



**Politecnico
di Torino**

Politecnico di Torino

Mechatronics Engineering

A.a. 2021/2022

Sessione di Laurea April 2022

**Machine learning approach for online
estimation of Li-ion battery State of
Health**

Battery Monitoring Techniques

Relatori:

Angelo BONFITTO

Candidati:

Nourhan ABDELRAHMAN



**Politecnico
di Torino**

Thesis summary

As the world moves more towards sustainability and reducing greenhouse effects, the age of electric cars is approaching in a quick pace and, batteries, which are the electric heart of these cars, are gaining more and more attention. This includes not only developments related to batteries' material and structure but also the ones related to charging and monitoring battery health. The analysis and monitoring of cell degradation in batteries is crucial for assuring the safety of those electric vehicles. Alongside this, the expanding advances made in machine learning techniques and their wide range of applications has made it a very powerful tool that can be utilized in understanding the complex behavior of systems like batteries.

The main objective of this thesis is to develop and test a machine learning algorithm capable of monitoring and estimating the battery State of Health (SOH) in real-time, with the possibility of deploying this algorithm on a hardware, like in the case of the battery management system of an electric vehicle. This work is composed of two main stages. Firstly, a current profile is applied and features are selected. For the purpose of online and real-time estimation, a discharge train of current pulses is applied through the aging process of Li-ion battery cells. Since terminal voltage of the aged battery drops faster than that of a fresh cell, a combination of features is extracted from the terminal voltage response to the current pulse test. Correlation analysis is carried to decide on the best features that can be used in the estimation of battery SOH. Secondly, the selected set of features is used to train an artificial neural network so that it can accurately classify the health of the battery cell. In order to decide on the best level of battery state of charge at which the training and SOH estimation should be done, two current pulses at three different levels of SOC are utilized and then compared to better understand the effect of SOC level and choose the optimal and more consistent level of SOC. Further analysis is done on tuning of the neural network hyperparameters using metaheuristic techniques like the particle swarm optimization (PSO), in order to find the best combination of number of layers, dimension and gradient descent values. The trained neural network is deployed with the use of code generation from Simulink model on a TI development board to test the real time performance of the trained network on classifying and estimating the SOH level.

Analysis of the obtained results showed that all candidate features extracted were highly correlated with the SOH of the battery. Using only five of these features; two knee points, slope, voltage difference and calculated internal resistance, the trained network is capable of classifying the SOH level with very low error rates and showed consistency in the performance when tested on other battery cells.

In conclusion, applying the short-term current pulse approach which only takes up few seconds in addition to the help of machine learning algorithm, the developed technique is capable of providing a real-time estimation for battery SOH with great accuracy across all the testing cells.

Contents

Thesis summary	i
Contents	ii
List of Figures.....	iii
List of Tables	v
1. Introduction.....	1
1.1 Problem analysis	1
1.2 State of the art review	2
1.2.1 SOH estimation based on Impedance.....	2
1.2.2 SOH estimation based on Capacity	3
1.2.3 SOH estimation based on Voltage and Current	4
1.3 Methodology	6
1.4 Thesis statement	7
2. Fundamentals of Li-ion battery	9
2.1 Standard testing techniques	10
2.2 Ageing of Li-ion cells	14
2.2.1 C-rate's influence on cell degradation.....	14
2.2.2 Temperature's influence on cell degradation	15
2.2.3 Aging of SANYO NCR18650GA Battery (Random Walk).....	16
3. Fundamentals of Artificial Neural Networks	20
3.1 Feed-forward Neural Network	20
4. Features Extraction and Selection.....	25
4.1 Current Profile	25
4.2 Features Correlation and Selection	28
5. ANN Training and Hyperparameters Tuning.....	33
5.1 ANN classifier training on the 3 different levels of SOC	34
5.2 Sensitivity analysis	39
5.3 ANN Hyperparameters Tuning with Meta-heuristic Techniques	42
5.3.1 Discrete Particle Swarm Optimizer.....	44
6. Hardware Implementation.....	47
7. Conclusion	50
References.....	51

List of Figures

Figure 1 Schematic Nyquist plot of one LiB impedance measurement in [7].	3
Figure 2 Comparison between ML and other methods in [7]	3
Figure 3 Capacity estimation performance at different temperatures, from [14].	4
Figure 4 The four types of features in [19]	5
Figure 5 Regression model fitting effect (a)LS (b)RF in [19]	5
Figure 6 SOH estimation result with f1 feature in [22]	6
Figure 7 Flowchart of the overall methodology of the thesis	7
Figure 8 CCCV tests. Data from [26], [27].	10
Figure 9 CCCV tests. Data from [26], [27] (cont.)	11
Figure 10 CCCV tests. Data from [26], [27] (cont.)	11
Figure 11 CCCV test for cells at different temperatures (a)LCO (b)LFP. Data from [28], [29]	12
Figure 12 CCCV test for cells at different temperatures (a)NCA (b)NMC. Data from [30], [31]	12
Figure 13 CCCV tests for cells at different ageing stages. Data from [26], [27].	13
Figure 14 Maximum capacity degradation for different discharge C-rate. Data from [26], [27]	15
Figure 15 Maximum capacity degradation for different temperatures. Data from [26], [27]	16
Figure 16 Different temperatures for the 6 cells	17
Figure 17 V, I and T of first 50 RW steps observed on battery cell RW9 in [37]	18
Figure 18 V, I and T of last 50 RW steps observed on battery cell RW9 in [37]	19
Figure 19 The structure of the feed-forward neural network (FFNN).	21
Figure 20 Activation functions (a) Sigmoid (b) Tanh (c) ReLU (d) leaky ReLU	22
Figure 21 The overall framework for battery SOH estimation using ML algorithms	24
Figure 22 Train of discharging pulses of 10A	25
Figure 23 Voltage response under current pulse test through aging process in [22]	26
Figure 24 Current and voltage response of one pulse	26
Figure 25 The candidate features in the voltage response under	27
Figure 26 Discharge resistance after 2s and 8s	28
Figure 27 Voltage points correlation (a)Pearson (b)Spearman	29
Figure 28 ΔV Pearson and Spearman correlation	30
Figure 29 Slopes K1 and K3 correlation (a)Pearson (b)Spearman	31
Figure 30 Smoothing of ΔV	32

Figure 31 Features range and spread (a)before (b)after Z-score standardization.....	33
Figure 32 Flowchart of the training procedure	34
Figure 33 Flowchart of the testing process	35
Figure 34 Performance comparison between using 7, 6 and 5 features for SOC 60-65%.....	37
Figure 35 SOC level [30-35] % on test cell 1	Figure 36 SOC level [60-65] % on test cell 1... 38
Figure 37 SOC level [80-85] % on test cell 1	38
Figure 38 Confusion matrix of Cell0 with 10 output classes.....	40
Figure 39 Confusion matrix of Cell1 to Cell5 with 10 output classes	41
Figure 40 Flowchart of Particle Swarm Optimization	44
Figure 41 Convergence of PSO global best.....	45
Figure 42 Accuracy of Network trained with PSO on Cell0	46
Figure 43 LAUNCHXL-F28379D Board Overview	48
Figure 44 Overall Simulink model of the system	48
Figure 45 Neural Network structure on Simulink.....	49

List of Tables

Table 1 Characteristic comparison between Li-ion chemistries.	9
Table 2 Battery Specifications of Sanyo NCR18650GA.....	17
Table 3 FNN configuration parameters	34
Table 4 Results of testing using 7 features	36
Table 5 Results using 6 features	36
Table 6 Results using 5 features	37
Table 7 Final results of the best networks for the 3 level of SOC	39
Table 8 Results of ANN training on 10 output classes	39
Table 9 Accuracy of Network trained with PSO on Cell1 to Cell5	46



**Politecnico
di Torino**

Chapter 1

1. Introduction

With the increasing awareness of climate changes and greenhouse gases effects along with the inevitable fossil-fuel reserves depletion, tremendous changes have to take place to ensure saving the critical environmental situation our planet is facing. The transportation field is transitioning into more sustainable solutions, moving to hybridization and electrification of vehicles' powertrains instead of internal combustion engine vehicles (ICEV), in order to cope with some of the environmental problems. Electric vehicles (EV) have been undergoing rapid developments in the recent years. A key component of these electric vehicles is the battery, especially Lithium-ion batteries (LIB). Compared to other existing types of batteries like Lead-acid or Ni-H batteries, LIB is the most promising and attractive energy storage device due to its higher energy density, long cycle life, the high-power density and the low self-discharge rate[1]. The fast-paced growth in the market of electric vehicles (EV) has put a huge demand on ensuring the high performance and the long-life span of these batteries. As batteries are complex nonlinear systems that gets affected by several factors internal and external (temperature, aging, etc.), they propose lots of challenges and problems that need to be solved.

1.1 Problem analysis

One of the main problems faced when dealing with LIB is to ensure a reliable and safe operation of the Li-ion battery under different conditions. Therefore, the performance and condition of the battery has to be monitored all the time. The Lithium-ion battery pack is equipped with a Battery Health Management system as part of the Battery Management System (BMS). Its main task is to provide a safe operation of the battery and accurately estimate the states of the battery like State of Charge (SOC), State of Health (SOH) and available power capability (SOP)[2]. Battery State of Health (SOH) indicates the condition of the battery compared to its ideal state, it defines the battery level of degradation[3]. The process of estimating the battery State of Health can be very challenging due to the nature of the battery being a complex, nonlinear system. However, it is a necessity; as it helps managing the life of each cell and decide when it is crucial for aged cells to be replaced before causing any hazardous accidents.

A real time estimation of battery State of Health is vital for lots of automotive applications. The State of Health affects significantly the other states of the battery like SOC and SOP, therefore the focus is emphasized more on the online estimation of battery SOH which can be obtained through the use of advanced algorithms and the measurements of battery's voltage, current and temperature. This information is then transmitted to the controller which helps in deciding key factors for the energy management and power distribution in the vehicle[4]. The next section discusses in more details the recent research and development that have been done in this topic in recent years.

1.2 State of the art review

As the world is progressively adopting the use of electric vehicles, battery-related research and development has witnessed an exponential growth in the recent years. Different approaches have been adopted for the modeling and for the battery states estimation. There exist three main indicators that can be used for SOH estimation and they are; the battery impedance, capacity and its internal resistance. Where the internal resistance and impedance of the battery indicate its power capability, the capacity represents the amount of energy that can be stored by a battery[3]. In order to accurately estimate these indicators, there can be three methods considered which are: Experimental approach, Model-based approach and Machine Learning. Due to the amount of time and the high cost of the experimental-based methods, they are less appropriate for real-time applications. On the other hand, Machine learning is a combined approach of experimental and model-based methods which uses as an input for training the data collected from vehicle's BMS to accurately predict the degradation level of the battery[5]. This gives the Machine Learning method the ability to be used in real-time estimation of battery states.

1.2.1 SOH estimation based on Impedance

As mentioned before, the battery impedance can be a strong indication to the battery degradation level and it has been widely used to estimate battery SOH. Defined as the value of the ratio of voltage to current at the terminals, the impedance is frequency dependent and experiments had shown that it gets affected by the aging and temperature of the battery[6]. The most common approach to measure it is with a technique called Electrochemical Impedance Spectroscopy (EIS). Through injecting sinusoidal voltage or current signal and then measure the corresponding current or voltage, the impedance can be measured reflecting the physical characteristics of the battery. Yet, the large time constant of the diffusion and relaxation process makes the full spectrum of EIS not suitable for online estimation[7]. To counter this problem, several alternative approaches has been adopted to measure the impedance. A pseudo-random binary sequence (PRBS) is fed to the battery in the work of [8] then fast Fourier Transform is used to obtain the impedance spectrum. The research also addresses the difference between the performance of PRBS of different bit lengths. Using similar technique was utilized in [9] where FFT was replaced with the use of a continues wavelet transform. That Kept the information of the time and frequency together which leads to major improvement in the time taken to complete the measurement. Taking 97 min in comparison with the single-sine method that would take 54h.

A technique using a step pulse as incident signal with the use of fast Fourier transform (FFT) allows for extracting a full impedance spectrum in a shorter time[10]. Another method is to inject a single sine and a multi-sine signal (MSSIT) where characteristics are only analyzed at specified frequencies[11]. Utilizing Machine learning into this problem has also led to great results. The combination of Machine Learning and MSSIT technique is able to obtain the full impedance spectrum while maintaining a short measurement time. This is the case of [7], where various sinusoidal current signals with different frequencies ($f_1, f_2, f_3, \dots, f_n$) are injected into the battery at the same time. The variations in battery's voltage is then measured. The following

Figure 1, is the Nyquist plot of one measurement where the blue circles represent actual data points extracted from MSSIT and the dotted line represents the full spectrum. The interpretation for the intervals between two consecutive points is solved using ML. Obtaining the full spectrum is possible, however; more points need to be recorded which means longer measurement time.

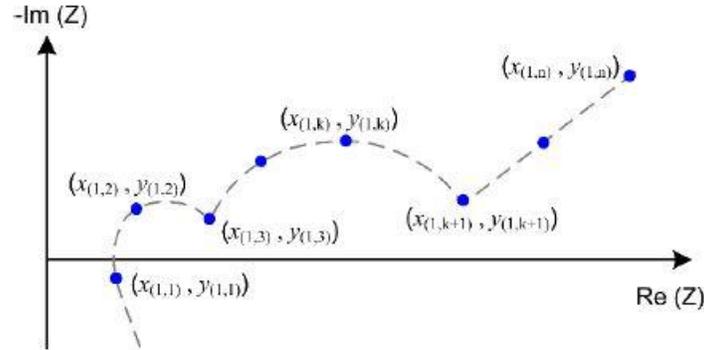


Figure 1 Schematic Nyquist plot of one LiB impedance measurement in [7].

As a result when ML technique is compared with other approaches in [7] like the piecewise cubic Hermite interpolation polynomial (Pchip), modified Akima piecewise cubic Hermite interpolation (Makima), and cubic spline interpolation (spline), it demonstrated the best accuracy and a stable performance across the different samples. As shown in Figure 2, using limited number of test points and ML has shown better performance also at low frequencies.

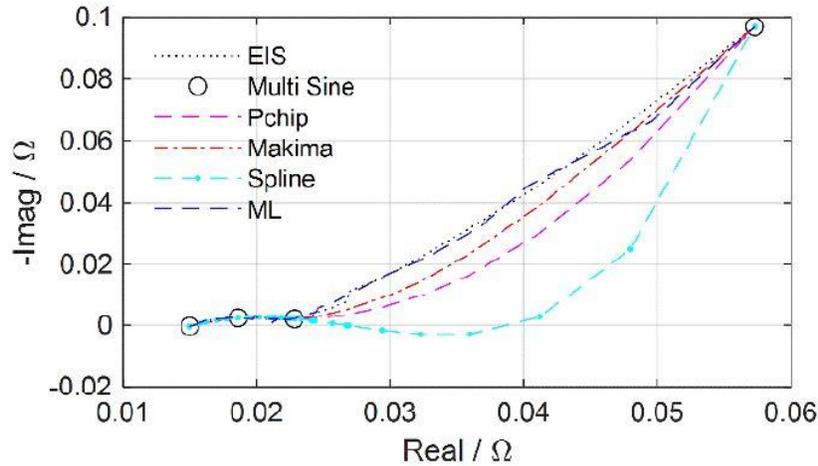


Figure 2 Comparison between ML and other methods in [7]

1.2.2 SOH estimation based on Capacity

Another powerful indicator of the battery degradation level is the capacity. The battery capacity declines with aging as the active materials in the cathode and anode gets reduced. Commonly, the relation between the open circuit voltage and SOC (OCV-SOC curve) is used to determine the degree of battery aging[12]. To obtain this curve, a process of fully charging and discharging of the battery at low rate needs to be done which is usually time consuming that makes it not suitable for real-time online estimation. As an alternative method, incremental

capacity (IC) and differential voltage (DV) curves analysis are introduced to estimate the capacity loss in a more time efficient way. Integrating the use of ML with these techniques has shown great improvement in results. In [13], ICA has proved to be a powerful tool along with Support Vector Machine algorithm (SVM) in predicting the capacity fading. In addition to reducing the computational load, the model that was built on a single cell has shown consistent good performance when tested on other cells with 1% error bound. A drawback of this method is that it still requires during the charging phase to gather a wide range of charging data, usually between 3 to 3.6 V. Using a linear function of the region capacity on the IC curve is proposed in [14] to present a fast and accurate estimation. The method results in RMSE of 2.5%. A better performance with Mean Errors ranging between 0.81% and 1.48% is shown in [15]. Between the battery capacity and the positions of features of interest (FOIs) on IC curves, a linear regression relationship is found. Yet, the curve has to be smoothed with Gaussian filter to be able to evaluate the characteristics of the IC curve. From measurements of current and voltage during charging phase, the capacity and then SOH is estimated using a long short-term memory RNN (LSTM-RNN) in [16]. The cells were aged at different temperatures using a driving cycle dataset. As shown in Figure 3, the estimations are accurately obtained over various operating situations.

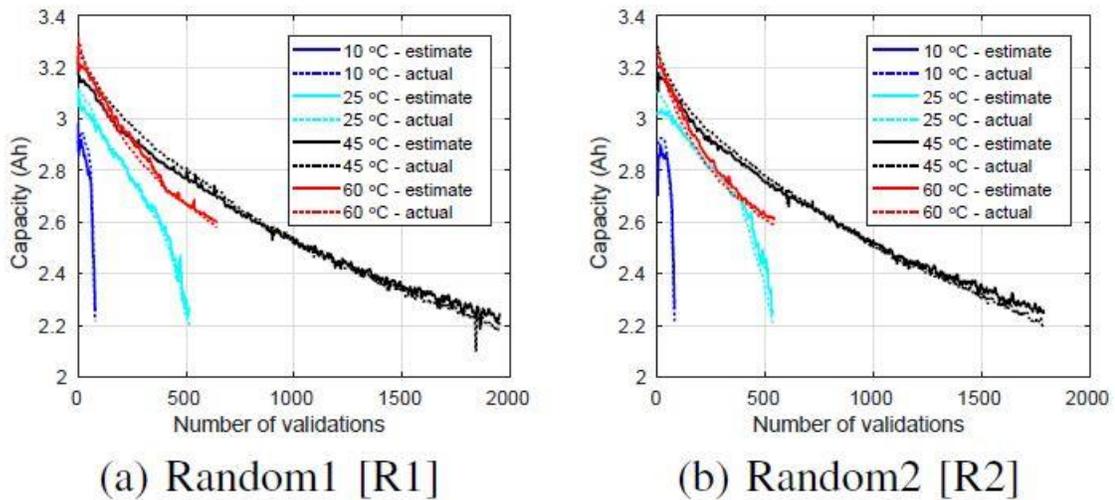


Figure 3 Capacity estimation performance at different temperatures, from [14]

1.2.3 SOH estimation based on Voltage and Current

In [17][18] using the NASA dataset and working on sample features extracted from the Constant Current, Constant Voltage curve (CC-CV), good performance of RMSE between (1.02% to 4.22%) and (0.78% to 3.45%), respectively, is obtained. However, this method requires utilizing the full cycle of going from 100% SOC down to 0% SOC and the process of features acquisition is lengthy. The work done in this paper [19], introduced a new set of features that can be extracted from the voltage and temperature curves. The four features that were used are illustrated in Figure 4 below, and they are; the time required for the voltage to reach its upper and lower cut-off limit during charging and discharging phase, respectively, and the time required for the temperature to reach its maximum value while charging and discharging.

