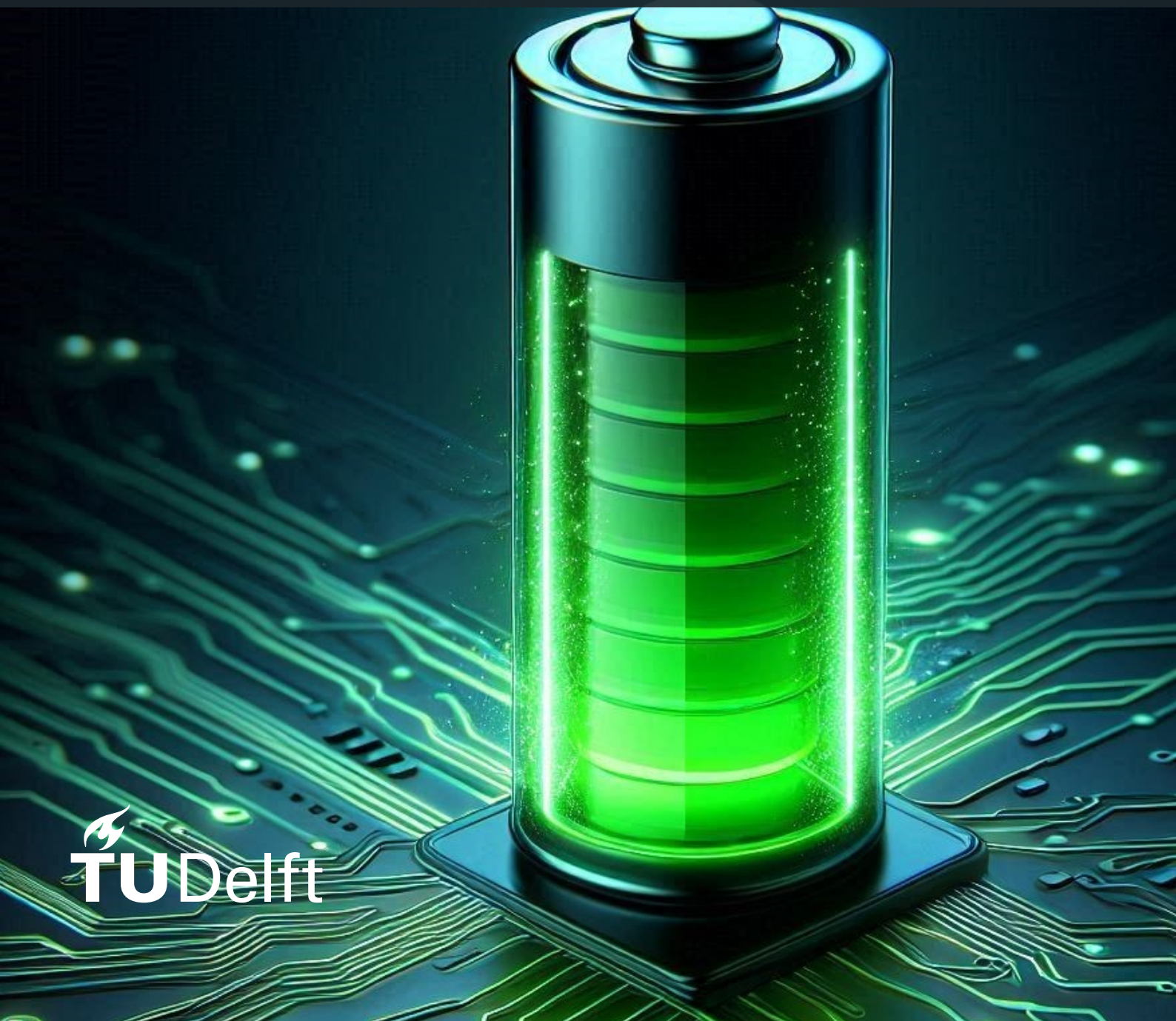
Battery State of Health Estimation

Estimating Aleatoric and Epistemic Uncertainty of the Battery State of Health Using Simultaneous Quantile Regression and Orthonormal Certificates

J.S. Bogaert



TU Delft

Battery State of Health Estimation

Estimating Aleatoric and Epistemic Uncertainty of the Battery State of Health Using Simultaneous Quantile Regression and Orthonormal Certificates

Thesis report

by

J.S. Bogaert

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on December 17, 2025 at 13:00

Thesis committee:

Chair:	Ir. P.C. (Paul) Roling
Supervisors:	Dr. I.I. (Ingeborg) de Pater
	MSc. J.S.H. (Sara) Habib
External examiner:	Dr. Ir. P. (Pieter-Jan) Proesmans
	Dr. D. (Donatella) Zappalá
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	February, 2025 - December, 2025
Student number:	5298601

An electronic version of this thesis is available at `https://repository.tudelft.nl/`.

Cover image generated using *Microsoft Designer*.

Faculty of Aerospace Engineering · Delft University of Technology

Preface

Today marks the end of my 9 month research journey into deep learning, uncertainty quantification and battery state of health modelling. Initially when I was looking for a master thesis topic back in January, I was particularly intrigued when I came across the topic titled: ”Estimating the state of health of electric aircraft batteries”. Even tough I knew very little about batteries and had never learned about deep learning. I had always been very interested in energy systems during my Bachelor and really wanted to learn more about deep learning. Looking back at decision, I am really glad I decided upon taking the challenge. This period has been an incredible learning experience.

I would like to thank numerous people for supporting me throughout this journey, without you this would not have been possible.

Firstly I would like to thank my family and parents for supporting me throughout the thesis and listening to the many difficulties and set-backs I was encountering. I am forever grateful for the opportunities which you provide to me. I would also like to thank my friends for all the help and advice. During working hours it was really enjoyable to work along side everyone on our own thesis and have lunch break together. While outside of university our weekly running sessions were always very nice way to end of our day.

I would also like to sincerely thank my main supervisor, Professor Ingeborg de Pater, and co-supervisor Sara Habib for their continuous guidance, feedback, and support throughout the entire master thesis. The meetings we had every Tuesday were always very valuable and insightful. I particularly enjoyed our long and creative discussion sessions on potential different analytical modelling choices and possible interpretation of our results. It has been a pleasure to be able to perform my thesis on battery state of health modelling under your supervision.

J.S. Bogaert
Delft, December 2025

Contents

List of Figures v

I Literature Review & Research Definition 1

1 Literature Review 2
1.1 Background Information . . . 2
1.2 Battery Systems . . . 3
1.3 Data Set Availability and Description . . . 7
1.4 Learning Methods . . . 10
1.5 Deep Learning Deep Dive . . . 14
1.6 Uncertainty Estimation Methods . . . 19
1.7 Performance Evaluation . . . 21

2 Formal Research Proposal 24
2.1 Introduction . . . 25
2.2 Research Questions . . . 26
2.3 Method, Tools, and Expected Results . . . 26
2.4 Planning . . . 28
2.5 Conclusion . . . 28
2.6 Supplementary Planning Material . . . 29

3 Research Questions 31

References 35

II Scientific Article 36

4 Scientific Paper 37

III Reflection 69

5 Reflection on the Work Performed During the Thesis 70
5.1 Design of Research . . . 70
5.2 Difficulty of Finding Appropriate Datasets . . . 71

Nomenclature

List of Abbreviations

ANN	Artificial Neural Network
BESS	Battery Energy Storage System
BEV	(Battery) Electric Vehicle
BMS	Battery Management System
BNN	Bayesian Neural Networks
BRR	Bayesian Ridge Regression
CC	Constant Current
CI	Confidence Interval
CMU	Cell Management Unit
CV	Constant Voltage
DCNN	(Deep) Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DoD	Depth of Discharge
ECE	Expected Calibration Error
ECM	Equivalent Circuit Modelling
EFC	Equivalent Full Cycle
EIS	Electrochemical Impedance Spectroscopy
EM	Electrochemical Model
EoL	End of Life
GPR	Gaussian Process Regression
GRU	Gated Recurrent Unit
IC	Incremental Capacity
ICA	Incremental Capacity Analysis
LB	Lower Bound
LCO	Lithium Cobalt Oxide
LFP	Lithium iron phosphate
LSTM	Long-Short Term Memory
MA(P)E	Mean Average (Percentage) Error
MBS	Monitoring and Balance System
MCD	Monte Carlo Dropout
ML	Machine Learning
MMU	Module Management Unit
MPIW	Mean width of the Prediction Interval
MSE	Mean Square Error
MVE	Mean Variance Estimation
NCA	Nickle Cobalt Aluminium Oxide
NLL	Negative Log Likelihood
NMC	Nickle Manganese Cobalt Oxide
OC	Orthonormal Certificate
OOD	Out of Distribution
PI	Prediction Interval
PICP	Prediction Interval Coverage Probability
PMU	Pack Management Unit
PPS	Power Processing System
ReLU	Rectified Linear Unit
RF	Random Forest
RMS(P)E	Root Mean Square (Percentage) Error
RNN	Recurrent Neural Network
RS	Reliability Score
RUL	Remaining Useful Life
RVM	Remaining Vector Machine
SEI	Solid Electrolyte Interphase
SGD	Stochastic Gradient Descent
SOC	State of Charge
SOH	State of Health
SQR	Simultaneous Quantile Regression
SVR	Support Vector Regression
UB	Upper Bound

List of Figures

1.1 Academic search results for ”battery AND capacity estimation” courtesy of WebOfScience. 3
1.2 Composition of a lithium-ion battery, extracted from [7]. 3
1.3 Battery degradation for different battery composition, as a results of changes in operating environment [12]. 4
1.4 Main responsibilities of a battery management system, visual extracted from [4] 6
1.5 A simple battery management system architecture extracted from [4] 6
1.6 Charge and discharge, battery sensor measurements as a function of time, for cycle 10 and 100 of battery ”B0005” in the NASA dataset. 8
1.7 Evolution of capacity for battery ”B0005” in the NASA dataset. 8
1.8 Charge and discharge, battery sensor measurements as a function of time, for cycle 10 and 100 of battery ”el150800737329” in the Toyota research dataset. 9
1.9 Summary of the most applicable methods for battery state of health estimation created by [16] . . . 10
1.10 Incremental capacity expressed in Ah/V as a function of voltage [21]. 11
1.11 A basic neural network with hidden layers. 15
1.12 Deep convolution neural network architecture, constructed for battery state of health estimation based on charge measurements [26] 15
1.13 Configuration of the standard long short term memory unit [43] 16
1.14 Neural network architecture developed for remaining useful life predictions of bearings [44]. 17
1.15 Hybrid neural network employed by [19] for predicted battery SOH. 18
1.16 A basic residual network component [34] 19
1.17 The architecture of the residual network based neural network as employed within [34] 19
1.18 Three stage procedure to constructing calibration plots [52] 23

2.1 Proposed model pipeline for the battery state of health estimation model. 24
2.2 project Gantt chart 29
2.3 Project work breakdown structure 30

Part I

Literature Review & Research Definition

1

Literature Review

Within this chapter the literature study can be retrieved, containing a summary of the research conducted during the initial phase of the master thesis. This chapter is structured in the following manner. First in Section 1.1 an introduction into the topic will be provided, highlighting the background and need of battery state estimation. Afterwards in Section 1.2 and introduction to battery management systems (BMS) will be provided. Within Section 1.3, a range of lithium-ion battery datasets will be introduced. Next Section 1.4 will treat the main state of health (SOH) methods which have been developed in research, followed by a more detailed section about deep learning in Section 1.5. Afterwards in Section 1.6 uncertainty quantification techniques commonly used in research will be provided. Lastly methods to evaluate the performance of a deep learning method will be highlighted in Section 1.7. In support of the literature study, the research proposal and research questions can be retrieved in Chapter 2 and Chapter 3 respectively.

