coefficient are not enough to guarantee a good prediction. Thus, this is the reason why the errors are still high when time horizon is of 20 or 30 minutes. To obtain a better prediction it is necessary to down sample to a coarse-grained dataset.

4.2 Performance comparison

By comparing the performances obtained in Section 4.1.1 and 4.1.2, when trying to predict real-time SOC, the training error approximately halves if data is sampled to one observation per minute, and the best prediction performance is obtained with applied recurrent dropout, when data is down-sampled to $\frac{1}{60}$Hz. Thus, the algorithm with more fine-grained data input is not able to capture time dependencies from the past, even with recurrent dropout applied, and therefore with downsampling performances are improved.

When time horizon is of 20 minutes, both training and validation errors does not show great improvements with data sampled at 1Hz, even when recurrent dropout is applied (Tables 4.1 and 4.2). Counterwise, when data is down-sampled to $\frac{1}{60}$Hz, even if training errors in Table 4.3 are similar to the ones in Table 4.4, when considering this sampling frequency and applying the recurrent dropout, there is a little improvement of some tenths of a percentage points in validation errors: test MAE is of 13.7%, while in all the other cases was above 14.5%, and also validation rmse is 26.3% while in all the other cases was above 28.0%, so there is an improvement of a few percentage point.

For time horizon of 30 minutes, the best performance is again when input data is sampled to one observation per minute, with recurrent dropout of 0.5 applied. The improvements obtained in test errors are not significant here, but train error MAE diminishes of about 2% and RMSE approximately halves.
When looking at results in Section 4.1.1 there is no improvement on both MAE and RMSE errors in training neither in test: on the contrary errors increase a little bit, confirming that the algorithm with this input data sampling frequency is not able to predict SOC within a time horizon of half an hour, or greater.

When the time horizon is of 10 minutes the lowest training errors are obtained with data sampled at 1Hz and a 0.7 recurrent dropout applied. But for the validation errors, the best set-up is when input data is sampled at one observation per minute and no recurrent dropout is applied.

This particular behaviour for time horizon of 10 minutes, is due to the overfitting effect: when data is sampled to 1Hz, train error is quite low, but the validation error is very high, similar to the ones obtained with greater time horizons. Applying the recurrent dropout of 0.7 does not reduce this effect, suggesting that this sampling frequency is too high to let the algorithm give accurate predictions of SOC. When input data is down-sampled to one observation per minute, training errors increase with respect to the ones obtained with 1Hz sampling, but validation errors, especially MAE, are lower, suggesting that this sampling frequency is more suitable for time horizon of 10 minutes. Unfortunately applying dropout in this setting is not useful because it causes information loss and test errors increase again. This suggests that recurrent dropout needs to be adapted both to the sampling frequency and to the time horizon.

For time horizon of 30 minutes the algorithm clearly overfits the data and this is more noticeable with sampling frequency of one observation per minute: for example RMSE halves compared to the one obtained at 1Hz, but then validation errors are higher than the two obtained in Table 4.1. By applying recurrent dropout of 0.5 training errors are the same but MAE loses 0.8% and RMSE loses 0.7%.

This overfitting effect is also related to the newtowk structure and the dimension of training and test set: if the number of parameters of the network is too high compared to the available data, the neural network may overfit to the training

data and have poor ability to generalize to new test data. The lack of significant improvements for time horizons of 20 and 30 minutes are due also to the too small dimension of training data. When data is sampled to 1Hz the dataset is:

- Train: 162.247 observations;
- Test: 63.055 observations.

When the input data is bigger the algorithm is more able to adapt to the data and reaches low values of train errors. But when tries to predict SOC the values obtained are quite high, suggesting that the algorithm overfits the data, because is not able to capture time dependencies with a too fine-grained dataset. On the other hand, when input data is sampled to one observation per minute, the dimensions are drastically reduced:

- Train: 1.865 observations;
- Test: 773 observations.