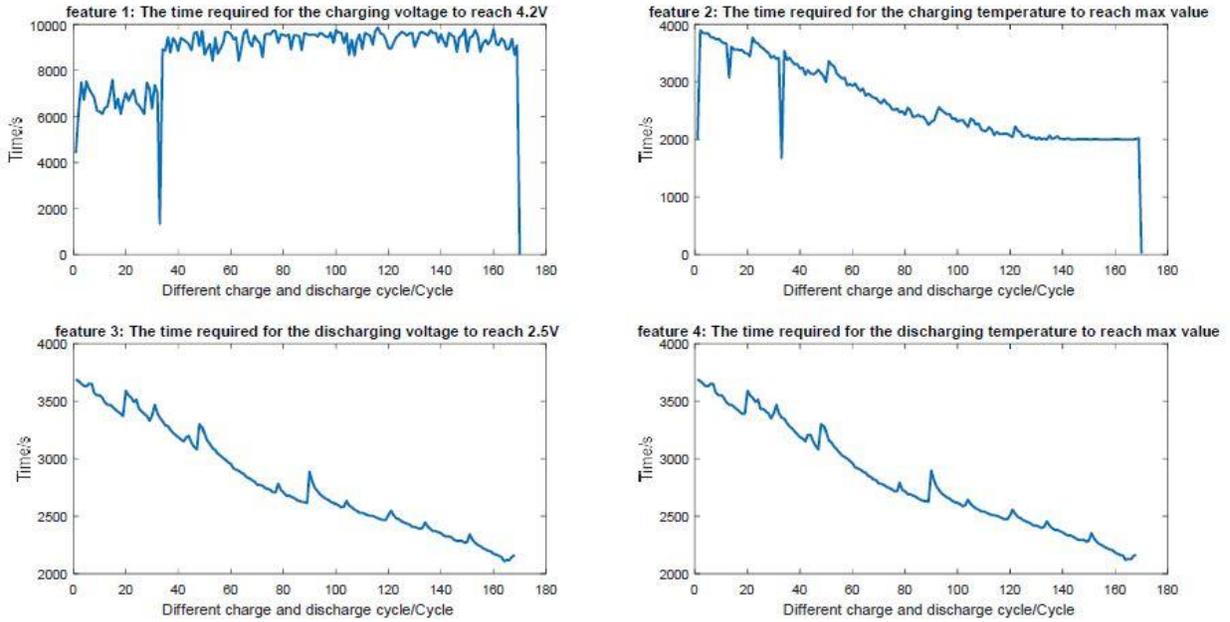


Figure 4 The four types of features in [19]

With the use of least square regression tree and random forest regression model, the selected features are able to estimate the level of SOH with mean square error (MSE) equals to 0.0006 and 0.0002, respectively. The following Figure 5 shows the fitting effect of the LS and RF regression model.

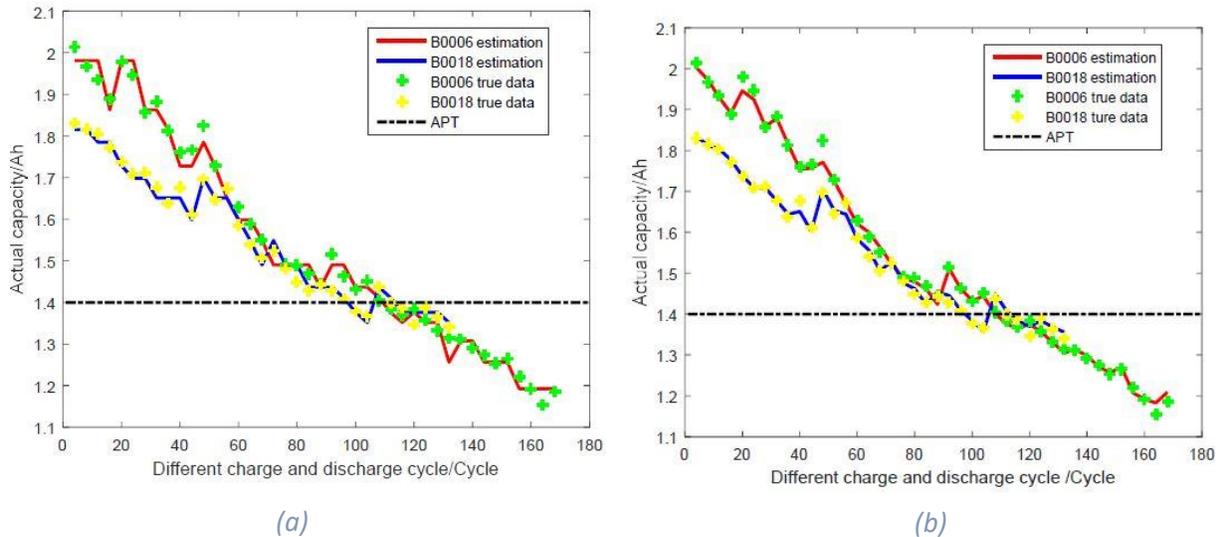


Figure 5 Regression model fitting effect (a)LS (b)RF in [19]

In [20], a novel method is proposed by using the Hybrid Pulse Power Characterization (HPPC) test technique. The sample features are constructed from the voltage data. Using Support Vector Regression (SVR) along with Genetic algorithm (GA), the model achieved RMSE between 1.91% and 1.31%. However, the work only took into consideration the battery calendar

aging and neglected the cyclic battery aging leading to the fact that this method cannot meet the real-time, online measurement requirements[21].

A new approach is presented in [22] where a short current pulse is used with Support Vector Machine (SVM). The extracted features included the knee point of the voltage response and the slopes between them. The LiFePO_4 battery is aged for 37 weeks and the obtained results found that the combination of all the knee points and slope between points B and C (V_A , V_B , V_C , V_D , KI) gave the best performance with MSE equals to 0.0039 Ah. The result of SOH estimation with these features ($f1$) combination is showed in the following Figure 6,

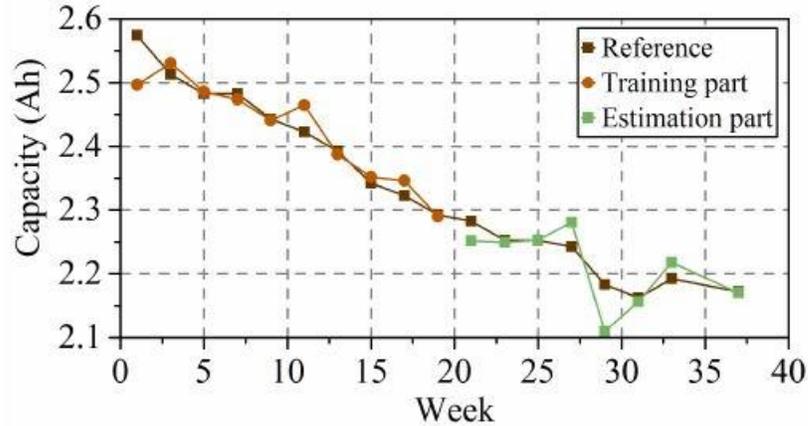


Figure 6 SOH estimation result with $f1$ feature in [22]

This approach is close to the method adopted in this thesis, however; in the work of [22], no analysis is included about the SOC level suitable for the current pulse application and the features extracted depended only on some voltage measurements. An overview of the use of ML techniques on battery states estimation is also presented in [23].

1.3 Methodology

As shown in the state-of-the-art review, there is still a place for improvement in terms of providing a more real-time and accurate estimation of the State of Health. The aforementioned work has also implied a great potential for the use of machine learning with the Battery Management System (BMS). Therefore, this thesis focuses on tackling some of these issues by working with a short-term current pulse profile and designing a machine learning algorithm that can quickly and accurately provide an online estimation of SOH. Then, test the designed algorithm on different cells to ensure a robust and accurate performance. The work focuses on obtaining a small set of features that can be used to train the Neural Network classifier for accurate and real-time estimation of the SOH. Further analysis is performed to optimize the tuning of the Neural Network hyperparameters with the use of genetic algorithm such as Particle Swarm Optimization (PSO). The distinguishing feature of the work presented in this thesis is the selected few features used in SOH estimation which ensures a real time performance, along with the use of AI which eliminates the use of complex models that include lots of parameters which can be affected by several factors decreasing the robustness and flexibility of the model. To

summarize, the following flowchart represents the steps followed throughout this thesis to design the Neural Network classifier for SOH estimation.

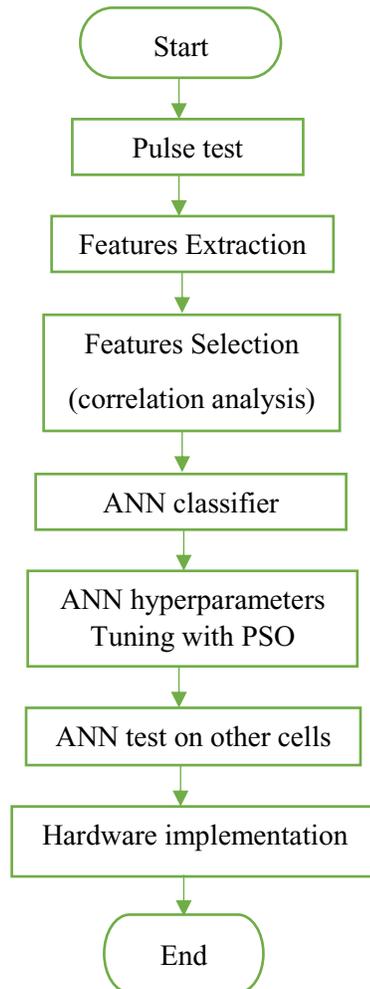


Figure 7 Flowchart of the overall methodology of the thesis

1.4 Thesis statement

Considering the information presented in the previous sections, the problem statement for this thesis is formulated as follows:

How can artificial intelligence approaches be applied to improve the online estimation of a battery's state of health in electric vehicles?

The rest of this thesis is organized as follows. An overview of the battery as a system and the process of battery aging is introduced in chapter 2. Chapter 3, presents the fundamental concepts of Artificial Neural networks and addresses in specific the feed-forward NN that is used for the further analysis performed in this thesis. Then the choice of the current profile to ensure a

real-time estimation of SOH is introduced in chapter 4 along with the process of features extraction and selection based on the correlation analysis. In chapter 5, the configuration and training of the ANN classifier is presented, in addition to, the tuning of the Neural Network hyperparameters is analyzed with the help of Particle Swarm Optimization technique, the working principle of the genetic algorithm and the obtained results are discussed. Finally, chapter 6, discusses the hardware implementation of the designed algorithm and gives the conclusion on the performance of the proposed designed model.

Chapter 2

2. Fundamentals of Li-ion battery

The short- and long-term performance of Lithium-ion batteries is influenced by various conditions. And because they are very complicated devices, understanding the effect of these elements and the distinctions between them is critical before attempting to construct any form of model or algorithm. For that aim, an overview of the primary testing techniques is analyzed along with the impact of several conditions on the aging of the battery. Finally, the mechanisms of the aging method used in this thesis is thoroughly investigated.

Li-ion batteries store energy in chemical form and subsequently convert it to electricity via a reduction-oxidation (redox) reaction. The anode is always used for oxidation, whereas the cathode is used for reduction. Electrons moves from the anode, the negative electrode (NE), during discharge and travel via the applied external load to the cathode, the positive electrode (PE), where they contribute to the production of the metal salt. Simultaneously, the electrolyte acts as a channel where anions and cations travel through to reach the NE/PE in order to donate and absorb electrons. During charging phase, electrons flow from the positive electrode (cathode) to the negative electrode (cathode) while cations flow via the electrolyte from PE to NE for reduction, and for oxidation anions drift from NE to PE. Negative electrodes are typically made of a mixture of graphite and solid lithium (LiC_6), in addition to that, lithium-titanium-oxide salts ($Li_4Ti_5O_{12}$, LTO) have also been employed. The most common positive electrode materials are lithium-cobalt-oxide ($LiCoO_2$, LCO), lithium-iron-phosphate ($LiFePO_4$, LFP), and lithium-manganese-oxide ($LiMn_2O_4$, LMO), as well as ternary salts such lithium-nickel-cobalt-aluminum-oxide ($LiNi_{1-x-y}Co_xAl_yO_2$, NCA) and lithium-nickel-manganese-cobalt-oxide ($LiNi_{1-x-y}Mn_xCo_yO_2$, NMC). The battery's chemistry type is mostly determined by the needs of a particular application since every type of them has unique properties.[22, p.250], [23, p.9], [26],[27]. Table 1 shows the key features of the different chemistries' batteries can have.

Table 1 Characteristic comparison between Li-ion chemistries.

Chemistry	Specific capacity	Capacity density	Voltage	Advantage	Disadvantage
LCO	150 A h kg^{-1}	550 x 10^3 A h m^{-3}	3.8 V	Low self-discharge	Lifespan
LFP	165 A h kg^{-1}	589 x 10^3 A h m^{-3}	3.4 V	Lifespan	Specific energy
LMO	120 A h kg^{-1}	596 x 10^3 A h m^{-3}	4.1 V	Specific power	Thermal stability
NCA	190 A h kg^{-1}	700 x 10^3 A h m^{-3}	3.7 V	Energy and power density	Cost
NMC	170 A h kg^{-1}	600 x 10^3 A h m^{-3}	3.7 V	Specific energy	Cost

2.1 Standard testing techniques

To observe and analyze the behavior of the battery and in addition collect important data that helps identifying the parameters of various models, different experimental tests have turned into standard tests in the industry. These kind of standardized tests are called as reference performance tests (RPT)[22, p.263],[23, p.33]. The following part of this thesis discusses those test and uses the data that can be found in the Battery Archive[28],[29], Sandia National Laboratory[30][31][32][33], Oxford University[34],[35], and McMaster University[36]. The purpose of these tests is to use excitation signals that are time-varying to analyze the device's short and long-term response. Depending on the case being studied, these tests can last anywhere from a few seconds to several days.

Constant-current, constant-voltage (CCCV)

The CCCV approach is the most popular charging method, it is used to determine the cell's maximum possible capacity at a given time and under specific conditions. It usually starts with a completely emptied battery and progresses in two stages. The first phase is the charging phase, it involves injecting a constant C-rate (e.g. 0.5C or 1C) into the battery until reaching the upper cut-off voltage. The second step involves charging the battery at constant voltage while the current gradually declines. When the current has fallen to roughly 0.05 C, the test is usually deemed complete. There has to be a resting interval between the charging and discharging phases which after a CC method with a negative current is applied for discharging till the lower cut-off voltage is attained, following this a CV phase takes place. SoC = 100 percent and SoC = 0 percent are the charge levels at the end of the charging and discharge processes, respectively. The test can be repeated for more reliable findings. After repeating for two or three times, the average is used to determine the final value. Figures [8-10] below shows the CCCV terminal voltage and current for five different batteries with different chemistries. The last CV discharge curve is not included as it wasn't available in the dataset.

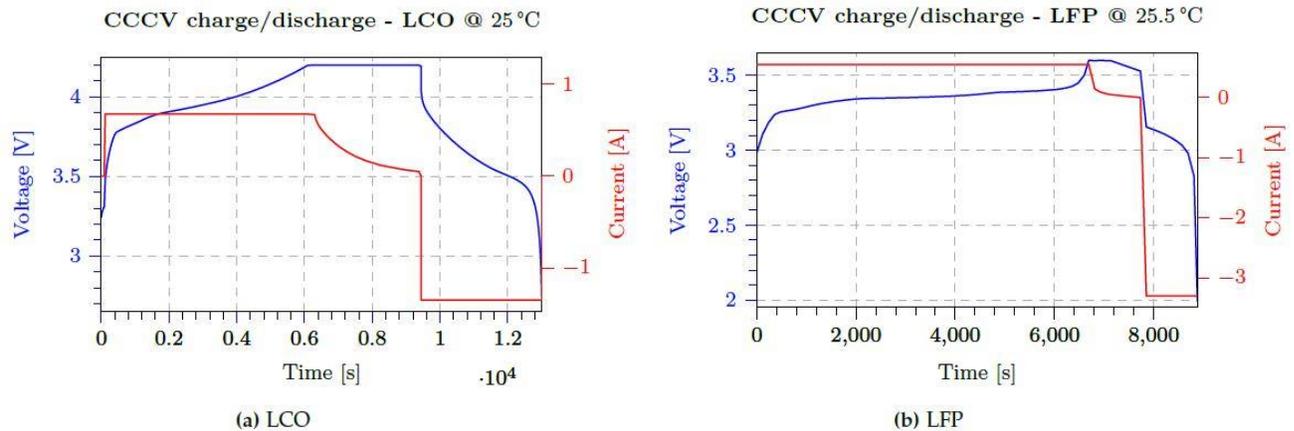


Figure 8 CCCV tests. Data from [26], [27]

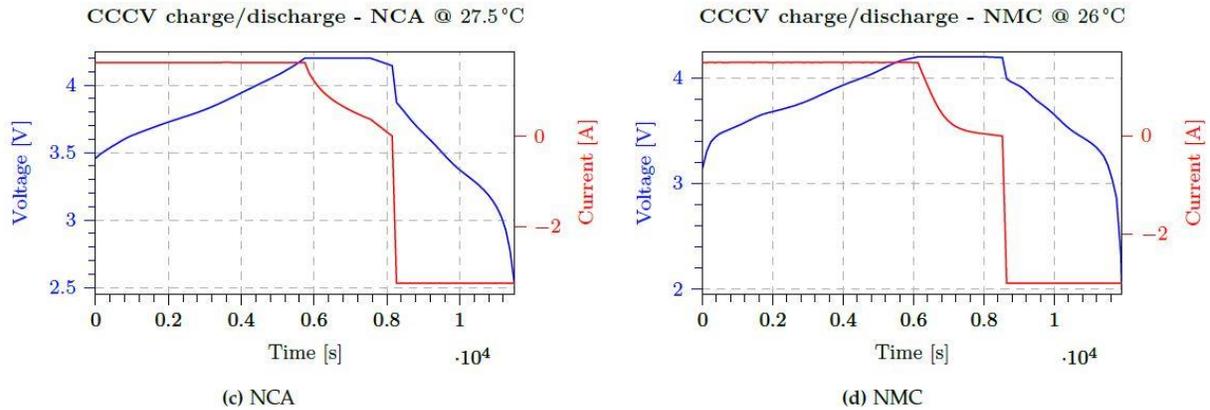


Figure 9 CCCV tests. Data from [26], [27] (cont.)