1.1. Background Information

Globally significant effort is being done to transition away from more traditionally used energy sources, such as oil and natural gas to more sustainable production methods such as wind and solar. Numerous countries around the world have exhibited clear interest, and are implementing targets to become more sustainable and reduce their CO_2 emissions, to adhere to the climate targets set within the Paris agreements [1]. To achieve this reduction in CO_2, not only the energy industry, but many other industries are required to make fundamental changes to their processes to become more sustainable.

Within a wide range of industries it can be noted that big changes are occurring, which are enabled through a substantial amount of innovation. For example in the energy production industry a large steady growth of both the rated solar and wind capacity can be observed [2]. While in transportation industry, more particularly the car industry, battery electrical vehicles (BEV) have seen an incredible increase in popularity. According to [3] the electrical vehicle market is projected to reach a 1 trillion (US) dollar market volume by the end of the decade. This increase in popularity is not only observable by the market size, academically a sharp increase in popularity can also be recognised as highlighted in Figure 1.1.

If we focus our attention to aviation, we observe this transition is occurring more slowly, even though numerous companies such as *Airbus*, *Boeing*, and *Embraer* exhibiting clear interests into the topic. This slow transition is primary because numerous challenges still exist, withholding widespread application of more sustainable methods. These challenges can be divided across a wide range of topics, and may include elements such as: research and development (R&D) of battery systems, safety, certification, and operations. One of the challengers which we will focus on in this study includes the determination of the state of health (SOH) of a battery, purely based on sensor measurements. Numerous authors have been able to develop successfully develop techniques for this [4]. However safety critical systems [5, 6], such as the ones within aviation, require accurate uncertainty estimations, which has largely been unexplored. This literature review therefore aims to provide to provide insight into a variety of components we performed research into, before starting with our own research.

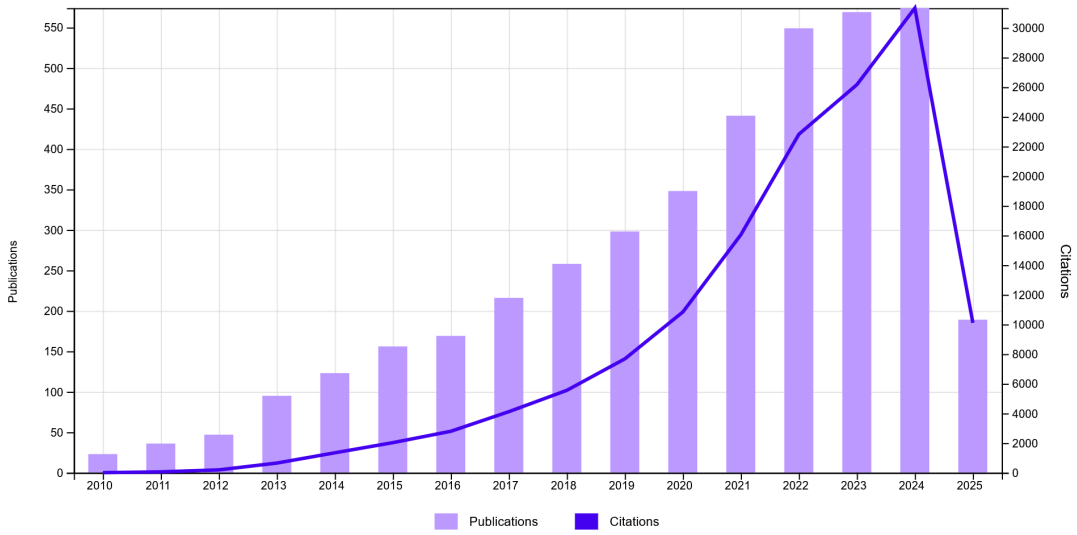


Figure 1.1: Academic search results for ”battery AND capacity estimation” courtesy of WebOfScience.

1.2. Battery Systems

This section aims to provided background information into some general components of a battery and battery system. Although in this master thesis the focus will be put on the developed of an estimation algorithm, it was deemed useful to perform research into the various components and architecture of a battery system. Within this section the focus will be put on two topics. Firstly in Section 1.2.1 a short description on batteries will be provided. Afterwards in Section 1.2.2 the battery management system will be introduced, which is the system enabling the construction of advance control and estimation processes.

1.2.1. Battery cell

The lowest level of any battery system is the battery cell. Regardless of its composition, a general generic battery is constructed out of the following components: electrolyte, anode, cathode, current collectors and a separator. Here current and power are delivered due to the potential created by the cathode and anode. Within Figure 1.2 a simple schematic of a battery is provided by [7].

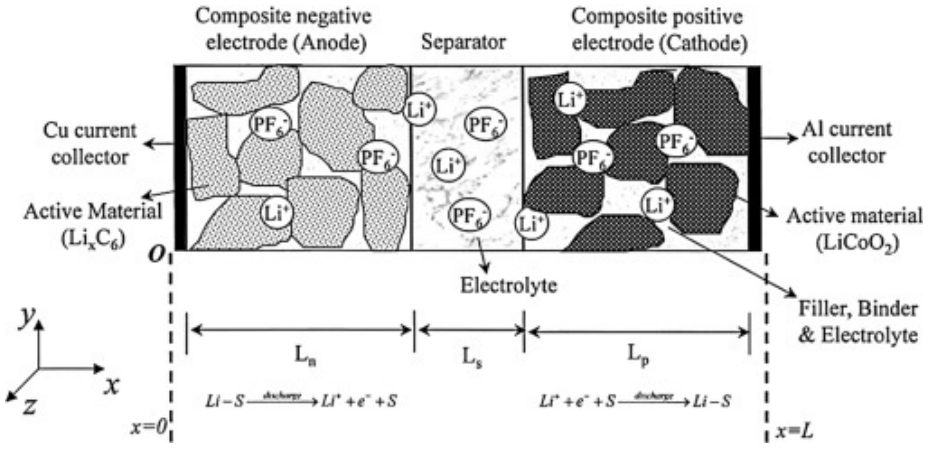


Figure 1.2: Composition of a lithium-ion battery, extracted from [7].

The most well known form of battery composition is the lithium-ion cell, where lithium is coupled with another ion/ group for creating a potential. Although many different types of battery compositions exist such as lead-acid, fuel cells, sodium-sulfur, etc... Lithium-ion batteries have shown to be a popular candidate, for a range of applications,

due to their favourable cost, energy density, performance and diverse operational window [8]. Within Section 1.3 more detailed information about the various types of lithium-ion batteries can observed.

One of the main disadvantages of lithium-ion batteries is their degradation or change in performance over time. A batteries degradation is covered by a range of complex chemical processes, which in turn result in a change in performance. Due to the complex nature of this process the topic is being actively researched, furthermore degradation is also leading for the research which will be conducted in this report. Below a range of degradation mechanisms are described according to [9, 10, 11].

The formation of a ”solid electrolyte interphase (SEI) film” [11] is typically regarded as one of the leading degradation mechanism for a lithium ion battery. This layer is generated early in the batteries life cycle, as a protective measure against corrosion, due to the connection between the electrolyte and the electrode [9]. However due to further growth of the layer, it results in a depletion of electrolytes, and electrode (anode and cathode) material, therefore reducing the batteries potential, and leading to a decrease in capacity or increase in resistance [11]. Depending on the research paper, the authors may also refer to external conditions either accelerating or decelerating the ageing process. For example in [10] the authors performed a comprehensive review on data driven battery prognostic. Here they mentioned the potential dangerous effects of over charging, over discharging, and excessively high current, on a batteries structure and health. Especially over charge could lead to structural damage, generation of heath, and depletion of battery material (both within the electrodes and electrolyte). Alternatively stress within the battery may lead to formation of cracks, fractures or delamination. While extreme temperatures, similarly lead to a reduction of performance.

Unfortunately regardless of how a battery is or is not used, it performance will reduce in time. This is degradation is due to the mechanism mentioned within the previous paragraph. Generally researchers refer to two global degradation processes over time cyclic ageing, and calendar ageing [9, 10]. Their main principle is relatively straightforward to understand, given their name and based on the basic understanding of battery degradation which has now been developed. Cyclic ageing, is due to active use (charge and discharge) of the battery. While calendar ageing, is a process which occurs regardless of battery use, and refers to general degradation in time. The exact dynamics of these degradation processes is however dependent on a range of elements. Within Figure 1.3 the results of an experiment performed by [12] can be observed, in which different lithium-ion battery composition were cycled (expressed as equivalent full cycles (EFC)).

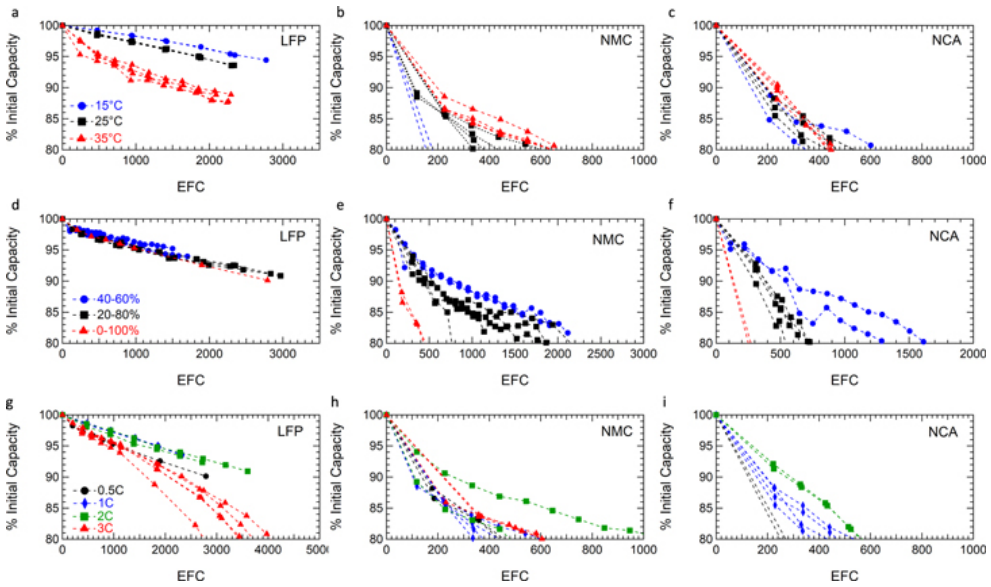


Figure 1.3: Battery degradation for different battery composition, as a results of changes in operating environment [12].

