Ali Al-Hilo

Aging Characterization of Li-ion Batteries

School of Technology and Innovations
Master's Thesis in Smart Energy

Vaasa 2022

Acknowledgments

I would like to thank Prof. Hannu Laaksonen for his guidance and continuous assistance throughout this master's thesis. During this journey, I gained remarkable insight into scientific research and academic writing. I really appreciate his patience and dedication toward the completion of this master thesis as well as every single step along the way. His encouragement, trust, and knowledge led me through this program.

I would also like to thank my co-supervisor. Chethan Parthasarathy, Ph.D. student, for his tremendous efforts and meaningful advice in both the theoretical and practical part of this work. Throughout this thesis, he has always been ready and patiently responded without hesitation to all my queries.

My sincere gratitude to the School of Technology and Innovations, Electrical Engineering, University of Vaasa, Finland, for giving me this great opportunity to study and work my master's thesis in their laboratory.

Last but not least, my deepest thanks to my parents, Mr. Chasib Al-hilo and Mrs. Asmaa Al-hilo, for their support and encouragement during all my study levels. Special respect and gratitude to my wife Yasmin Al-hello for her unlimited assistance and care.

UNIVERSITY OF VAASA
School of Innovations and Technology
Author: Ali Al-Hilo
Title of the thesis: Aging Characterization of Li-ion Batteries
Degree: Master of Science in Smart Energy
Discipline: Smart Energy
Supervisor: Hannu Laaksonen
Co-Supervisor: Chethan Parthasarathy
Year: 2022 **Pages:** 72

ABSTRACT:

The depleting nature of fossil energy resources accompanied by increasing demand for energy and their influence on increasing the climate change rates and greenhouse gas emissions motivated the governments and decision-makers to adopt energy sources that are less harmful to the environment and community. Hence, exploiting the potential power of sunlight, water, wind, and underground energy was the cornerstone player in achieving cleaner and more sustainable energy systems. Unfortunately, it is almost impossible to rely on these resources to match the energy demand because of their erratic behavior. Thus, efficient storage systems are needed to enhance reliability and increase the dependency on these energy production systems by storing excess energy and releasing it when needed.

In the essence of this development, lithium-ion rechargeable batteries stand out to be a practical solution for different energy uses, starting from small appliances, transportation, and even grid applications. Although at different levels of utilization, all types of Lithium batteries experience the problem of aging and capacity degradation. LiBs aging is a unique and complicated phenomenon influenced by the interdependency between different internal and external factors. Also, the degradation rates, modes, and mechanisms are affected by the battery's design, production process, and application field. Hence, it has become indispensable to simulate the battery functionality accurately to indicate and fix the possible failures in advance.

Different approaches were employed for tracing and estimating the capacity fade.
Some models rely on an offline investigation of the battery cell(s), while others succeed in scoring high estimation results while the battery is in operation. Data-driven model (DDM) is an example of an online model that doesn't need a comprehensive understanding of the battery components and the chemical reactions inside. Hence, a specific machine learning type of DDM approach called the long-short term memory algorithm (LSTM) was utilized in this thesis. LSTM excels in solving a time-dependent problem; hence, it is adopted here as battery aging is an accumulated problem affected by its previous state.

Herein, a Lithium titanate oxide (LTO) pouch battery cell was employed and subjected to experimental characterization under two different C-rates at a temperature of 25 ^{o}C. The test results were then treated to extract specific health indicators that are studied in the literature. The estimation model was built in a python programming language with the help of data manipulation, machine learning, and visualization libraries. The model's outputs were then evaluated using metrics like mean absolute error (MSE), and root mean squared error (RMSE).

KEYWORDS: Lithium-ion battery, Aging, LSTM, Capacity fade, Estimation model, Feature extraction

Contents

Figures

Tables

List of Abbreviations

Ah	Ampere hour
SoC	State of Charge
ESS	Energy storage systems
BESS	Battery Energy Storage Systems
VRES	Variable renewable energy sources
BMS	Battery Management System
EMS	Energy management system
LiB	Lithium-ion battery
LTO	Lithium Titanate Oxide
ECM	Equivalent circuit model
PBS	Physical- based model
DDM	Data driven model
CCCV	Constant current/constant voltage
CC	Constant current
GHG	Greenhouse gas
DOD	Depth of discharge
ANN	Artificial neural network
FFNN	Feedforward neural networks
FBNN	Backpropagation neural networks
LSTM- RNN	Long-short term memory recurrent neural network
LAM	Loss of active material
LLI	Loss of lithium ion
IR	Internal resistance
Li+	Lithium ions
EV	Electrical vehicle
SEI	Solid electrolyte interphase
SSE	solid-state electrolyte

SSBs	solid-state batteries
TMD	Transition metal dissolution
CEI	Cathode electrolyte interphase

1 Introduction

This chapter aims to explain the background and the scope of this thesis, identify the addressed questions, and presents the main contribution.

1.2 Context and motivation

In recent years, the greenhouse gas (GHGs) emissions rates from energy activities have been grown rapidly to records that have not seen before (37 629.7 MtCO2eq compared to 34 187.1 MtCO2eq in 2010) (Greenhouse Gas Emissions from Energy Data Explorer – Analysis, 2021). Although there was a remarkable decline in values during 2020 following the economic and social restrictions during the Covid-19 pandemic time, scientists expected the emissions to bounce again starting from 2021 (Tollefson, 2021). This increased emission rate results in the aggravation of climate change phenomena. Climate change is considered among the most significant risks that threaten human society and all species kinds on Earth. This phenomenon's direct and indirect impacts can be seen as increasing the global temperature, diminishing the ice area, more droughts seasons, frequent flash floods, and fierce storms (Khan et al., 2019). However, carbon dioxide gas CO2, which mainly emits fossil-fuel-based energy production and transportation, accounts for 78% of GHG emissions (Climate Change Indicators: Greenhouse Gases, 2021). Some researchers claim that if the CO2 emissions rates continue to follow the current manner, the global temperature will increase by 6°C by the end of this century (Gallo et al., 2016). However, to keep the global temperature growth under 2 °C, not more than 33% of the discovered fossil fuel reserves can be consumed by 2050. In addition to its effect on the environment, hydrocarbon fuel is a finite energy resource, and it is expected to deplete in the coming few decades. Thus, the transition toward sustainable energy systems has become an urgent task to mitigate the severe effects of climate change and greenhouse gas (GHG) emissions and secure the energy supply in parallel with expanding energy demand (Kalair et al., 2020).

As a result, many governments and organizations have agreed on setting some pledges and commitments to restore the environment. More integration of renewable energy shares in the energy systems is one of the most viable solutions for this problem. However, it is impossible to control the production of some clean energy resources such as hydro, wind, and solar energy, i.e., variable renewable energy sources (VRES), because of their stochastic nature. Hence, the need for energy storage systems (ESS) has risen to absorb the extra power generated during the off-peak times and release it during the critical periods (Gallo et al., 2016b).

Depending on the working principle and the utilized technology, ESS can be classified mainly into five categories: mechanical, electrochemical, chemical, electrical, and thermal energy storage. They distinguish from each other by many factors, including power and energy densities, charge/discharge time, response time, self-discharge rate, storage period, efficiency, lifespan, and capital and maintenance costs (Sufyan et al., 2019).

Among different kinds of ESS, Battery Energy Storage Systems (BESS) have shown many advantages in applications that require fast response and independency of geographical demands. They are also successful in supplying power and energy for both a fast and short time (Yang et al., 2018).

Lithium-ion batteries (LiBs) are rechargeable batteries that excel in different applications such as electronic devices (e.g., mobile phones and laptops), medical appliances, and EV sectors. They also showcase great potential to be integrated into the electric grid applications, especially as ancillary services providers. Compared to other batteries, LiBs feature in providing high energy and power densities regarding their size and weight; they are capable for fast charging and can work under a wide ambient temperature range; they are almost 100% efficient and have a shallow self-discharging rate (Akinyele & Rayudu, 2014)

Nevertheless, like other batteries, Li-ion batteries experience the aging phenomena. It is an inevitable process that happens because of the side effect of chemical reactions inside the battery cells. The degradation is a highly complex problem. It depends on the variation of many external factors like temperature, depth of discharge (DOD), battery loading, charging/discharging current rates, cut-off voltage, etc. These factors are interconnected and affect each other. Hence, the determination of the battery life is considered a challenging task.

However, scientists have applied different kinds of experiments and tests on various LiBs cells to track their aging modality. They aim to a comprehensive understanding of battery aging to help manufacturers choose the most suitable materials and structure of the battery, develop the cell production processes, and establish a guided design of battery systems for various uses (Guo et al., 2021). As a result, the users would have more reliable and safer battery systems (Lin et al., 2015).

In contrast, an accurate lifetime prediction model must be included in the battery management system (BMS). Therefore, the users would be able to trace the degradation behaviors of their batteries, avoid failures by applying maintenance processes in advance, and try to exploit their potential capacity efficiently. Thus, different estimation approaches have been investigated in both academia and industry. The most applied models can be divided into three main types: physical-based model (PBS), equivalent circuit model (ECM), and data-driven model DDM (Ng et al., 2020); however, the focus of this thesis would be on adopting DDM for capacity fade estimation.

1.2 Thesis purpose

he primary purpose of this thesis is to examine the performance of a specific data-based approach to track and predict the capacity fade of a lithium-ion battery cell. The

thesis focuses on the cycle aging effect and neglects the effect of calendar aging on the capacity fade. Two systematic sequential laboratory tests have been run on the cell under 1C and 5C, respectively. The thesis will utilize the data extracted from the tests and perform data engineering, including data cleansing and feature extraction, to prepare data to be injected into the prediction model. Long-short term memory recurrent neural network (LSTM- RNN) algorithms will be adapted to achieve the prediction task. LSTM is a powerful tool in dealing with time series problems. Hence, it will be executed on each of the two tests separately.

1.3 Thesis contribution

The thesis will investigate two main issues related to capacity fade prediction. The first one is the feature extraction problem. There is a lack of battery testing datasets, so it is crucial to extract as many features as possible from the available datasets to have an accurate and practical battery cell model. Second, this thesis tries to predict the capacity fade of the battery cell using a novel machine learning approach LSTM (long-short term memory).

The objectives of this thesis are summarized as follows

1. Theoretical review on the aging phenomena and impact factors affecting battery performance.
2. Create a systematic test technique for Lithium-ion battery
3. Monitor the degradation behavior of the lithium-ion cell under constant temperature and two different C-rates.
4. Employ data engineering on the battery datasets to clean the data and extract the most efficient features.
5. Build a capacity fade estimation model using Python programming language and other related libraries (e.g., Pandas, Numby, Sciketlearn, Matplot, etc.), evaluate the outputs' accuracy, and discuss the results.

The applied approach and the results are limited to the cell level of lithium-ion batteries. However, the results cannot be generalized as they are done on only one cell in a laboratory environment.

1.4 Structure of the thesis

The thesis is structured as follows:
Chapter 1 introduces the background of the subject, explains the motivation and goals of the study, and defines the thesis's contribution.

Chapter 2 provides a theoretical overview of the aging phenomena in Li-ion batteries. It defines factors that expedite the aging process and relates them to their effect on the battery components.

Chapter 3 briefly describes Artificial neural network (ANN) and focuses on (LSTM- RNN) theory and functionality.

Chapter 4 is assigned to explain the practical part of this thesis. It will introduce the methodology and procedure used to estimate the capacity fade of a Li-ion LTO (lithium-titanate-oxide) cell. It illustrates the test procedure done on the cell, the feature engineering method, and the model parameters.

Chapter 5 explains the model's results and compares them with the output of the experimental process.

Chapter 6 concludes the study, explains its limitations and suggests development for future work.

2 Overview of li-ion battery ageing

Lithium-ion (Li-ion) battery aging is a nonlinear and irreversible reaction that occurs with time and use inside the battery cell caused by a combination of external and internal factors (Xu et al., 2018). Generally, there are two types of aging: calendar aging when the battery is in standby mode, i.e., rest or storage mode, and cycle aging when the battery is under operation, i.e., charge/discharge mode. The battery degrades in different ways and rates under each mode mentioned above. There is a wide range of degradation mechanisms; some are triggered instantaneously with the early moments of the battery operation, while others can occur later or result from the formation of other mechanisms (Edge et al., 2021). In contrast, two leading physical indicators of battery aging are capacity fading and resistance increases. Capacity fade results from the loss of active material (LAM) of electrodes, loss of cyclable lithium (LLI), or the growth of internal resistance (IR). In contrast, the increase in the cell resistance downgrades the available power that the battery can discharge.

This chapter defines the components of Lithium-ion batteries and the working principle, presents the causes behind battery aging, identifies the concept of calendar and cycle aging, and explains the key factors that expedite the degradation.

2.1 Component of LIBs and working principle

Lithium-ion batteries comprise four main components: cathode, anode, separator, and electrolyte.

Depending on the required characteristics of the battery, materials and compounds used for manufacturing the cell components are varied. The cathode is the positive electrode usually made of metal oxide active material coated into an aluminum foil that acts as a current collector. The most used commercial cathode compound is lithium cobalt oxide (LiCoO2). However, usually commercial batteries are named after the lithium donator material in the cathode (Zubi et al., 2018). In contrast, the anode,

i.e., the negative electrode, is manufactured of active carbonaceous material such as, e.g., graphite, lithium titanate (LTO), or silicon (Si) coated into copper current collector foil (Kabir & Demirocak, 2017). Lithium electrons placed in the cathode are released through an external path to be stored in the anode structure during charging operation. In the meantime, a process called intercalation happens when Lithium ions (Li+) pass among the electrolyte and through the separator to join lithium electrons into the graphite layers of the anode. In the discharge phase, electrons and lithium ions move back to the cathode through the same path they utilized in the charging process.

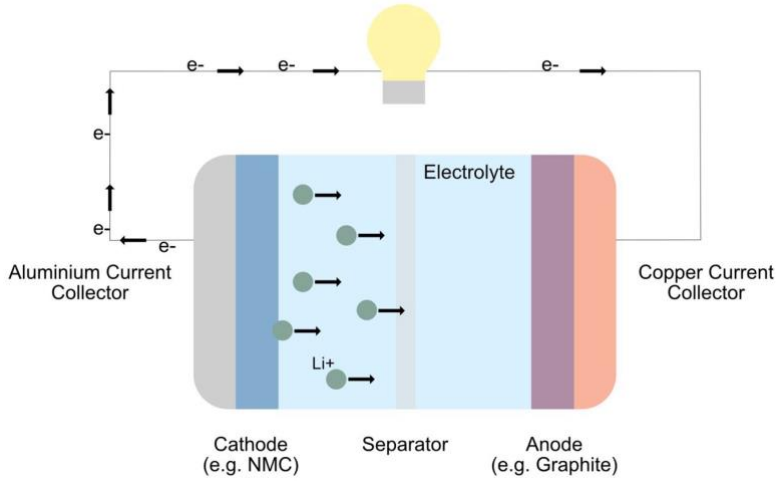


Figure 1 A schematic of a lithium-ion battery and its components (L. Guo et al., 2021)

The electrolyte is a solvent located between cathode and anode to be the medium that allows the shuttling of lithium ions between the two electrodes. The electrolyte is usually comprised of material dissolved in salt, and it usually is not involved in the chemical reaction when charging /discharging the battery cell (Song et al., 2021).