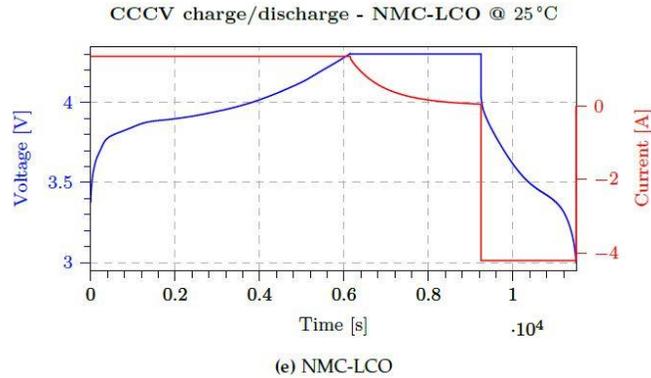
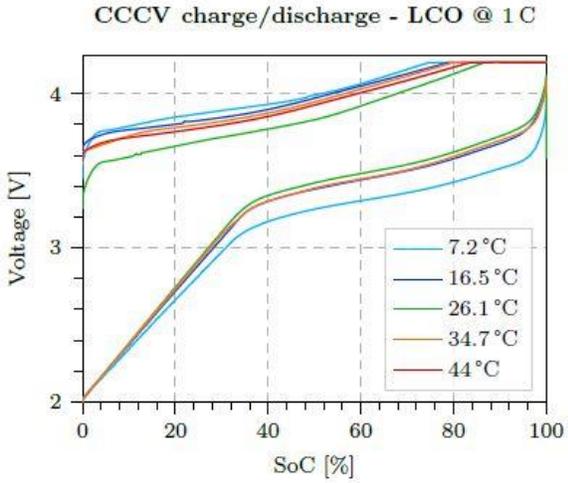


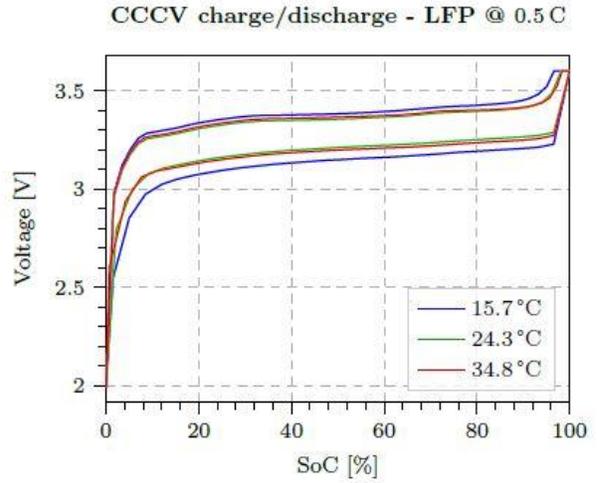
Figure 10 CCCV tests. Data from [26], [27] (cont.)

The behavior of the batteries is heavily dependent on the materials used in a battery, as shown in these graphs. When comparing the voltages of the LFP and LCO cells during charging, it is clear that the LFP has a fairly flat shape with a small voltage interval, whereas the LCO has a steeper contour and a larger voltage interval. Another notable distinction is that the CV stage of the LFP cell is substantially smaller in comparison to the rest.

To investigate the effect of the temperature on the behavior of the voltage response, numerous cells need to be cycled to better understand the significance of it. As shown in Figures 11 and 12, The upper-half curves in each diagram correspond to the charge process (growing SoC) and the lower-half curves to the discharge process (decreasing SoC). Curves during charge displace higher and the opposite while discharging as the temperature drifts away from 25 °C. This displacement is almost non-existent in the case of LFP, but it is significant in the case of NCA and NMC.

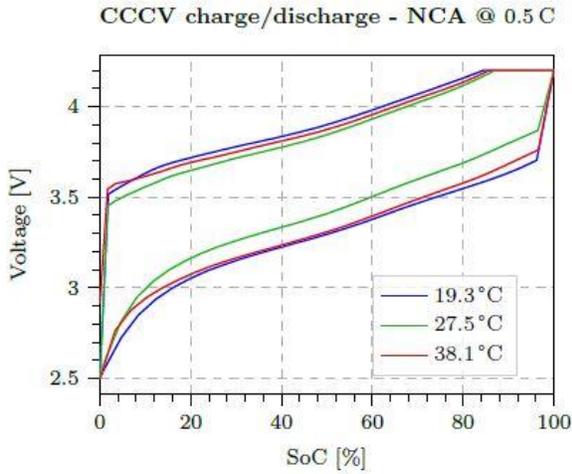


(a)

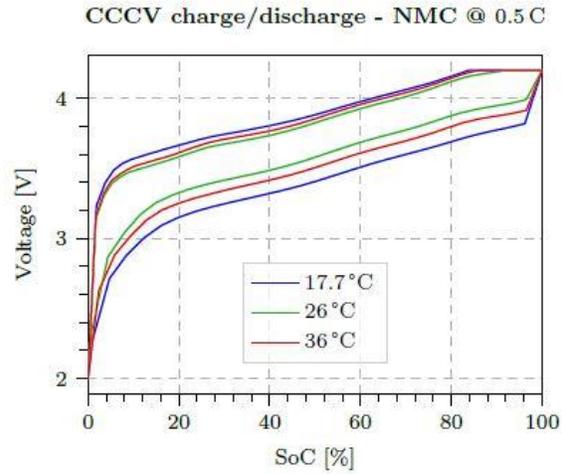


(b)

Figure 11 CCCV test for cells at different temperatures (a)LCO (b)LFP. Data from [28], [29]



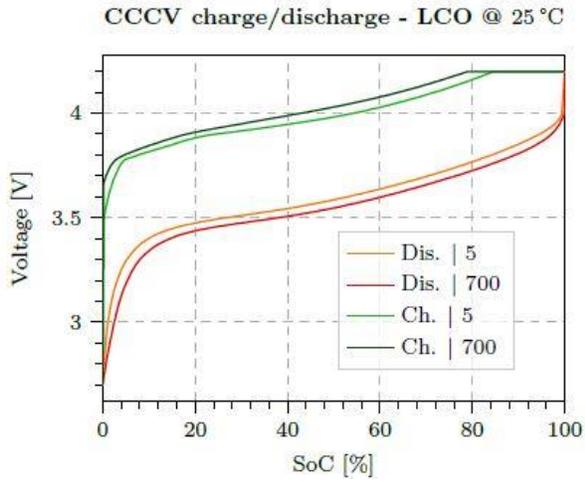
(a)



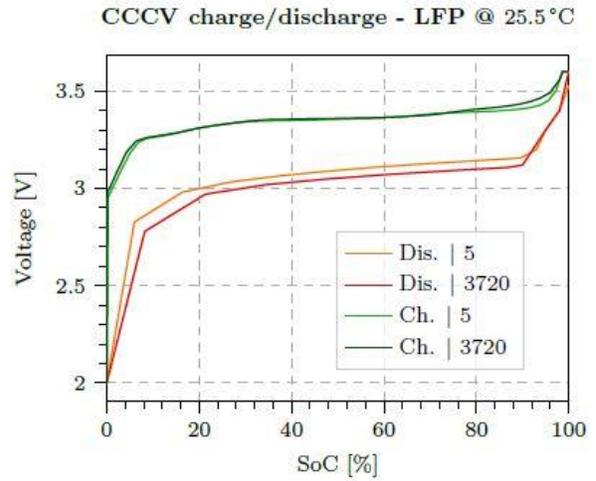
(b)

Figure 12 CCCV test for cells at different temperatures (a)NCA (b)NMC. Data from [30], [31]

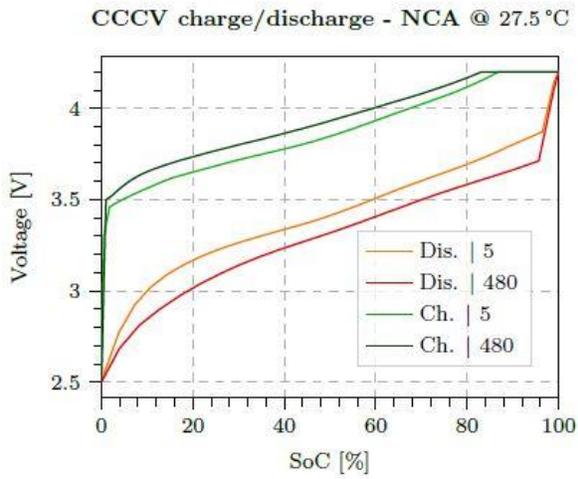
Finally, Figure 13, depicts how the CCCV voltage curves vary as the battery degrades after the aging process with the different cycles. The maximum usable capacity goes down roughly 20% compared to the rated value, which means that all these figures are related to the battery's end of life (EoL).



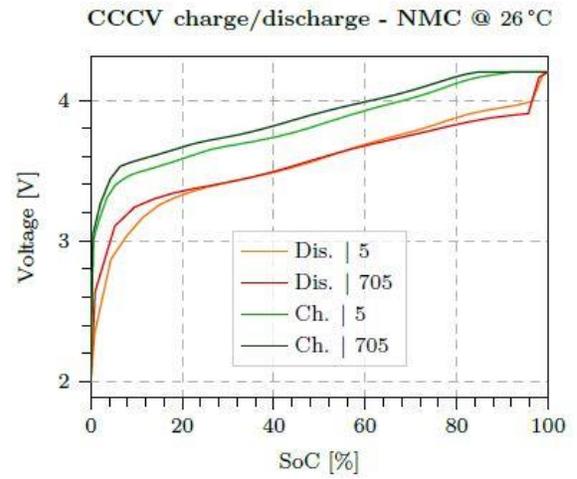
(b) LCO



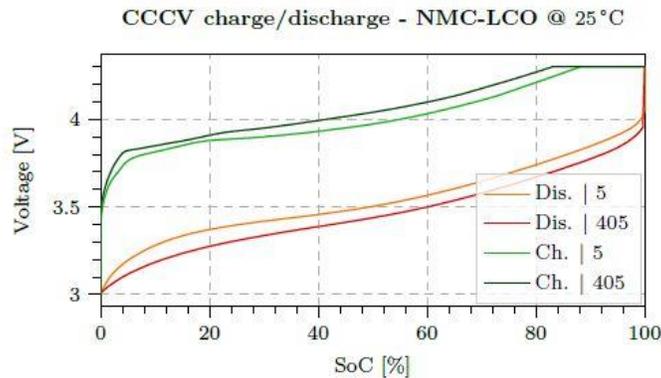
(a) LFP



(d) NCA



(c) NMC



(e) NMC-LCO

Figure 13 CCCV tests for cells at different ageing stages. Data from [26], [27]

The relationship between the terminal voltage and the degradation level is obvious. That for the same level of SoC and during charging, it increases with the deterioration of the degradation level, and decline when it is discharging. As a result, the cut-off voltages are reached earlier, resulting in a decreased maximum capacity. The effects of degradation are influenced by cell chemistry once again. Even when cells have been heavily cycled, LFP curves remain close to each other, whereas NCA batteries have larger gaps. In CCCV experiments, the charged (or discharged) capacity is often determined by integrating the current measurement (coulomb counting), however; there are several other approaches that can be used as well. A difference between the value of the charging and discharging capacity will show that the battery isn't efficient, then, it is possible to calculate an efficiency factor between charge and discharge case.

2.2 Ageing of Li-ion cells

Batteries as an electrochemical system deteriorate as they are being used. The alternations happening to its internal structure and the damage it builds up with time is because of a phenomena called “solid-electrolyte inter phase (SEI) film formation” which means the active material is being depleted [37]. There exist several factors that can impact the decline of battery's SOH, some of them include the C-rate and temperature. Even if the same aging process is applied to two different cells for example, it is still expected to have slightly different response due to manufacturing reasons and not just the aging process itself. However, some common behavior can be generalized is that there exist usually two phases in the degradation process. First, the rapid decline in capacity level during few cycles this can be associated to the SEI film formation. The second is that the decrease takes a linear form with a smaller slope after the film is entirely formed[23, p.55].

2.2.1 C-rate's influence on cell degradation

It is known that there exists a proportional relation between the torque required and the corresponding current usage. An electric vehicle would need more power when going up a hill or accelerating and hence withdrawing more current from the EV battery[38]. As for the current fed into the battery, it depends on the properties of the charging station since this is the source of current for the battery. Analyzing the effect of C-rate on three different batteries with three different chemistries, in Figure 14, it is evident that the C-rate has a crucial impact on the deterioration of the capacity available. The type of battery is still playing a role also, this is clear in the durability of the LFP cell; maintaining a longer lifespan in comparison to other types.

For both types LFP and NCA, the degradation in the value of capacity is minimum with smaller rates such as 0.5C and 1C. for LFP, at 2C, it might seem that the cell is performing worse however it is only due to the initial value of the capacity being lower than other rates. The only noticeable change happens at the highest rate of 3C, the capacity declines quickly. As for NMC type of battery, the first 100 cycles seem identical for all rates. Only after that, a slight difference can be seen, however; it can be safe to say that the NMC didn't get much affected by the value of the C-rate.

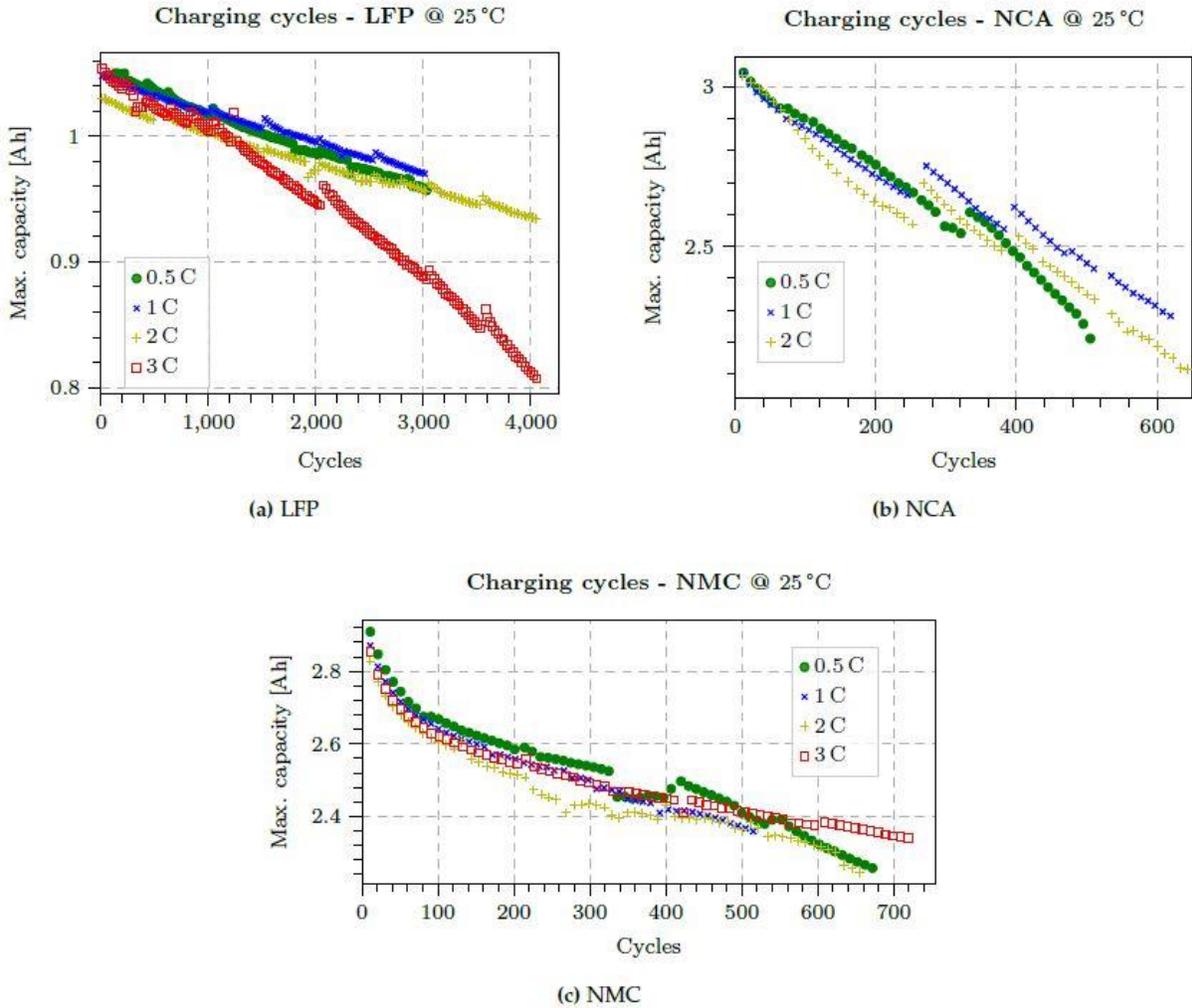


Figure 14 Maximum capacity degradation for different discharge C-rate. Data from [26], [27]

2.2.2 Temperature's influence on cell degradation

The ambient temperature can be also a major factor impacting the ageing process of the battery. The internal chemical reactions inside the battery is expected to be influenced by the internal and external temperatures. The following Figure 15, analyze the relation between the capacity degradation and the temperature. By maintaining a constant temperature during the charging and discharging process, three different temperatures illustrates the influence on the maximum capacity available.

One disadvantage of this dataset is that the range of the temperatures used for the analysis is very narrow, perhaps a wider more inclusive range would help to really understand the pattern of the temperature influence on the battery's degradation. However, from the Figure, it can be noticed that for LFP cell, the capacity declines the quickest when the temperature rises to 34.8 °C

even though the initial capacity is higher at that temperature. while the behavior at 15.7 °C and 24.3 °C has similar trend. The NCA cell shows a similar pattern, the level of degradation increases with the increase in temperature. The last type NMC shows a totally different trend, where the degradation is worst at lower temperatures and similar performance is maintained for 26 °C and 36 °C

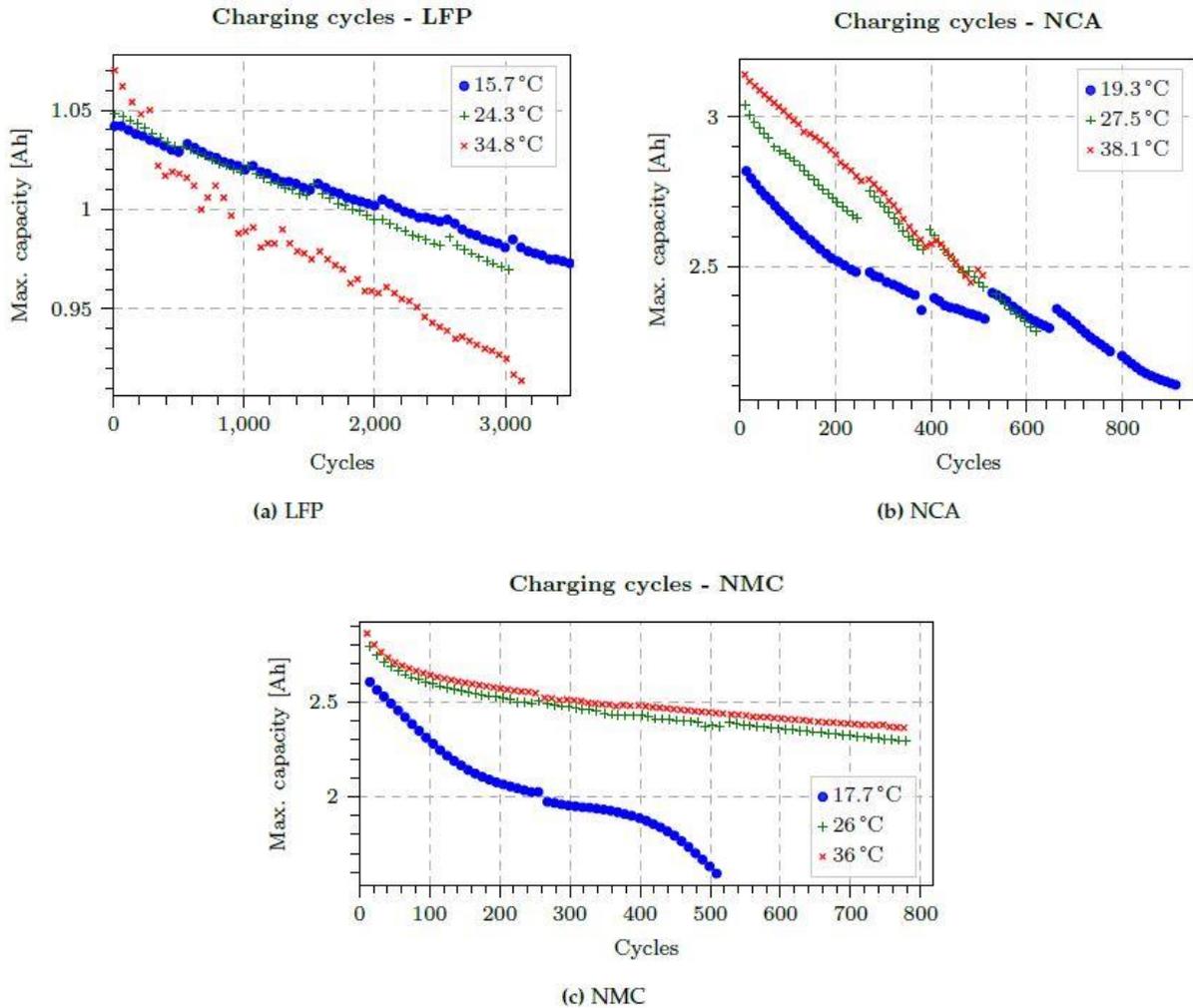


Figure 15 Maximum capacity degradation for different temperatures. Data from [26], [27]

2.2.3 Aging of SANYO NCR18650GA Battery (Random Walk)

The available dataset used throughout this thesis belongs to 6 different cells. The same aging process is performed for all of them, however, some changes are present for example in the value of the ambient temperature for each cell. Figure 16, shows the different ambient temperature for the 6 cells. Cell0 will be used in training, the temperature is maintained around

20 °C. while the other 5 cells are used for the testing after the design of the ANN classifier which will be discussed in more details in later chapters.