Within Figure 1.3 it can be observed that the dynamics greatly differ based on cell composition, the ambient temperature, discharge rate and lastly the depth of discharge (DoD, proportion of battery capacity which is used). It can for example be observed that the LFP battery is effected less by the DoD and combined with the higher cycle

count, in comparison to the other batter types. The EFC metric, can be confusing since in reality the batteries with a shallow DoD undergo more cycles, however less ”capacity” is used. From an comparison and performance point of view, it is a useful metric since the total used capacity can be compared [12]. Note that a discussion on the different battery compositions, in the category of lithium-ion batteries is included in Section 1.3.

Due to the wide amount of failure mechanisms which could be dangerous and detrimental to a batteries health, it is critical to have a system which can limits these effects. Within Section 1.2.2 a subsection is devoted to describing the BMS, which is an essential system for guaranteeing battery performance.

1.2.2. Battery Management System and Battery Pack Design

In the previous subsection the battery was introduced, although a single battery cell is useful, its capabilities are inherently limited. If we look at the standard 18650 cell for example, which is often used in experiments such as [13, 14, 15]. We observe that the capacity is generally in the region of 1 to 2 Ah, operating at a voltage of around 3 to 4 V. These figures are of course nowhere near what would be required to operate an electrical car, electrical aircraft or energy storage system. To be able to construct suitable battery packs adhering to certain design requirements, manufacturers will connect a large amount of batteries in both series and parallel to form battery modules (afterwards connected into packs). Here series connections lead to an increase voltage, while capacity and current remain constant, while for parallel connections to opposite behaviour can be noted. This effectively leaves us with a large amount of battery cells connected in series and parallel. Now to be to able reliably, safely and optimally use them within large systems, a battery management system (BMS) is required [4] [1].

Battery storage systems have been successfully deployed in numerous different industries [4]. In the car industry, batteries modules have historical been used, to power electrical systems and start a car its engine. However more recently, through the development and advancements made, Lithium ion batteries have become the main power sources for BEV. Alternatively battery modules have shown to be powerful method for energy storage within electrical infrastructure. For example for ENGIE (an energy company), battery storage system or frequently referred to as battery energy storage systems (BESS), have played a crucial role in creating grid stability[2]. This could be achieved due to dynamically charging, and afterwards redeploying the energy into the electrical grid. Regardless however of their use case, as stated in [4] the BMS play a vital role in managing and controlling the different batteries in a battery system.

The BMS is responsible for 2 main tasks: control and state estimation of the various cells in a battery system. In Figure 1.4 an overview of the tasks of a BMS can be retrieved. Within control various elements are included, which can al be summarised into the following categories: reliability, safety, life enhancement and optimal operations. This can be achieved due to the BMS’ direct contact with the battery module, through which it can measure battery parameters and control temperature, charge and discharge procedures. Through this manner the battery can be protected from entering harmful operating conditions, but also it can be ensured that it is operating as close to the design point. Alternatively it is also capable of active load balancing between cells. Therefore keeping the voltage and capacity in balance, and avoiding over or undercharge of the battery module [4].

Within the category of state estimation, the BMS’ main responsibilities includes the determination of health related parameters or battery specific related terms. Typically researchers in the field of prognostics refer to three main metrics: Battery State of Charge (SOC), Battery State of Health (SOH) and Remaining Useful Life (RUL). The SOC, is a measure for the amount of energy which is currently in the battery. While the SOH, refers to the total chargeable capacity relative to the rated capacity, taking into account battery degradation. Lastly the RUL refers to the amount of time, often described as the amount of charge-discharge cycles, until the batteries SOH drops below 80% of the rated capacity [16, 8, 12].

[1] Insightful website concerning battery pack design, introducing various important concepts and including reference to state-of-the-art methods and procedures. URL: `https://www.batterydesign.net/`

[2] `https://www.engie.nl/zakelijk/verduurzamen/energie-encyclopedie/battery-energy-storage-systems-bess` [viewed on 24-3-2025]

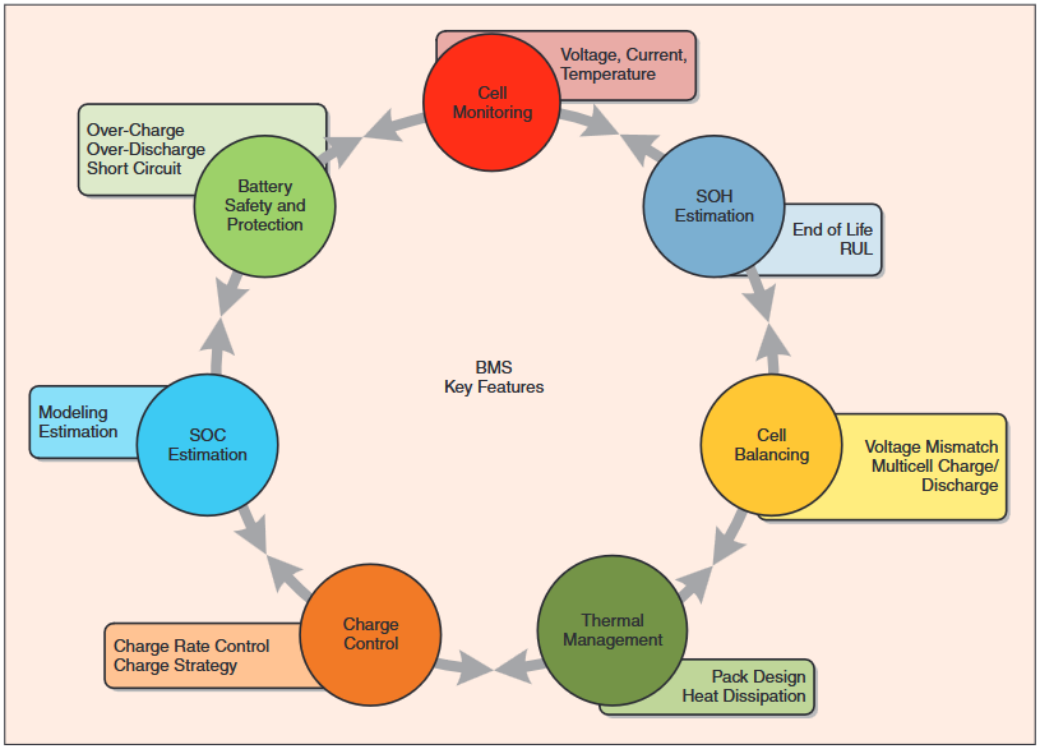


Figure 1.4: Main responsibilities of a battery management system, visual extracted from [4]

To achieve this functionality the BMS is in direct contact with the different cells of the battery pack, through which it can directly measure voltage (U), current(I), and temperature(T) levels, which are then used as an input to the control and estimation strategy. Indirectly it is possible to compute the change in capacity. Utilising the individual measurements the BMS can make informed prediction on the battery health metrics. To provide an indication into the working principle of a BMS a typical architecture of said system can be retrieved in Figure 1.5.

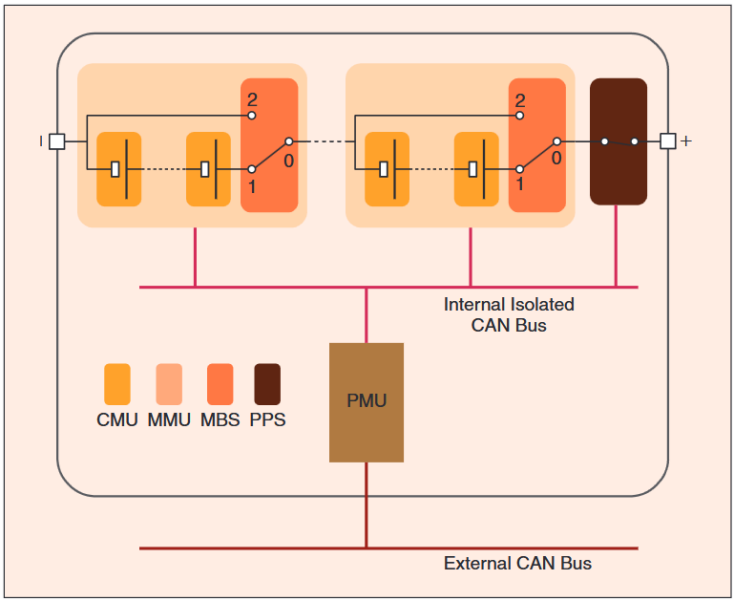


Figure 1.5: A simple battery management system architecture extracted from [4]

Figure 1.5 presents a simplified version of the various components required for a BMS system. At the lowest level

their exists the CMU or the cell management unit (highlighted in yellow in the figure). This unit is responsible for monitoring the individual cells, more specifically it is responsible for measuring the voltage, current and temperature of said sell. Next there is the module management unit (MMU), which collect information from the CMU, processes this, and communicates this to the pack management unit (PMU). Within the MMU three distinct components can be found, namely battery cells and its respective CMU connected in series, and the monitoring and balance system (MBS). The MBS is responsible for the previously mentioned load balancing, where in case of a defect, the switch is activated offering a bypass to the flow. The individual MMUs are then connected together with a power processing system (PPS). Around all of these components the PMU is constructed which is responsible for pack related monitoring [4].

The CMU is a vital component for any state estimation method, since it provides the main processing units with the required cell parameters. Utilising these signals the processing unit can then among other elements, estimate prognostic related data. The role of the CMU, MMU and PMU can be summarised under the terms monitoring and control. While the PPS an MBS are integrated into the system for control and safety reasons.

1.3. Data Set Availability and Description

Previous it was described that due to their favourable performance, lithium-ion batteries are currently the most commonly used for BEV applications. To be able to construct the any learning method described in Section 1.4, it will therefore be critical to find a large amount of high quality data. Due to its popularity in both research and industry, a wide range of different data sets have been made public, which are suitable for the development of deep learning models. The aim of this section is to provide an overview and description of the different datasets which are available.

In 2021, [17], performed research into the wide variety of datasets which are available. The aim of the report was to provide better visibility in the domain. Furthermore the authors highlighted the importance of high quality datasets and monitoring, for both research and design, as well as efficient operations [17]. Generally in the various data sets a two major points of distinction can be noted: battery type, and data type.