This is the reason why the improvements are quite low: the train and test dimensions are not enough compared to the number of parameters that the network needs to learn. Thus, even if training errors decrease suggesting that even with those small amount of data, with a more coarse-grained input data, the algorithm can better understand time dependencies between input features. But when tries to predict, because of the high number of parameters, is not able to properly generalize and adapt to new test data. As showed in Figure 4.2, the number of learnable parameters is 731 and it is very close to the dimension of the test set. Thus, the performance of the network is negatively affected by this, and this is the reason why even if recurrent dropout is applied to data sampled at $\frac{1}{60}$Hz, there are no improvements.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 512, 10)           720

 dense (Dense)               (None, 512, 1)            11

=================================================================
Total params: 731
Trainable params: 731
Non-trainable params: 0
```

Figure 4.2: Output snippet showing the network structure and parameters.

Generally, if the number of parameters is too high relative to the amount of available data, the neural network may overfit to the training data and have poor generalization ability on new test data. On the other hand, if the number of parameters is too low relative to the complexity of the problem, the neural network may not be able to capture the underlying relationships in the data and have poor predictive ability.

Chapter 5

Conclusion

LIBs are the most commonly used batteries especially in the electric vehicle industry. They have become more prevalent in recent years primarly because of their great efficiency, that is one of the main advantages. They have a longer lifespan (measured by the number of cycles) and a lower self-discharge rate compared to traditional cells, thus LIBs will help to enable sustainability and circularity in electric industry, and will also help reducing air pollution.
State of charge estimation is crucial for electric batteries usage, because it provides information about the remaining energy in the battery, which is essential to guarantee safe and efficient operation of many battery-powered devices and systems. Knowing the correct SOC of a battery, allow users to estimate how long the battery will last before needing to be recharged or replaced, and to avoid discharging the battery below a safe or optimal level, which can cause damage to the battery or reduce its lifespan. SOC estimation is particularly important in electric vehicles, where accurate estimation of the battery's remaining charge is critical for predicting the vehicle's range and ensuring that it can reach its destination safely. SOC prediction algorithms can be integrated for anomaly detection in the battery management system based on future predictions of SOC, can be used for a predictive maintenance plan for the battery, and are crucial to assign an appropriate target for the second life of the battery to make the most of its remaining useful life, favoring sustainability an circularity. Thus, accurate SOC prediction gives advantages both to the battery manufacturers and to the end user.
There are several methods for estimating SOC, including Coulomb counting, voltage measurement, Kalman filter, among others. Due to the non-liner behaviour of LIBs, SOC estimation is quite challenging. Accurate SOC estimation requires a deep understanding of the battery's behavior and characteristics, as well as the development of sophisticated algorithms and models that can account for various factors that affect the battery's performance and behavior over time.

Thanks to advances in machine learning, and to the increasing availability of

battery data, data-driven methods are increasingly replacing or in some cases are combined with traditional theory-based methods.
Among data-driven methods, neural networks are most popular. In particular recurrent neural networks, according to experts, are more able to capture time dependencies of LIBs, because of their internal loop that allows the reuse of information of previous steps in the training phase.