In contrast, the separator, a semipermeable and electronic insulating membrane, plays an essential role in cell safety by preventing short-circuiting cathode and anode. It has big enough holes that permit Li+ to pass back and forth. Figure 1 shows the components of one LTO cell (L. Guo et al., 2021).

2.2 Ageing types

LiBs undergo the effect of aging under both operating and non-operating conditions. Calendar aging occurs when the battery degrades under a non-operation state, while cycle aging is caused by operating the battery (Zubi et al., 2018). Both aging types affect each other; the rate and the causes of any type affect the rates and the causes of the other type. Although calendar aging rates are slower than cycle aging, it is deemed the most influential type of battery degradation, specifically in some applications when batteries spend less time operating than rest, such as in EV batteries (Sui et al., 2021). Calendar aging is mainly driven by external factors such as temperature and the battery's state of charge (Soc). Storing the battery at high temperature and SoC rates expresses the SEI layer's formation, growth, or rebuilding on the anode side.

On the other hand, cycle aging is a more complex type than calendar aging due to various factors that influence its occurrence. However, cycle aging is mainly catalyzed by lithium plating of the anode. It is proved that operating the battery with elevated charge rates, and low temperatures accelerate the lithium plating rates (Redondo-Iglesias et al., 2018).

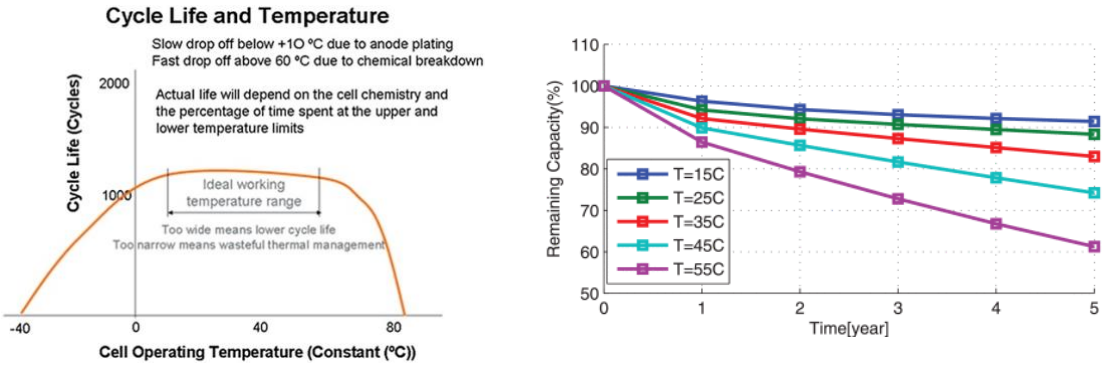
2.3 External ageing factors of Li-ion batteries

Understanding the external factors that trigger and evolve Li-ion battery capacity and power deterioration is vital to slow down the aging rate and allow the maximum usage of their capacity. This section discusses how factors like temperature, SOC, C-rate, etc. influence the battery aging rate during storage and operation.

2.3.1 Impact of temperature

Abnormal ambient temperature values have a detrimental effect on the functionality, safety, and age of the Li-ion battery. However, the impacts of low temperatures are

different from those of high-temperature values. In general, the impacts of the temperature manifest more clearly in cell failure than in the reduction of its cycle life. Figure 2a) shows that the cell works ideally in temperatures between 10°C and 60°C. Operating the cell beyond this range leads to a sharp decline in the number of available cycles (Kabir & Demirocak, 2017). Figure 2b) (Xu et al., 2018) presents the aging calendar rate of five battery cells stored at different temperatures at 50% SoC.



a) b)

Figure 2. a) Expected cell cycle life under varied operating temperatures. b) Calendar aging at 50% Soc under different temperatures.

Operating or storing the battery under high-temperature values leads to incremental parasite reactions on the anode, the cathode, and the electrolyte (Amine et al., 2005). The SEI layer will increase on the anode side, causing a loss in the lithium inventory (LLI) and a rise in internal cell resistance. In contrast, metal dissolution from the cathode and electrolyte solution decomposition will occur, leading to loss of active material (LAM) and LLI (Li et al., 2019). In the worst scenario, the battery can be threatened by thermal runaway risk (Finegan et al., 2015). It can be noticed clearly that high-temperature factors trigger all degradation modes; thus, the battery will lose power and capacity when cycling or resting at elevated temperatures.

Nevertheless, low temperature affects the performance and the age of the battery. On the one hand, the performance declines as the transportation rate of Li+ becomes

slower, the conductivity of the electrolyte weakens, and the diffusion rate of the lithium-ion diminishes (Jaguemont et al., 2016). On the other hand, lithium plating on the anode may happen along with LLI when charging the battery under high C-rate and low temperatures. This phenomenon's continuity may lead to the growth of lithium dendrites, which may pierce the separator, causing a short circuit inside the cell. In contrast, all other aging mechanisms mentioned above are hindered or not influential. However, Storing the battery at low temperatures does not have a remarkable effect on the aging rate (Li et al., 2019), while cycling the battery at low temperatures results in an irreversible loss of capacity and power (Jaguemont et al., 2016).

2.3.2 Impact of high/low state of charge (SoC)

Both high and low SoC may cause severe damage to both electrodes. When the cell is overcharged, the cathode becomes fully delithiated, and all active lithium ions migrate to the anode causing the over lithiation problem. As a subsequent implication, the cathode structure undergoes irreversible damage. A reduction in the active material, decomposition of electrolyte, rising internal resistance, dissolution of transition metal ions, and substantial increase in cell temperature are further consequences (He et al., 2011) and (Li et al., 2019). Also, the cell may experience a thermal runway when overcharging it with a high C-rate (Ohsaki et al., 2005).

At over-discharging conditions, an unusual increase of anode potential arises, leading to a dissolution of the current collector. When recharging the cell, after the above process, copper dendrites can grow, which may short circuit the cell. This will result in a loss of available capacity and power, and the effect will amplify when accompanied by high C-rate charging conditions (Sun et al., 2022).

It is recommended to operate the battery at a specific range between 20% and 90 and store it at around 50% (Kabir & Demirocak, 2017). Figure 3 illustrates that the battery stored at 100% SoC experienced a higher degradation rate than those stored at 60 % SoC.

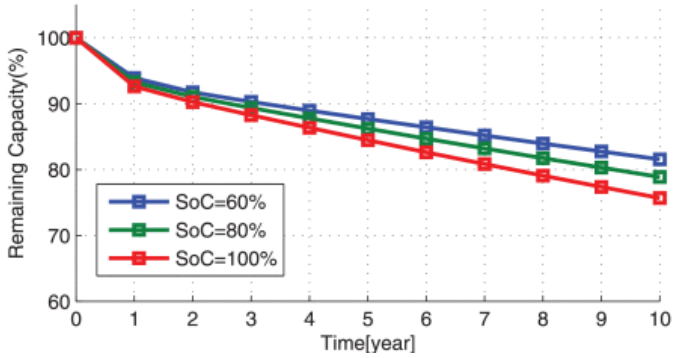


Figure 3 Calendar aging at 25 °C for different SoC (Xu et al., 2018)

2.3.3 Impact of high C-rate

Cycling the battery with rapid C-rates has a similar influence on the cell of the overcharge/over-discharge. High charging/discharging current is always accompanied by heat waste, which raises the internal cell temperature, thereby accelerating aging rates. This factor becomes more influential with some material of cell components. For instance, organic electrolytes have low heat capacity; hence, their temperature will increase rapidly when a high current flows through (Chandrasekaran, 2014). In addition, Lithium plating can be initialized on the graphite anodes as they cannot absorb the massive amount of Li ions when transferred at high rates (Keil & Jossen, 2016). Also, electrodes may experience partial overcharge/over-discharge behavior as lithium ions accumulate on specific parts of the anode or the cathode.

(Dubarry et al., 2011) studied the impact of charging different battery cells under different C-rates on the capacity fade and ascribed the proportions of different degradation modes over battery lifespan. Figure 4 reveals that in the first 500 cycles, the cells degrade linearly under five different C-rates. The authors suggest that LLI mode drives the cells' aging in this stage because of some side reactions that permit the formation of SEI layer on the surface of the electrodes. After 500 cycles, LAM is claimed to be the responsible mode for accelerating capacity degeneration.

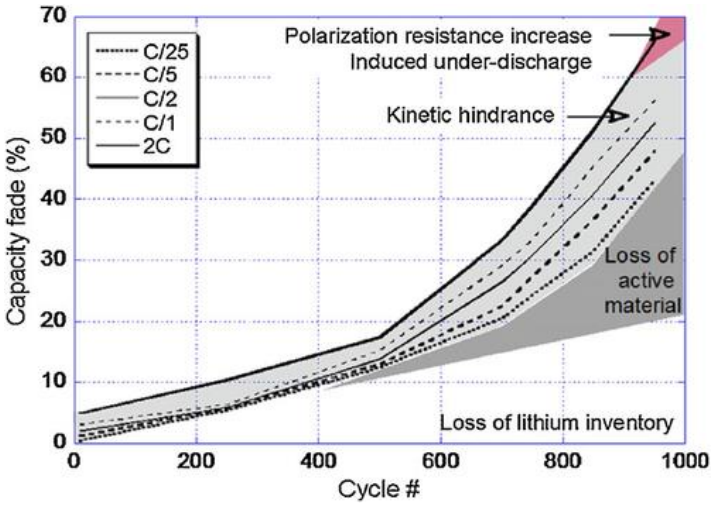


Figure 4 Specific battery cell aging under different C-rates at fixed room temperature. (Dubarry et al., 2011)

2.3.4 Impact of mechanical stresses

Along with its life, from the manufacturing phase to recycling, Li-ion batteries experience mechanical tension or pressure, including volume increasing of electrodes, gas evolution in some battery packs, and an undesired load during operation. Electrodes' sides near the separator are mostly affected by these factors causing them to crack or fracture. At high fracture or cracking rates, the electrodes' structures fail and cause severe cell degradation (Li et al., 2019).

The cause-consequence relation is presented in Figure 5 below, summarizing the contributions of some aging factors to the power and capacity loss of the battery. In practice, a battery management system (BMS) is assigned to control and mitigate the effect of aging factors on large-scale battery applications. BMS, e.g., develops suitable charging protocols, manages cell temperature, and prevents over-charging/ discharging conditions.

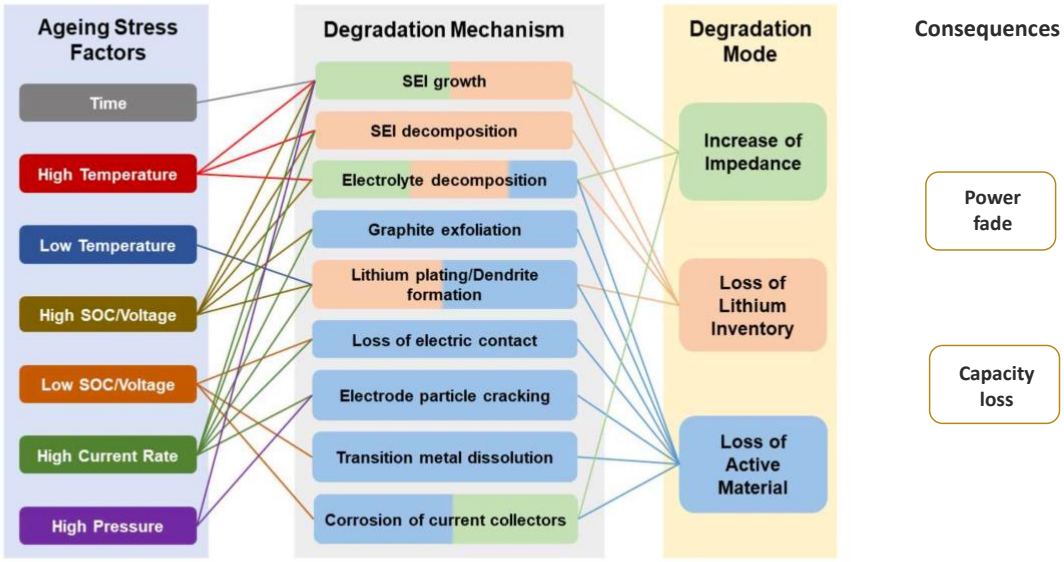


Figure 5 Cause-effect relation of some aging factors on battery degradation, adopted from (Li et al., 2019).

2.4 Description of common aging mechanisms

Each of the four main components of the lithium cell wears out in different ways and rates. Accordingly, each battery cell degrades uniquely. Some degradation mechanisms influence only one of these components, while some mechanisms have an associated broad effect on many cell components. Figure 6 summarizes the most common mechanisms that impact the battery components. This chapter will explain and classify the most influencing degradation mechanisms according to their effect on the cell components. In context, the cause-effect relation between the most critical aging mechanisms and capacity fade or power loss will be presented. However, the influence of the external factors, e.g., temperature and SoC during cyclic and calendar aging, will be linked with the corresponding aging mechanism.

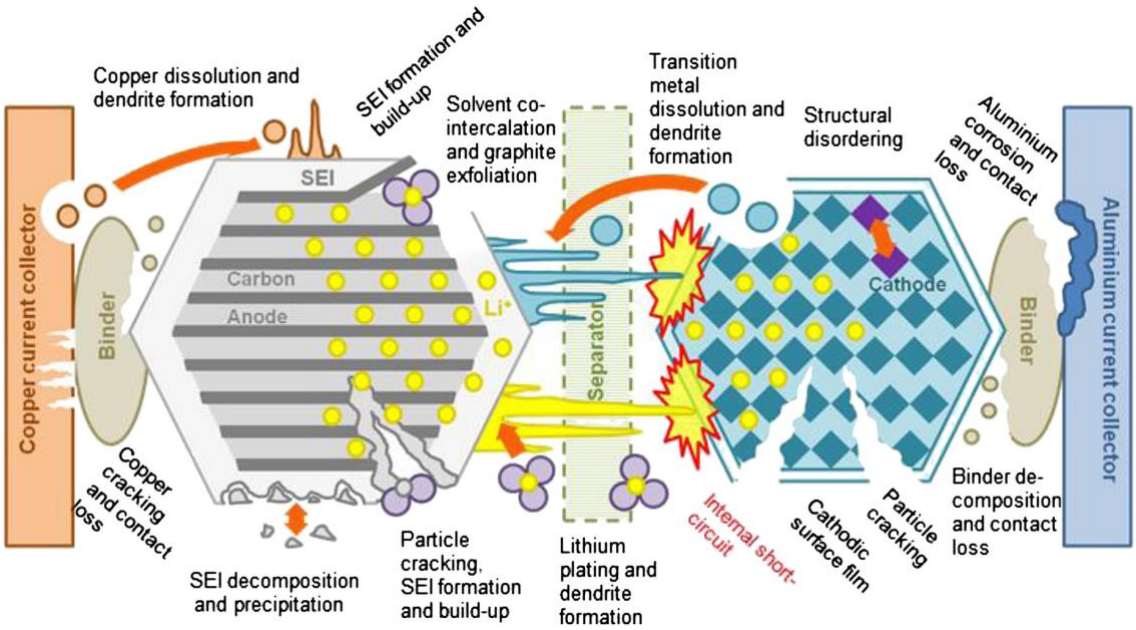


Figure 6 Schematic of component-wise mechanism of li-ion batteries

2.4.1 Solid electrolyte interphase (SEI) layer

A layer initiates due to chemical reactions in the contact area between the electrolyte liquid and the conductors on the anode surface. This reaction leads to a breakdown of the reductive electrolyte and causes a reduction in Li-ions. The output particles of this reaction will gather on the negative electrode surface, forming the SEI layer. Although it may affect the cathode, SEI is primarily considered an anode degradation mechanism. It starts directly on the cell's first charge when li ions travel from cathode to anode among the electrolytes, and it will keep growing even if the cell is in storage mode (Edge et al., 2021). SEI has dual effects on battery performance; during the initial stage of its formation, cells lose around 10% of the lithium inventory because of the rapid loss of the electrolyte and the reduction in li-ions. However, the positive effect of SEI is to allow the passage of Li+ in/out of the electrodes while blocking the passage of electrons and electrolyte particles, preventing further corrosion and reduction of electrolytes and electrons. (Vetter et al., 2005). The thickness of the SEI layer could grow if a new anode surface exposes to electrolyte or the solvent molecules diffuse inside a formed SEI layer.