Firstly related to battery type all of the datasets presented in the report, can be classified within the lithium ion category. However, an additional distinction can then be made based on the cell architecture/ composition. As the name suggests a lithium-ion battery is constructed out of a lithium (reductant), however variation can occur due to different oxidant it is coupled with. The authors mention four main battery compositions. Firstly there are the phosphate based battery compositions, accounting for 40% of the market share in EVs. This high market share is primarily due to the recent growth of the BEV market in China, where the phosphate based cells are most common [18]. Lithium iron phosphate $LiFePO_4$ (LFP), in this case the lithium (Fe^{+}) is coupled with iron phosphate group ($FePO_4^{-}$). Another composition includes lithium cobalt oxide $LiCoO_2$ (LCO), where the reductant is CoO_2^{-}. Lastly there are two nickel based compositions, lithium nickel cobalt aluminium oxide (NCA) and Lithium nickel manganese cobalt oxide (NMC). These cells have historically been the more popular compositions in the rest of the world [18, 17].

Secondly a division can be constructed based on the data type itself. Here three main categories are frequently referred to by [17], namely cyclic degradation data, drive cycle data, and lastly calendar degradation data. Each of these categories are setup to investigate the effect of a different operational environments, on the batteries performance. This is done to perform research into global mechanism which have an effect on a batteries age and performance [12].

Within all three data types an experimental test setup is created, where a certain parameter is altered, to then investigate the influence on the batteries health, and remaining useful life. These parameters among other, may include ambient temperature of testing environment, by also the manner in which the battery is used (Depth of Discharge or current). Within the cyclic data sets the focus is laid on performing cyclic tests, where additionally the charge and discharge profiles/ settings are altered. Secondly in BEV applications the conditions and energy demand may greatly differ to the “ideal” testing environment in a laboratory. To capture this dynamic nature, the category of drive cycle data was created. Here a standardised test procedure, to better mimic “real life” operations. Lastly there is calendar ageing, this refers to the general degradation of batteries over long periods of time, as highlighted in Section 1.2.1. In this project the cyclic datasets will be of most interest. The remaining subsections introduce a few popular datasets are described, in general they follow a similar principle [17].

1.3.1. NASA PCoE dataset

One of the most popular, and highly cited lithium-ion datasets in literature, are the ones by the NASA prognostics centre of excellence [13, 14]. Due to its ease of use, reliability and the high quality data, numerous authors such as

[19] have utilised their datasets in their research. The dataset itself aims to perform research into cyclic degradation, where the ambient temperature and (dis)charge profiles are varied.

The first of the two NASA datasets is more simple in nature, making it particularly interesting for benchmark comparison in research [13]. The data set contains 30, 2Ah battery cells which undergo cyclic loading until their end of life criteria is met (as previously mentioned in literature often an 80% threshold is utilised). Within the dataset 3 distinct ambient temperature settings are utilised, namely a low setting ($4°C$), medium setting ($24°C$) , and a high temperature setting ($43°C$). During charging the constant current, constant voltage procedure is utilised, discharge is done at a constant current unit a set voltage threshold. The second dataset, takes a more random approach to the problem, here for both charge and discharge a random constant current approach is chosen [14, 17]. In both cases current, temperature, voltage, and capacity values are present. Within Figure 1.6 an example of the data can be retrieved for both charge and discharge. Additionally Figure 1.7 highlight the decay in capacity over time of battery B0005.

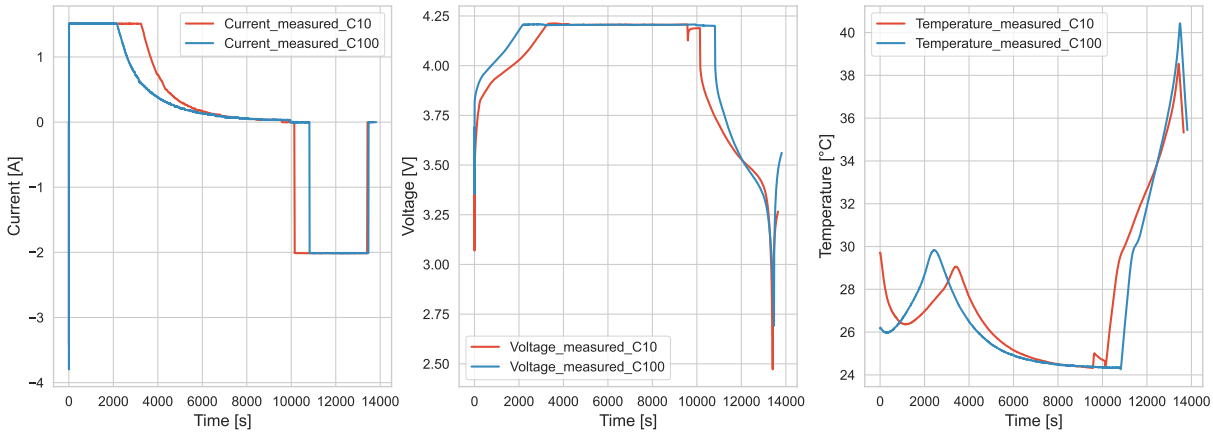


Figure 1.6: Charge and discharge, battery sensor measurements as a function of time, for cycle 10 and 100 of battery ”B0005” in the NASA dataset.

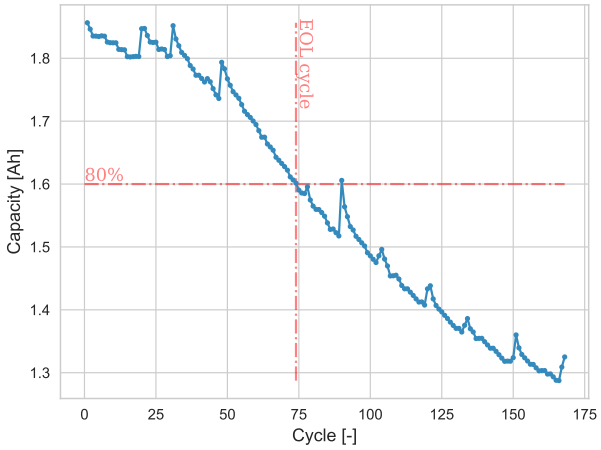


Figure 1.7: Evolution of capacity for battery ”B0005” in the NASA dataset.

Within Figure 1.6 the sensor reading for current, voltage and temperature, for two distinct full charge-discharge cycles, can be viewed as a function of time. Observing the current and voltage plots, the previously mentioned constant current, constant voltage charge procedure can immediately be observed. After a certain period of constant voltage charging it can be noted that the current nears 0 A, indicating the battery is fully charged. The discharge procedure is then started, here the simple constant current approach is used.

Figure 1.7 present the degradation of cell ”B0005” as a function of the amount of cycles. It can be observed

that after around 75 charge-discharge cycles the battery capacity drops below the 80% threshold, for the first time. Typically in a real life scenario the battery would be declared EOL, however here the experiment was continued. Remarkably it can also be observed that, occasionally the capacity of a battery increases. This phenomena might appear, when cycling is halted, after prolonged period of use [20].

1.3.2. Toyota research

Similarly to the NASA dataset highlighted in Section 1.3.1, the dataset created by Toyota research also performed in to the cyclic behaviour of batteries. Here the researchers used a phosphate based cell of 1.1Ah . Contrary to the NASA dataset the researchers utilised a different charging procedure tailored towards fast charging and the temperature was kept constant at $30°C$. The discharge however was performed using the standard constant current procedure. Within the dataset itself more than one hundred cells where tested, and the data was afterwards provided in a clear structured format. One downside however to the dataset in comparison to the NASA dataset, is that some additional work is required to split the charge and discharge data, however the additional amount of quality data makes up for this shortcoming [15]. Within Figure 1.8 the measurements for two distinct cycles can be observed.

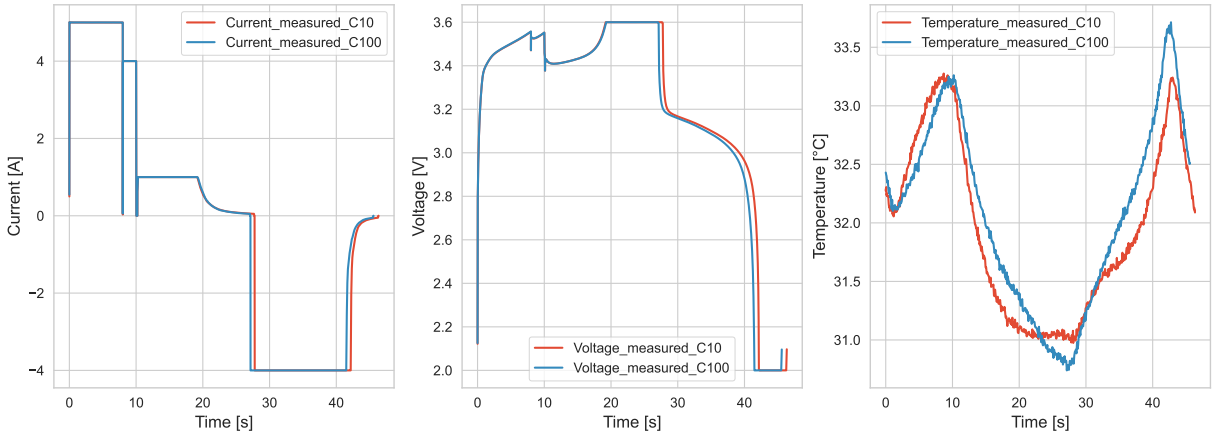


Figure 1.8: Charge and discharge, battery sensor measurements as a function of time, for cycle 10 and 100 of battery ”el150800737329” in the Toyota research dataset.

Within Figure 1.8 the sensor reading for current, voltage and temperature, for two distinct full charge-discharge cycles, can be viewed as a function of time. Contrary to the NASA data set, it can immediately be recognised that, a more complex charging procedure is being utilised. The discharge procedure remains the same, here the traditional constant current approach is used, until the battery is fully depleted

The dataset provided by Toyota can be especially useful to investigate the model its performance to changing charging procedures. In particular it will be interesting to research if the model developed for Section 1.3.1, in which a more traditional charging setup is utilised, also performs well for the non-traditional charging procedure. Since the development of electrical aircraft is currently very early on in its development cycle, their are still many things design decisions which can be made. Therefore a flexible model, which is able to extract and learn deep features regardless of the charging procedure is highly preferred.

1.3.3. Other datasets

Due to the immense popularity of lithium-ion batteries numerous other institutions have also developed key interest in analysing the effect effect of certain parameters on a batteries performance. For example Oxford published their own dataset focussing on more long term dependencies, while CALCE performed similar research to NASA [17].

The researchers at the Sandia National Laboratory within the paper highlighted within Figure 1.3 [12] also made the data of their publication available through the battery archive [3]. Due to the wide variety of factors which where researched in this study, such as the effect of the ambient temperature, battery chemistry and DoD on a batteries capacity. Its content is of immense value for researching how models behave in certain different operational envelopes.