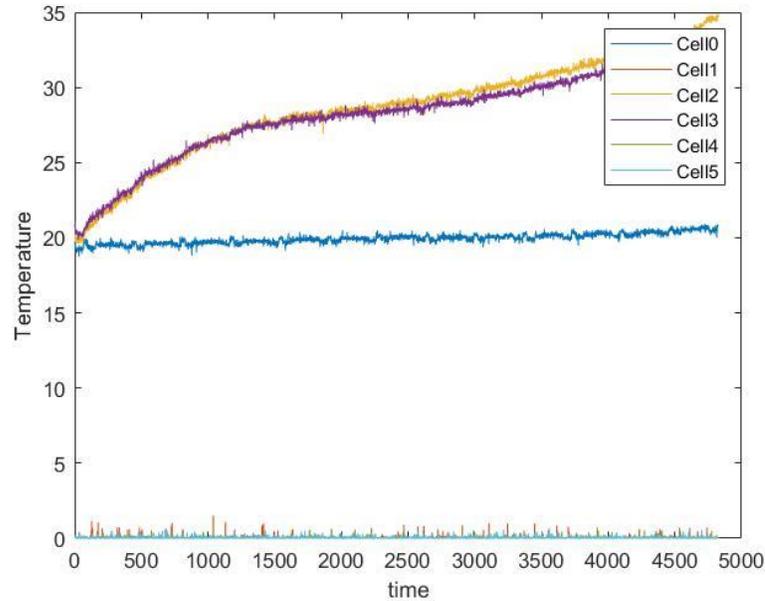


Figure 16 Different temperatures for the 6 cells

As shown cell 2 and 3 have an increasing temperature that goes up from 20 °C until below 35 °C. the other 3 cells (1,4 and 5) are kept at low temperatures close to zero. While cell0 have an almost constant temperature of 20 °C. The key characteristic of the battery cells used during this thesis is included in the following table 2,

Table 2 Battery Specifications of Sanyo NCR18650GA

Battery Properties	Specifications
Manufacturer	Sanyo
Battery Chemistry	Li-ion
Nominal capacity	3000 mAh
Voltage	3.6 V
Discharge current	10 A

In attempt to capture the real-life operation conditions for the battery in electric vehicle, the battery cells are aged with a sort of dynamic load that includes different charging and discharging current profiles that are randomly selected for the cycles. Mainly all the six battery cells undergo the same aging process and it consist of two stages or profiles. The first is the Random Walk (RW) method and in addition to it a reference charge and discharge profile.

The Random Walk approach consists of a random sequence of current steps that ranges in values between -4.5 A and 4.5 A. The current profile is fed to the battery cells for every five minutes. The charging currents are represented by negative values while the positive represents a discharging profile. A new, random value for the current profile is chosen every five minutes

period. For safety reasons the voltage is limited to its maximum and minimum threshold. During the process of applying the current step, if the voltage passed the limited range, the operation stops and a new random value for the current step is applied to the battery cell. A delay of 1 s is allowed after each step to be able to decide on the new value of the current step. One cycle of the RW profile include around 1500 RW steps and same number for rest periods between the steps. Between each RW profile, reference charging and discharging profile is implemented to analyze the batteries capacity.

The reference profile is applied prior to every RW cycle and it consist of a sequence of charging and discharging profiles. A constant current is applied while the voltage increases until it reaches the maximum limit, then, the current is allowed to drop under the application of a constant voltage. Similarly, a constant load is fed to the battery up till the voltage decline to its minimum cut-off value. This reference profile is used to measure the capacity by integrating the discharge current using the following equation,

$$Q_{present}(t) = \int_0^t I_d(t) dt$$

During the implementation of RW profile, the BMS's measurements of V, I, T and time is recorded, the following Figures 17 and 18, shows the first 50 RW steps and then the last 50 steps used in [39],

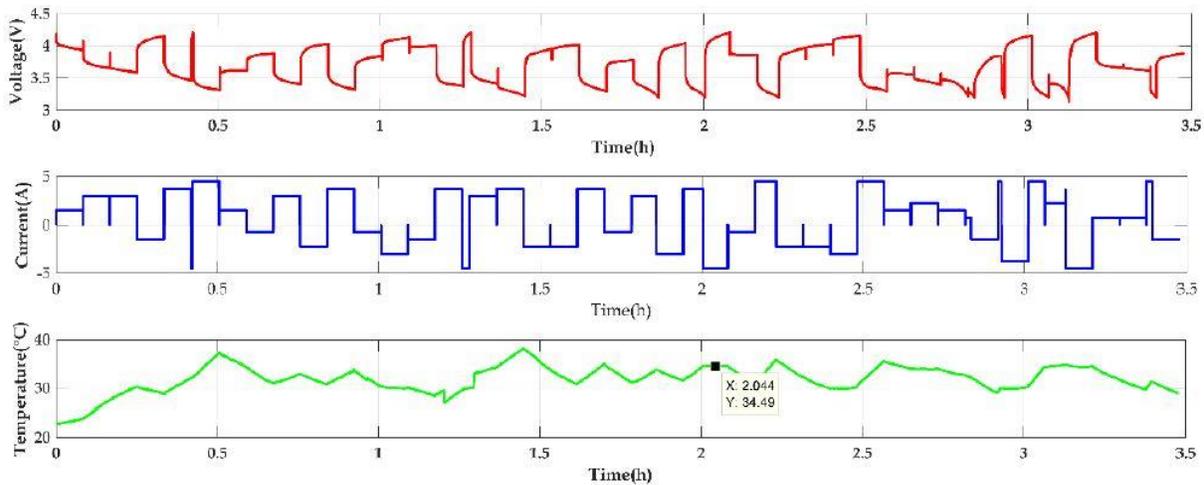


Figure 17 V, I and T of first 50 RW steps observed on battery cell RW9 in [37]

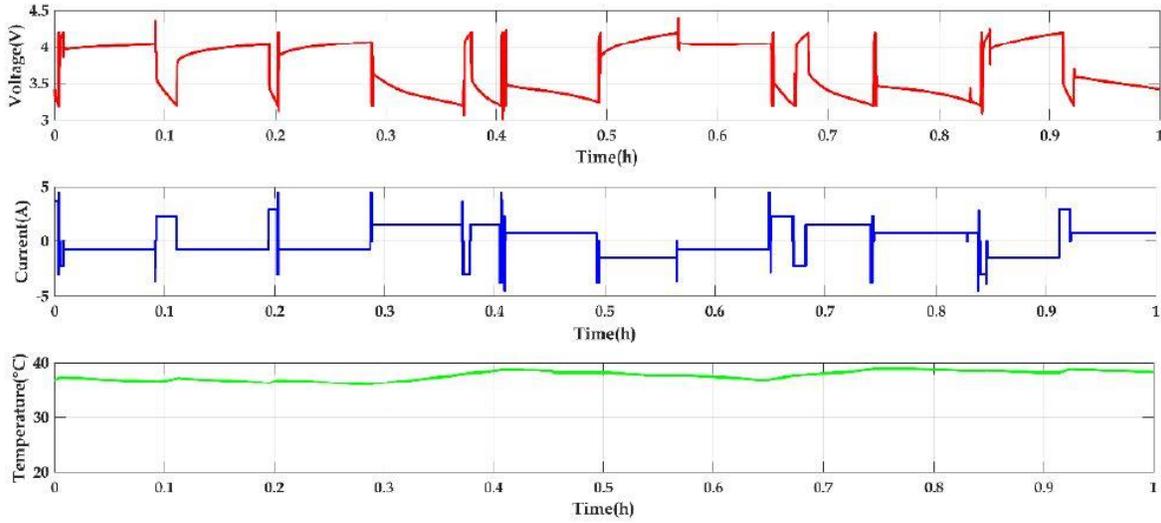


Figure 18 V , I and T of last 50 RW steps observed on battery cell RW9 in [37]

As it can be noticed from Figure 18, the time taken to finish the last 50 RW steps is shorter in comparison to the first 50 RW steps and that is because the battery is reaching the limits of the voltage threshold more frequently, which causes the algorithm to move on to a new step and a new value of the current. This is a clear indication to the degradation of the battery State of Health.

The six cells are all undergoing the same ageing process, they are cycled for 55 cycles, with also a current pulse applied for the further analysis of the real time estimation of the SOH which will be discussed thoroughly in the next two chapters.

Chapter 3

3. Fundamentals of Artificial Neural Networks

In this chapter a detailed theoretical background of the working principle of Neural Networks is presented. With a focus on the structure implemented in this thesis, which is the feed-forward neural network.

Artificial intelligence enables computers and machines to mimic the decision-making and problem-solving capabilities of human beings. AI is a branch of science concerned with the creation of computational systems that can comprehend a situation, and adjust their behavior to learn rather than just perform a single, predetermined task. In the last two decades, there has been an exponential growth in AI usages and applications. AI includes within itself several fields like fuzzy logic[40], machine learning[41] and deep learning[42]. The general concept of the three is similar, however; each technique has its own set of properties that make it more suitable for specific applications including autonomous driving, control and system modeling, computer vision, and natural-language processing[23, p. 46],[22, p. 443]. The unique aspect of these algorithm is their ability to try and learn from the data and previous experiences in order to, in the future, apply this knowledge into new scenarios.

This thesis utilizes in specific machine learning technique and the use of artificial neural network[44]. ANN was created to mathematically simulate the activity of the human brain. The general structure of a neural network is an input layer, multiple hidden layers and an output layer[45]. The data is received by the input layer, which then passes it on to the hidden layer. Each hidden layer consists of several neurons. Through the activation function, these neurons perform a weighted linear combination computation and propagates the information to the next hidden layer. Different hidden layers may have different number of neurons and also these neurons can have different activation functions. This process goes on between the hidden layers until the output layer is reached, which predicts the model's target.

3.1 Feed-forward Neural Network

A typical structure of a feed forward neural network is shown in Figure 19. Because information flows through the function being evaluated from \mathbf{x} , through the intermediate computations in the hidden layers, and finally to the output y , these models are called feedforward. There are no feedback or loop connections in which output can be feedback into the model. If such feedback existed these networks would have been called recurrent neural network.

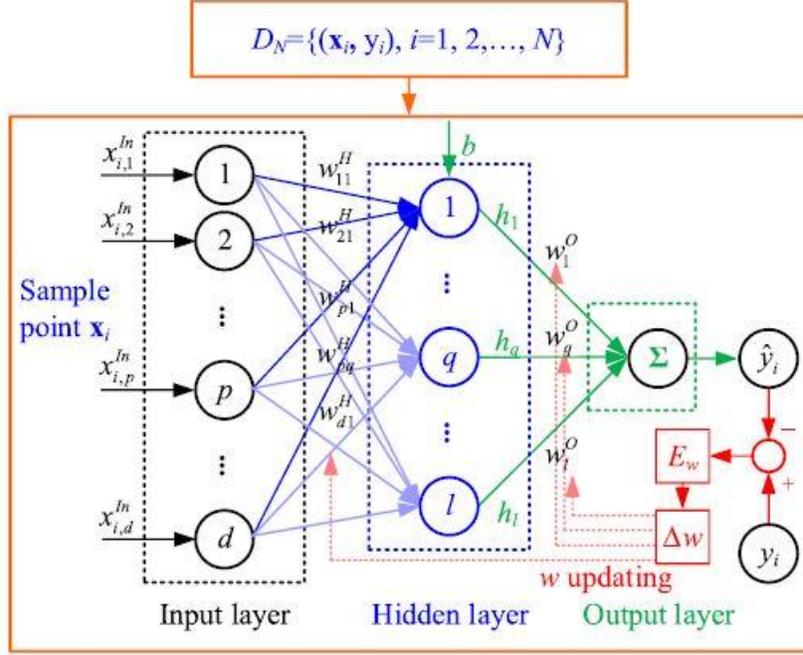


Figure 19 The structure of the feed-forward neural network (FFNN).

The input to the network is a d-dimensional features. While the hidden layer's input is the sum of the inner-product and the bias according to eq.1 below.

$$h_q = g \left(\sum_{p=1}^d x_{i,p}^{In} w_{pq}^H + b_q \right) \quad q = 1, \dots, l$$

Where $x_{i,p}^{In}$ is the pth feature of the ith sample data in the input layer, and w_{pq}^H is the weight connecting pth input neuron and the qth hidden neuron, b_q is the bias of the qth hidden neuron, in the hidden layer l is the number of the neurons. Lastly, $g(\cdot)$ is the activation function. This shows that intrinsically the neural network is performing a simple linear algebra operation which makes them very efficient for hardware implementation. Another attractive aspect in ANN is that many calculations are performed independently at each layer which allows for parallel execution that speeds up the whole process. Therefore, usually these networks are deployed on concurrent-computing hardware like graphics processing units (GPU) and field-programmable gate arrays (FPGA)[23, p.156].

One of the important steps in the design of neural network is the choice of the activation function. This decision will determine the complexity of the operations that can be achieved by the algorithm. A simple example is a system without an activation function or with threshold-like one, it will only be able to produce linear discrimination functions at the output, leaving it unable to solve more complex, nonlinear operations. While the use of non-linear activation functions gives the network the ability to perform more complex tasks. However, not only the activation

function determines the level of complexity of operations that the network can perform but also the number of the hidden layers and the number of neurons present in each layer, this is defined as the network's capacity.

Several options of activation function exist, the most popular are the linear, sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU), and the leaky ReLU. A graphical representation of the functions can be seen in Figure 20.

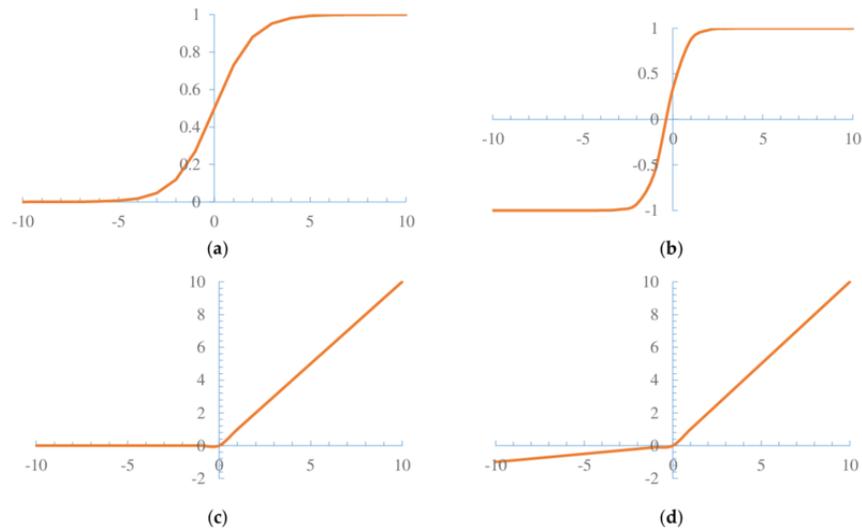


Figure 20 Activation functions (a) Sigmoid (b) Tanh (c) ReLU (d) leaky ReLU

Their mathematical expressions are as follows,

$$\text{sigmoid } g(u) = \frac{1}{1+e^{-u}}$$

$$\text{Tanh } g(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

$$\text{ReLU } g(u) = \begin{cases} 0, & \text{for } u \leq 0 \\ u, & \text{otherwise} \end{cases}$$

$$\text{Leaky ReLU } g(u) = \begin{cases} a \cdot u, & \text{for } u \leq 0 \text{ (} 0 < a < 1 \text{)} \\ u, & \text{otherwise} \end{cases}$$

A crucial criterion in the choice of activation function is their derivative to be simple and easy to compute. The value of these derivatives should not be too large or too close to 0 so that it would avoid the “vanishing gradient” problems[23, p. 129]. The sigmoid function is one of the most popular and widely used activation functions. Its derivative is always non-zero which makes the gradient descent effective at every step during the training phase. The Tanh function has similar behavior when it comes to the gradient descent also its zero-centered output lead to increasing the convergence speed. Compared with other functions the ReLU has higher computational efficiency. It is widely used in deep learning because of its ability to create sparse solutions. A drawback of this function is a case called dying ReLU, it happens when most neurons output zero which means the gradient cannot be backpropagated, resulting in part of the neurons becoming inactive and outputting only zero for any input. The Leaky ReLU was introduced as an enhanced version of the existing ReLU which solves the problem of the dying ReLU. However, compared to ReLU, it doesn't create as many sparse solutions.

The final output of the output layer is calculated according to the following equation:

$$\hat{y} = g\left(\sum_{q=1}^l h_q w_q^o\right)$$

Where h_q represents the output, and w_q^o is the output weight of the q th hidden neuron. As there is no analytical process or expression to calculate these weights in a single step, an iterative optimization process is used. The most commonly used method is backpropagation. It mainly consists of three phases: the forward phase where input is passed and allowed to propagate forward through the network. At first the weights are randomly initialized, then the values are updated of each neuron h_q . The loss function which is the difference between the predicted and measured output is computed. The second stage is the backward phase. After computing the loss function, the error is passed in reverse, layer by layer from the output layer all the way to the input layer while calculating the gradient of the loss function. To compute the error at a neuron within a specific hidden layer, the errors in the following layer's neurons is multiplied by the connecting weights. Last phase is updating the parameters. The gradient of the loss function is utilized to update the values of the network's weights and biases. The optimization process depends on the steepest-descent. After the computation of the loss function gradient, the updating of the parameters is performed in the opposite direction of the direction resulting in the largest gradient [23, p. 21]. The mean square error, $E_w = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2$ is usually a common choice for the loss function.

The proper selection of the learning strategy is a crucial factor that has to be considered when designing intelligent algorithm. For example, the learning rate (γ) is one of the hyperparameters of the network that doesn't change or update throughout the training, unlike the weights and biases, which poses a problem in choosing the proper value for it. In addition, maintaining a constant rate of learning during the whole learning process can lead to poor performance. One way to go around that is to set the initial value of the learning rate to be large so that an optimal solution can be reached quickly. Then, by reducing the value of the rate, the system avoids to oscillate or drift from the optimal solution. Another method that can lead to better results is to use the history of the cost function's gradient not just its latest value. These approaches are called learning rate decay and gradient momentum. Some common examples like RMSProp and Adam can be found in[23, p. 134].