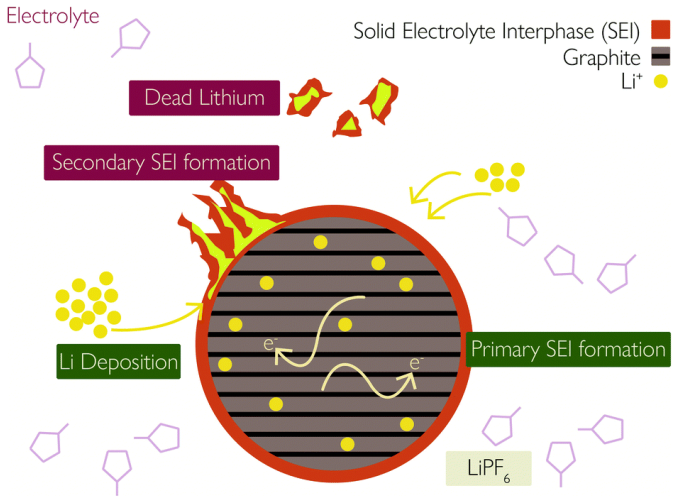


Figure 7 Schematic of SEI layer formation (Edge et al., 2021)

SEI formation and growth are claimed to be the leading cause of lithium-ion battery degradation. Elevated temperature expedites the diffusion rate of lithium ions resulting in the development of the SEI layer. Moreover, cycling the cell with a high C-rate leads to particle cracking and the formation of new SEI layers (Kabir & Demirocak, 2017). The formation and growth of SEI cause an irreversible loss of capacity and increase the cell's internal resistance. Some types of anode material do not experience the SEI process under normal conditions, such as Lithium titanate anode (LTO) (Chahbaz et al., 2021).

2.4.2 Lithium plating

Lithium plating is a parasitic reaction that occurs on the graphite anode during the charging process forming a spiky, granular, and/or mossy metallic Li+ on its surface (X. Lin et al., 2021). It happens in two forms; thermodynamic and kinetic plating. Thermodynamic plating happens when the NE is fully lithiated and has no space for Li-ion to be absorbed. Kinetic plating occurs when the cell is charged with a high C-rate triggering more side reactions from the primary intercalation reaction (Edge et al., 2021). However, the metallic lithium can be restorable or not restorable. The restorable portion indicates the deposited lithium with an electrical contact with the

anode surface. It can be restored during the relaxation or discharging phase after the lithium plating by a lithium stripping process, which leads to the reinjection of these lithium metals into the anode. In contrast, irreversible lithium reacts with electrolyte and form a secondary SEI layer or lithium film. The latter is a composite of dead lithium with no electrical contact with the anode. The effect of irreversible lithium is seen as an increase in internal resistance, decline of the energy density, and acceleration of capacity fade (X. Lin et al., 2021).

Generally, lithium plating is ascribed to three main enablers: (i) unsafe operating environments, (ii) cell manufacturing flaws, and (iii) Cell aging (X. G. Yang et al., 2018). These factors are further detailed in table 1.

Factors	Causes and Conditions
Hazardous Operating Conditions	(a) Low temperatures (b) High charging C-rates (c) High SOCs
Cell Defects	(a) Cell properties and poor design, such as imbalanced negative to positive ratio
Aging of the Cell	(a) Leading to cell unbalance (b) Kinetic degradation (Capacity fade, energy fade, CE decrease, energy efficiency fade and resistance increase)

Table 1 Factors enhance Lithium plating (X. G. Yang et al., 2018)

Lithium plating is a potential safety risk as it could tear the separator, causing a hazardous short circuit between anode and cathode, which could lead to a thermal runaway and an instant failure of the cell. It also causes the loss of lithium ions and consumption of electrolyte solvent leading to capacity fade and restricting the fast-charging capability (X. Lin et al., 2021).

Lithium plating can be detected during cell operation by studying the voltage curve when discharging or resting the cell after the charging process. A plateau in voltage

value shows because of the stripping phenomena when the deposited Li metal return to its typical case (Figure) (X. G. Yang et al., 2018).

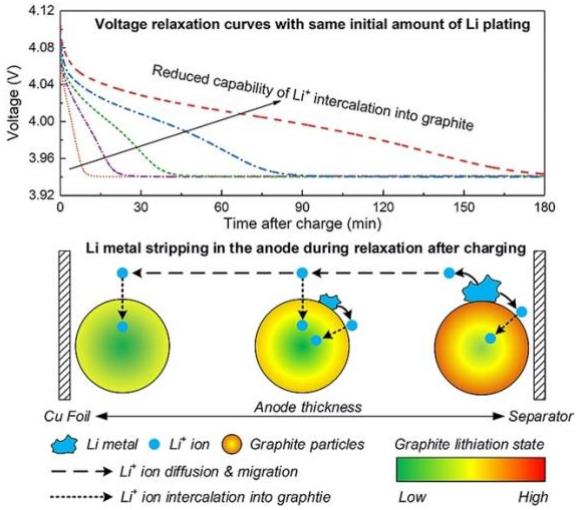


Figure 8 Voltage plateau during relaxation phase (X. G. Yang et al., 2018)

The lithium plating mechanism obstructs further development and adaptation of lithium-ion batteries. In some applications, batteries are demanded to fulfill some requirements such as the capability of fast charging, charging at low temperatures, high level of safety, long lifespan, and self-recovery ability. Unfortunately, all these requirements enable Lithium plating in the battery (X. Lin et al., 2021b).

2.4.3 Particle fracture

Operating the cell leads to expanding and diminishing electrodes' volume in response to the intercalation/ deintercalation process. The continuity of this process, along with specific intrinsic factors including "(particle size, morphology, element composition, structure, pre-existing defects, and mechanical properties) and extrinsic factors (C-rate, cutoff voltage, cycling time, and thermal treatment)" drives particle fractures in the electrodes. However, a solid-state electrolyte (SSE) in solid-state batteries (SSBs) is also susceptible to examining particle fracture phenomenon (P. Li et al., 2020). It can

be described as tiny cracks spread on the electrodes and the solid electrolyte body and is considered one of the essential contributors that cause capacity fade and consequently accelerate the battery degradation. It is usually located near the separator's electrodes' side because of the high stress caused by the high local current densities (Hendricks et al., 2015).

The particles crack in two ways; intergranular fractures, where the fracture occurs in the borders between the primary particles, and intragranular fractures, which affect the material's primary particles. The formation and growth rate of each of these types depends on the electrodes' structure and materials and the operation conditions of the battery (P. Li et al., 2020).

These cracks expose new surfaces of active materials to react with electrolytes forming new SEI layers. In contrast, some cracks can produce idle areas of active materials that cannot react with other parts of the electrode and /or with electrolytes. Hence, impactful amounts of active materials and lithium inventory (when isolated parts contain Li+) will be lost as a result (Birkl et al., 2017). Moreover, the advent of these cracks is responsible for the instability of the electrodes' structure, which will cause the cracks to continue forming until the electrodes collapse (J. Guo et al., 2021).

2.4.4 Transition metal dissolution (TMD)

his mechanism mainly influences the Mn-based cathodes (e.g., The spinel lithium manganese oxides (LMO)), and it is accelerated when operating the cell under high SoC or at elevated temperatures. Although other transition-metal-oxides-based cathodes such as lithium-nickel-cobalt-mixed-oxides can examine this degradation mechanism it is usually neglected as it occurs only under extreme operation conditions (J. Guo et al., 2021).

The dissolution of Mn-based cathode particles in the electrolyte release water and hydrofluoric acid (HF). Mn dissolution affects cell aging in three directions. First, the formed HF keeps the dissolution of TM and Li+ on the cathode's surface, promoting the consumption of the active materials and contributing to a substantial loss of capacity. Second, the dissolved ion of magnesium will reach the anode and accumulate either on its surface or/and on the formed SEI layer. This would increase the thickness of the SEI layer, reduce the electrolyte amount and promote self-discharge of carbon-based anodes. Third, some electrically inert particles, e.g., MnF2 and MnCO3, will precipitate on the cathode, causing a continuous increase in its impedance (Kabir & Demirocak, 2017).

2.4.5 Cathode electrolyte interphase (CEI)

his layer forms as a response to a complex chemical surface reaction between the cathode and the electrolyte when operating the cell under high voltage values. These operation conditions, along with high temperature, ease the electrolyte decomposition. This will generate particles such as HF that enhance the cathode deterioration and accumulate the ingredients that form the CEI. Usually, the thickness of the CEI layer is less than that of the SEI layer and does not affect the whole surface of the cathode. However, CEI continues to grow in parallel to the cycling process of the cell, which finally may cause an irreversible loss of capacity (Xiong et al., 2020).

2.4.6 Binder decomposition

Utilizing the battery under elevated temperature is the key influencer of this failure. The decomposition of the binder and the dissolution of the current collector will cause instability in the cathode compounds. Cathode instability happens because of the rise in the internal resistance that is triggered by the loss of contacts in the electrode (Xiong et al., 2020).

3 Artificial neural networks

Artificial neural network (ANN), also referred to as neural network (NN), is a branch of artificial intelligence that simulates the structure and function of the nervous system in the human brain using computational models. In essence, ANNs are applied to model nonlinear functions to obtain outputs with an acceptable level of accuracy. Structurally, the basic neural network consists of an input layer, a hidden layer, and an output layer. Sometimes input layer is considered a virtual layer. There could be more than one hidden layer in a more complex structure where the network is called a deep learning network. All layers are interconnected, so neurons in a particular layer are connected to neurons in the next layer. The connection lines between nodes work similarly to the synapses in the human brain as they transfer the information through the model layers. The input layer collects the data and feeds them to the hidden layer, where the information is handled, and the relationship between the input and the targeted output is built. The output layer receives the information from the hidden layer(s) and interprets them to be the result of the system (Abiodun et al., 2018).

ANN is mainly classified into feedforward neural networks (FFNN) and feed backward or backpropagation neural networks (FBNN). In FFNN, the data transmit in one direction and do not include feedback connection like in FBNN. There are different forms of NN under these two classes. Examples of FFNN are multi-layer perceptron and radial basis function networks. In contrast, Bayesian regularized neural network and self and recurrent neural network (RNN) are examples of FBNN. Describing all types of ANN is beyond the scope of this thesis, and the focus will be on the recurrent neural network (RNN), which will be discussed in the following paragraph (Abiodun et al., 2018).

RNN is a feed backward neural network that uses loop features in its structure to handle historical observations among data. RNN can memorize sequences of data and handle them accordingly. Thus, it is broadly employed to solve time series problems,

where prior values affect the posterior outcomes. The basic structure of RNN is presented in figure 9. RNNs can be seen in three different forms, many-to-many, one-to-many, and many-to-one. Many-to-many means the model gets a sequence of inputs and results in a sequence of outputs. A one-to-many network takes a single input and produces a sequence of outputs. Many-to-one is the opposite of the previous type; the network requires a sequence of inputs and yields one output (Kaur et al. 2019).

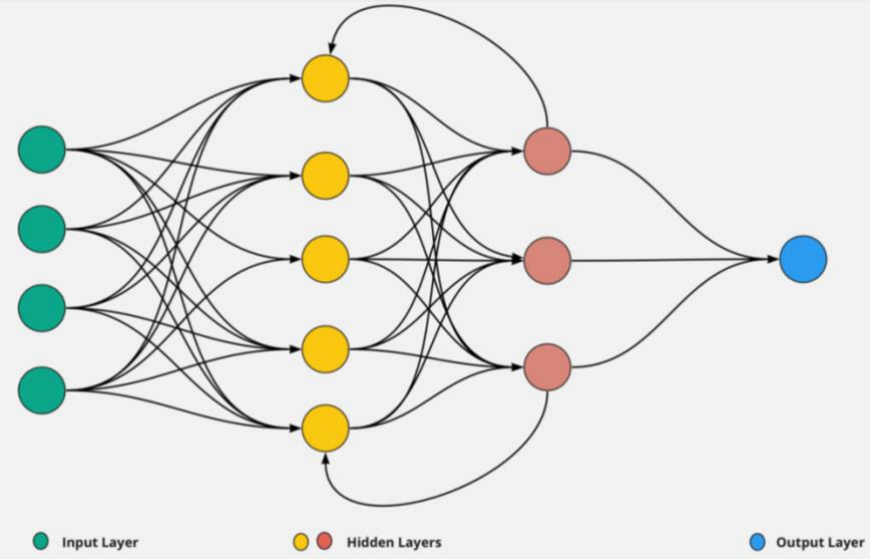


Figure 9 Structure of simple RNN (Dataasprint, 2020)

In contrast, basic forms of RNN could experience vanishing or exploding gradient problems, making training and getting desired results challenging. The vanishing gradient problem appears when the model gets stuck between specific values and loses optimal solutions during the long training phase. As a result, the parameters could not update, and the network becomes useless. Therefore, in 1997 Hochreiter & Schmidhuber have developed a model called LSTM to overcome this flaw of RNN (Chung et al. 2014).

3.1 LSTM

ong-short-term memory (LSTM) is a versatile and commonly used recurrent neural network in solving a host of machine learning and deep learning tasks. LSTM can look back for lengthy time steps, making it the ideal choice for this task compared to traditional neural. Because of this merit, this thesis adopted LSTM for capacity degradation estimation, as the capacity of a battery cell degrades with time under various long- and short-term conditions (van Houdt et al., 2020).

Compared with other RNN (recurrent neural network) types, LSTM excels in two points. Firstly, its ability to mitigate the exploding and vanishing gradient problem. Secondly, it can learn the information from the data more effectively by looking back at the long dependencies. LSTM algorithm could contain one or more of the LSTM units. Each LSTM unit has three gates of the same shape. These gates can memorize the data according to their importance. So that the data with high importance will be kept, while the others will be forgotten. This process aims to pass the most wanted information along the sequence to achieve an accurate prediction (van Houdt et al., 2020).