[3] `https://batteryarchive.org/study_summaries.html` [viewed on 14-4-2025]

1.4. Learning Methods

The degradation and unpredictability described previously poses some concerns, especially for safety critical systems, which rely on exact formulation [5]. Due to the complex dynamical chemical processes involved in a battery, it is often not trivial to determine the health related parameters of a battery. For this reason numerous researchers have developed methods through which a batteries health can be evaluated

As previously mentioned in Section 1.2.2 generally there are three important metric related to batter health, namely SOH, SOC, and RUL. This report aims the primary focus will be laid on SOH estimation methods. Within the field of battery prognostics, SOH (among the two other metrics) estimation methods are an active area of research with numerous researchers and companies contributing to the development of state-of-the-art estimation methods. Typically research contributions can be summarised into two main categories experiment based approaches and model based techniques [16, 8]. A nice table created by [16] summarising the most popular methods can be observed in Figure 1.9.

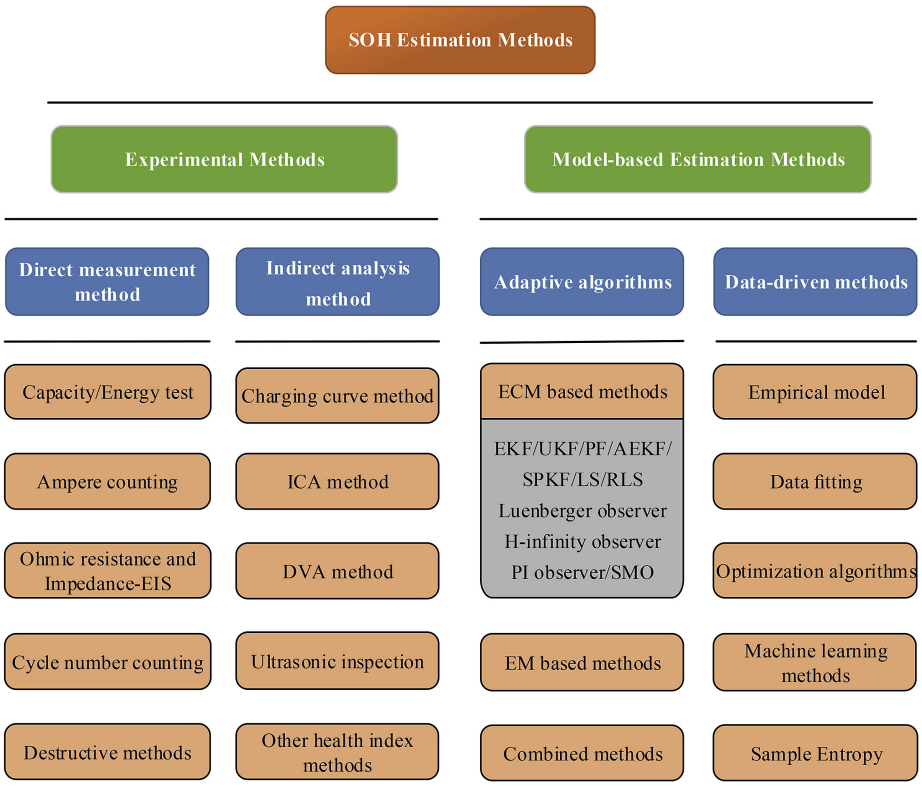


Figure 1.9: Summary of the most applicable methods for battery state of health estimation created by [16]

Within Figure 1.9 the main division between the experimental and model driven techniques, quickly becomes apparent. Generally the experimental techniques are suitable in laboratory environments, while model based techniques are more applicable in an practical or operational setting. The two categories are however heavily linked. Since the total chargeable capacity by itself itself is not a directly measurable parameter, methods are required to be able to infer the capacity. The data driven techniques for example, heavily rely on experimental methods to obtain labels (capacity measures), which can then be used together with sensor measurements to construct a model.

Before highlighting the various methods which could be observed in Figure 1.9, it is important to highlight common definitions. Generally there are two main criteria based on which the battery SOH is evaluated; capacity (C) or internal resistance (R_I). The latter is often not presented or chosen due to the more intuitive capacity criteria. Below in Equation 1.1 the formulation for both can be retrieved [19]:

$$SOH = \frac{C_{current}}{C_{initial}}\,[\%]\,, \quad SOH = \frac{R_{I,EOL} - R_{I,current}}{R_{I,EOL} - R_{I,initial}}\,[\%]\,. \tag{1.1}$$

Here $C_{initial}$ is the rated capacity of the battery in ampere-hour (Ah), and can thus be considered the point of reference. Whereas $C_{current}$ represent the capacity of battery in its fully charged state, in the current cycle, thus taking into account degradation. Similarly the second formulation relies on the internal resistance of the battery, where $R_{I,initial}$ is the initial resistance of the battery expressed in Ohm (Ω). $R_{I,current}$ represent the over time increasing internal resistance and lastly $R_{I,EOL}$ is the resistance at EOL (80% capacity) [19].

1.4.1. Incremental Capacity Analysis (ICA)

Incremental capacity analysis (ICA) is a popular experimental based technique which has been applied by a variety of researchers. For example [21] made use of the technique to analyse calender ageing, under different conditions, purely based on sensor measurements. As initial motivation for the research, the authors noted interests into developing a methods with low computational expense, but high accuracy, such that it would be feasible to perform real time computations in a BMS. The computational complexity is often a big disadvantage of more modern data driven, or high fidelity physics methods. Alternatively [22] utilised ICA in combinations with circuits to predict battery SOH. The method was capable of making high accuracy predictions in the case of both a battery cell and pack. The authors of this report highlighted the importance of integrating electrochemical features into a model, such that informed capacity predictions can be made.

ICA is a technique which makes use of derivatives, based on the originally measured voltage, and capacity signals, for its predictions. The base expression is defined by Equation 1.2:

$$IC = \frac{dQ}{dU}. \tag{1.2}$$

By then relating the incremental capacity to voltage, the following relation can be achieved.

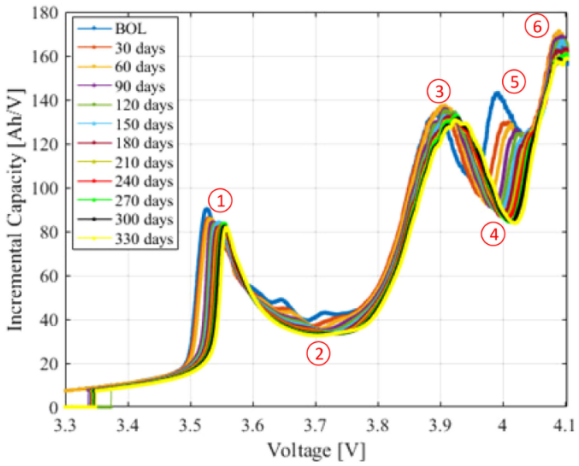


Figure 1.10: Incremental capacity expressed in Ah/V as a function of voltage [21].

Within Figure 1.10 it can be observed that a relation ship exists between the age of a battery, and the incremental capacity chart. More specifically it was found that both the IC peaks, and valleys, in point 1, 2 and 4 hold valuable information relating to the battery SOH. A regression was then be performed relating the capacity to the amplitude of IC peak or value, to obtain a generalised relationship. Utilising the new found expression it becomes possible to determine the SOH as function of the ICA peaks. Both reports were tested and verified against experimental data. Similarly the differential voltage technique relies on the inverse of Equation 1.2

The advantage of the ICA technique is that it is more simple in nature, meaning that is is suitable for real time applications in a BMS. Furthermore through the method it is possible to integrate physical battery phenomena, to make predictions. A downside of the technique is that the method is battery specific [11], and its accuracy and validity may depend on the operating conditions and environment [21].

1.4.2. Ah counting

Ah counting can be defined as the easiest experimental technique, through which SOH can be estimated. Due to its simplicity it can often be used as a validation technique, however its application areas are limited. Ah counting relies on either a full charge or discharge to determine capacity. The technique in essence is simply performing ”counting” operations on the current signal. The generic expression for determining capacity is defined as follows:

$$C = \int_0^t I dt. \tag{1.3}$$

In reality the method is mostly suitable for SOC estimation, and less applicable for real life operation [16].

1.4.3. Electrochemical Impedance Spectroscopy

Electrochemical Impedance Spectroscopy (EIS), is an experimental based, data driven technique, which relies on impedance for its prognostic judgement. As previously mentioned in the introduction of this section, SOH can be assessed through either capacity or internal resistance. Through the use of EIS researchers aim to determine the internal resistance (Equation 1.1), which can then be utilised to determine the state of health [16].

Numerous researchers have utilised the technique, and have achieved strong results. For example resreacher in [20] utilised the technique to predict SOH values in an experiment. Afterwards these values could then be utilised to perform SOH forecasts, using a recurring neural network (RNN). Similarly authors in [23] used EIS as a basis for SOH forecasts, however now the authors relied on particle filtering. In both cases impedance measurement are performed, which are then used to construct the Nyquist plot, from which the internal resistance can be extracted.

From a method point of view EIS is mostly applicable in a laboratory environment, since it can be used to determine SOH. Therefore offering an alternative or complementary method to Ah counting, for determining SOH or capacity, in a practical setting.

1.4.4. Model based techniques

Besides the more experimental based methods highlighted in the previous subsections, the second group of methods rely more heavily on physics informed formulations. Alternatively this group of methods also includes techniques such as machine and deep learning, which will be further introduced in the next subsections. Since the focus of this literature study will primarily be on deep learning techniques, a brief description will be provided on the physics based models.

The physics informed model typically rely on either equivalent circuits or electrochemical models (ECM and EM) or high fidelity models. In first two cases the goal is to represent the complex mechanism of a battery through the formation of ”simplified” expressions. Within ECM, simplified electrical circuits are constructed, which are then used to find the relation between various parameters. For example, as highlighted in [24] an equivalent circuit can be constructed, which on its turn can be used to compute the open circuit voltage and electromotive force, which are a measure of SOC (through the use of refrence tables). Alternatively another popular approach in literature is to combine ECM with existing filtering techniques, such as the H-infinity or the Kalman filter. For example the authors in [25] proposed the use of an extended Kalman filter in combination with ECM for SOC estimation. The implemented method achieved an accuracy of 1.5% in comparison to the 2% achieved by the previous method. The methodology of EM based techniques is similar to ECM, however in this case partial differential equations are used instead [16].

1.4.5. Machine learning

The previously described methods can produce highly accurate results, however especially the physics based model driven techniques, rely on complex formulation. Furthermore it is not always trivial or possible to capture the intricate nature of a battery its internal chemical mechanisms [26]. Additionally techniques such as Ah counting, ICA or any other experimental techniques are not always possible in a practical setting. Because they either rely on specialised equipment, or atypical measurements [22]. As a results the data driven approaches such as classical machine learning techniques (ML) and deep learning (DL) have become increasingly popular. Within this subsection a general introduction to the topic is provided. Whereas within Section 1.5 deep learning is investigated in more detail.