This thesis shows how different sampling frequencies of input data collected from testing equipment or, from BMS in case of real-time vehicle monitoring, can provide different performances. Results demonstrate that when data is too fine-grained, even if the available data is of large amount, the network is more sensitive to overfitting: the algorithm is able to learn its own parameters and gives low training errors, but cannot accurately generalize to new test data. Moreover, the input data frequency must be chosen in accordance to the time horizon of the SOC prediction. In general, errors obtained in this thesis are not very low, and this is due to the fact that the chosen parameters were kept constant and were not properly optimized to the two considered sampling frequencies, but they show that a proper sampling frequency combined with a recurrent dropout, can improve the network performance and give more accurate SOC prediction. The more distant is the considered horizon, the more we need to aggregate the data and collect them into a coarse-grained dataset. This must be combined with a proper recurrent dropout, to let the network properly capture the time dependencies of the features: results shows that with a fine-grained dataset a higher recurrent dropout is more suitable, and reduces the overfitting risk; on the other hand, with a coarse-grained dataset a lower recurrent dropout is more suitable and gives better results (because an higher recurrent dropout can favor information loss in training phase).

Another important aspect to be considered when down-sampling the input data, is the dimension of the training and test set: poor improvements obtained for large time horizons with sampling frequency of one observation per minute are due to the drastically reduced number of observations of train and test set. Neural networks require large amount of data to be properly trained, thus when down-sampling this aspect should be considered: the more distant the time horizon is, the more input data needs to be down-sampled to a coarse-grained dataset, the more the covered period of collection of input data needs to be larger, in order to derive proper models that can accurately predict SOC within different time horizons.

5.1 Future developments

The parameters of the LSTM sequential model were kept constant in all experiments. This is the reason why training and test errors are not good enough: networks' parameters necessitate further improvement to make the algorithm fitting better

the input data and derive more accurate SOC predictions. There are a lot of parameters to consider: number of neurons, number of layers, activation function and weights initialization, and so on. In Section 2.1.4 there are some hints provided, but this area of research is huge and NNs optimization is still complex. A great example of this is provided by the optimization of the recurrent dropout of the LSTM layer: it has been investigated in Section 3.2 how a proper recurrent dropout can improve errors of SOC prediction.
This thesis is focused on showing the effect of sampling frequency of input data in prediction within different time horizons. Thus the model does not consider some aspects that can be included in future works such as the effect of different temperatures on SOC: in Section 3.2 is explained how SOC behaviour changes as the ambient temperature is different. Therefore, it is necessary that SOC algorithms are trained with battery measurements collected at different ambient temperatures. Unfortunately, the majority of innovative ideas in this area were developed in the laboratory environment, thus further improvements are needed in the area of real-time vehicle monitoring. To address this problem, the EU regulation will help monitoring the entire life cycle of the batteries that are placed on the market in the European Union.
Thus, monitoring the state of charge of electric vehicle batteries is fundamental and more accurate algorithms are needed. Those models can be implemented for real-time battery monitoring by BMS and can be then used for some business developments related to various aspects:

- battery life: the definition of an appropriate target for the second life of the battery is crucial to make the most of its remaining useful life.
- Performance:a predictive maintenance plan for the battery can be developed to avoid battery damage.
- Safety: the integration of anomaly detection techniques in the battery management system to avoid dysfunctions and failure of the battery.
- User experience: the implementation of a user interface that makes it easier for the owner of an electric vehicle to monitor the battery parameters.

Bibliography

[1] Philip Kollmeyer, Carlos Vidal, Mina Naguib, and Michael Skells. «LG 18650HG2 Li-ion Battery Data and Example Deep Neural Network xEV SOC Estimator Script». In: 2 (2020). DOI: 10.17632/CP3473X7XV.2 (cit. on p. 29).

[2] Carlos Vidal, Phillip Kollmeyer, Ephrem Chemali, and Ali Emadi. «Li-ion battery state of charge estimation using long-short-term-memory recurrent neural network with transfer learning». In: (2019) (cit. on p. 31).