LSTM unit structure is shown figure 10. the subscript t refers to the current state of each vector. In our application t denotes the cycle number. At each *t* moment, there are three inputs should be passed to each LSTM-RNN unit:

- X_t is the input vector of the current timeframe.
- h_{t-1} is the output value from the previous step,
- C_{t-1} is the cell state pf the previous step.

Meanwhile, the LSTM-RNN produce two outputs: the output value h_t and cell state C_t at present time *t*. It can be noticed from the figure that LSTM cell is compromised from four main components that will be explained in the following paragraph (Chen et al. 2020).

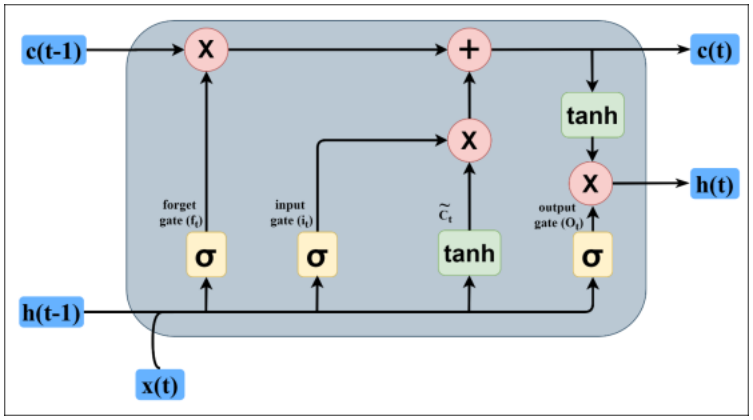


Figure 10 The architecture of LSTM unit (P.Venugopal et al. 2021)

Forget gate: his is the first step in LSTM. This gate decides which data are passed to the next unit and which are thrown away. It compares the current input Xt and the previous output ht-1 by passing them to the Sigmoid function (σ).

The output of the Sigmoid function ft varies between zero and one. Hence, the result that gets a value closer to one will be passed. The forget gate works as follows:

$$f_t = \sigma\,(W_f\,.\,[\,h_{t-1}\,,\,X_t] + b_f) \qquad (1)$$

W_f and b_f stand for the weight matrix and the bias of forget gate, respectively.

Input gate and input node: The cell state will be updated in this stage. First, the current input *Xt* and the previous state *ht-1* feed into a sigmoid layer called "input gate" and the tanh function called "input node" separately. The output tanh function ($\tilde{C}t$), which fluctuates between -1 and 1, will be multiplied by the output of the sigmoid function *it*, resulting in a new vector of candidates injected into the cell state. Equation (2) denotes the output of the tanh function, while equation 3 calculates the outputs of the sigmoid function.

$$\tilde{C}_t = tanh\,(W_g.\,[\,h_{t-1}\,,\,X_t] + b_g\,) \qquad (2)$$

$$i_t = \sigma\,(W_i.\,[\,h_{t-1}\,,\,X_t] + b_i\,) \qquad (3)$$

W_i and b_i denotes for the weight matrix and the bias of input gate, respectively.

Cell State: he function of this component is to transfer the relevant data across all time steps. The transferred data depends on the output of the forget and the input gates. The equation below shows how *Ct* is calculated.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Output gate: It determines the output of the unit(*ht)*. Similar to forget gate, current input *Xt* and the previous state *ht-1* are passed to a sigmoid function, while the newest cell state will be passed to the tanh function. Then the output from these functions will be multiplied to determine the output.

$$O_t = \sigma (W_o . [h_{t-1} , X_t] + b_o) \quad (5)$$

$$h_t = tanh (C_t) * O_t \quad (6)$$

W_o and b_o denote the weight matrix and the bias of output gate, respectively.

4 Modelling of Li-ion LTO cell capacity fade

4.1 Methodology

The practical aspects of the study include designing and implementing battery cell tests, gathering and preparing data, and constructing and evaluating the model. After completing the test on the cell, the data has been extracted and prepared using some feature extraction and data manipulation techniques. The data are then split into training and testing sets. Training sets will help the model learn the hidden dependencies between features and capacity fade patterns, whereas testing sets will be used to validate the model. Subsequently, the results will be evaluated using some evaluation metrics. Figure 11 shows the framework and the flowchart of the capacity fade prediction.

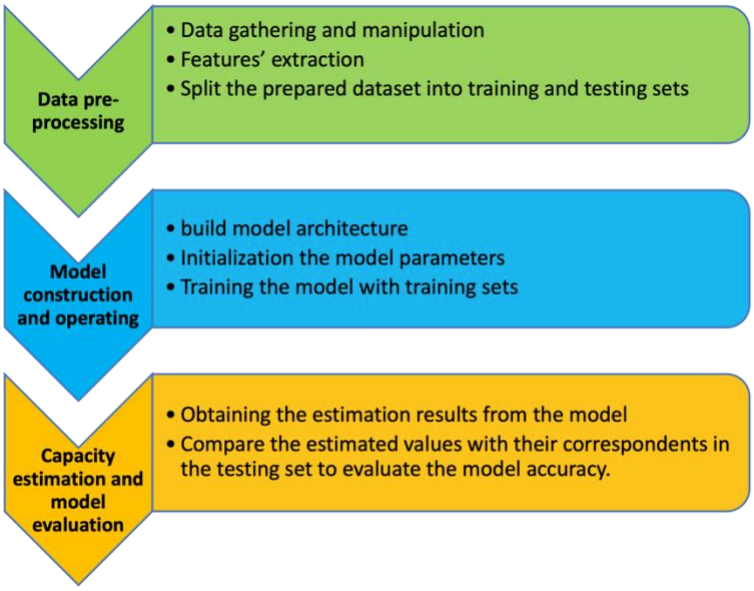


Figure 11 Framework of capacity fade estimation

Python programming language is used for coding in this study. However, other open-source libraries are employed as well. Pandas (a library designed to manipulate and analyze data) and Numby (a library that uses high-level mathematical functions for handling arrays and matrices) are operated for data preprocessing (Pandas, 2008) and

(NumPy, 2006). Scikit-learn (a library that contains a complete set of functions for machine learning problems and provides tools for measurement and evaluation tasks) is involved in data scaling, model construction, and performance assessment (Scikit-Learn, 2007). TensorFlow and Keras (libraries that focus on deep learning tasks) are used for building the model (TensorFlow, 2015) and (Keras, 2015). Plotting and visualization are carried out using Seaborn and Matplotlib (plotting libraries for python programming language) (Seaborn, 2012) and (Matplotlib, 2003).

4.2 Test arrangement and procedures

The battery cell was tested under two different C-rates at a fixed ambient temperature value of 25 ^{o}C. First, the cell is subjected to a cycling test under 1 C-rate. After completing 700 cycles, it was then rested and exposed to a cycling test under 5 C-rate for 700 cycles. The primary purpose of these tests is to identify the degradation characteristics of the cell when it is operated at two different C-rates.

Product name	LTO battery cell
Nominal capacity	≈ 3.09 Ah
Nominal voltage	2.4 V
Work Voltage (min-max)	1.7 v- 2.9 v
Dimension	W63 × D14 × H97 mm
Max charging current	10 C
Max discharging current	10 C

Table 2 Specification of the battery cell

This experiment is conducted with a used cell. Therefore, the maximum capacity reached during the first cycle will be considered the nominal voltage, while all other

values will be adopted from the factory data sheet. The specifications of the tested cell are shown in table 2 above.

During the first stage of the test, the cell was fully charged and discharged at 1 C-rate. Charging and discharging time lasted around 1 hour for each. The cell was rested for 30 minutes after each charging or discharging phase. The test has been conducted due to the following procedure:

1. Charging the cell from 0 % to 100% SoC under 1 C-rate using Constant Current Constant Voltage (CCCV). CCCV values are set to current= 2.9 A and voltage= 2.65 V. The charging process ended either when the cell reached the upper voltage limit of 2.65 v or is fully charged.
2. Resting for 30 minutes to achieve stability.
3. Discharging the cell completely using Constant Current (CC) technique with a current equal to 2.9A. The discharge phase is stopped by the cell is fully discharged or reached the lower voltage limit at 1.7 V.
4. Resting for 30 minutes.
5. Repeating the above process for 100 cycles.
6. Applying Hybrid Pulse Power Characterization (HPPC) test profile on the cell to estimate cell DC resistance and its adjoining parameters.
7. The tests have been terminated after completing 700 cycles under 1 C-rate.

The cell maximum capacity then degrades from 3.09 to 3.05 Ah. 3.05 Ah is considered the nominal capacity for this stage of the test. As in the first stage, the temperature was fixed at 25 ^{o}C. However, the cell was fully charged/discharged at a 5 C-rate using the following steps:

1. Charging the cell from 0 % to 100% SoC under 5 C-rate using CCCV. The charging current is set to 14.5 A while the voltage value stays at 2.65 V.

The charging process ended either when the cell reached the upper voltage limit of 2.8 v or when it was fully charged.

2. Resting for 30 minutes.
3. Discharging the cell completely using Constant Current (CC) technique with a current equal to 14.5 A. The discharge phase was stopped when the cell fully discharged or reached the lower voltage limit at 1.7 V.
4. Resting for 30 minutes
5. Repeating the above process 20 times.
6. Charging and discharging the cell under 1 C conditions.
7. Applying the HPPC test.
8. Repeating the above sequence until reaching 700 complete cycles under 5 C-rate.

The equipment used in this experiment includes WEISS, Neware Battery Cell testing equipment, and a computer. Technik climate test chamber was used to sustain the ambient temperature around the cell. Neware Battery Cell Testing Equipment (BTS8000 model) was used to cycle the cell. Neware was connected to PC, where test parameters were controlled, and the results were recorded. The data were recorded every second during the test. Figure 12 below shows the configuration of the test equipment.

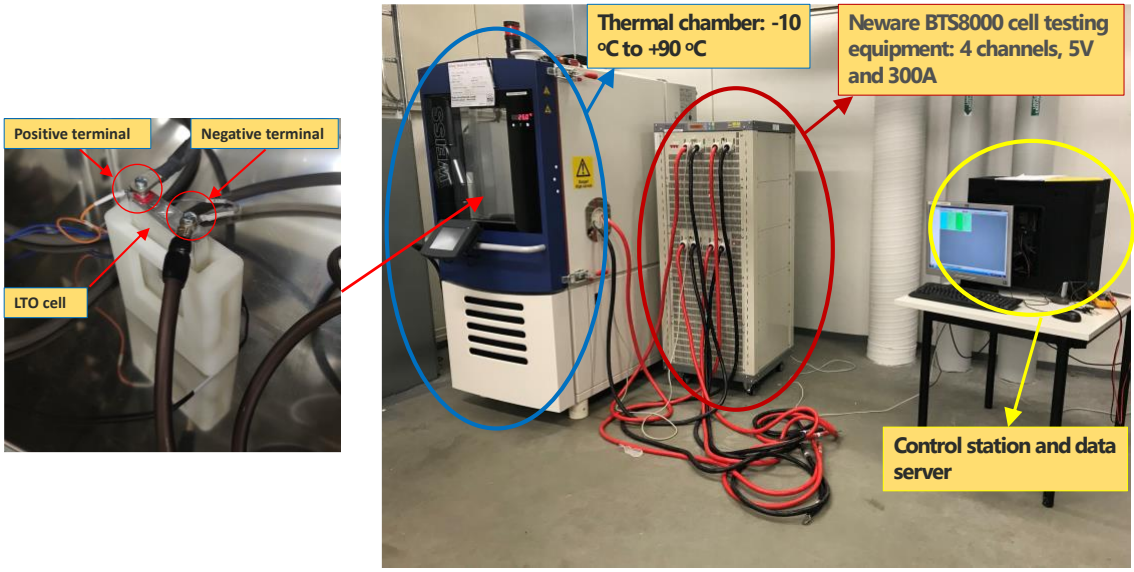


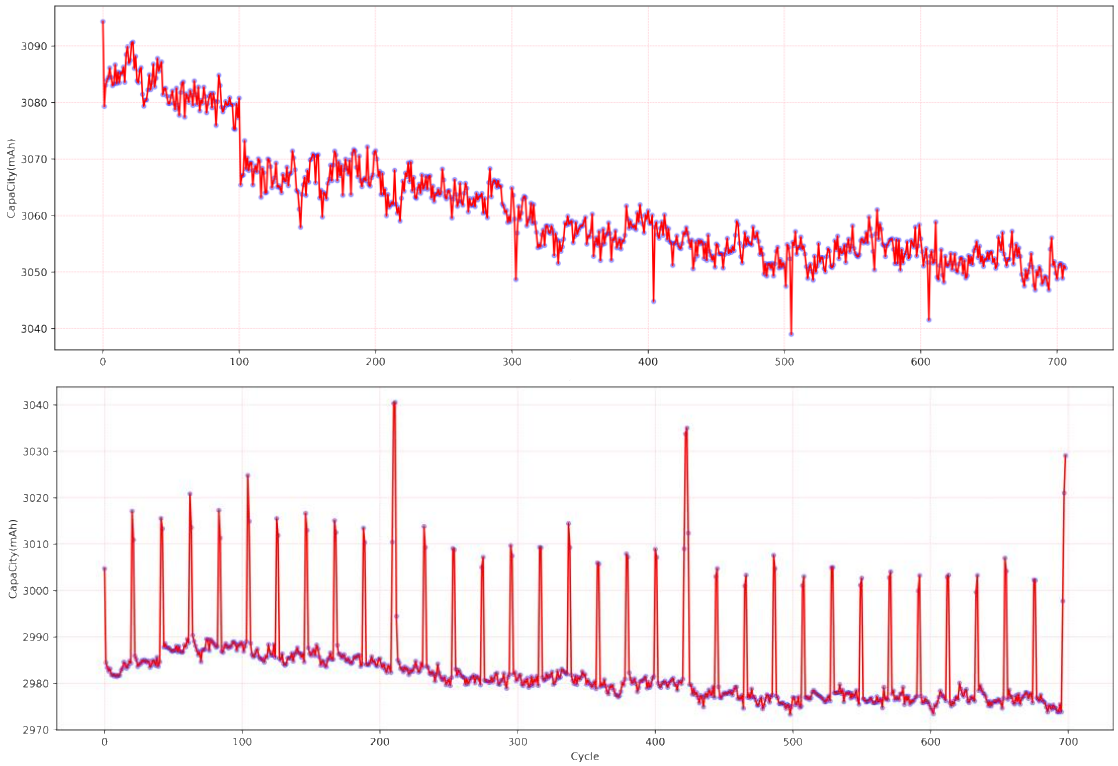
Figure 12 Equipment used for cell testing

4.3 Data preprocessing

here were two sets of data gathered from the test; the first set is the results of the 1C cycling test, while the second set is the output of 5C. The original 1C set contains 3355347 samples and 12 columns, while the 5C sets contained 4150048 samples and 12 columns. Each sample in both tests represents the recorded values for one second. The most valuable parameters obtained from the tests are voltage (V), charging/discharging current(mA), capacity (mAh), energy (mWh), cycle number, status (Rest, CCCV charge, CC charge, CC discharge).