Machine learning and deep learning both share a similar methodology. Given a training dataset, learn or represent an underlying principle in the data. Such that given a test data set, the model is able to make informed predictions for

this new data set. In its simplest form the problem can be defined as follows. Given a problem, where y is a function of x and an error, find the function (g) which can be utilised to approximate y, here denoted as ($\hat{y}$) [27]. The general problem is defined as:

$$y = f(x) + \epsilon \quad \text{(Original)}\,, \qquad \hat{y} = g(x) \quad \text{(Approximation)}\,. \tag{1.4}$$

Numerous different ML and DL techniques have been developed, however in essence they can all be linked back to Equation 1.4. Their main benefit is that they can be utilised to learn complex mechanism, purely based on sensor measurements. Additionally due to their mathematical nature the uncertainty of the predictions can also be quantified in some cases. Due to its importance in this report, Section 1.6 is fully devoted to this topic. Regardless of the ML or DL method which is being used, a common development procedure exists, such as the one outlined in [27], which is described here.

1. Feature, and label preparation
2. Model Training, Testing and Validation
3. Model performance reporting

Feature and label selection
Before any model can be develop, features and labels should be collected. Features also referred to as inputs, is the information which is afterwards utilised to predict a certain category or quantity (label). In the cases the voltage, temperature and current measurements made by a sensor could be considered the model features, while the capacity values (target variable) are considered the labels of the model. Referring back to Equation 1.4, x is the input, while y is the output.

Although not strictly required, it is usually recommended to perform analysis into the data, features and labels, before developing a model, to identify certain patters or anomalies. For example a review paper by [28] highlighted the importance of investigating of utilising metrics, to evaluate the performance and relevance of certain indicators. Although the report was highly focused around remaining useful life, it does provide valuable insight into potential pre-processing and analysis techniques.

Model training, testing, validation
Next, given the data has been pre-processed and the features and labels have been acquired, the next step is to trained a selected model. Depending on the machine learning technique, the exact optimiser may differ, however the main the main goal of training is to estimate model coefficients. Ideally after training, the model should then be able to make accurate and representable predictions given an input.

Two important elements in the model training include the optimiser and a loss function. The loss function is the to be minimised expression which depends on the problem, while the optimiser is the component responsible for getting to a local minimum. Popular loss functions utilised in deep learning are for example the mean squared error (MSE) and negative log likelihood (NLL). While stochastic gradient descent and the Adam optimiser are popular optimisers in the domain.

Validation and testing are both essential parts in the development of any machine learning model. Firstly it useful to understand how a model performs during the training phase. Secondly it becomes possible to determine if the model is suffering from over fitting. To diagnose and ruce over-fitting numerous methods have been proposed such as k-fold cross validation, early termination of training, L1/L2 normalization, etc...

Model performance reporting
To evaluate a models performance generally two perspectives can be taken. Firstly the performance of the point predictions can be made, secondly the quality of the uncertainty predictions can be made. For both categories, a wide range of metrics has been constructed which can be utilised to evaluate predictions. Within Section 1.7 an overview for both is provided. In both cases the metrics are a quick and reliable manner through which it becomes possible, to determine the predictive quality of the model. Note that to a large extent this aspect largely overlaps with the previous step. During training for example, the RMSE metric is evaluated over time.

Applicable research
In [29] a machine learning and deep learning framework was developed using hand crafted features. Through the use of these models battery SOH could be predicted, and the uncertainty of these predictions was quantified. The authors developed and compared 4 procedures: deep neural networks (DNN), Random Forest (RF), Gaussian Process Regression (GPR) and lastly Bayesian Ridge Regression (BRR), to investigate their suitability for SOH predictions. As input to the model an automated feature selection framework was develop, which would sample the most relevant handcrafted features derived from the charge and discharge curve. The features were then provided to the each of the four algorithms, together with the labels (capacity). The models were then trained, validated (k-fold cross validation), and tested on wide variety of datasets. It was deemed beneficial to investigate the models performance in a range of conditions. Firstly the common NASA [13] dataset was utilised, containing sensor measurements of experiments performed on the standard 18650 cell with a capacity of 2Ah [17]. Additionally a dataset by Toyota research [15] was used, which investigated the effects of fast charging. Lastly the CALCE and Oxford dataset were also used. To evaluate the predictive performance the MAPE, and RMSPE metrics were used. The uncertainty was evaluated using calibration plots, C score, sharpness value, and α, β values [29].

Similarly [30] performed research into GPR for battery prognostics, using the NASA dataset. Here as input feature, values from the ICA curve (Section 1.4.1) were sampled, while the experimental capacity figures were used as labels. The paper however failed to perform analysis into the validity of the confidence intervals. Traditionally the ICA curve exhibits considerable fluctuations, and as results it may not be possible to extract either the peaks or valleys from the graph. Furthermore these fluctuations could influence the accuracy of the created model. To solve the issue of noise, the authors present a pre-processing approach relying on the moving average filter or Gaussian filter. Both MAE and RMSE were used as performance metrics, where a maximum error of <1.5 % was reported [30]

The above mentioned ML techniques although popular are not the only techniques which have been utilised in the field. However due to their ability to quantify uncertainty as expressed in [29] they are preferred over methods support vector regression (SVR) [31]. Another popular method which is able to compute uncertainty is the relevance vector machine (RVM). The authors in [32] proposed the use of a RVM and achieved really strong results using relatively simple features. Such as initial and final voltage level or the total charge within a certain discrete period.

1.5. Deep Learning Deep Dive

Deep learning is subset of machine learning, which has received a great amount of attention in recent years, due to its performance and versatility [33]. In comparison to ML, deep learning utilises a biologically inspired (deep) neural network structure for its predictions. With the key benefit being that deep learning in comparison to classical machine learning, is its ability to learn and extract deep features from measurements on its own [34].

The development of the first neural networks can be dated back to halfway the 20th century, with researchers taking inspirations from how brains work [35]. The simplest form of neural network, called a perceptron was developed in 1943 by researchers in [36]. Although very simple, and at the time the capabilities were very limited. Over time researchers developed numerous additions, such as giving it the ability to learn, through the use of back propagation and stochastic gradient descent (SGD) [33, 37]. Now novel architecture and techniques are being applied in a wide range of industries [35].

Within Figure 1.11, a simple neural network can be observed, with two hidden layers, relying on two input features (x) and one output ($\hat{y}$). Each of the lines connecting the ”neurons” or nodes, represents a weight. The main reason how neural networks are able to learn, and approximate complex systems is due to their reliance on non-linear activation functions for mapping intermediate inputs to outputs. The typical structure is included in Figure 1.11, here after summation at the neuron, the new value is inserted into the activation function. Multiple different activation function have been developed such as the sigmoid, rectified linear unit (ReLU), hyperbolic tangents [33, 38]. Given any structure the basic training loop can be summarised as: on the forwards pass given x, $\hat{y}$ is computed based on the current weights. Then based on the difference between the $\hat{y}$ and y (loss function), the weights can then be updated using for example SGD as optimiser and back propagation algorithm, during the backward pass.

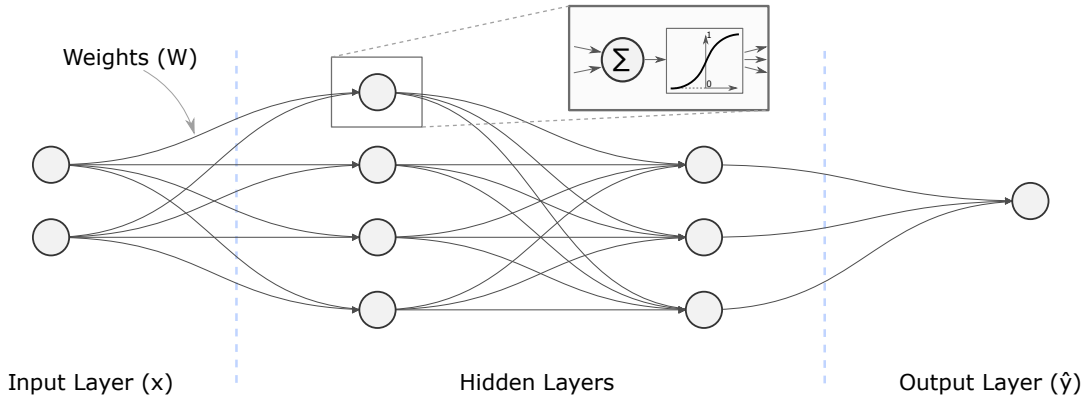


Figure 1.11: A basic neural network with hidden layers.

Due to its success in other domains, researchers also started developing methods to estimate battery state of health, through the use of neural network. Numerous different methods have been developed, each offering unique functionality. The remainder of this section is divided on based on the exact architecture which the researchers utilised. Note that often methods can be of hybrid form, therefore a weak separation is employed in the next subsections.

1.5.1. Convolutional Neural networks

Convolutional neural network (CNN) were initially developed by Y. LeCun for application in image recognition, early references to concept include [39]. Since its development the convolution operation has also found use in other domains, such as the one being researched here. Generally CNN can be seen as one of the important advancements, enabling the automatic extraction and learning of deep features in data [39].

Researchers in [26] developed a deep convolutional neural network (DCNN), where the input features where created based on charge measurements. As motivation for their researcher the authors highlight the ability of a neural network to learn and represent the complexity of battery, as a main driver over traditional methods.

For the DCNN, as input the time dependent voltage, current, and current derived change in capacity were taken at N discrete locations for all cycles (K), during the charging protocol. Whereas for the labels, the standard experimentally derived capacity values were utilised for each cycle. The change in capacity or ΔQ can be computed using Equation 1.3, with adjusted bounds. Now utilising the constructed input and output elements (Equation 1.5), a model was constructed (Figure 1.12):

$$\text{Features}^{(K,N,3)} \Rightarrow \begin{bmatrix} U_{1,k} & I_{1,k} & \Delta Q_{1,k} \\ U_{2,k} & I_{2,k} & \Delta Q_{2,k} \\ \vdots & \vdots & \vdots \\ U_{N,k} & I_{N,k} & \Delta Q_{N,k} \end{bmatrix}^{k \in K}, \qquad \text{Labels}^{(K,1)} \Rightarrow \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_K \end{bmatrix}. \tag{1.5}$$

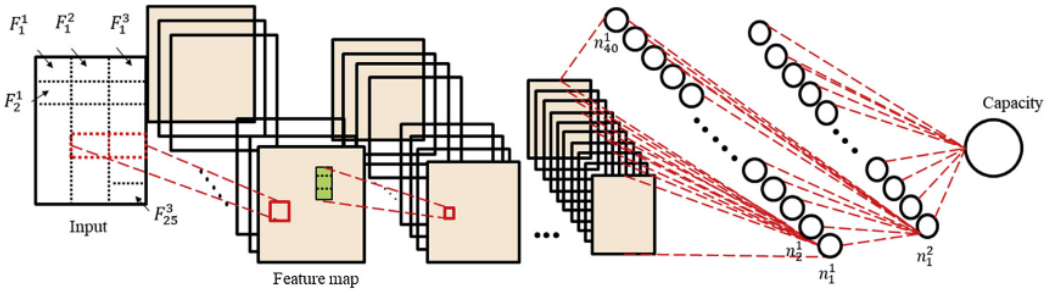


Figure 1.12: Deep convolution neural network architecture, constructed for battery state of health estimation based on charge measurements [26]

The authors integrated additional hidden layers using convolution, max-pooling, batch normalisation, and ReLUs into the final model. This was done to be able to get a deeper representation of the underlying non-linear data. The model itself was trained, validated and testing using k-fold, on the NASA (Section 1.3.1) and a self composed dataset (No reference found). The latter dataset included prolonged cyclic measurements, with the familiar constant current, constant voltage charge and constant current discharge approach being used. For the model, a cropping procedure was employed on top of the data, to reduce dependency on the initial phase of charging. In both cases the data was acquired in a controlled environment, however elements such as DoD and Temperature dependence where not included in the model, therefore transferability remains a question. The model was trained using SGD with momentum, where a generic cost-function with the integration of L2 was used. The final model achieved a root mean square error (RMSE) smaller than 0.5% was achieved, with a maximum error of 5%.