One of the main problems that need to be avoided during training is overfitting. It happens when the algorithm rather than learning the underlying properties and relation of the data, it memorizes the input-output data. This usually happens when the dataset is too low in diversity. It is possible to detect overfitting while training, by feeding a new, unseen dataset commonly known as test dataset and observe the behavior and trend of the error. Usually, when a system is suffering from overfitting, it will show good accuracy with the training data, but very bad performance and an increase in the test error when given a new, unknown dataset. To avert this problem, many approaches have been implemented, such as early stopping, batch training and neuron drop-out. In early stopping technique, when the validation error doesn't improve after a certain number of epochs, the training process is forced to stop before any overfitting occurs[23, p. 27]. Usually the weights get updated whenever a new set of inputs is used. Using batch training approach, the optimization process runs only after finishing the processing of a

number of randomly selected samples. As a result, instead of individual errors, the average error of the entire batch is utilized to update the weights. Because the effect of outlier values is offset by the good ones in the same batch, the system has a more robust path towards the optima. This also can help reducing the computational cost. Finally, as the name implies, neuron drop-out involves deactivating certain neurons in each layer across the whole network at random. As a result, the algorithm improves its generalization capabilities, and the impact of individual paths in the same batch is minimized[23, p.188].

The process of estimating SOH based on Machine Learning technique usually consists of two main parts, the training phase and the estimation phase. Figure 21 represent the overall framework of the system. During the aging process of the battery, data like (V, I, T and t) is collected from the BMS. Based on this collected data, features should be extracted. These features have to contain enough aging information that makes it relate to the real level of battery degradation. The extracted features with the real SOH values represent the training dataset. The training phase is often performed offline, on the other hand, the estimation phase can be performed online or offline. The Machine learning algorithms tries to learn the nonlinear relationship between the input features and the output values of SOH.

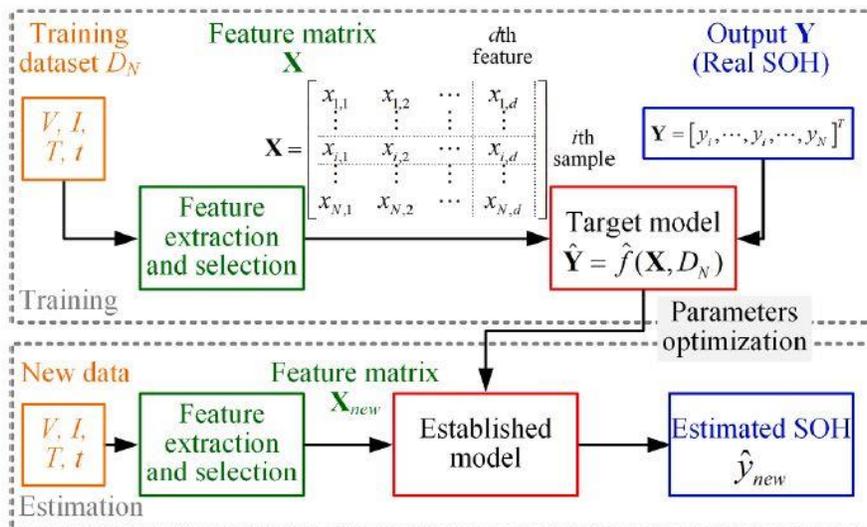


Figure 21 The overall framework for battery SOH estimation using ML algorithms

In the field of AI there exist several advanced and more complex architectures of algorithms. The algorithms and their applications are constantly being improved. In this chapter, the basic and fundamental concepts is introduced. For a more in-depth analysis of the topic, the reader can refer to these references[40]-[43], [46]. In this thesis, the FFNN structure is chosen for its simplicity and good performance in various applications. The use of FFNN in SOH estimation has been proofed to be efficient by the BMS's one-year real-time data collected[47] and the calendar aging performed at a variety of degradation conditions[48]. After the basic background of the batteries and neural networks fundamentals, the following chapters discuss in details the design and development of the algorithm, the choice of current profile and the features selected for the SOH estimation.

Chapter 4

4. Features Extraction and Selection

The definition of SOH is usually associated with the maximum available capacity of the battery or the change in the battery's internal resistance. The capacity fade and the increase of the internal resistance usually are strong factors indicating the degradation of the battery State of Health. These two factors help setting up some sort of criteria to judge the condition of the battery, it helps identifying end-of-life (EoL) thresholds, meaning that when the battery passes this threshold it is thought to have lost major performance capabilities and should no longer be used. When it comes to the capacity, the end-of-life is defined when the available capacity is around 80% of the nominal one, while for internal resistance the battery needs to be replaced when the value reaches 200% of the original resistance value. The SOH mathematical expressions based on these two definitions are shown through the following equations,

$$SOH_Q = \frac{Q_{max}}{Q_{rated}} \cdot 100\%$$

$$SOH_R = \frac{R_0}{R_{0,pristine}} \cdot 100\%$$

4.1 Current Profile

As was shown in the literature review of section 1.2, there exist many methods for SOH estimation from the use of CCCV curve to IC and DV curves. The proposed approach in this thesis is to use a short-term current pulse to ensure the online real-time performance of the system. The following Figure 22, shows the train of pulses used throughout the development of this thesis. The train of pulses is discharging pulses of 10A, every pulse lasts for 10s. Between each two pulses, exist a discharging pulse of 4.314A.

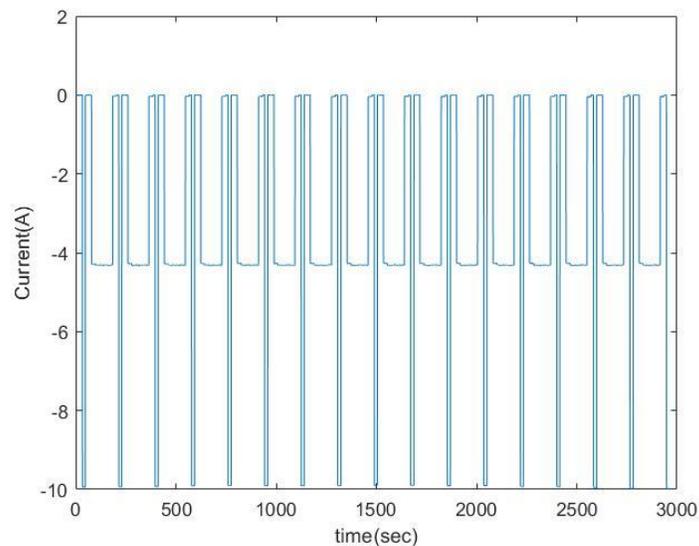


Figure 22 Train of discharging pulses of 10A

Each discharging phase has between 14 to 17 pulses going from 100% level of SOC down to 0% level of SOC. The approach ensures designing a model that is able to estimate battery's SOH using only some short-term data extracted from the voltage response of the battery. As a result of the aging of the battery, the voltage response changes with the battery level of degradation. As shown in Figure 23, the voltage of an old battery drops more quickly than that of a new cell[22]. This is an indication of the increase in the battery's internal resistance and it is power and capacity fade due to the aging process.

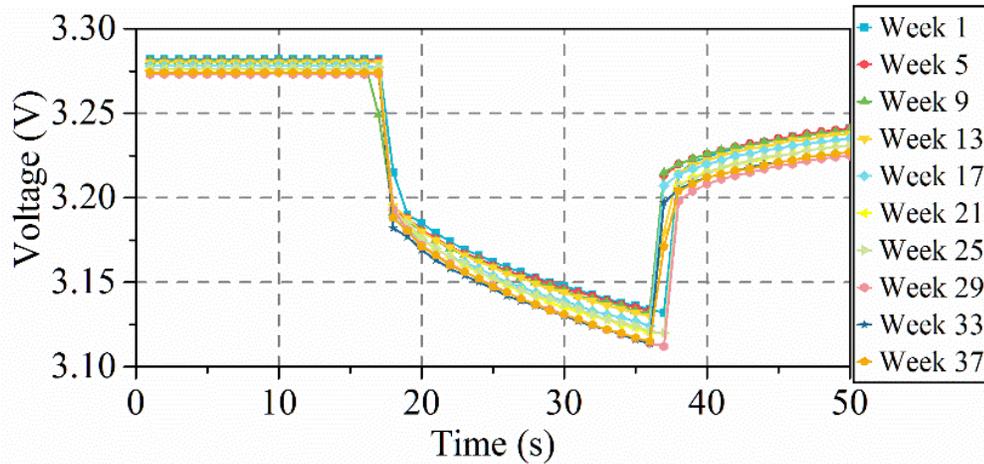


Figure 23 Voltage response under current pulse test through aging process in [22]

There are 4 main points on the voltage response curve that serve as good candidates for features to be used in the SOH estimation. Figure 24 below identifies the 4 knee points on the voltage response curve, voltage point A and B are found at the rising edge of the pulse while points C and D are near the falling edge of the pulse.

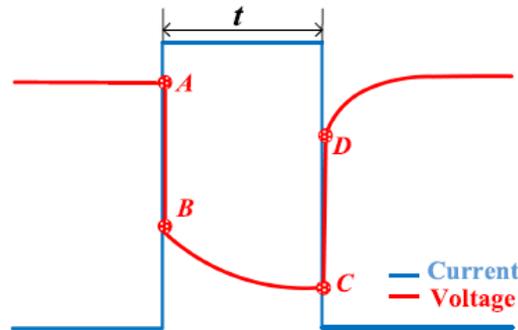


Figure 24 Current and voltage response of one pulse

In addition to the 4 knee points of the voltage response curve, other features like the slope between the voltage points has shown to be an effective features in the process of SOH estimation in[22]. The knee voltage points refer to the geometric change in the terminal voltage curve, while the slopes represent the differential or relative position of these points on the curve. For the ease of access and extraction, in this thesis, 3 knee points on the voltage curve are

considered and they are points A, B and C. Along with the slope between points B and C named as K1, and slope between points A and C named as K3. In attempt to estimate the SOH, a two pulse method was adopted in[49] where the voltage difference between point A and point C was considered as an effective feature in the estimation of SOH. The voltage difference is regarded as the 6th candidate feature. Figure 25 shows a graphical explanation of the first 6 candidates features considered in this thesis.

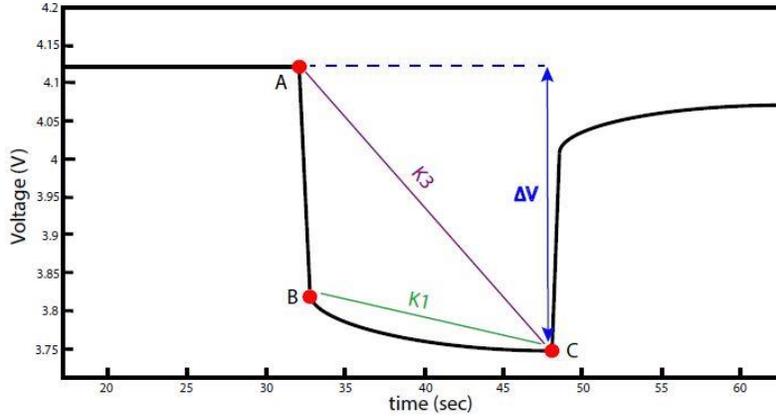


Figure 25 The candidate features in the voltage response under

Lastly, as discussed before the internal resistance of the battery is a great indication of its State of Health. Hence, the last candidate feature considered in this thesis is the internal resistance calculated according to ohm's law which defines resistance as the change in the voltage over the change in the current as shown in the equation below given two points on the discharge curve,

$$R_i = \frac{U_1 - U_2}{I_2 - I_1}$$

According to the VDA (Verband der Automobilindustrie, Frankfurt am Main, Germany) which sets a collection of procedures for cells measurements for the automotive application[50], the discharge resistance of a cell can be measured by applying a constant current discharge pulse for 18 seconds with a rate of 20C. Then, allowing for a 40 seconds rest period, a 16.6C charging pulse is applied to measure the charge resistance. The current profile was applied to the battery and the cell voltage drop was measured after 2 seconds and after 8 seconds to calculate the discharge resistance as the last and 7th candidate feature considered. The discharge resistance is calculated according to the following equations,

$$Ri_{discharge,2s} = \left| \frac{U_1 - U_2}{I_{discharge}} \right| \quad Ri_{discharge,8s} = \left| \frac{U_1 - U_3}{I_{discharge}} \right|$$

Figure 26 illustrates the increase in the calculated discharge internal resistance with the aging of the battery. The value of internal R after 2 seconds ranges from 19.36 m Ω to 28.33 m Ω and the internal R calculated after 8 seconds has the value between 23.43 m Ω and 34.14 m Ω through the aging process.

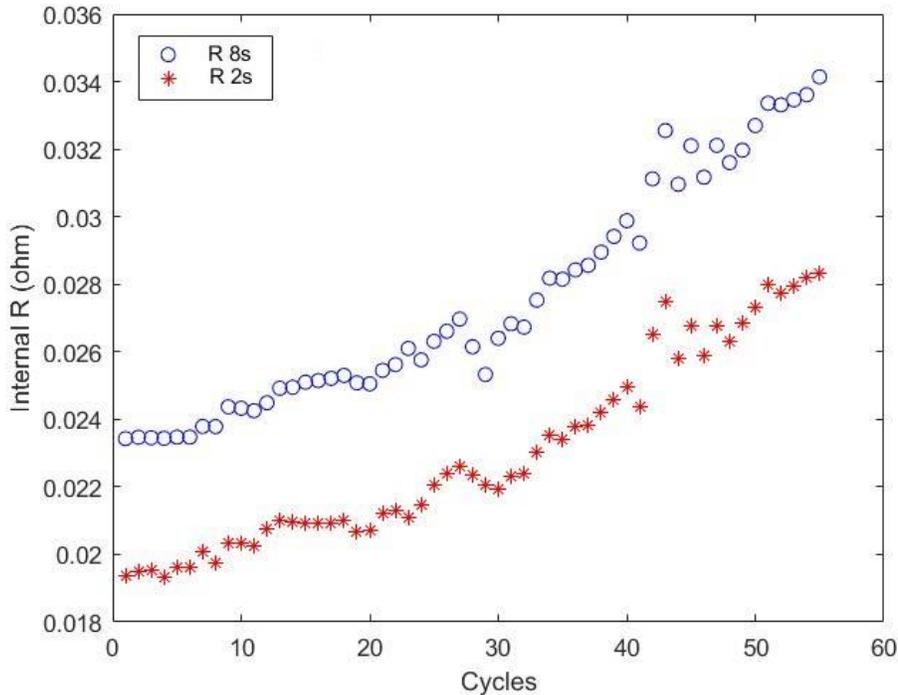


Figure 26 Discharge resistance after 2s and 8s

4.2 Features Correlation and Selection

According to the literature review conducted, these 7 candidate features (VA, VB, VC, ΔV , K1, K3 and R) has shown good results when used for battery's SOH estimation and showed good relation with the level of degradation of the battery. However, further analysis needs to be performed to investigate the relation between these features and the SOH of the battery used during this work. And whether all these features should be used or a combination of some of them.

A simple and effective approach is to analyze the correlation between these candidate features and the battery's SOH. Correlation defines the association between two variables; describing how the change in one variable can affect the behavior of the other variable. A positive correlation exists when two variables increase or decrease in parallel, whereas a negative correlation happens when one variable increase while the other decreases. The two variables have a zero correlation if a change in one variable has no effect on the other. Correlation type statistical measure is used to evaluate the relation between each input feature and the output (SOH) to select features having the strongest relationship with the target output. For this purpose,

the use of Pearson Correlation and Spearman Correlation is utilized. Both methods measure the correlation, however; the fundamental difference lies in the type of analysis required. The Pearson correlation is used for determining the linear relationship between two continuous variables. When a change in one variable causes a proportional change in the other, the relationship is said to be linear. On the other hand, the Spearman correlation determines if two continuous or ordinal variables have a monotonic connection. The variables in a monotonic connection tend to change together, but not always at the same rate. Rather than using raw data, the Spearman correlation coefficient is based on the ranked values for each variable.

Pearson and Spearman correlation coefficients might be anywhere between +1 and -1. When one variable grows and the other variable increases by a consistent amount, according to Pearson the correlation coefficient would equal to +1. A perfect line is formed by this interaction. In this scenario, the Spearman correlation coefficient is similarly +1. The Pearson correlation coefficient is positive but less than +1 if one variable grows and the other variable also increases, but the rate of increase is not consistent. In this situation, the Spearman coefficient is still +1. Both correlation coefficients are essentially 0 when a relationship is random or non-existent. Both correlation coefficients are -1 if the correlation is a perfect line for a declining relationship. The Pearson correlation coefficient is negative but greater than -1, if one variable drop while the other grows, but the amount is not consistent. In this scenario, the Spearman coefficient is still -1.

Starting with analyzing the correlation of the three voltage points (VA, VB and VC) with the battery's State of Health. In Figure 11, it shows the Pearson and Spearman correlation of the 3 points. As shown in the Figure 27, the 3 points maintain a very good correlation with the value of SOH across the various levels of SOC. Except for the bad performance of point A at the first two pulses, almost all points have illustrated a Pearson correlation more than 0.8 and less than 0.9. on the other hand, using Spearman correlation, similar behavior is shown. Yet, the value of correlation is higher, mostly between 0.9 and 0.95. This might indicate that the relation between the input features and SOH is more of a monotonic increase rather than a linear relation.

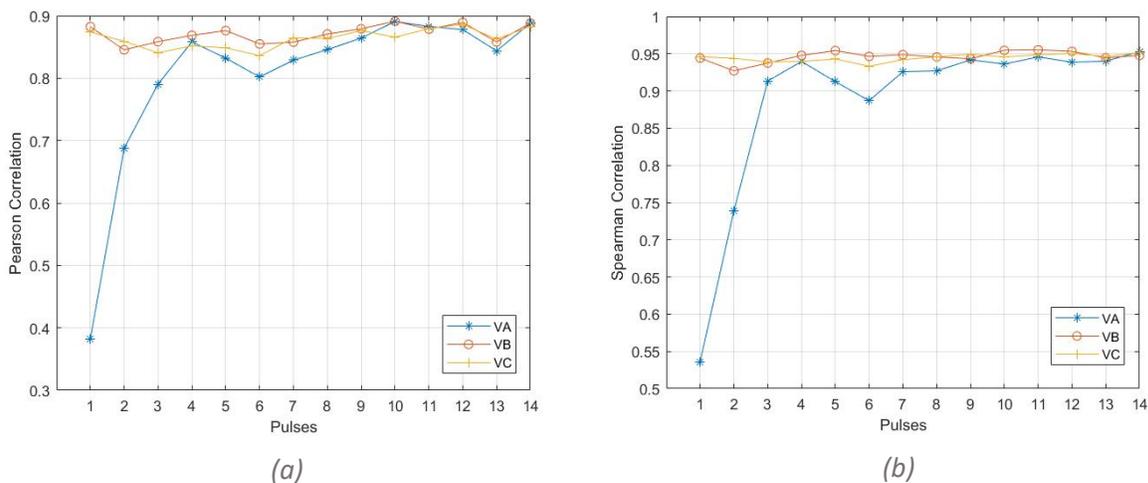


Figure 27 Voltage points correlation (a)Pearson (b)Spearman

The Pearson and Spearman correlation between ΔV and SOH is shown in Figure 28. The correlation here is negative which means the grow in one variable correspond to the decline of the other. This fits the fact that the voltage response of the battery drops with the aging process which result in an increase of the ΔV while the level of SOH decrease. Same as the voltage points, Spearman shows higher correlation with SOH meaning the relation is probably not linear and more monotonic. Spearman correlation ranges between -0.9296 and -0.9645, while the Pearson correlation is between -0.8415 and -0.9068. Overall, the ΔV shows a good correlation with SOH level which implies that it can be a good feature to consider further on.