To train the model, only charging data under the CCCV protocol are used. There are two fundamental reasons for this. First, the cell has been through repeated and symmetrical charging/discharging patterns, so any effect of the charging phase on capacity will be seen during the discharging phase and vice versa. Therefore, it is unnecessary to include data from both phases. Second, using only CCCV charging data reduces the compute power required for training the model, which has been considered an obstacle due to the limited computing power resources. Thus, all data related to the discharging phase, HPPC test phase, and rest phase are excluded from the set of data injected into the model. Moreover, only data related to each cycle's maximum capacity has been selected to further reduce the needed computation power for training the model. This reduced the data samples for 1C and 5C to 703 and 699 samples, respectively. Figure 13 shows the capacity fade for each cycle for 1C and 5C, respectively. The steep decline or rise in 1C test figures corresponds to the capacity values following the HPPC test phase, while spikes in values for the 5 C-rate test represent the values when cycling under 1 C-rate.

a)



b)

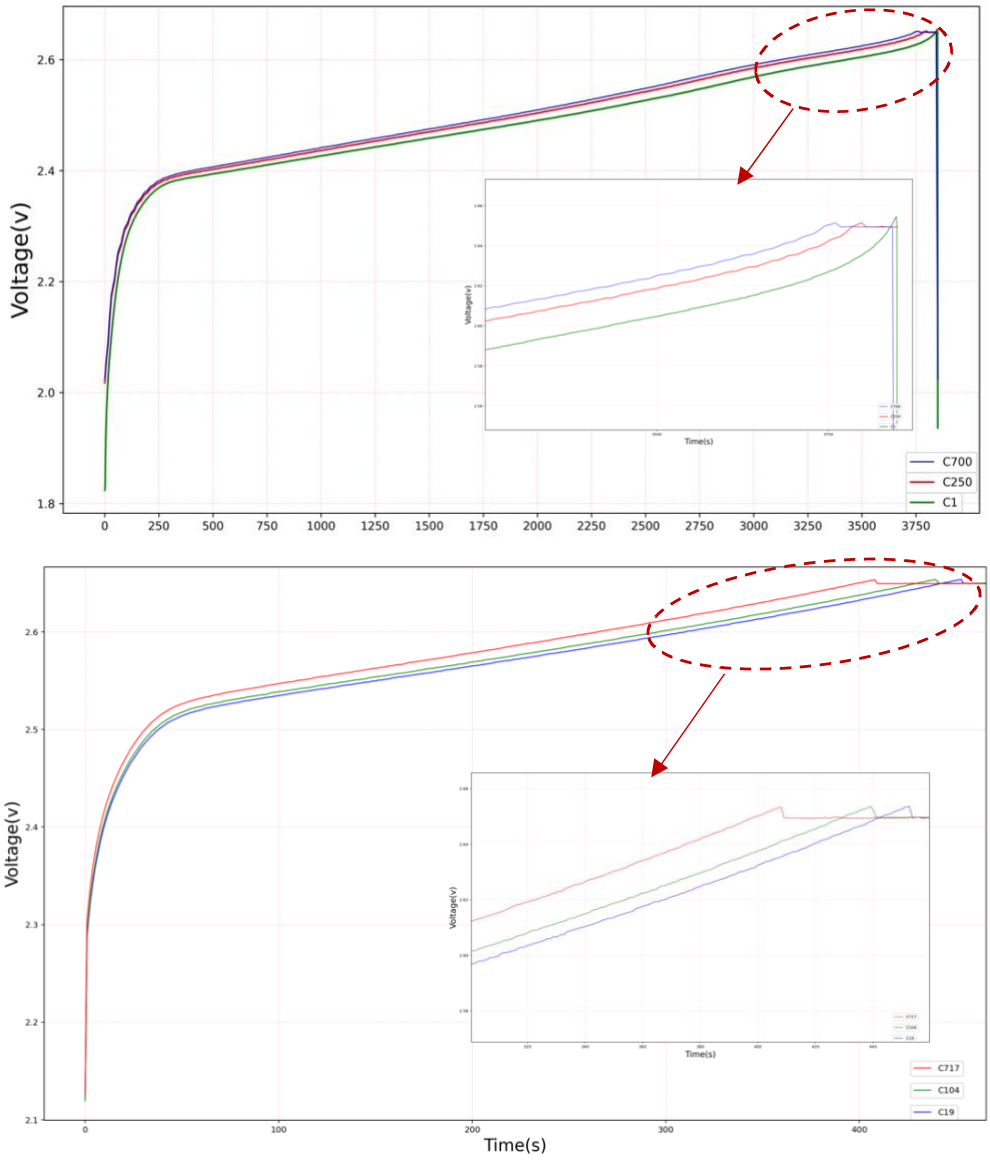
Figure 13 Capacity degradation over cycle for a) 1C test b) 5C test

4.5 Feature extraction and selection

t is a process where new sets of features are extracted from the original data to capture more insights from the dataset. Therefore, due to the limitations of the battery data parameters, it is essential to extract some features relevant to capacity degradation (Zhang et al., 2017). In this work, some features have been obtained from analyzing voltage and energy values over time for both 1C and 5C tests. It is noticed that at the last testing cycles, the voltage reaches the upper cut-off value faster than in the first cycles. Figures 14a and 14b show a decline in the time required to reach from one voltage to another as the number of cycles increases for both tests. This behavior can be noticed clearly at the final stage of the curve when the cell voltage approaches the upper cut-off voltage of 2.65 v.

(Li et al., 2021) investigated this phenomenon and considered it as an indicator of capacity degeneration.

a)



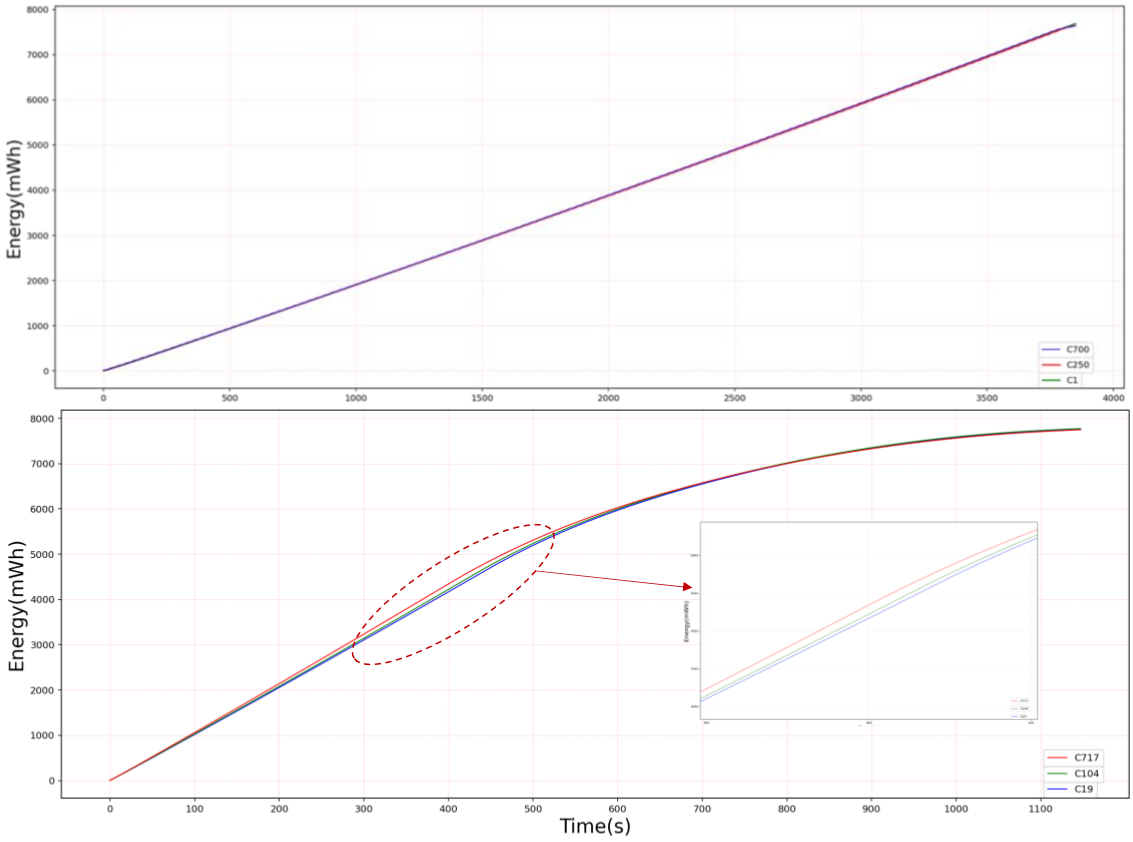
b)

Figure 14 The cell voltage during CCCV phase for a) 1C test and b) 5C test at different degradation levels.

In the same way, the amount of energy stored in the cell during the charging phase varies over cycles. As the cell degrades, its ability to store energy reduces, and loss in

its energy occurs faster than in initial cycles. Herein, this is considered an indicator of capacity fade phenomena. As such, the change in the stored energy amount over cycles is extracted as a feature. Figure 15 shows the pattern of the energy stored in the cell over different cycles for 1C and 5C testing systems.

a)



b)

Figure 15 The cell energy during CCCV phase for a) 1C test and b) 5C test at different degradation levels.

Thus, this study extracted features from voltage and energy variables. Equation 7 was used to measure the transit time between V_i and V_{i+1}.

$$DT = |t_{v2} - t_{v1}| \quad (7)$$

DT stands for the change in time for specific voltage intervals, t_{v1} is the time at v1, and t_{v2} is the time at v2. In this way, the voltage data is divided into six ranges for each cycle, with the time passed for each interval considered a feature.

Likewise, the energy produced during the same previous voltage ranges was extracted to include them in the dataset. The amount of energy stored in a specific range can be calculated as follow.

$$ES = |E_{v2} - E_{v1}| \quad (8)$$

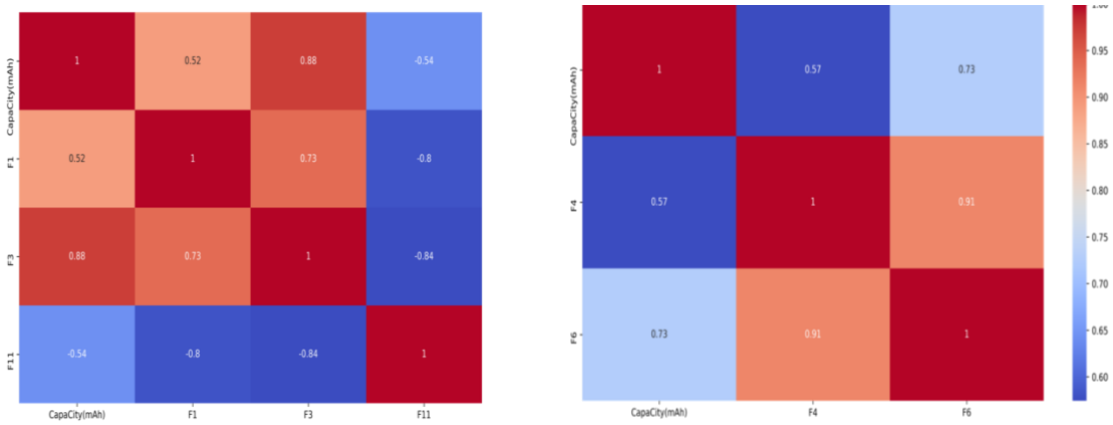
Where *ES* represent the energy stored in the cell during a specific voltage range between *v1* and *v2*, E_{v2} represents the energy at the end of voltage range *v2*, and E_{v1} represents the energy at the beginning of the same voltage range *v1*. Table 3 presents the extracted features.

Time-based features	Description	Energy-based Features	Description
F1	Time the cell needed to transit from 2 to 2.2 volts	F7	Energy stored during voltage range 2 to 2.2v
F2	Time the cell needed to transit from 2.2 to 2.4 volts	F8	Energy stored during voltage range 2 to 2.4 volts
F3	Time the cell needed to transit from 2.4 to 2.5volts	F9	Energy stored during voltage range 2.4 to 2.5 v
F4	Time the cell needed to transit from 2 to 2.5volts	F10	Energy stored during voltage range 2 to 2.5 v
F5	Time the cell needed to transit from 2.5 to 2.6 volts	F11	Energy stored during voltage range 2.5 to 2.6 v
F6	Time the cell needed to transit from 2.6 to 2.7 volts	F12	Energy stored during voltage range 2.6 to 2.7 v

Table 3 The extracted features

After extracting the features, it is necessary to choose only features that score high correlation values with the capacity using a specific filtering method. Herein, the Pearson correlation method was employed. It is commonly used for numerical parameters to measure the linear relationships between them (Williams et al., 2020). Values between 1 and -1 are assigned, with 0 indicating no correlation, 1 indicating total positive correlation, and -1 indicating a total negative correlation. In other words, positive values over 0.8 mean that the variables have a substantial direct relationship, while negative values of less than -0.8 imply a strong inverse relationship (Berman,

2016). However, in this study, thresholds between (-)0.5 and (-)0.95 have been set to filter the features. Therefore, any feature that scored value out of the threshold range with capacity is removed from the datasets. Similarly, either feature will be deleted if it scores a value out of the threshold range with another feature. The correlation has been applied only to the training set to overcome overfitting and underfitting problems. Figure 16 below shows the correlation maps for both tests after filtering the original sets with the specified thresholds.



a) 1C b) 5C

Figure 16 Correlation map for both test

4.6 Implementation of the LSTM

After finishing data pre-processing and feature extractions, the datasets become ready to be trained and tested on the model. Nevertheless, the data needs to be transformed into a standard scale since LSTM is a scale-sensitive. Thus, the min-max normalization method in equation 9 has been applied to scale data into the range [0,1] (Witten et al., 2016).

$$yi = \frac{x_i - \min(x)}{\max(x) - \min(x)} \qquad (9)$$

Where yi corresponds to xi value after scaling, and max(x) and min(x) represent the maximum and the minimum observations in the dataset, respectively.