1.5.2. Long Short Term Memory

The above presented form of neural networks, are incredibly powerful. In combination with other specialised components such as convolution layers, they can be used to solve a wide range of complex problems. However a key limitation is that these DNN or CNN may not always be appropriate for time series analysis, since they require the input to be of equal size. To combat this issue the recurrent neural network was proposed, offering a framework or ”building blocks” which are more suitable for time series. The RNN had one important and well recognised downside, referred to as the ”vanishing and exploding gradient” problem. As previously described, Neural networks highly rely on gradients during training to update their model parameters. However due to the ”vanishing and exploding gradient” problem, training an RNN was often not possible or difficult due to the failure to converge [37]. The issue only becomes more apparent when dealing with longer time series data (which is the cases for battery prognostics) [40]. As an attempt to combat this issue the long short term memory (LSTM) [41] was presented.

In 1997 researchers in [41] proposed the concept of an LSTM network, with an improved internal structure greatly reducing the gradient problems. Since is proposal they have been successfully applied in for example language processing applications [42]. LSTM, and a variation called GRU, can both be categorised in the group of RNN. RNN are a type of neural network especially useful for time index data due to their ability to remember and pass through information [40]. To enable this form of memory, an architecture such as Figure 1.13 is required.

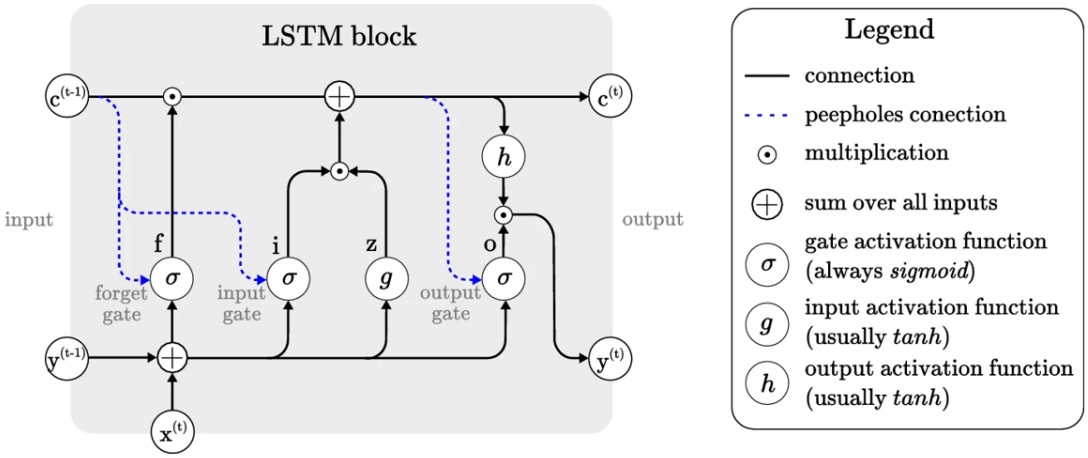


Figure 1.13: Configuration of the standard long short term memory unit [43]

Due to its representation in Figure 1.13 the underlying equations can readily be extracted based as seen in [43]. Note that for now the peephole part of the diagram may be ignored. As previously mentioned the LSTM was initially constructed for time indexed problems, in the diagram this behaviour is denoted by t. The value x refers to the input, y to the output and c refers to the memory of the cell (often also referred to as cell state).

First the LTSM determines which information shall be forgotten (referred to as ”f” in Equation 1.6), and which information shall be remembered (referred to as ”i” in Equation 1.7). Both elements are computed based on the current input x, the previous output y and the respective weights W and bias b, which are altered during the training process.

$$f_t = \textbf{sigmoid}(W_{f,y}y_{t-1} + W_{f,x}x_t + b_f) \tag{1.6}$$

$$i_t = \textbf{sigmoid}(W_{i,y}y_{t-1} + W_{i,x}x_t + b_i) \tag{1.7}$$

Next z is computed, which is referred to as the actual information, which is afterwards combined with f and i to determine the value of c at the next time index.

$$z_t = \textbf{tanh}(W_{z,y}y_{t-1} + W_{z,x}x_t + b_z) \tag{1.8}$$

$$o_t = \textbf{sigmoid}(W_{o,y}y_{t-1} + W_{o,x}x_t + b_o) \tag{1.9}$$

Lastly the the value the output at timestamp t can be computed using Equation 1.10 and Equation 1.11

$$o_t = \textbf{sigmoid}(W_{o,y}y_{t-1} + W_{o,x}x_t + b_o) \tag{1.10}$$

$$y_t = \textbf{tanh}(c_{t-1} \odot f_t + i_t \odot z_t) \odot o_t \tag{1.11}$$

Numerous authors have been able to successfully apply neural networks based on LSTM in their research, highlighting the possible benefits of the method. For example [40] utilised a lstm, to perform SOC predictions. As input features: voltage, current and temperature measurements were taken, and as labels the capacity was used. Through the use of the technique the authors achieves SOC predictions with a RMSE of below 1%, and a maximum error of 2.6%. Although the technique was applied to SOC, the technique proved to give promising results applied to batteries.

Similarly in [44] researchers made use of an LSTM base neural network for remaining useful life studies related to bearing failure. The main difference of this approach, was that the authors decided upon combining the convolution and LSTM operation into a singular layer. This decision was motivated by the possibility that this structure would allow for more representative and high quality features to be extracted from the sensor measurements. The structure employed by researchers can be observed within Figure 1.14.

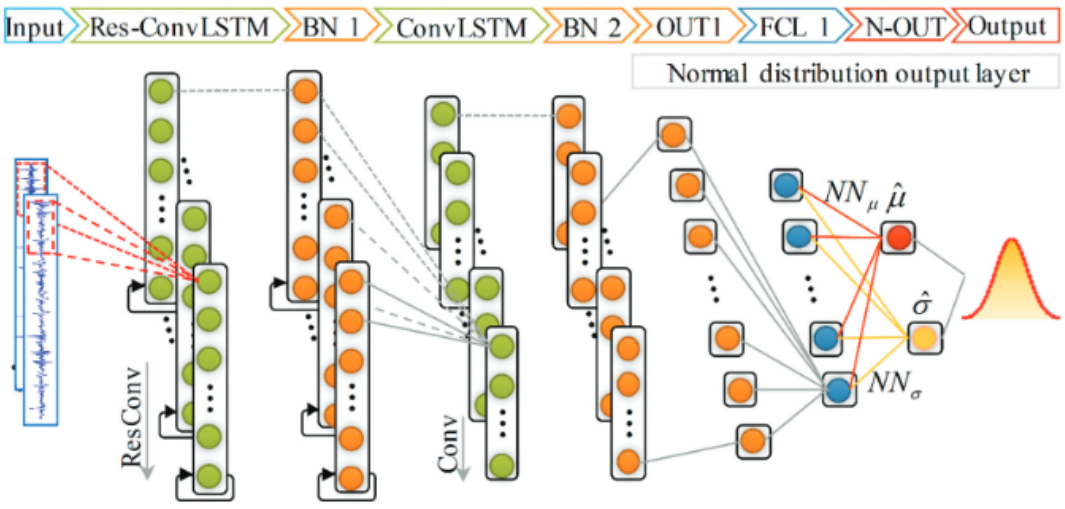


Figure 1.14: Neural network architecture developed for remaining useful life predictions of bearings [44].

Although the topic of bearing useful life predictions is different to battery SOH predictions. Bearing prognostic is an active research area where many novel techniques are developed, such as the one expressed in Figure 1.14. As

a results many interesting observations can be made from the research occurring within this field. An interesting observation which can be made within Figure 1.14 is the output layer which the authors employ. Contrary to most research which has been observed up to now, most researchers only perform point predictions. Here through what the authors call a normal distribution output layer, both a point prediction and standard deviation is predicted. To allow for this, the negative log likelihood (NLL) function is utilised, containing the predicted standard deviation ($\hat{\sigma}$), and the mean ($\hat{\mu}$) [44]:

$$loss \Rightarrow \frac{1}{2}\sum_{i=1}^{N}\left[\ln(\hat{\sigma}_i^2) + \frac{(y_i - \hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right]. \tag{1.12}$$

In [19] researcher formulated a clever approach to estimating battery SOH. Here a 2-part hybrid neural network was developed, one part of the network is based on CNN, whereas the other part makes use of a gated recurrent unit (GRU). The authors believed that combining the CNN and GRU would result in an increased performance. Since CNN have shown strong performance in pattern recognition, while the GRU is directly coupled to time series analysis. Although the internal structure of a GRU is not exactly equivalent to a LSTM (Figure 1.13). The main working principle and motivation is identical to an LSTM. Similar to the first paper discussed within this subsection, the authors made using of the sensor readings as input feature and relied on the capacity as labels. An illustration created by the authors of the neural network can be observed within Figure 1.15.

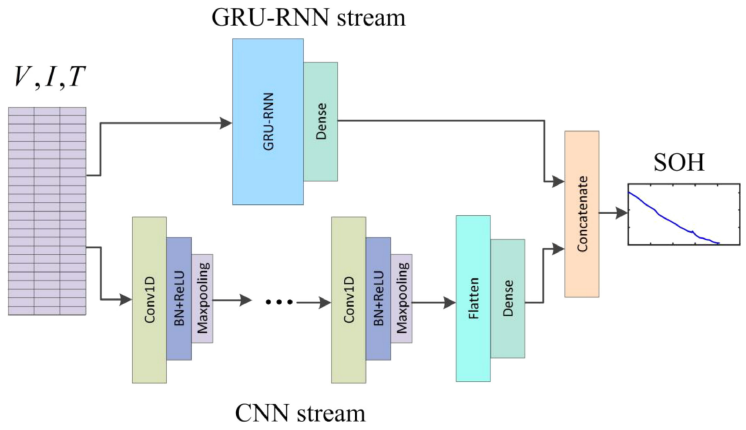


Figure 1.15: Hybrid neural network employed by [19] for predicted battery SOH.