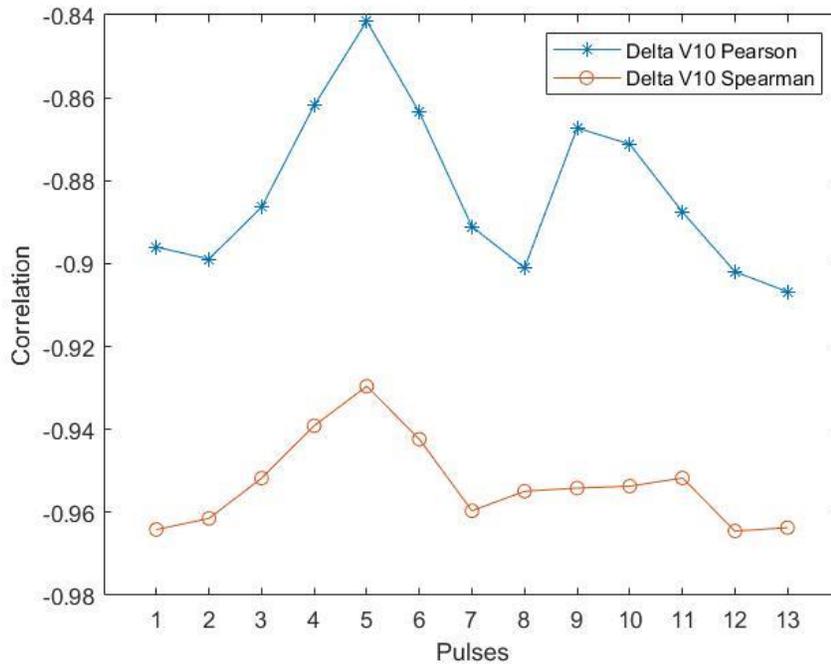


Figure 28 ΔV Pearson and Spearman correlation

As for the slopes K1 and K3, the following Figure 29 illustrates the Pearson and Spearman correlation. Through most of the various SOC levels, the slope K1 has shown worse performance compared to the other slope K3 especially on 100% level SOC at the first pulse. Excluding the first pulse, K1 Pearson correlation is between 0.7567 and 0.9106. While K3's Pearson correlation is in the range of 0.80398 and 0.9159. According to Spearman correlation the trend of the performance is almost the same however, the value of correlation is higher. With K1 Spearman correlation is ranging from 0.7945 to 0.9595 (excluding pulse 1), K3 Spearman correlation is between 0.8493 and 0.9646.

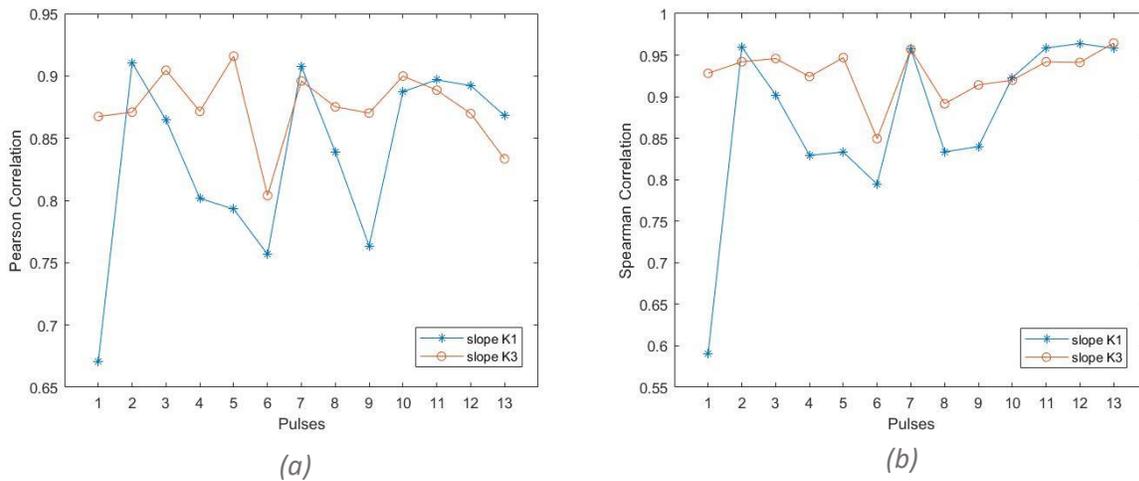


Figure 29 Slopes K1 and K3 correlation (a)Pearson (b)Spearman

Analyzing the Spearman correlation of the calculated internal resistance, it demonstrates a -0.9490 correlation with resistance measured after 2 seconds and a correlation of -0.9508 for the resistance measured after 8 seconds. The correlation is negative indicating that with the decrease and degradation of SOH, the internal resistance increases. Further on, through this thesis, the resistance measured after 8 seconds is considered for its slightly better performance and correlation with SOH level.

Moving forward for the Artificial Neural Network training, 3 different level of SOC are considered for the analysis to decide on which level gives the best results while estimating the SOH. Two pulses are considered at each level of these 3 cases. Analyzing all the features correlation with SOH. The three levels giving the highest correlation over all the features are pulses 12 and 13 corresponding to SOC level between 30% and 35%. The second case considers pulse 7 and 8 where the SOC level is between 60% to 65%. Finally, the last level of SOC considered is between 80% and 85% at pulses 3 and 4.

For a final step before moving on to the ANN training, features are smoothed to remove outlier and zero values. That will help ensuring better and more realistic results. Figure 30 shows the before and after curve of the smoothing of one of the features (ΔV).

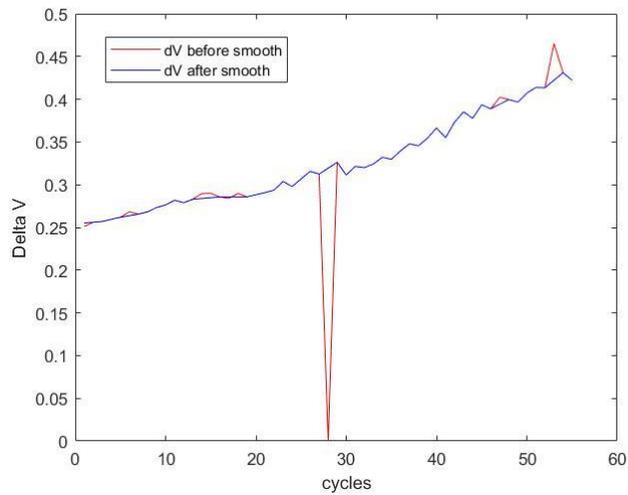


Figure 30 Smoothing of ΔV

The next chapter discusses the preprocessing performed for the features before the training of ANN. It contains in details the procedures followed for the training of the model and the features considered. Lastly, the tuning of the Neural Network hyperparameters is illustrated using genetic algorithm (PSO). Results of the trained network and the test results on the new cells is also included.

Chapter 5

5. ANN Training and Hyperparameters Tuning

The objective of this chapter is to start the training process of the Neural Network classifier for the purpose of identifying the optimal and best SOC level to be further used for the SOH estimation. In addition, iteratively decide on the number of features and their combination necessary for ensuring the accuracy of the system. Lastly, the best SOC level obtained from the first step is going through further analysis through using genetic algorithm (PSO) to decide on the hyperparameters of the Neural Network and analyze if it is possible to obtain better results using this method.

One last step before starting the training of ANN classifier, is additional preprocessing for the features. Analyzing the range and spread of the features' values, it shows a bit of wide spread between some features having values around 4 and others have values as small as -0.02. Standardization which is a scaling technique for all the features is performed to ensure a better and more uniform performance across all battery cells. For this purpose, the Z-score standardization method is used. The Z-score standardize the data to have a zero mean value and a standard deviation of 1. This is achieved through the following equation,

$$X' = \frac{X - \mu}{\sigma}$$

Where μ is the mean of feature values and σ is the standard deviation of the values. The following Figure 31 shows the difference in the spread of the features before and after the Z-score standardization.

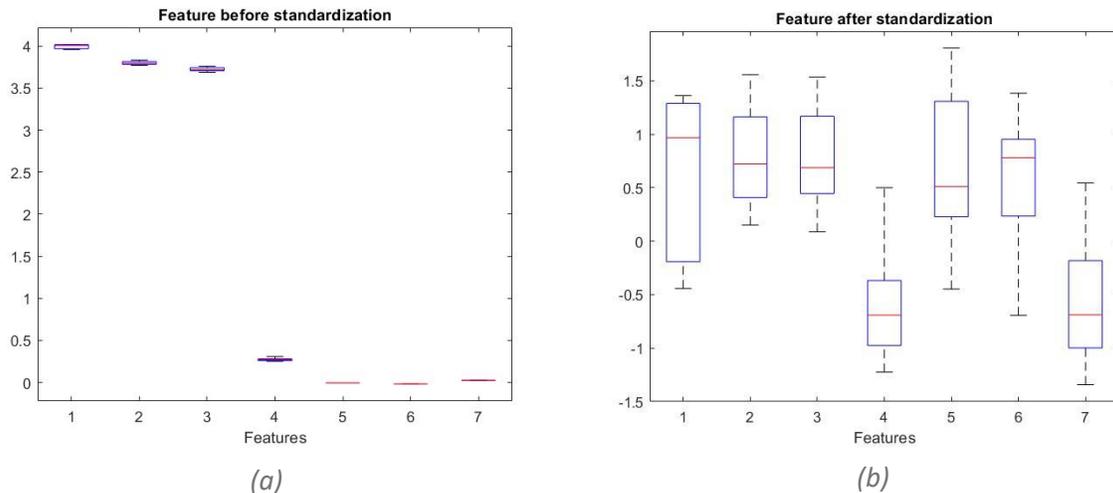


Figure 31 Features range and spread (a) before (b) after Z-score standardization

As discussed in section 3.1, the algorithm designed and tested through this thesis is the feed-forward Neural Network. The configuration for algorithm in the training phase is shown in table 3,

Table 3 FNN configuration parameters

Early stopping	Training function	Optimizer	Epochs	Performance function	Train/Validation split
50 epochs	trainlm	Levenberg-Marquardt	1000	Cross-Entropy	70% / 30%

The network training function is called trainlm, it updates the weights and biases according to Levenberg-Marquardt optimization. It is considered one of the fastest backpropagation algorithms in MATLAB toolbox. The cross entropy is used as the loss function to optimize the model during training. The objective is to minimize the model’s cross entropy. Cross entropy is usually utilized with multi-class and multi-label classification. This thesis aims to classify the SOH into 3 categories either being a new, mid-life or old battery cell.

5.1 ANN classifier training on the 3 different levels of SOC

In order to compare the performance between the different levels of SOC, the same training and testing procedures are repeated for the various levels. The training is performed on Cell0 then to test the network the 5 other cells are used. Network hidden layers dimension is [20, 40]. For each level of SOC, the training is repeated 10 times where only networks with accuracy above 98% are saved. The sequence shown in the flowchart, Figure 32, illustrates the training process, this is also repeated for minimum 5 times in total to better understand the trend of each level of SOC.

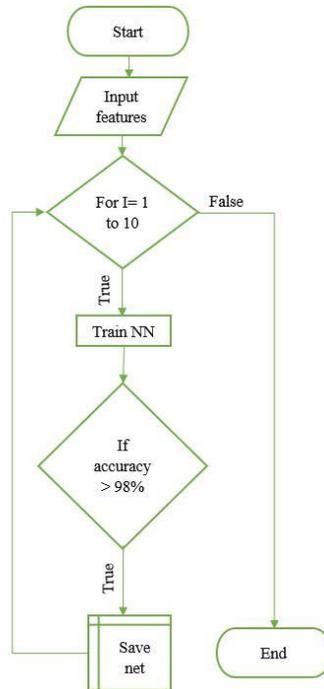


Figure 32 Flowchart of the training procedure

The testing process is performed on the new features obtained from different 5 cells. The initial training results showed a high performance achieving an accuracy higher than 98%. The process

of testing is described using the flowchart in Figure 33, for each level of SOC, the saved network with accuracy 98% or above is fed the new features of the different 5 cells. Then, the average accuracy of the 5 cells is calculated. If there is more than one network out of the 10 iterations with accuracy above 98%, the average accuracy over all these networks is calculated. The same process is repeated for 5 times.

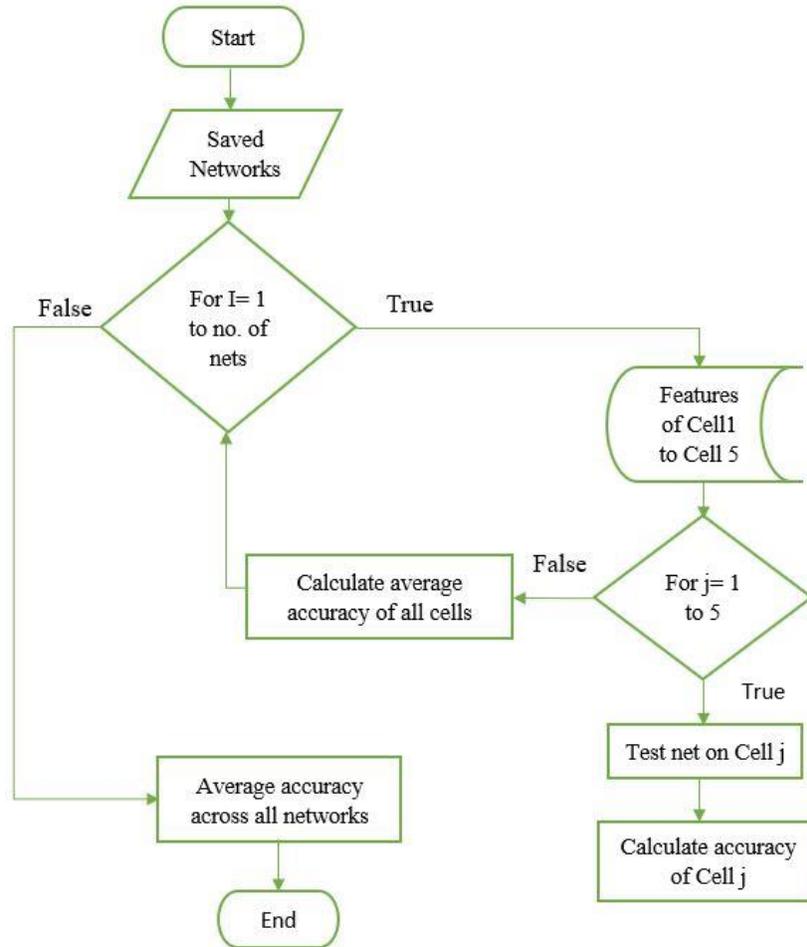


Figure 33 Flowchart of the testing process

To better understand, consider the example of the first trial reported in the first row of table 4 below. Each row of the table represents 10 training iterations on cell0. The integer number refers to the number of networks found with accuracy above 98% during the training phase. In this example, for SOC level 30% to 35%, 2 networks are found. Each network is tested on the other five cells. The first network resulted an average accuracy over the 5 cells equals to 94.73, while the second network resulted in an average accuracy of 95.818 over the 5 test cells. Then the average of these 2 networks $((94.73+95.818)/2)$ is calculated and reported on the first row, first column of the table.

This table presents the results when all the 7 features (VA, VB, VC, ΔV , K1, K3 and R) are used in the training and testing. *Number indicates number of times existed a network with accuracy > 98% out of 10 iterations (**During Training**).

Table 4 Results of testing using 7 features

	SOC 35-30% [p12-13]	SOC 65-60% [p7-8]	SOC 85-80% [p3-4]
1	*2 95.2728	4 95.6365	6 96.6667
2	0	3 96.0605	5 97.2
3	0	2 96.8180	7 96.8831
4	0	2 96.1818	4 96.4546
5	3 92.0607	4 95.9092	6 97.0909

It can be noticed that training and testing using pulses with the lowest level of SOC is performing the poorest. While pulses at higher SOC (80-85%) is showing the highest frequency of obtaining a network with accuracy greater than 98%. The performance of the mid-range SOC and high SOC level is comparable and showing quite similar results. Since the available set of features are not large, the analysis for the different features' combination is done through trial and error by excluding one by one the features performing the least in the correlation analysis. The same procedure for training and testing is repeated but considering only 6 out of the 7 features, excluding VA, the 6 features considered are VB, VC, ΔV , K1, K3 and R. the results are shown in the following table 5,

Table 5 Results using 6 features

	SOC 35-30% [p12-13]	SOC 65-60% [p7-8]	SOC 85-80% [p3-4]
1	3 91.9394	4 95.4546	6 96.1212
2	0	2 95.1818	9 96.6667
3	1 91.9091	4 96.5454	7 96.9610
4	0	5 94.4364	6 96.8486
5	0	1 97.2727	7 96.6493

The degradation in the performance after removing VA is not significant. While the trend of the performance is still the same for the low level of SOC giving the poorest performance while level 60% SOC and 80% SOC are having very close behavior. The final feature to be removed is K1 due to the fact that it has one of the lowest performances in correlation compared to other features. Repeating the process of the training and testing with 5 features (VB, VC, ΔV , K3 and R). The results are reported in table 6,

Table 6 Results using 5 features

	SOC 35-30% [p12-13]	SOC 65-60% [p7-8]	SOC 85-80% [p3-4]
1	4 94.5	3 96.0606	1 94.5455
2	1 90.9	6 96.333	6 96.2121
3	1 91.4545	2 95.8182	7 96.0779
4	1 90.9091	4 96.1364	7 96.2597
5	1 92.5455	4 97.2727	3 95.4545

The overall performance has actually slightly improved after removing the slope feature K1. The trend of the performance of the different SOC levels is still the same. For better visualization of the effect of removing VA and K1, the mid-level of SOC is taken as an example here, arranging the results in ascending order and comparing the performance with all features and with 6 then 5 features. The Figure 34 below illustrates the slight degradation that happened after removing VA, however; improvement is noticed with the use of 5 features after removing K1 which perform even better in most of trials than the full use of the 7 features.

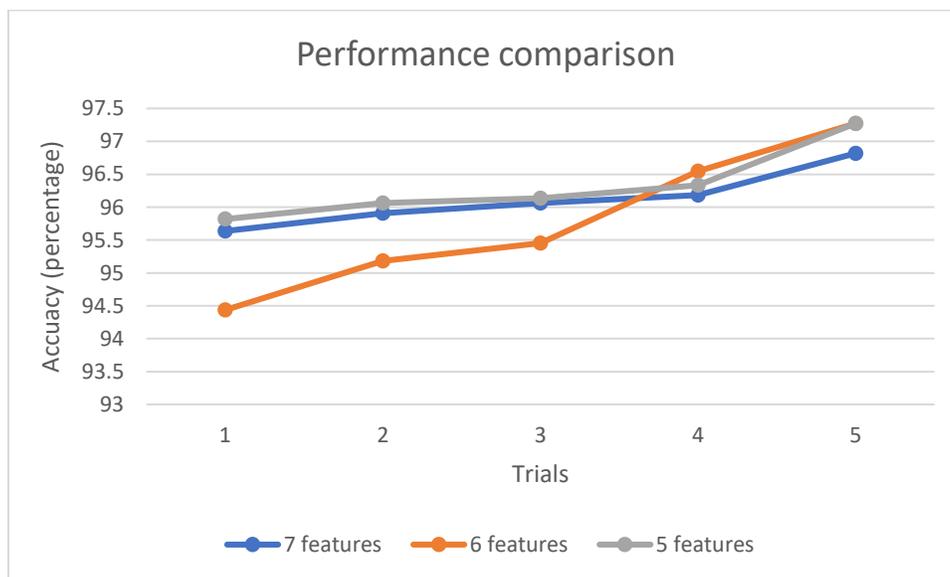


Figure 34 Performance comparison between using 7, 6 and 5 features for SOC 60-65%

As a result, the 5 features (VB, VC, ΔV, K3 and R) are considered for the final features input to the Neural Network. Choosing the best 3 networks for the 3 different levels of Soc, trained with 5 features. The following confusion matrices are shown in the Figures [35-37] below for testing the networks on cell 1.