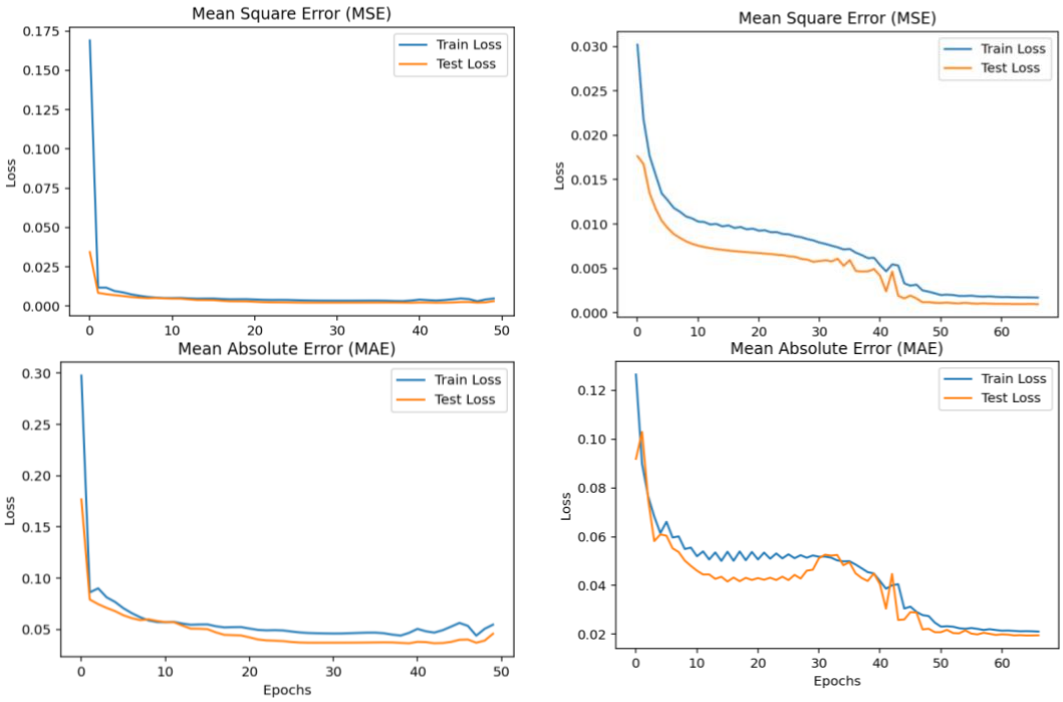
The model was constructed with one input layer, one hidden layer of 40 LSTM units, followed by one dense output layer. Linear function was used as an activation function. Adam optimizer was applied, while mean squared error was employed as a loss function. The model was trained on 70 percent of the scaled data. The remainder of the data was assigned for the testing part. LSTM algorithm requires three-dimension data. Hence, before injecting the data, they have been modified to 3-dimensional arrays representing features, samples, and timestep of 30 cycles as dimensions. Afterward, Model fitting was done for 50 training epochs with a batch size of 20 to estimate the next capacity value based on values of 30 previous cycles. The model parameters are summarized in table 4 below.

Parameter	Value
No. of hidden layers	1 layer
No. of LSTM units	40
Data split	70 % for training, 30% for testing
Epoch	50
Batch size	20
Loss method	MSE, MAE
Metric	MAE, RMSE, R^2
Optimizer	Adam
Activation function	Linear

Table 4 LSTM model parameters

5 Results and discussion

This LSTM-RNN model follows an online capacity face estimation method; hence the previous values of the capacity were always available to be included as input. Two loss functions were implied to minimize the error of the model; mean square error (MSE), which is the most common loss function used for regression problems, and mean absolute error (MAE). Figure 17 shows the convergence curves of these two functions over the epochs for both tests. From all graphs, it can be seen how the model learned from both the train set and test set, where the error reduced significantly to lower values.



a) b)

Figure 17 Loss function of a) 1C test, b) 5C test

The model showed good skills in tracking and estimating the capacity fade pattern for both tests. Three criteria were implemented to measure the prediction performance. Root mean squared error (RMSE), mean absolute error (MAE), and goodness-of-fit (R 2). Values change between zero and infinity for the two first metrics, where zero is the

idle result. Instead, R^2 output alters within 0 and 1, where 1 is the perfect value. The following formulas mathematically define the metrics mentioned above.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \quad (10)$$

$$MAE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|} \quad (11)$$

$$R^2 = 1 - \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 / \sum_{i=1}^{m}(y_i - \bar{y}_i)^2 \quad (12)$$

$\hat{y}_i, y_i, \bar{y}_i$ represent the estimated value, the actual value, and the average value, respectively. While m refers to the total number of samples.

RMSE is the prediction error's standard deviation, which calculates how data are distributed around the regression line. MAE measures the average of the absolute difference between actual capacity and predicted capacity. In contrast, R^2 defines the precision of the model (Chen et al., 2020).

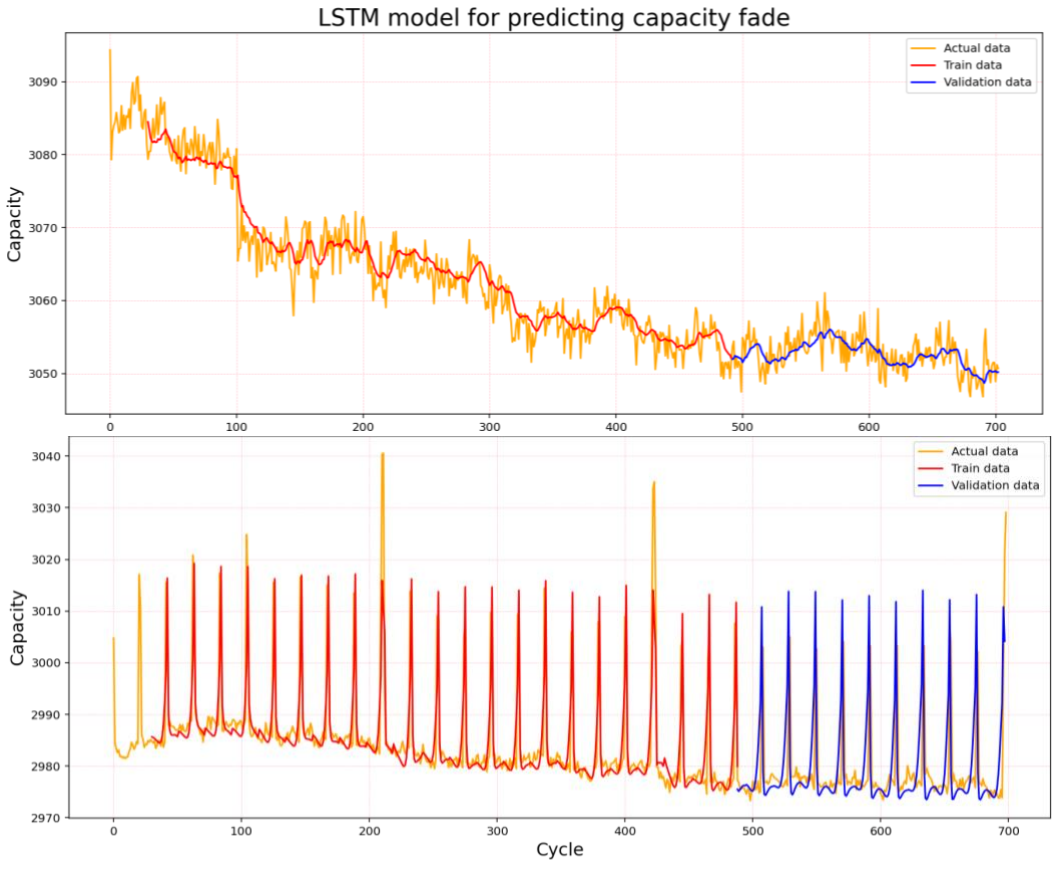
	1C test		**5C test**	
	Training set	Testing sets	Training set	Testing sets
MAE	1.99	1.713	2.637	2.483
RMSE	2.57	2.186	4.814	4.265
R^2	0.91	0.285	0.777	0.766

Table 5 comparison of model results for both tests.

The same parameters in table 4 are applied for training the models of both tests. In table 5, a comparison between the results of both tests for the training and testing set is shown. For the 1 C-rate cycling test, the model scored 1.99 for MAE and 2.57 for

RMSE; these values indicate a narrow estimation error range. The model has performed well also in the testing set considering MAE and RMSE values. R2 scores are dissimilar with 0.91 and 0.285 for training and testing sets, respectively. These values show 91% similarities between the predicted and actual values for training sets and only 28.5% for testing sets. On the contrary, for the 5C model, R^2 results are almost similar for the test set and train set, while the values of MSE and MAE are slightly higher than for the 1C test. Figure 16 presents the prediction performance of the model for both tests.

a)



b)

Figure 18 prediction performance for a) 1C model & b) 5C model

6 Conclusion

The adoption of Lithium-ion batteries as energy storage is hindered by various factors, precisely the complexity and the nonlinearity nature of their deterioration behavior. Hence, efforts have been dedicated to first developing these batteries and making them last longer by tolerating different critical operation factors. Second, building models that replicate the functionality of these batteries, estimating the degradation rate and the end of life in advance to overcome the failures and perform the maintenance duties in advance.

This study reviewed the most common mechanisms and factors that initiate and enhance the degradation phenomena of Lithium-ion batteries during storage and operation states. It also related the discussed mechanisms to factors that impact them.

In the practical part, this thesis applies organized tests in a successful version of these batteries called Lithium titanate cell LTO to estimate its degradation rate using LSTM-RNN machine learning model. Each test completed around 700 cycles that were interspersed with rest periods and HPPC tests.

Under the 1C test, the cell degrades around 1.5 % from the nominal value of 3090 mAh to the last recorded maximum value of 3050 mAh. The last maximum recorded value under the 1C test is considered the nominal value of the 5 C test. Accordingly, the cell lost only 0.7% of its capacity under the 5 C test. In sum, the cell degrades almost 2.2% over around 1400 completed cycles.

The model may be enhanced by extracting geometrics features from the measurements' curves and by including internal temperature and internal resistance values.

Nevertheless, the model results cannot be generalized and are limited to this thesis experiment for many reasons. Firstly, the experiment has been done on only one cell, so the degradation rates of the first test influence the rates of the second one. Secondly, the experiments were done to accelerate the degeneration of the cell. Hence they do not replicate the real-world applications of batteries. Finally, although the model succeeded in tracking the general trends of capacity fade, the model experienced a lack of ability to predict precisely when it comes to the details. LSTM-RNN model that is used here was a simple one with only one hidden layer, and its hyperparameters were not tuned to the perfect values. Hence, the model could perform better when increasing the hidden layers, and the hyperparameters are optimized. Also, the data used to train the model was insufficient, and there was also a low correlation between features and capacity values. Thus, to achieve better results, one can work on optimizing the model parameters, adding more experimental data, reducing the filtering thresholds, and extracting features that are more relevant to capacity fade. The previous limitations points can be considered for future research to get better and more practical results from the model. Also, to get a clearer view of LSTM model performance, it can be compared with another machine learning model performance such as Support Vector Machine (SVM).

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938

Akinyele, D., & Rayudu, R. (2014). Review of energy storage technologies for sustainable power networks. *Sustainable Energy Technologies and Assessments*, *8*, 74–91. https://doi.org/10.1016/j.seta.2014.07.004

Amine, K., Liu, J., & Belharouak, I. (2005). High-temperature storage and cycling of C-LiFePO4/graphite Li-ion cells. *Electrochemistry Communications*, *7*(7), 669–673. https://doi.org/10.1016/j.elecom.2005.04.018

Berman, J. J. (2016). Understanding Your Data. *Data Simplification*, 135–187. https://doi.org/10.1016/b978-0-12-803781-2.00004-7

Birkl, C. R., Roberts, M. R., McTurk, E., Bruce, P. G., & Howey, D. A. (2017). Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, *341*, 373–386. https://doi.org/10.1016/j.jpowsour.2016.12.011

Chahbaz, A., Meishner, F., Li, W., Ünlübayir, C., & Uwe Sauer, D. (2021). Non-invasive identification of calendar and cyclic ageing mechanisms for lithium-titanate-oxide batteries. *Energy Storage Materials*, *42*, 794–805. https://doi.org/10.1016/j.ensm.2021.08.025

Chandrasekaran, R. (2014). Quantification of bottlenecks to fast charging of lithium-ion-insertion cells for electric vehicles. *Journal of Power Sources*, *271*, 622–632. https://doi.org/10.1016/j.jpowsour.2014.07.106

Chen, Z., Xue, Q., Wu, Y., Shen, S., Zhang, Y., & Shen, J. (2020). Capacity Prediction and Validation of Lithium-Ion Batteries Based on Long Short-Term Memory Recurrent Neural Network. *IEEE Access*, *8*, 172783–172798. https://doi.org/10.1109/access.2020.3025766

Climate Change Indicators: Greenhouse Gases. (2021, July 14). US EPA. Retrieved February 17, 2022, from https://www.epa.gov/climate-indicators/greenhouse-gases#ref1

Dataasprint. (2020). [Graph]. Dataaspirant. https://dataaspirant.com/how-recurrent-neural-network-rnn-works/

Dubarry, M., Truchot, C., Liaw, B. Y., Gering, K., Sazhin, S., Jamison, D., & Michelbacher, C. (2011). Evaluation of commercial lithium-ion cells based on composite positive electrode for plug-in hybrid electric vehicle applications. Part II. Degradation mechanism under 2C cycle aging. *Journal of Power Sources*, *196*(23), 10336–10343. https://doi.org/10.1016/j.jpowsour.2011.08.078

Edge, J. S., O'Kane, S., Prosser, R., Kirkaldy, N. D., Patel, A. N., Hales, A., Ghosh, A., Ai, W., Chen, J., Yang, J., Li, S., Pang, M. C., Bravo Diaz, L., Tomaszewska, A., Marzook, M. W., Radhakrishnan, K. N., Wang, H., Patel, Y., Wu, B., & Offer, G. J. (2021). Lithium ion battery degradation: what you need to know. *Physical Chemistry Chemical Physics*, *23*(14), 8200–8221. https://doi.org/10.1039/d1cp00359c

Finegan, D. P., Scheel, M., Robinson, J. B., Tjaden, B., Hunt, I., Mason, T. J., Millichamp, J., di Michiel, M., Offer, G. J., Hinds, G., Brett, D. J., & Shearing, P. R. (2015). In-

operando high-speed tomography of lithium-ion batteries during thermal runaway. *Nature Communications*, *6*(1). https://doi.org/10.1038/ncomms7924

Gallo, A., Simões-Moreira, J., Costa, H., Santos, M., & Moutinho Dos Santos, E. (2016). Energy storage in the energy transition context: A technology review. *Renewable and Sustainable Energy Reviews*, *65*, 800–822. https://doi.org/10.1016/j.rser.2016.07.028

Greenhouse Gas Emissions from Energy Data Explorer – Analysis. (2021, November 10). IEA. Retrieved February 17, 2022, from https://www.iea.org/articles/greenhouse-gas-emissions-from-energy-data-explorer

Guo, J., Li, Y., Pedersen, K., & Stroe, D. I. (2021). Lithium-Ion Battery Operation, Degradation, and Aging Mechanism in Electric Vehicles: An Overview. *Energies*, *14*(17), 5220. https://doi.org/10.3390/en14175220

Guo, L., Thornton, D. B., Koronfel, M. A., Stephens, I. E. L., & Ryan, M. P. (2021). Degradation in lithium ion battery current collectors. *Journal of Physics: Energy*, *3*(3), 032015. https://doi.org/10.1088/2515-7655/ac0c04

Haykin, S. (2022). *Neural Networks And Learning Machines* (3rd ed.). Pearson Education; Third edition (1 April 2016).