[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok. «A Novel Ultrathin Elevated Channel Low-temperature Poly-Si TFT». In: 20 (Nov. 1999), pp. 569–571.

[4] JJ. *MAE and RMSE — Which Metric is Better? | by JJ | Human in a Machine World | Medium*. 2016. URL: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d.

[5] European Parliament. *EUR-Lex - 52020PC0798R(01) - EN - EUR-Lex*. 2021. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%5C%3A52020PC0798R%5C%2801%5C%29&qid=1673380177492.

[6] *optuna.samplers.TPESampler — Optuna 3.1.0 documentation*. URL: https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html#optuna.samplers.TPESampler.

[7] *pandas.Series.autocorr — pandas 1.5.3 documentation*. URL: https://pandas.pydata.org/docs/reference/api/pandas.Series.autocorr.html.

[8] *pandas.DataFrame.asfreq — pandas 1.5.3 documentation*. URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.asfreq.html.

[9] *pandas.DataFrame.rolling — pandas 1.5.3 documentation*. URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rolling.html.

[10] HighPower Cell Development Team. «Technical Information of LG 18650HG2 (3.0Ah)». In: (Dec. 2014).

[11] *Recurrent dropout | Machine Learning for Finance*. URL: https://subscription.packtpub.com/book/data/9781789136364/4/ch04lvl1sec62/recurrent-dropout.

[12] Gautam Vermani. *What is Recurrent dropout in neural network*. URL: https://www.projectpro.io/recipes/what-is-recurrent-dropout-neural-network.

[13] Crissman Loomis. *Using Optuna to Optimize TensorFlow Hyperparameters | by Crissman Loomis | Optuna | Medium*. 2020. URL: https://medium.com/optuna/using-optuna-to-optimize-tensorflow-hyperparameters-57b6d4d316a2.

[14] «A Vision for a Sustainable Battery Value Chain in 2030 Unlocking the Full Potential to Power Sustainable Development and Climate Change Mitigation». In: (2019). URL: www.weforum.org.

[15] *NMC Battery Material for Li-ion Cells (LiNiMnCoO2) | Targray*. URL: https://www.targray.com/li-ion-battery/cathode-materials/nmc.

[16] Atsushi Baba and Shuichi Adachi. «SOC estimation of HEV/EV battery using series kalman filter». In: *Electrical Engineering in Japan (English translation of Denki Gakkai Ronbunshi)* 187 (2 Apr. 2014), pp. 53–62. ISSN: 04247760. DOI: 10.1002/eej.22511.

[17] *Battery Cell, Module or Pack. What's the difference? [Infographics] | Automotive Cells Company*. 2022. URL: https://www.acc-emotion.com/stories/battery-cell-module-or-pack-whats-difference-infographics.

[18] *LSTM RNN in Tensorflow - Javatpoint*. URL: https://www.javatpoint.com/long-short-term-memory-rnn-in-tensorflow.

[19] *Carriages preview | Legislative Train Schedule*. URL: https://www.europarl.europa.eu/legislative-train/theme-a-european-green-deal/file-revision-of-the-eu-battery-directive-(refit).

[20] *Fiche de procédure: 2020/0353(COD) | Observatoire législatif | Parlement européen*. URL: https://oeil.secure.europarl.europa.eu/oeil/popups/ficheprocedure.do?lang=fr&reference=2020/0353(COD).

[21] *10.1. Long Short-Term Memory (LSTM) — Dive into Deep Learning 1.0.0-beta0 documentation*. URL: https://d2l.ai/chapter_recurrent-modern/lstm.html#fig-lstm-3.

[22] Gonçalo dos Reis, Calum Strange, Mohit Yadav, and Shawn Li. *Lithium-ion battery data and where to find it*. Sept. 2021. DOI: 10.1016/j.egyai.2021.100081.

[23] Carlos Vidal, Pawel Malysz, Phillip Kollmeyer, and Ali Emadi. *Machine Learning Applied to Electrified Vehicle Battery State of Charge and State of Health Estimation: State-of-the-Art*. 2020. DOI: 10.1109/ACCESS.2020.2980961.

[24] Mrinal Tyagi. *Deep Dive: Artificial Neural Network | by Mrinal Tyagi | Towards Data Science*. 2021. URL: https://towardsdatascience.com/deep-dive-artificial-neural-network-e77aa627dc1b.

[25] Ruifeng Zhang, Bizhong Xia, Baohua Li, Libo Cao, Yongzhi Lai, Weiwei Zheng, Huawen Wang, and Wei Wang. «State of the Art of Lithium-Ion Battery SOC Estimation for Electrical Vehicles». In: *Energies* 11 (7 2018). ISSN: 1996-1073. DOI: 10.3390/en11071820. URL: https://www.mdpi.com/1996-1073/11/7/1820.

[26] European Parliament. «Texts adopted - Batteries and waste batteries ***I - Thursday, 10 March 2022». In: *OECD Due Diligence Guidance for Responsible Supply Chains of Minerals from Conflict-Affected and High-Risk Areas* (Apr. 2016). DOI: 10.1787/9789264252479-EN. URL: https://www.europarl.europa.eu/doceo/document/TA-9-2022-0077_EN.html.

[27] «Regulation». In: *OECD Due Diligence Guidance for Responsible Supply Chains of Minerals from Conflict-Affected and High-Risk Areas* (Apr. 2016). DOI: 10.1787/9789264252479-EN. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%5C%3A52020PC0798R%5C%2801%5C%29&qid=1673380177492.