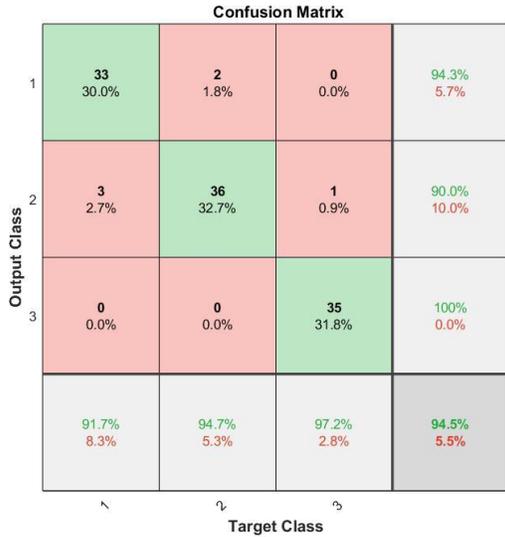


Figure 35 SOC level [30-35] % on test cell 1

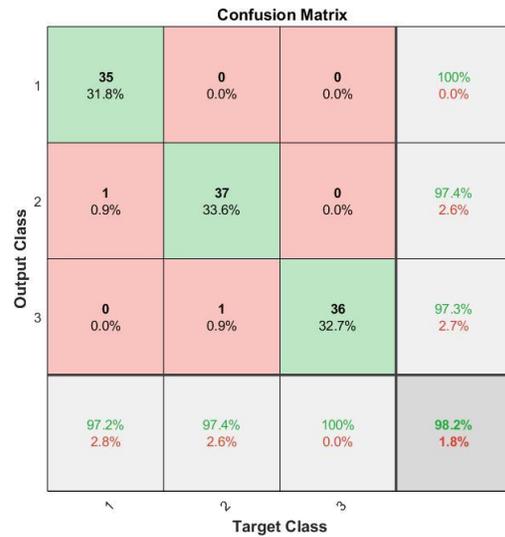


Figure 36 SOC level [60-65] % on test cell 1



Figure 37 SOC level [80-85] % on test cell 1

The final analysis performed with the best networks from the 3 levels of SOC is shown in table 7, the network designed at SOC 30-35%, had an average performance across all cells equals to 96.2%. While at SOC 60-65%, the network's average performance across all cells is 98% and lastly, SOC level at 80-85% has network with average performance across all cells equals to 97.8%. To conclude, the mid-level of SOC around 60-65% represents the best performance and most suitable level of SOC to continue the analysis and the SOH estimation. It is to be noted that all these networks have an accuracy of 98.2% on cell0, the cell that is used during training.

Table 7 Final results of the best networks for the 3 level of SOC

	SOC 30-35%	SOC 60-65%	SOC 80-85%
Average accuracy	96.2%	98%	97.8%

Further analysis is then performed with the features extracted from SOC level of 60-65% to tune the hyperparameters of the Neural Network. These hyperparameters include the number of layers, layers dimension and the value of the gradient descent. The following section discuss in more details the genetic algorithm used, its logic and the results obtained.

5.2 Sensitivity analysis

To understand the relation between the Neural Network performance and the number of output classes and if it has a significant impact on it, sensitivity analysis is considered. Previously, the output was categorized into 3 classes (New, Mean and Old) with the SOH ranging from 100% to around 80%. The previous analysis for the training and testing is repeated for larger number of output classes. To answer the question of will the estimated performance hold for larger number of classes and if not, how bad will it degrade. The output classes are divided to 10 classes, when training the Neural Network with the same configuration used previously, the following results in the table 8 below are found, where the best obtained networks of the different SOC levels are reported.

Table 8 Results of ANN training on 10 output classes

SOC level*	Training	Testing
2&3	96.3	89.636
2	94.54	89.45
3	95.45	90.18
2	96.36	90.545
2	94.545	88.36

*1: pulse 12-13 [SOC 35-30%], 2: pulse 7-8 [SOC 65-60%] and 3: pulse 3-4 [SOC 85-80%]

The mid-level of SOC has yielded the best results once again, however; it is evident that the overall performance has degraded. On the other hand, the lowest level of SOC hasn't recorded any networks to be performing better than other SOC levels. The best network has an accuracy of 96.4% in the training cell (Cell0) and an average accuracy across the other cells equal to 90.545. The following Figures 38 and 39 illustrate the confusion matrices obtained from this network.

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	12 10.9%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.3% 7.7%
3	0 0.0%	0 0.0%	11 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	1 0.9%	0 0.0%	81.8% 18.2%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	10 9.1%	0 0.0%	0 0.0%	90.9% 9.1%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.2%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	100% 0.0%
	100%	100%	91.7%	100%	100%	90.0%	90.0%	100%	90.0%	100%	96.4%
	0.0%	0.0%	8.3%	0.0%	0.0%	10.0%	10.0%	0.0%	10.0%	0.0%	3.6%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

Figure 38 Confusion matrix of Cell0 with 10 output classes

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	11 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	7 6.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	1 0.9%	0 0.0%	12 10.9%	2 1.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
5	0 0.0%	0 0.0%	5 4.5%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	66.7% 33.3%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	1 0.9%	0 0.0%	81.8% 18.2%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	0 0.0%	90.0% 10.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	100% 0.0%
	100%	91.7%	58.3%	100%	83.3%	90.0%	90.0%	100%	90.0%	100%	90.0%
	0.0%	8.3%	41.7%	0.0%	16.7%	10.0%	10.0%	0.0%	10.0%	0.0%	10.0%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

(b) Cell1

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	12 10.9%	1 0.9%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
3	0 0.0%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	11 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	1 0.9%	0 0.0%	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.3% 7.7%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	8 7.3%	0 0.0%	0 0.0%	0 0.0%	88.9% 11.1%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	90.9% 9.1%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	0 0.0%	90.9% 9.1%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	100% 0.0%
	100%	100%	83.3%	91.7%	100%	90.0%	80.0%	100%	100%	100%	94.5%
	0.0%	0.0%	16.7%	8.3%	0.0%	10.0%	20.0%	0.0%	0.0%	0.0%	5.5%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

(a) Cell2

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	12 10.9%	3 2.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0%
2	0 0.0%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
3	0 0.0%	0 0.0%	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
4	0 0.0%	0 0.0%	0 0.0%	10 9.1%	2 1.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	83.3%
5	0 0.0%	0 0.0%	0 0.0%	2 1.8%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	81.8%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	90.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	10 9.1%	0 0.0%	0 0.0%	0 0.0%	90.9%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	3 2.7%	0 0.0%	76.9%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 6.4%	0 0.0%	100%
	100%	75.0%	100%	83.3%	75.0%	90.0%	90.0%	100%	100%	70.0%	88.2%
	0.0%	25.0%	0.0%	16.7%	25.0%	10.0%	10.0%	0.0%	0.0%	30.0%	11.8%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

(d) Cell3

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	12 10.9%	3 2.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0%
2	0 0.0%	9 8.2%	4 3.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	69.2%
3	0 0.0%	0 0.0%	8 7.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
4	0 0.0%	0 0.0%	0 0.0%	12 10.9%	2 1.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	85.7%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	90.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	0 0.0%	100%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	2 1.8%	0 0.0%	75.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	8 7.3%	88.9%
	100%	75.0%	66.7%	100%	83.3%	90.0%	90.0%	100%	90.0%	80.0%	87.3%
	0.0%	25.0%	33.3%	0.0%	16.7%	10.0%	10.0%	0.0%	10.0%	20.0%	12.7%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

(c) Cell4

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
2	0 0.0%	12 10.9%	2 1.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	85.7%
3	0 0.0%	0 0.0%	10 9.1%	0 0.0%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.9%
4	0 0.0%	0 0.0%	0 0.0%	12 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	9 8.2%	0 0.0%	1 0.9%	0 0.0%	81.8%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	10 9.1%	0 0.0%	0 0.0%	90.9%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 7.3%	0 0.0%	100%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	10 9.1%	90.9%
	100%	100%	83.3%	100%	83.3%	90.0%	90.0%	100%	80.0%	100%	92.7%
	0.0%	0.0%	16.7%	0.0%	16.7%	10.0%	10.0%	0.0%	20.0%	0.0%	7.3%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										

(e) Cell5

Figure 39 Confusion matrix of Cell1 to Cell5 with 10 output classes

As shown in Figure. 39, both Cell3 and Cell4 are performing the poorest bringing down the average of the accuracy. Further analysis is performed where classes are divided into seven categories, the overall performance is found to be better with training accuracy of 98.18% and an average testing of 91.82%. it showed a slight improvement in the average test accuracy over the cells however, compared to the initial 3 classes the performance has declined noticeably.

5.3 ANN Hyperparameters Tuning with Meta-heuristic Techniques

When it comes to the choice of the number of hidden layers in a Neural Network or the number of neurons contained in it, there is no 'right' or evident solution. A deeper network may perform better than a shallower one, but it's also possible that the total opposite happens, it all depends on the network's design. The number of hidden layers, neurons and the gradient descent value are part of the hyperparameters that needs to be decided. When creating a network for a specific job, these hyperparameters are usually configured manually, relying on previous experience. However, if there is no prior experience with the specific type of data, it would be very useful to find a method that can automatically select the hyperparameters with little to no previous knowledge.

In classical machine learning techniques an automatic hyperparameter selection is also possible, such as optimization for the learning rate and regression regularization. In the case of regression, the optimization is rather straightforward: several values for each parameter are tested using a grid search, and the best result is chosen at the end[51]. Because the complexity of the search space is very modest — in the aforementioned example, only two dimensions: learning rate and regularization — this strategy works well for regression. Assuming that ten different values for each dimension need to be tested. In this example, a full grid search would necessitate the training of 10^2 models. Considering a similar grid search strategy when designing a deep neural network having 26 layers for example. We already want to limit ourselves to only optimizing the number of neurons in each layer with ten test values each, we'll need to train 10^{26} models. Even the most optimistic estimates of how long it would take to complete this work would be longer than all of our lives put together.

Stochastic gradient descent in [51] is another alternative for optimization approach. The model is evaluated for one set of parameters when employing gradient descent for hyperparameter optimization. The model is then re-evaluated using random parameter changes. The gradient, or change in performance, then defines which parameters will be tested in the next iteration. At the end an optimum will be reached by the set of parameters. There are, however, some disadvantages to the procedure. When the evaluated performance is concave or near concave in nature, so it has one pronounced optimum, the approach performs well. When analyzing classical regression models and shallow neural networks, this is usually the case. The nature of the performance function will not approximate concave behavior when the depth of a network and hence the complexity of the optimization problem increases. It will instead be a function with multiple local minima. When using gradient descent, the optimization will still converge to an optimum, but the chances of it being a particularly good one, much alone the global optimum, are extremely tiny and entirely dependent on chance.

To summarize, optimizing hyperparameters of the neural networks is difficult due to their complexity, and it requires a method that goes beyond what has previously worked for classical machine learning optimization. A vast search space needs to be covered to avoid being stuck in a local minimum, but the number of models trained need to be kept to a manageable number.

Many-population based stochastic search techniques have been proven to be the most promising in such a setting. Genetic algorithms, differential evolution, fruit fly optimization, ant colony optimization, and particle swarm optimization are some of the techniques used.

All of the above algorithms have one thing in common: they all create a group of search entities that do some sort of local optimization while coordinating with one another to reach a globally optimized solution. The majority of the algorithms are part of the swarm intelligence category, which is inspired by biological swarm behavior, such as how a flock of birds or an ant colony responds to particular situations.

Particle swarm optimization (PSO) [52] is a well-known swarm intelligence algorithm. It was first introduced in 1995, and it is based on how a flock of birds in quest of food sources changes its position based on their individual previous positions as well as the position of their swarm. The technique has been tested on a variety of high-dimensional real-world applications and has shown to be efficient and reliable[53]-[54]. It's a population-based meta-heuristic optimization technique, which means it starts with a set of individual searches 'particles,' each of which represents a potential solution. An evolutionary process then modifies the placements of this population of particles. PSO has an advantage over other swarm intelligence algorithms in that it can explore a broad, multi-dimensional search space in a straightforward but efficient manner. While it cannot guarantee that it will find the global optimum solution, it is very likely to find a solution close to it in a few iterations.

PSO has a completely linked swarm in its approach, which means that all particles share information and that any particle knows the best place ever visited by any particle in the swarm. Particles with no background data are initially given a random position P and velocity V . The velocity is changed throughout each iteration by taking into consideration the prior velocity, the particle's individual optimum $Pbest$, and the population's current best solution $Gbest$. Following that, based on the current position and velocity, the position is updated. The position and a velocity of each particle are calculated as follows:

$$x_i(it + 1) = x_i(it) + v_i(it + 1)$$

$$v_i(it + 1) = w * v_i(it) + C_1 * Rnd(0,1) * [pb_i(it) - x_i(it)] + C_2 * Rnd(0,1) * [gb(it) - x_i(it)]$$

w describes the inertia weight, while C_1 and C_2 are the acceleration coefficient to scale the effect and contribution of the cognitive and social component respectively. The combination of these 3 coefficients control the tradeoff between the singular particle's learning rate and the learning rate of the PSO population. The random, stochastic value used between 0 and 1 ensures the extend of the search algorithm to cover more space. The following flowchart (Figure 40) summarize the working principle of PSO.

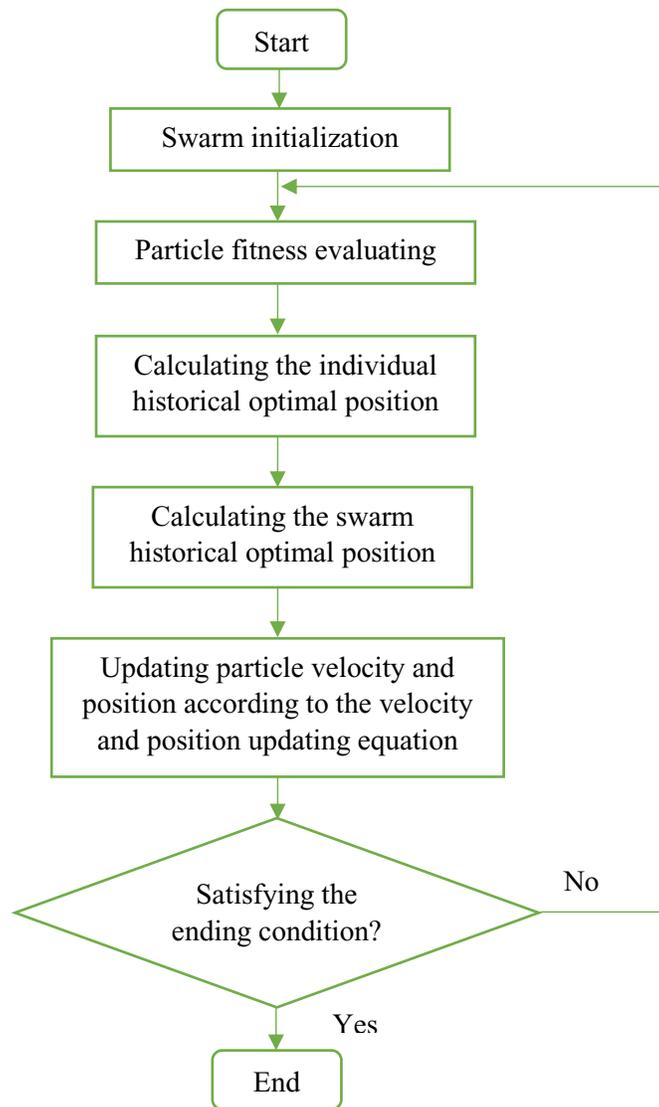


Figure 40 Flowchart of Particle Swarm Optimization

5.3.1 Discrete Particle Swarm Optimizer

Particle swarm optimization is usually used to obtain the global optimum of a continuous mathematical function. When converting the technique to be applied on a neural network, a problem arises that the majority of the hyperparameters, notably the number of layers and their sizes, are discrete integers rather than continuous. As a result, the algorithm must return integer numbers, which may be easily accomplished by rounding the calculated velocities. Rounding, on the other hand, creates an inherent velocity threshold of one below which a particle becomes trapped and will no longer update or change its position. Although this situation might not

happen as often but as a precaution, there has to be an early retirement criterion implemented for the particles fall below this condition.

Taking a closer look at the velocity equation, it is can be noted that when a particle's current position equates to or is close to their local and/or global optimum, the particles are more likely to fall below the threshold. The velocity, or more accurately the inertia weight w , must be dynamically modified because it is important that these particles will be continuing and optimizing their current optimum. Various methods for weighing the inertia dynamically have been developed [17–20]. In conclusion, the optimum method for dynamically changing inertia is very reliant on the optimization problem. In the proposed approach and results, a simple strategy is implemented in which the distance from the particle's local optimum is weight exponentially:

$$w_{dyn} = w + 0.5 * e^{-|p^{i-1} - p_{best}^{i-1}|}$$

Instead of becoming stuck prematurely, this increase in inertia permits the particle to go further past their current local optimum.

The algorithm is then applied on the Neural Network to optimize the number of layers, number of neurons and the value of the gradient descent. The parameters of the PSO are set to w equals 0.7298 and $c1=c2= 1.4962$. The population size is set to 100 particles and 1000 iterations. The algorithm trains on features from cell0 and then test the designed network on the other 5 cells. The objective is to minimize the cost function which is the mean of the classification error on the new 5 cells. The following Figure 41 shows the convergence of the best solution while the particles optimize the global best solution.

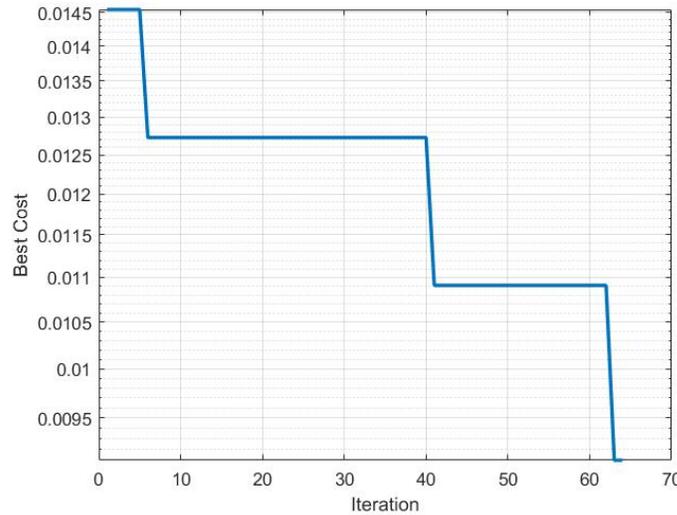


Figure 41 Convergence of PSO global best

The best cost obtained is 0.0091 and the best configuration of the hyperparameters found is to use two layers with the dimension of 52 and 41 neurons. The value of the gradient descent is $1e-16$. The training accuracy on cell0 is found to be 99.1%, as shown in Figure 42, and the test accuracy on the other cells is illustrated on table 9 below.