He, Y. B., Ning, F., Yang, Q. H., Song, Q. S., Li, B., Su, F., Du, H., Tang, Z. Y., & Kang, F. (2011). Structural and thermal stabilities of layered Li(Ni1/3Co1/3Mn1/3)O2 materials in 18650 high power batteries. *Journal of Power Sources*, *196*(23), 10322–10327. https://doi.org/10.1016/j.jpowsour.2011.08.042

Hendricks, C., Williard, N., Mathew, S., & Pecht, M. (2015). A failure modes, mechanisms, and effects analysis (FMMEA) of lithium-ion batteries. *Journal of Power Sources*, *297*, 113–120. https://doi.org/10.1016/j.jpowsour.2015.07.100

Jaguemont, J., Boulon, L., & Dubé, Y. (2016). A comprehensive review of lithium-ion batteries used in hybrid and electric vehicles at cold temperatures. *Applied Energy*, *164*, 99–114. https://doi.org/10.1016/j.apenergy.2015.11.034

Kabir, M. M., & Demirocak, D. E. (2017). Degradation mechanisms in Li-ion batteries: a state-of-the-art review. *International Journal of Energy Research*, *41*(14), 1963–1986. https://doi.org/10.1002/er.3762

Kalair, A., Abas, N., Saleem, M. S., Kalair, A. R., & Khan, N. (2020). Role of energy storage systems in energy transition from fossil fuels to renewables. *Energy Storage*, *3*(1). https://doi.org/10.1002/est2.135

Keil, P., & Jossen, A. (2016). Charging protocols for lithium-ion batteries and their impact on cycle life—An experimental study with different 18650 high-power cells. *Journal of Energy Storage*, *6*, 125–141. https://doi.org/10.1016/j.est.2016.02.005

Keras: the Python deep learning API. (2015, March 27). Keras. Retrieved April 27, 2022, from https://keras.io/

Khan, N., Dilshad, S., Khalid, R., Kalair, A. R., & Abas, N. (2019). Review of energy storage and transportation of energy. *Energy Storage*, *1*(3). https://doi.org/10.1002/est2.49

Li, P., Zhao, Y., Shen, Y., & Bo, S. H. (2020). Fracture behavior in battery materials. *Journal of Physics: Energy*, *2*(2), 022002. https://doi.org/10.1088/2515-7655/ab83e1

Li, Y., Liu, K., Foley, A. M., Zülke, A., Berecibar, M., Nanini-Maury, E., van Mierlo, J., & Hoster, H. E. (2019). Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. *Renewable and Sustainable Energy Reviews*, *113*, 109254. https://doi.org/10.1016/j.rser.2019.109254

Li, Y., Stroe, D. I., Cheng, Y., Sheng, H., Sui, X., & Teodorescu, R. (2021). On the feature selection for battery state of health estimation based on charging–discharging profiles. *Journal of Energy Storage*, *33*, 102122. https://doi.org/10.1016/j.est.2020.102122

Lin, C., Tang, A., & Wang, W. (2015). A Review of SOH Estimation Methods in Lithium-ion Batteries for Electric Vehicle Applications. *Energy Procedia*, *75*, 1920–1925. https://doi.org/10.1016/j.egypro.2015.07.199

Lin, X., Khosravinia, K., Hu, X., Li, J., & Lu, W. (2021). Lithium Plating Mechanism, Detection, and Mitigation in Lithium-Ion Batteries. *Progress in Energy and Combustion Science*, *87*, 100953. https://doi.org/10.1016/j.pecs.2021.100953

Lu, C., Tao, L., & Fan, H. (2014). Li-ion battery capacity estimation: A geometrical approach. *Journal of Power Sources*, *261*, 141–147. https://doi.org/10.1016/j.jpowsour.2014.03.058

Matplotlib. (2003). Matplotlib. Retrieved April 27, 2022, from https://matplotlib.org/

Meda, U. S., Lal, L., M, S., & Garg, P. (2022). Solid Electrolyte Interphase (SEI), a boon or a bane for lithium batteries: A review on the recent advances. *Journal of Energy Storage*, *47*, 103564. https://doi.org/10.1016/j.est.2021.103564

Ng, M. F., Zhao, J., Yan, Q., Conduit, G. J., & Seh, Z. W. (2020). Author Correction: Predicting the state of charge and health of batteries using data-driven machine learning. *Nature Machine Intelligence*. https://doi.org/10.1038/s42256-020-0191-4

NumPy. (2006). Numpy. Retrieved April 27, 2022, from https://numpy.org/

Ohsaki, T., Kishi, T., Kuboki, T., Takami, N., Shimura, N., Sato, Y., Sekino, M., & Satoh, A. (2005). Overcharge reaction of lithium-ion batteries. *Journal of Power Sources*, *146*(1–2), 97–100. https://doi.org/10.1016/j.jpowsour.2005.03.105

pandas - Python Data Analysis Library. (2008, January 11). Pandas. Retrieved April 27, 2022, from https://pandas.pydata.org/

Redondo-Iglesias, E., Venet, P., & Pelissier, S. (2018). Calendar and cycling ageing combination of batteries in electric vehicles. *Microelectronics Reliability*, *88–90*, 1212–1215. https://doi.org/10.1016/j.microrel.2018.06.113

Scikit-learn. (2007, June). Scikit-Learn. Retrieved April 27, 2022, from https://scikit-learn.org/stable/

Seaborn. (2012). Seaborn. Retrieved April 27, 2022, from https://seaborn.pydata.org/

Song, Y., Liu, X., Ren, D., Liang, H., Wang, L., Hu, Q., Cui, H., Xu, H., Wang, J., Zhao, C., Zuo, X., Xu, G., Amine, K., & He, X. (2021). Simultaneously Blocking Chemical Crosstalk and Internal Short Circuit via Gel-Stretching Derived Nanoporous

Non-Shrinkage Separator for Safe Lithium-Ion Batteries. *Advanced Materials*, *34*(2), 2106335. https://doi.org/10.1002/adma.202106335

Sufyan, M., Rahim, N. A., Aman, M. M., Tan, C. K., & Raihan, S. R. S. (2019). Sizing and applications of battery energy storage technologies in smart grid system: A review. *Journal of Renewable and Sustainable Energy*, *11*(1), 014105. https://doi.org/10.1063/1.5063866

Sui, X., ŚWierczyński, M., Teodorescu, R., & Stroe, D. I. (2021). The Degradation Behavior of LiFePO4/C Batteries during Long-Term Calendar Aging. *Energies*, *14*(6), 1732. https://doi.org/10.3390/en14061732

Sun, P., Zhang, X., Wang, S., & Zhu, Y. (2022). Lithium-ion battery degradation caused by overcharging at low temperatures. *Thermal Science and Engineering Progress*, *30*, 101266. https://doi.org/10.1016/j.tsep.2022.101266

Tensorflow. (2015, November 9). Tensorflow. Retrieved April 27, 2022, from https://www.tensorflow.org/

Tollefson, J. (2021). Carbon emissions rapidly rebounded following COVID pandemic dip. *Nature*. https://doi.org/10.1038/d41586-021-03036-x

van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Vetter, J., Novák, P., Wagner, M., Veit, C., Möller, K. C., Besenhard, J., Winter, M., Wohlfahrt-Mehrens, M., Vogler, C., & Hammouche, A. (2005). Ageing

mechanisms in lithium-ion batteries. *Journal of Power Sources*, *147*(1–2), 269–281. https://doi.org/10.1016/j.jpowsour.2005.01.006

Williams, B., Halloin, C., Löbel, W., Finklea, F., Lipke, E., Zweigerdt, R., & Cremaschi, S. (2020). Data-Driven Model Development for Cardiomyocyte Production Experimental Failure Prediction. *Computer Aided Chemical Engineering*, 1639–1644. https://doi.org/10.1016/b978-0-12-823377-1.50274-3

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. (2016). *Data Mining: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data Management Systems)* (4th ed.). Morgan Kaufmann. https://doi.org/10.1016/C2015-0-02071-8

Xiong, R., Pan, Y., Shen, W., Li, H., & Sun, F. (2020). Lithium-ion battery aging mechanisms and diagnosis method for automotive applications: Recent advances and perspectives. *Renewable and Sustainable Energy Reviews*, *131*, 110048. https://doi.org/10.1016/j.rser.2020.110048

Xu, B., Oudalov, A., Ulbig, A., Andersson, G., & Kirschen, D. S. (2018). Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment. *IEEE Transactions on Smart Grid*, *9*(2), 1131–1140. https://doi.org/10.1109/tsg.2016.2578950

Yang, X. G., Ge, S., Liu, T., Leng, Y., & Wang, C. Y. (2018). A look into the voltage plateau signal for detection and quantification of lithium plating in lithium-ion cells. *Journal of Power Sources*, *395*, 251–261. https://doi.org/10.1016/j.jpowsour.2018.05.073

Yang, Y., Bremner, S., Menictas, C., & Kay, M. (2018). Battery energy storage system size determination in renewable energy systems: A review. *Renewable and Sustainable Energy Reviews*, *91*, 109–125. https://doi.org/10.1016/j.rser.2018.03.047

Zhang, L., Liu, L., Yang, R., Wang, K., Chen, Z., & Li, M. (2017). A roadmap for modeling and feature extraction of energy storage battery pack for marine energy power station. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*. https://doi.org/10.1109/phm.2017.8079289

Zubi, G., Dufo-López, R., Carvalho, M., & Pasaoglu, G. (2018). The lithium-ion battery: State of the art and future perspectives. *Renewable and Sustainable Energy Reviews*, *89*, 292–308. https://doi.org/10.1016/j.rser.2018.03.002

Appendices

Appendix 1. Python programming code

1. Importing Libraries and Data

```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
In [2]:
#Import data
df= pd.read_csv ('final_df.csv')
print('data shape is', df.shape)
print(df.info()) # shows some information about the data
df.head()
In [3]:
df['Time'] = pd.to_datetime(df['Absolute Time']) # convert the 'Absolute Time' column type from obje
ct to datetime
print(df.info())

In [4]:
#pd.set_option('display.max_rows', 5000) # show 5000 samples
a= df[df.Status == 'CCCV_Chg'] # choose the only 'CCCV_Chg' data
a= a[a['Cur(mA)'] != 0]      # Remove current values equal to zero
a= a[a['C-rate'] == 1]       # Choose only the tesst values unde 1C
a['Voltage(V)'] = (round(a['Voltage(V)'], 1))  """ round the voltage to one decimal value, to get more
                                                   similar records between cycles"""
a=a.reset_index(drop=True) # reset index and drop the old one
a=a.drop(columns=["Step","Status","Absolute Time"]) # drop unuseful columns
a.head(5000)

In [5]:
g=a.groupby('Voltage(V)').count() # count the data amount based on the recorded voltage values
g.sort_values("Voltage(V)", ascending = True).head(5000)
```

2. Features Extraction

```
In [6]:

v1 = a.where(a['Voltage(V)'] == 2) # extract data that recorded at voltage = 2 volt
v2 = a.where(a['Voltage(V)'] == 2.2)
v3 = a.where(a['Voltage(V)'] == 2.4)
```

```
v4 = a.where(a['Voltage(V)'] == 2.5)
v5 = a.where(a['Voltage(V)'] == 2.6)
v6 = a.where(a['Voltage(V)'] == 2.7)


v1=v1.dropna() # Drop non-number records
v2=v2.dropna()
v3=v3.dropna()
v4=v4.dropna()
v5=v5.dropna()
v6=v6.dropna()

v1.shape,v2.shape,v3.shape,v4.shape,v5.shape,v6.shape #shape of each datasets

In [7]:
x1=v1.groupby(['Cycle'])[["Voltage(V)","Time","Energy(mWh)"]].nth(0) """ Group the data by cycle
 and choose the first recorded data when V=2v"""
x2=v2.groupby(['Cycle'])[["Voltage(V)","Time","Energy(mWh)"]].nth(0)  # V=2.2
x3=v3.groupby(['Cycle'])[["Voltage(V)","Time","Energy(mWh)"]].nth(0)  # V=2.4
x4=v4.groupby(['Cycle'])[["Voltage(V)","Time","Energy(mWh)"]].nth(0)  # V=2.5
x5=v5.groupby(['Cycle'])[["Voltage(V)","Time","Energy(mWh)"]].nth(0)  # V=2.6
x6=v6.groupby(['Cycle'])[["Voltage(V)","Time","Energy(mWh)"]].nth(0)  # V=2.7

x1.shape,x2.shape,x3.shape,x4.shape,x5.shape,x6.shape
```

2.1 DT1(F1) & ES1(F7) (2v-2.2v)

```
In [8]:

result = pd.concat([x1, x2], axis=1, join="inner")
print(result.shape)
result.head()

In [9]:
# change in time dt1
start=result.iloc[:,1:2]
end= result.iloc[:,4:5]
dt1=(end - start)
dt1['Time']= dt1['Time'].astype('timedelta64[s]')
dt1=dt1.dropna()
print(dt1.head())

#change in energy de1
start=result.iloc[:,2:3]
end= result.iloc[:,5:]
```

```
de1=(end - start)
de1=de1.dropna()
print(de1.head())

In [10]:
# change in time dt1
start=result.iloc[:,1:2]
end= result.iloc[:,4:5]
dt1=(end - start)
dt1['Time']= dt1['Time'].astype('timedelta64[s]')
dt1=dt1.dropna()
print(dt1.head())

#change in energy de1
start=result.iloc[:,2:3]
end= result.iloc[:,5:]
de1=(end - start)
de1=de1.dropna()
print(de1.head())

In [11]:
#Plot
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twinx()
ax1.scatter(result.index ,dt1['Time'],  label='Time(s)',c='g',alpha=1,linewidth=1) #voltage
ax2.scatter(result.index , de1['Energy(mWh)'],c='r', label='Energy(mwh)', alpha=1,linewidth=1)  #
energy

plt.title("Time and enenrgy variation over cycle for charging vlaues (2v-2.2v)",size = 20)
ax1.legend(loc=(0.01,0.9)); ax2.legend(loc=(0.01,0.95))
ax1.set_xlabel('Cycle',size=(15))
ax1.set_ylabel('Time(s)',size=(15))
ax2.set_ylabel('Energy(mwh)',size=(15))

#ax1.set_xticks(np.arange(779, 1500, 50));
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

2.2 DT2(F2) & ES2(F8) (2.2v-2.4v)

```
In [12]:
result1 = pd.concat([x2, x3], axis=1, join="inner")
print(result1.shape)
result1.head()
```

```
In [13]:
# dt2
start=result1.iloc[:,1:2]
end= result1.iloc[:,4:5]
dt2=(end - start)
dt2['Time']= dt2['Time'].astype('timedelta64[s]')
dt2=dt2.dropna()
print(dt2.head())

#de2
start=result1.iloc[:,2:3]
end= result1.iloc[:,5:]
de2=(end - start)
de2=de2.dropna()
print(de2.head())

In [14]:
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twinx()

ax1.scatter(result1.index ,dt2['Time'],  label='Time(s)',c='g') # dt2
ax2.scatter(result1.index , de2['Energy(mWh)'],c='r', label='Energy(mwh)')  # de2

ax1.legend(loc=(0.01,0.9));
ax2.legend(loc=(0.01,0.95))

plt.title("Time and enenrgy variation over cycle for charging vlaues (2.2v-2.4v)",size = 20)
ax1.set_xlabel('Cycle',size=(15))
ax1.set_ylabel('Time(s)',size=(15))
ax2.set_ylabel('Energy(mwh)',size=(15))
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)

#fig.savefig('dv1/dt, de1/dt.png', dpi=600)
```

2.3 DT3(F3) & ES3(F9) (2.4v-2.5v)

```
In [15]:
result2 = pd.concat([x3, x4], axis=1, join="inner")
print(result2.shape)
result2.head()
```

```
In [16]:
# dt3
start=result2.iloc[:,1:2]
end= result2.iloc[:,4:5]
dt3=(end - start)
dt3['Time']= dt3['Time'].astype('timedelta64[s]')
dt3=dt3.dropna()
print(dt3.head())