Cell0

Output Class	1	36 32.7%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	38 34.5%	1 0.9%	97.4% 2.6%
	3	0 0.0%	0 0.0%	35 31.8%	100% 0.0%
		100% 0.0%	100% 0.0%	97.2% 2.8%	99.1% 0.9%
		1	2	3	
		Target Class			

Figure 42 Accuracy of Network trained with PSO on Cell0

Table 9 Accuracy of Network trained with PSO on Cell1 to Cell5

Cell1	Cell2	Cell3	Cell4	Cell5	Average
97.27	100	98.18	100	100	99.09

To conclude, the neural network hyperparameters tuned with the help of PSO has shown better results compared to the analysis done manually with trial and error. The high accuracy of estimating the SOH class for the trained cell and then the test cells, shows a great potential for the usefulness of the PSO when optimizing the choice of the ANN hyperparameters. The next chapter present the process of hardware implementation to verify the accuracy and speed of the designed system in real situations.

Chapter 6

6. Hardware Implementation

In this chapter, after obtaining the results of the ANN classifier trained with PSO, which showed a great performance on both the training and testing cells. One step further is to implement the designed model on the real hardware in order to verify the accuracy and the speed of the whole system.

In order for the designed model to fulfill its purpose, which is to be able to estimate SOH online and to analyze the real-time performance, it is necessary to be able to deploy this model into an embedded system. The deployment of the whole system is divided on two computing systems, the first one is responsible for the application of the pulse to the battery cell and collecting the measured data of the current, voltage and time, while the second one's task is to take in the input features and implement the trained Neural Network to obtain the predicted result of the SOH classification.

The choice of the embedded system is crucial and depends on the application or the task required. The device needs to be reliable and with low latency to ensure a quick and immediate response. To balance between all the requirements and choose the right type of embedded system, few things can be considered. Analyzing the whole task can help realizing the scope of the system which helps in selecting the right processing solution. The amount of processing needed to fulfil the objective of the application is important to be realized because using a very powerful processor at any task regardless of its complexity might lead to unnecessary increase in power, size or price. A good choice is an integrated system on chip (SoC). SoC incorporates a variety of elements that would include a wide range of embedded applications. It is also a powerful system able to execute deep learning algorithms because of the variety of processing components that it contains. Some of these components are a display, video acceleration and industrial networking capabilities[55].

In this thesis the microcontroller used is a Texas instrument C2000 Delfino LaunchPad Board (LAUNCHXL-F28379D). "The C2000 real-time MCU uses the C28x DSP (Digital Signal Processor) core as the main processing unit. Capable of both 32-bit float or fixed-point operations with dedicated instructions tailored to real-time control applications." [56] As the reputation of the C2000 revolves around being a real time controller with low latency and high switching frequencies, it makes it a good fit for this application. An overview of the board is shown in Figure 43.

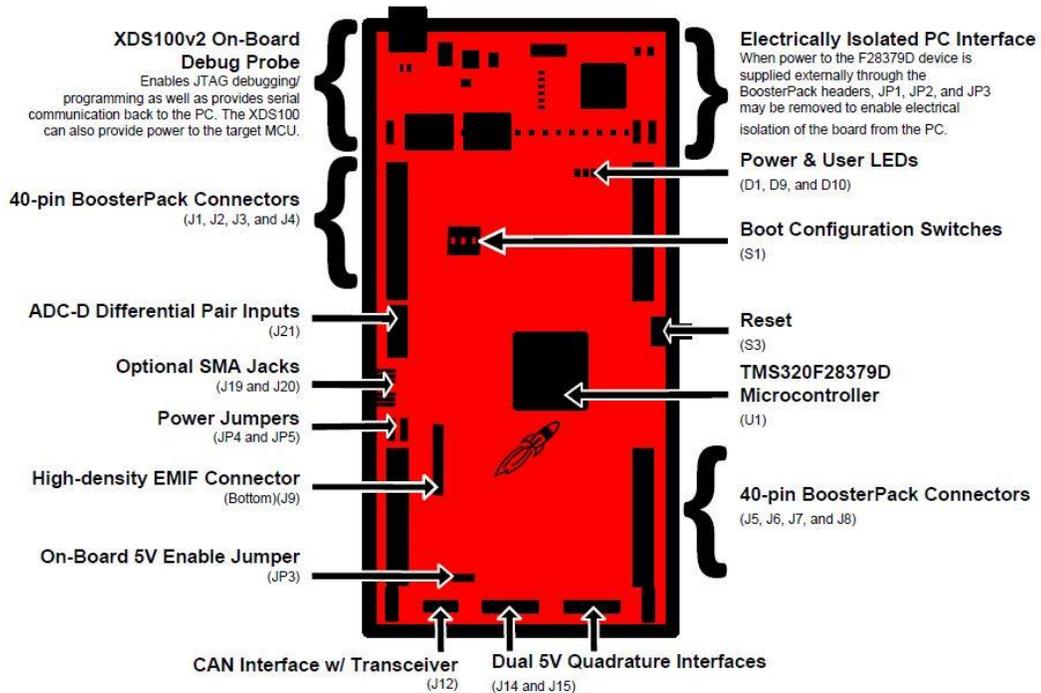


Figure 43 LAUNCHXL-F28379D Board Overview

To accomplish this task, the main approach is to use the automatic code generation of Simulink/MATLAB to run the code on the target hardware. This requires the implementation of the ANN on Simulink along with the preparation of the MATLAB code file that will extract the useful features from the raw data of current, voltage and time. The Figure 44 below shows the overall structure of the entire model on Simulink.

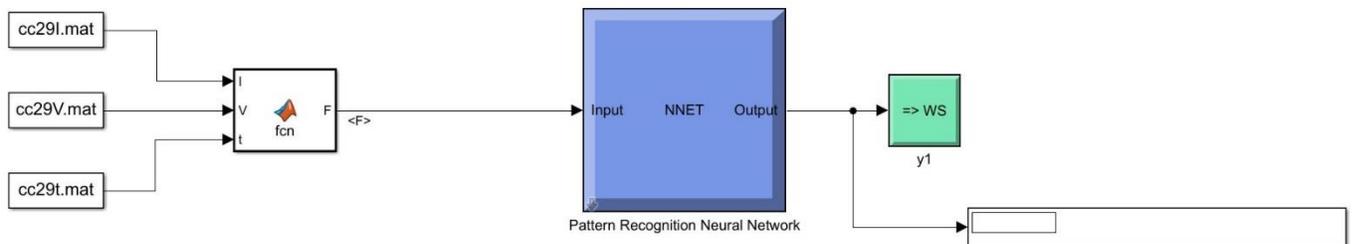


Figure 44 Overall Simulink model of the system

As shown in the Figure 44 above, the three files of raw data (V, I and t) are passed to the MATLAB file which then extract a vector of the five features (VB, VC, ΔV , K3 and R) that is propagated forward to the Neural Network previously designed with the help of the genetic algorithm (PSO). The output for one pulse is a vector [3x1] which contains the class of the predicted SOH. A closer look into the structure of the ANN can be realized in Figure 45.

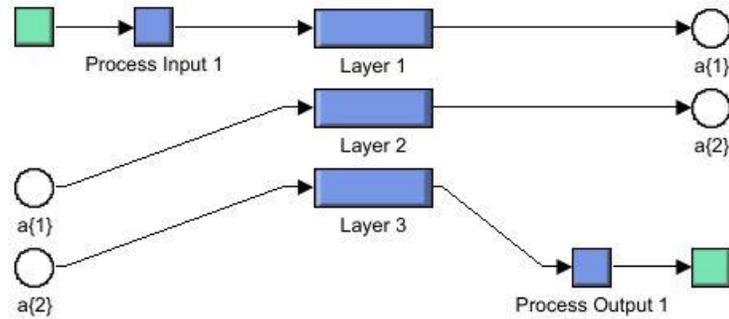


Figure 45 Neural Network structure on Simulink

The analysis is carried with different pulses through the lifespan of the battery, it has proven to maintain the same accuracy and performance on the embedded system while also maintaining the real-time performance. The next chapter summarizes the findings of this thesis and gives an overview of the results obtained and the analysis implemented.

Chapter 7

7. Conclusion

The objective of this thesis as discussed previously in section 1.4 is to analyze the use of machine learning techniques in improving the work done in the field of online SOH estimation for electric vehicles. In this light, various online techniques are investigated and in-depth analysis is done to decide on the most suitable current profile for the online, real time application. Several candidate input features are evaluated and the most optimal combination is used. The training and then testing of the designed Neural Network have indeed proved that the algorithm is capable of accurately estimating the level of degradation with a very short measurement time.

This thesis represents a unique approach in the topic of online SOH estimation. Depending only on a short pulse test fed into the battery cell, the ANN is able to detect with 99.1% accuracy the class of the SOH. And still maintain an equally great performance when tested on other cells. The average error across all the five cells is less than 1%. Without the need of lengthy measurements or waiting for an entire charging or discharging process, the analysis performed for the choice of the candidate features and their correlation to the SOH level ensured a good performance and also included a further analysis into the effect of SOC level and which level can be said to be the most suitable for the task. It was also shown how the genetic algorithm can be a very helpful optimization technique that aids in eliminating the uncertainty stemming from the manual tuning of the Neural Network hyperparameters. And finally, to further ensure the reliability of this designed model, real hardware implementation was performed.

Lastly, the work is one of the few to take into consideration the analysis of current pulse test with the aim of analyzing also the influence of SOC level. And continues the attempt to analyze the use of meta-heuristic optimization techniques that doesn't usually get utilized in this type of context.

References

- [1] Y. Nishi, "Lithium ion secondary batteries ; past 10 years and the future," vol. 100, pp. 101–106, 2001.
- [2] R. U. I. Xiong, S. Member, J. Cao, Q. Yu, and S. Member, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles," *IEEE Access*, vol. 6, pp. 1832–1843, 2018.
- [3] N. Noura, L. Boulon, and S. Jemeï, "A Review of Battery State of Health Estimation Methods : Hybrid Electric Vehicle Challenges," 2020.
- [4] R. Yang, R. Xiong, H. He, H. Mu, and C. Wang, "A novel method on estimating the degradation and state of charge of lithium-ion batteries used for electrical vehicles," *Appl. Energy*, vol. 207, pp. 336–345, 2017.
- [5] V. Mierlo, "Data-driven health estimation and lifetime prediction of lithium-ion batteries : A review Data-driven health estimation and lifetime prediction of lithium-ion batteries : a review," 2019.
- [6] and M. W. Z. Geng, T. Thiringer, Y. Olofsson, J. Groot, "On-board Impedance Diagnostics Method of Li-ion Traction Batteries Using Pseudo-Random Binary Sequences," in *20th European Conference on Power Electronics and Applications*, p. P.1–P.9.
- [7] C. Chen, "Machine learning approach for full impedance spectrum study of Li-ion battery," pp. 3747–3752, 2020.
- [8] M. W. Z. Geng, T. Thiringer, Y. Olofsson, J. Groot, "On-board Impedance Diagnostics Method of Li-ion Traction Batteries Using Pseudo-Random Binary Sequences," in *20th European Conference on Power Electronics and Applications*, 2019, p. P.1–P.9.
- [9] P. B. G. Nusev, D. Jurić, M. Gaberšček, J. Moškon, "Fast Impedance Measurement of Li-ion Battery Using Discrete Random Binary Excitation and Wavelet Transform," *IEEE Access*, pp. 1–1, 2021.
- [10] J. A. A. Qahouq, S. Member, Z. Xia, and S. Member, "Single-Perturbation-Cycle Online Battery Impedance Spectrum Measurement Method with Closed-Loop Control of Power Converter," vol. 0046, no. c, 2017.
- [11] H. Zappen and F. Ringbeck, "Application of Time-Resolved Multi-Sine Impedance Spectroscopy for Lithium-Ion Battery Characterization," no. 1, pp. 1–18, 2018.
- [12] M. A. Roscher, S. Member, J. Assfalg, and O. S. Bohlen, "Detection of Utilizable Capacity Deterioration in Battery Systems," vol. 60, no. 1, pp. 98–103, 2011.
- [13] C. Weng, Y. Cui, J. Sun, and H. Peng, "On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression q," *J. Power Sources*, vol. 235, pp. 36–44, 2013.
- [14] X. Tang *et al.*, "Short communication A fast estimation algorithm for lithium-ion battery state of health," *J. Power Sources*, vol. 396, no. June, pp. 453–458, 2018.
- [15] Y. Li *et al.*, "A quick on-line state of health estimation method for Li-ion battery with

- incremental capacity curves processed by Gaussian filter,” vol. 373, no. October 2017, pp. 40–53, 2018.
- [16] G. You, S. Park, and D. Oh, “Diagnosis of Electric Vehicle Batteries Using,” vol. 0046, no. c, 2017.
- [17] Y. L. Guo PY, Cheng Z, “A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction.,” *J Power Sources*, pp. 412: 442-450, 2019.
- [18] D. Yang, X. Zhang, R. Pan, Y. Wang, and Z. Chen, “A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve,” *J. Power Sources*, vol. 384, no. September 2017, pp. 387–395, 2018.
- [19] Y. P. and L. S. Huaqing Xu, “Health State Estimation Method of Lithium Ion Battery Based on NASA Experimental Data Set Health State Estimation Method of Lithium Ion Battery Based on NASA Experimental Data Set,” 2018.
- [20] L. Cai, J. Meng, D. Stroe, G. Luo, and R. Teodorescu, “An evolutionary framework for lithium-ion battery state of health estimation,” *J. Power Sources*, vol. 412, no. December 2018, pp. 615–622, 2019.
- [21] B. Xiao, “Rapid measurement method for lithium-ion battery state of health estimation based on least squares support vector regression,” no. May 2020, pp. 3–5, 2021.
- [22] J. Meng, L. Cai, G. Luo, D. Stroe, and R. Teodorescu, “Lithium-ion Battery State of Health Estimation with Short-term Current Pulse Test and Support Vector Machine,” pp. 3–8, 2018.
- [23] C. Vidal, S. Member, P. Malysz, S. Member, and P. Kollmeyer, “Machine Learning Applied to Electrified Vehicle Battery State of Charge and State of Health Estimation : State-of-the-Art,” *IEEE Access*, vol. PP, p. 1, 2020.
- [24] A. Emadi, “Advanced Electric Drive Vehicles,” *CRC Press*, 2015.
- [25] R. Xiong, *Battery Management Algorithm for Electric Vehicles*. Springer Singapore, 2020.
- [26] B. M. Armand, P. Axmann, D. Bresser, M. Copley, K. Edström, C. Ekberg, D. Guyomard, M. W.-M. Lestriez, P. Novák, M. Petranikova, W. Porcher, S. Trabesinger, and and H. Zhang, “Lithium-ion Batteries – Current State of the Art and Anticipated Developments,” *Lithium-ion Batter. – Curr. State Art Anticip. Dev.*, vol. 479, 2020.
- [27] A. Masias, *Lithium-ion battery design for transportation*. Springer International Publishing, 2018.
- [28] U. S. D. of Energy, “Battery Archive.”
- [29] V. De Angelis, Y. Preger, and B. R. Chalamala “Battery Lifecycle Framework: A Flexible Repository and Visualization Tool for Battery Data from Materials Development to Field Implementation.”
- [30] Sandia National Laboratories, “Battery Cell Testing Data Archive,” 2017.
- [31] H. Barkholtz, A. Fresquez, B. Chalamala, and S. F. “A Database for Comparative Electrochemical Performance of Commercial 18650-Format Lithium-Ion Cells,” *J. Electrochem. Soc.*, vol. 164, no. 12, 2017.

- [32] Sandia National Laboratories, “Battery Cell Aging Data Archive,” 2020.
- [33] S. F. Y. Preger, H. Barkholtz, A. Fresquez, D. Campbell, B. Juba, J. Romàn-Kustas and B. Chalamala, “Degradation of Commercial Lithium-Ion Cells as a Function of Chemistry and Cycling Conditions,” *J. Electrochem. Soc.*, vol. 167, no. 12, 2020.
- [34] T. Raj, “Path Dependent Battery Degradation Dataset Part 1,” <https://ora.ox.ac.uk/objects/uuidde62b5d2-6154-426d-bcbb-30253ddb7d1e>, 2020.
- [35] T. Raj, A. Wang, C. Monroe, and D. H. “Investigation of Path Dependent Degradation in Lithium-Ion Batteries,” *Batter. Supercaps*, vol. 3, no. 12, pp. 1377–13851, 2020.
- [36] P. Kollmeyer, “Panasonic 18650PF Li-ion Battery Data,” <http://dx.doi.org/10.17632/wykht8y7tg.1>, 2018.
- [37] M. Bercibar, I. Gandiaga, I. Villarreal, N. Omar, and P. V. den B. J. Van Mierlo, “Critical Review of State of Health Estimation Methods of Li-ion Batteries for Real Applications,” *Renew. Sustain. Energy Rev.*, vol. 56, pp. 572–587, 2016.
- [38] A. Barragán Moreno, “Optimal Control of the Ecoracer Powertrain: Design and evaluation of efficient and robust control algorithms,” 2020.
- [39] P. Venugopal and T. Vigneswaran, “State-of-Health Estimation of Li-ion Batteries in Electric Vehicle Using IndRNN under Variable,” 2019.
- [40] R. D. Driankov, P. W. Eklund, and A. L. “Fuzzy logic and fuzzy control,” *Springer Berlin Heidelberg*, no. isbn: 978-3-540-48602-2., 1994.
- [41] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2021.
- [42] I. Goodfellow, Y. Bengio, and A. C. *Deep Learning*. MIT Press, 2016.
- [43] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018.
- [44] M. K. Jain AK, Mao J, “Artificial neural networks: A tutorial,” *Comput Mar*, vol. 29(3), pp. 31–44, 1996.
- [45] H. MH, “Fundamentals of artificial neural networks,” *MIT Press*, 1995.
- [46] G. Rebala, A. Ravi, and S. C. “An Introduction to Machine Learning,” *Springer Int. Publ.*, no. 978-3-030-15729-6, 2019.
- [47] G. You, S. Park, and D. Oh, “Real-time state-of-health estimation for electric vehicle batteries : A data-driven approach,” *Appl. Energy*, vol. 176, pp. 92–103, 2016.
- [48] A. G. Kashkooli, H. Fathiannasab, Z. Mao, and Z. Chen, “Application of Artificial Intelligence to State-of-Charge and State-of-Health Estimation of Calendar-Aged Lithium-Ion Pouch Cells,” vol. 166, no. 4, pp. 605–615, 2019.
- [49] M. Coleman, W. G. Hurley, and C. K. Lee, “An Improved Battery Characterization Method Using a Two-Pulse Load Test,” vol. 23, no. 2, pp. 708–713, 2008.
- [50] “VDA-initiative energy storage system for HEV, test specification for Li-ion battery systems for hybrid and electric vehicles,” *VDA Frankfurt, Ger.*, 2007.
- [51] C.-C. C. and C.-J. L. Chih-Wei Hsu, “A Practical Guide to Support Vector Classification,” *BJU Int*, vol. 101, pp. 1396–400, 2008.

- [52] J. K. and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 — International Conference on Neural Networks*, 1995, pp. 1942–1948.
- [53] A. C. Z. de S. A. A. A. Esmim, G. Lambert-Torres, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. Power Syst.*, vol. 20, pp. 859–866, 2005.
- [54] Z.-L. Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," *IEEE Trans. Energy Convers.*, vol. 19, pp. 384–391, 2004.
- [55] M. Nadeski, "Bringing machine learning to embedded systems," *Texas Instruments*, 2019.
- [56] M. Pate, "The Essential Guide for Developing With C2000™ Real- Time Microcontrollers," no. September 2020, 2021.