#de3
start=result2.iloc[:,2:3]
end= result2.iloc[:,5:]
de3=(end - start)
de3=de3.dropna()
print(de3.head())

In [17]:
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twinx()

ax1.scatter(result2.index ,dt3['Time'],  label='Time(s)',c='g') # dt2
ax2.scatter(result2.index , de3['Energy(mWh)'],c='r', label='Energy(mwh)')  # de2

ax1.legend(loc=(0.01,0.9));
ax2.legend(loc=(0.01,0.95))

plt.title("Time and enenrgy variation over cycle for charging vlaues (2.4v-2.5v)",size = 20)
ax1.set_xlabel('Cycle',size=(15))
ax1.set_ylabel('Time(s)',size=(15))
ax2.set_ylabel('Energy(mwh)',size=(15))
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

2.4 DT4(F4) & ES4(F10) (2v-2.7v)

```
In [18]:
result3 = pd.concat([x1, x4], axis=1, join="inner")
print(result3.shape)
result3.head()

In [19]:
# dt4
```

```
start=result3.iloc[:,1:2]
end= result3.iloc[:,4:5]
dt4=(end - start)
dt4['Time']= dt4['Time'].astype('timedelta64[s]')
dt4=dt4.dropna()
print(dt4.head())

#de4
start=result3.iloc[:,2:3]
end= result3.iloc[:,5:]
de4=(end - start)
de4=de4.dropna()
print(de4.head())

In [19]:
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twinx()

ax1.scatter(result3.index ,dt4['Time'],c='g',  label='Time(s)') # dt2
ax2.scatter(result3.index , de4['Energy(mWh)'],c='r', label='Energy(mwh)')  # de2

ax1.legend(loc=(0.01,0.9));
ax2.legend(loc=(0.01,0.95))

plt.title("Time and enenrgy variation over cycle for charging vlaues (2v-2.5v)",size = 20)
ax1.set_xlabel('Cycle',size=(15))
ax1.set_ylabel('Time(s)',size=(15))
ax2.set_ylabel('Energy(mwh)',size=(15))
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

2.5 DT5(F5) & ES5(F11) (2.5v-2.6v)

```
In [20]:
result4 = pd.concat([x4, x5], axis=1, join="inner")
print(result4.shape)
result4.head()

In [21]:
# dt5
start=result4.iloc[:,1:2]
end= result4.iloc[:,4:5]
dt5=(end - start)
dt5['Time']= dt5['Time'].astype('timedelta64[s]')
```

```
dt5=dt5.dropna()
print(dt5.head())

#de5
start=result4.iloc[:,2:3]
end= result4.iloc[:,5:]
de5=(end - start)
de5=de5.dropna()
print(de5.head())
In [22]:
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twinx()

ax1.scatter(result4.index ,dt5['Time'],c='g',  label='Time(s)') # dt2
ax2.scatter(result4.index , de5['Energy(mWh)'],c='r', label='Energy(mwh)')  # de2

ax1.legend(loc=(0.01,0.9));
ax2.legend(loc=(0.01,0.95))

plt.title("Time and enenrgy variation over cycle for charging vlaues (2.5v-2.6v)",size = 20)
ax1.set_xlabel('Cycle',size=(15))
ax1.set_ylabel('Time(s)',size=(15))
ax2.set_ylabel('Energy(mwh)',size=(15))
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

2.6 DT6(F6) & ES6(F12) (2.6v-2.7v)

```
In [23]:
result5 = pd.concat([x5, x6], axis=1, join="inner")
print(result5.shape)
result5.head()

In [24]:
# dt6
start=result5.iloc[:,1:2]
end= result5.iloc[:,4:5]
dt6=(end - start)
dt6['Time']= dt6['Time'].astype('timedelta64[s]')
dt6=dt6.dropna()
print(dt6.head())

#de6
start=result5.iloc[:,2:3]
```

```
end= result5.iloc[:,5:]
de6=(end - start)
de6=de6.dropna()
print(de6.head())
```

```
In [25]:
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twinx()

ax1.scatter(result5.index ,dt6['Time'],c='g',  label='Time(s)') # dt2
ax2.scatter(result5.index , de6['Energy(mWh)'],c='r', label='Energy(mwh)')  # de2

ax1.legend(loc=(0.01,0.9));
ax2.legend(loc=(0.01,0.95))

plt.title("Time and enenrgy variation over cycle for charging vlaues (2.5v-2.6v)",size = 20)
ax1.set_xlabel('Cycle',size=(15))
ax1.set_ylabel('Time(s)',size=(15))
ax2.set_ylabel('Energy(mwh)',size=(15))

ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

2.7 Final Dataframe

```
In [26]:
#print the shape of each feature
print('dt1 shape=',dt1.shape,'; dt2 shape=',(dt2.shape),'; dt3 shape=',(dt3.shape),'; dt4 shape=',(dt4.
shape) ,'; dt5 shape=',(dt5.shape),'; dt6 shape=',(dt6.shape))

print('de1 shape=',de1.shape,'; de2 shape=',(de2.shape),'; de3 shape=',(de3.shape),'; de4 shape=',(d
e4.shape)  ,'; de5 shape=',(de5.shape),'; de6 shape=',(de6.shape))

In [27]:
#combine all features in one dataset
Dv= pd.concat([dt1,dt2,dt3,dt4,dt5,dt6,de1,de2,de3,de4,de5,de6], axis=1, join="inner")
Dv.columns = ['F1', 'F2','F3','F4','F5', 'F6','F7','F8','F9','F10','F11','F12']
print(Dv.shape[0])
Dv

In [28]:
#Extract the maximum recorded capcity for each cycle
b=a
```

```
b['CapaCity(mAh)'] = round(b['CapaCity(mAh)'], 3)
b=b.groupby('Cycle')['CapaCity(mAh)'].max()
print(b.shape[0])
b.head()
```

```
In [29]:
#combine b and DV
DV= pd.concat([b,Dv], axis=1, join="inner")
DV.head()
```

3. Plots

3.1 Ploting the maximum capacity per cycle for CCCV charging phase

```
In [30]:
#1.prepare the data
df1=df[df['C-rate'] == 1]
df1=df1[df1.Status == 'CCCV_Chg']
df1=df1.groupby(['Cycle'])[["Voltage(V)","Energy(mWh)",'CapaCity(mAh)']].max()  # V=2
df1=df1.reset_index(drop=True)

print(df1.shape)
print(df1.tail())
```

```
In [31]:
df1.head()
```

```
In [32]:
#2.plotting part
%matplotlib inline
fig= plt.figure(figsize=(18,6))
sns.lineplot(x=df1.index, y=df1['CapaCity(mAh)'], color='orange',legend=False)
plt.scatter(df1.index,df1['CapaCity(mAh)'], c='blue',s=15,)
plt.title("Capacity evolution over cycles",size = 20)
plt.xlabel("Cycle",size = 10)
plt.ion()
plt.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

3.2 Ploting the variation in time and energy at different voltage ranges

```
In [33]:
#1. prepare the data
df2=df[df.Status == 'CCCV_Chg']
df2=df2[df2['C-rate'] == 1]
df2=df2[df2['Cur(mA)'] != 0]
```

```
df2=df2.reset_index(drop=True)
print(df2.shape)
df2.head()

In [34]:
# 2. voltage ploting
%matplotlib inline

fig, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twiny();ax3=ax2.twiny();

ax1.plot(df2.index[:3852],df2.iloc[0:3852,6:7],c='g',label='C1')
ax2.plot(df2.index[785285:785795],df2.iloc[785285:785795,6:7],c='r', label='C250')
ax3.plot(df2.index[994536:995047],df2.iloc[994536:995047,6:7],c='b', label='C700', linewidth=1)

ax1.legend(loc=(0.92,0));ax2.legend(loc=(0.92,0.04));ax3.legend(loc=(0.92,0.08))
#ax1.set_xticks(np.arange(0, 4000, 250))
ax2.set_xticks([])
ax3.set_xticks([])

ax1.set_xlabel('Time(s)',size=(20))
ax1.set_ylabel('Voltage(v)',size=(20))
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)

In [35]:
#3. Energy ploting

%matplotlib inline

fig1, ax1 = plt.subplots(figsize=(18,8))
ax2 = ax1.twiny();ax3=ax2.twiny();

ax1.plot(df2.index[:3851],df2.iloc[0:3851,8:9],c='g',label='C1')
ax2.plot(df2.index[785285:785794],df2.iloc[785285:785794,8:9],c='r', label='C250')
ax3.plot(df2.index[994536:995046],df2.iloc[994536:995046,8:9],c='b', label='C700', linewidth=1)

ax1.legend(loc=(0.92,0));ax2.legend(loc=(0.92,0.04));ax3.legend(loc=(0.92,0.08))
#ax1.set_xticks(np.arange(0, 4000, 250))
ax2.set_xticks([])
ax3.set_xticks([])

ax1.set_xlabel('Time(s)',size=(20))
ax1.set_ylabel('Energy(mWh)',size=(20))
ax1.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
```

4. LSTM model

In [36]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
```

In [37]:

```
DV.head()
```

In [38]:

```
# split data into train and test sets to check the correlation on the training set
n_train = int(len(DV)*0.7)
trainset = DV.iloc[:n_train, :]
testset  = DV.iloc[n_train:,:]
print(trainset.shape),print(testset.shape);
```

In [39]:

```
#correlation heat map on the train set
%matplotlib inline
fig= plt.figure(figsize=(14,7))
sns.heatmap(trainset.corr(), annot=True, cmap='coolwarm')
fig.savefig('correlation map(1C).png', dpi=600)
```

In [40]:

```
"""Drop the coulumns that have correlation value in range less than -0.95>, bigger than 0.95 and betw
een 0.5 and-0.5 with the capacity and between each other"""

dn=DV.drop(columns=['F2','F4','F5','F6','F7','F8','F9','F10','F12'])
dn=dn.reset_index(drop=True)

# split data into train and test sets
n_train = int(len(DV)*0.7)
trainset = dn.iloc[:n_train, :]
testset  = dn.iloc[n_train:,:]
print(trainset.shape),print(testset.shape);
```

In [41]:

```
# correlation maps of the choosen features
```

```
%matplotlib inline
fig= plt.figure(figsize=(14,7))
sns.heatmap(trainset.corr(), annot=True, cmap='coolwarm')
fig.savefig('correlation map(1C).png', dpi=600)

In [42]:
#Scaling data into values between 0 and 1, for better performances.
scalar = MinMaxScaler(feature_range=(0, 1))
dnscaled = scalar.fit_transform(dn)
dnscaled[:2]

In [43]:
#define function to seprate input and output for btih test and train set.
def create_dataset(data, n_past):
  X, Y = [], []
  for i in range(n_past, len(data) ):
    X.append(data[i - n_past:i,:])
    Y.append(data[i :i + n_future, 0])

  return np.array(X),np.array(Y)

n_future = 1
n_past=30

# split data into train and test sets
n_train = int(len(dnscaled)*0.7)
train= dnscaled[:n_train]
test = dnscaled[n_train-n_past:]

trainX,trainY =create_dataset(train, n_past)
testX,testY = create_dataset(test, n_past)
print("Shape of input training set:",trainX.shape)
print("Shape of output training set",(trainY.shape))
print("Shape of input testing set:",testX.shape)
print("Shape of output testing set",(testY.shape))

In [44]:
# define the Autoencoder model

model = Sequential()
model.add(LSTM(40, activation='linear', input_shape=(trainX.shape[1], trainX.shape[2])
          ,return_sequences=False))

model.add(Dense(1, activation='linear'))
```

```
model.compile(optimizer='adam', loss='mse',metrics=['mae'])
model.summary()

In [45]:
#fitting the model
history = model.fit(trainX, trainY, epochs=49, batch_size=20, validation_data=(testX, testY), verbos
e=1, shuffle=False)

In [46]:
%matplotlib inline

# plot mse during training
plt.figure(figsize=(6,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('Mean Square Error (MSE)')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(loc='upper right')
plt.show()

# plot mae during training
plt.figure(figsize=(6,4))
plt.plot(history.history['mae'], label='Train Loss')
plt.plot(history.history['val_mae'], label='Test Loss')
plt.title('Mean Absolute Error (MAE)')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(loc='upper right')
plt.show()

In [47]:
# evaluate the model
_,train_mse = model.evaluate(trainX, trainY, verbose=0)
_,test_mse = model.evaluate(testX, testY, verbose=0)
print('Train: %.3f, Test: %.3f' % (train_mse, test_mse))

In [48]:
# Doing the  prediction and return data to original values by doing an inverse scale.

#1.for training set
train_predict = model.predict(trainX)
trp_copies = np.repeat(train_predict, dnscaled.shape[1], axis=-1)
```

```
train_predict = scalar.inverse_transform(trp_copies)[:,0]


#2.for testing set
test_predict = model.predict(testX)
tsp_copies = np.repeat(test_predict, dnscaled.shape[1], axis=-1)
test_predict = scalar.inverse_transform(tsp_copies)[:,0]

#3. return value if training output to its original value
trp_copies = np.repeat(trainY, dnscaled.shape[1], axis=-1)
trainy = scalar.inverse_transform(trp_copies)

#4. return value if testing output to its original value
tsp_copies = np.repeat(testY, dnscaled.shape[1], axis=-1)
testy = scalar.inverse_transform(tsp_copies

In [49]:
#calaclulate and print the model metrics
print('Train Mean Squared Error: %.3f'% mean_squared_error(trainy[:,0], train_predict))
print('Train Mean Absolute Error: %.3f'% mean_absolute_error(trainy[:,0], train_predict))
print('Train Root Mean Squared Error: %.3f'%np.sqrt(mean_squared_error(trainy[:,0], train_predic
t)))
print('Train r2_score: %.3f'% r2_score(trainy[:,0], train_predict))

print('\nTest Mean Squared Error: %.3f'% mean_squared_error(trainy[:,0], train_predict))
print('Test Mean Absolute Error: %.3f'% mean_absolute_error(testy[:,0], test_predict))
print('Test Root Mean Squared Error: %.3f'% np.sqrt(mean_squared_error(testy[:,0], test_predict))
)
print('Test r2_score: %.3f'% r2_score(testy[:,0], test_predict))

In [50]:
#prepare data fram of originl and predicted training sets
trainfd=pd.DataFrame({"Actual":trainy[:,0], "Predicted":np.float64(train_predict)})
trainfd['Predicted']=round(trainfd['Predicted'],3)
trainfd.index=trainfd.index+30
trainfd.tail()

In [51]:
#Plot the prediction results
%matplotlib inline
fig= plt.figure(figsize=(15,6))
plt.title('LSTM model for predicting capacity fade',size=20,color='black')
plt.xlabel('Cycle',size=15)
plt.ylabel('Capacity',size=15)
```

```
sns.lineplot(x=dn.index, y=dn.iloc[:,0],color='orange',label='Actual data')
sns.lineplot(x=trainfd.index, y=trainfd.iloc[:,1],color='red',label='Train data')
sns.lineplot(x=testfd.index, y=testfd.iloc[:,1],color='blue',label='Validation data')


plt.legend()
plt.grid(color = 'pink', linestyle = '--', linewidth = 0.5)
plt.show()
```