Within Figure 1.15 a clear separation can be observed between the CNN, and GRU-RNN part, which is unique approach, when compared to procedures utilised by other researchers within the domain of battery analysis. It must be noted that the CNN here, is the same as the one expressed within Figure 1.12. The neural network was trained, validated and tested on the datasets published by NASA and Oxford. Similar to the pure CNN based network, the authors did not perform explicit research into temperature or depth of discharge dependency of the model. After training using the Adam optimiser, the point predicted quality was similar to the results obtained by the previous authors.

The implementation referred to at the beginning of this subsection is referred to as the standard implementation of LSTM. Numerous different variant of the base LSTM have been developed such as the previously described GRU, which was applied to battery SOH in [19]. A convolutional LSTM was implemented within [44], and peepholes were described within [43]. Lastly a bi-direction LSTM was utilised within [45] and a CNN-LSTM was used within [46] to predict batter SOH and uncertainty.

1.5.3. Residual Network

Similarly to LSTM, residual networks were created to mitigate the formation of extremely large, or small gradient values [47]. To thus try to limit this gradient problem, researchers in [48] created a concept know as the residual network. Similarly to the other methods discussed within this section, the residual network is another element which can be integrated into a neural network. The working principle of a residual network is straightforward, and may be observed within Figure 1.16. Contrary to regular/ typical neural networks, a bypass is constructed over a discrete

amount of neural network layers, thus creating multiple paths for the signals. The bypass is often referred to as the identity path (since nothing changes), while the ”original” path is referred to as the residual path. When both paths merge, their results are summed together [34].

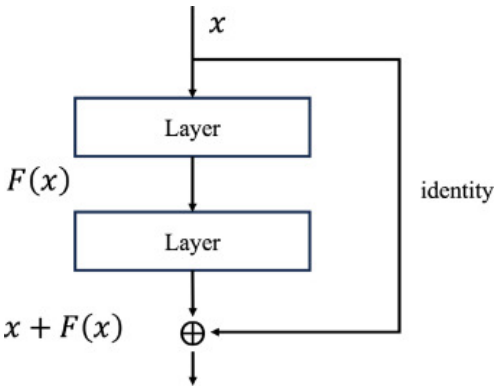


Figure 1.16: A basic residual network component [34]

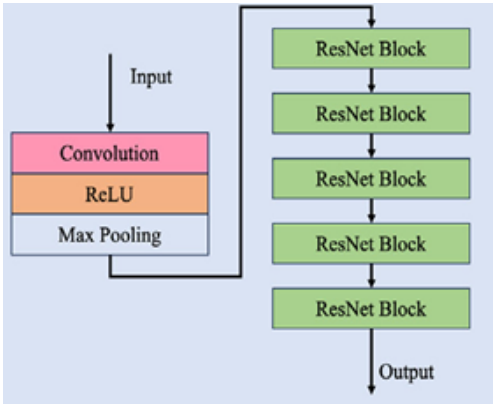


Figure 1.17: The architecture of the residual network based neural network as employed within [34]

Based on the residual network component, [34] constructed Figure 1.17, for battery SOH point and uncertainty predictions. The underlying network which is described within the figure, is similar to the CNN approach which was taken within Section 1.5.1, with the exclusion of the residual layers build around the original sub-groups of the network. The model was trained, validated (k-fold), and tested on the charging and capacity values within the NASA dataset.

Although a new method was proposed by the authors of this report, other contributions were made, which are more valuable compared to the model architecture itself. Firstly the authors acknowledge the variation which may exist between batteries. Not all batteries operate or behave the same, even if the battery type and operating conditions are the same. As a result the authors reiterated the importance of integrating uncertainty analysis into the framework. The uncertainty analysis procedure will be highlighted within Section 1.6.

1.6. Uncertainty Estimation Methods

Within the previous sections a range of methods where presented through which SOH can be determined. In practical applications many methods are infeasible, therefore measurements are used to infer a batteries SOH. Although numerous publications have achieved high accuracy for this estimation problem, the results remain approximations or predictions, which is furthermore presented by the maximum errors which still may occur for well trained methods. For this reason it is vital highlight the uncertainty of any given method, the importance of this becomes especially relevant for safety critical systems [6, 5]. Since uncertainty quantification is not a straightforward process, this section is fully devoted to the topic.

Uncertainty quantification involves two types of error, namely aleatoric and epistemic uncertainty. Epistemic uncertainty, also frequently referred to as reducible uncertainty, is a type of error which can be attributed to the lack of knowledge about the data, model selection, and data quality. While on the other hand Aleatoric uncertainty, also defined as irreducible error, arises from the principle that the problem is an approximation problem, and there thus is intrinsic unpredictable uncertainty engrained within the data. Books such as *”Introduction to statistical learning”* by [27] provide a nice mathematical formulation to describe this uncertainty [27, 49, 50, 51].

$$
\begin{aligned}
E\left[(y-\hat{y})^2\right] &= E\left[(f(x)+\epsilon-g(x))^2\right] && \text{(inserting Equation 1.4)}\\
&= E\left[(f(x)-g(x))^2-2\epsilon(f(x)-g(x))+\epsilon^2\right]\\
&= E\left[(f(x)-g(x))^2\right]-2E\left[\epsilon(f(x)-g(x))\right]+E\left[\epsilon^2\right]\\
&= E\left[(f(x)-g(x))^2\right]+var(\epsilon)+\left(E\left[\epsilon\right]\right)^2\\
&= \underbrace{E\left[(f(x)-g(x))^2\right]}_{\text{Epistemic}}+\underbrace{var(\epsilon)}_{\text{Aleatoric}}
\end{aligned}
$$

Within the field of uncertainty quantification the main goal is to be able to accurately represent and quantify the two errors, described above. Within literature numerous approaches have been developed for application within deep learning, ranging from simple implementations to complex formulations. Based on literature the popular methods can be summarised, where the main goal is to predict a mean ($\hat{\mu}$) and the standard deviation ($\hat{\sigma}$) or a variant of it, such that the distribution can be reconstructed.

To be able to get insight into the above mentioned uncertainty numerous different methods exist. Bayesian neural networks (BNN), are typically regarded as a more complex form of neural network. Here instead of using the traditional input output neurons, entire distributions (probability density functions) are utilised, to predict the distribution (posterior) from which the actual data is sample, based on a certain assumed prior distribution. Even though the BNN are able, to accurately represent uncertainty. They are often complex and expensive to train. In contrast two popular methods which are frequently utilised due to their ease of implementation, are Monte Carlo dropout (MCD) and neural network ensemble learning [52].

MCD was initially proposed by [53] as a measure to combat over fitting during training. However, more recently researchers [54] have also found that during testing this drop out layer can also provide good estimates of uncertainty. This is achieved through the use of drop out layers, which are incorporated after every dense layer. Within said dropout layers, neurons will randomly temporarily be inactive within the network, and as a result it can temporarily not contribute. Due to its ease of implementation it is particularly favoured over other more complex methods. However it exhibits poor performance for out of distribution points, and the results can often vary based on the selected hyper-parameters [52].

Quantifying uncertainty through the use of the Ensemble learning methodology [55], similarly to MCD, is a relatively simple technique to implement. Within literature two distinct version of the technique may often be found. To predict uncertainty, a multiple number of neural networks are trained, based on different training data, sampled from the same distribution. Firstly instead of making point predictions, an estimate is made on the mean and standard deviation, similar to Equation 1.12. The results are afterwards averaged over the multiple trained neural networks [52]. Another variant of the technique is often referred to as the bootstrap technique. Here no predictions are made on the mean, and standard deviation. Rather through the different predictions being made by the different neural networks of the same point, the uncertainty con be constructed in a standard manner, as summarised in [56]. Regardless ensemble learning suffers from the same shortcomings compared to MCD, combined with the inherent computational expense of training multiple neural network models [52, 56].

1.6.1. Quantile and Expectile Regression

Quantile regression has emerged as new technique to estimate uncertainty, through the use of a flexible framework. Quantile regression is by no means a new technique, it relies on a 2 part expression, to be enable estimate of quantiles, which on there turn can be used to reconstruct a distribution. The governing to be minimised equation can be expressed as Equation 1.13 based on the comprehensive analysis of the technique which has been conducted within [57]. Within the equation α refers to the α^{th} quantile ($\in [0,1]$), g refers to the regression model, y describes the labels, and $\hat{y}$ refers to the made predictions,

$$
\mathcal{L}\left(y, \hat{y}_\alpha \sim g(x)\right) = \sum_{i=1}^{N} w_{i,\alpha}|y_i - \hat{y}_{i,\alpha}|, \qquad \text{where: } w_{i,\alpha} = \begin{cases} 1-\alpha & y_i - \hat{y}_{i,\alpha} < 0 \\ \alpha & y_i - \hat{y}_{i,\alpha} \geq 0 \end{cases}. \tag{1.13}
$$

Using Equation 1.13 as a base, the researchers within [49] developed a procedure through which quantile regression could be modified for use within deep learning. The prosed method, named simultaneous quantile regression (SQR),

performance the minimisation across all different quantiles at once. To then attain a regression model which is able to model uncertainty through the prediction of $\hat{y}_{i,\alpha}$. Here $\hat{y}_{i,\alpha}$ is a function of $g(x)$ for a specific quantile. Note that in comparison to Equation 1.13 the authors of SQR use the mean, over varying quantile levels:

$$g^* \in \arg\min_g \left[\frac{1}{NK} \sum_{i=1}^{N} \sum_{\alpha \in \{0,\dots,1\}}^{K} w_{i,\alpha} |y_i - \hat{y}_{i,\alpha}| \right], \quad \text{where: } w_{i,\alpha} = \begin{cases} 1-\alpha & y_i - \hat{y}_{i,\alpha} < 0 \\ \alpha & y_i - \hat{y}_{i,\alpha} \geq 0 \end{cases}. \tag{1.14}$$

One of the main benefits of the technique is that in principle it can be utilised to predict specific quantiles, without enforcing a certain distribution. This allows for the distribution to be reconstructed within the post processing phase. The authors described the method to deliver well calibrated results, while ensuring the predicted distribution are monotone (not guaranteed when employing the standard quantile loss formulation)[49]. Based on the research above, two recent researcher papers were constructed applied to battery SOH analysis. In both cases the more traditional quantile loss term was used. The authors of [34] used the traditional pinball loss (Equation 1.13), in combination with a residual neural network. While in [45] a similar approach was taken, instead relying on a bi-directional LSTM in combination with a CNN.

Alternatively as also mentioned by [57], there also exists a technique referred to as expectile regression (Equation 1.15). The formulation is constructed in a near identical fashion, however now a the square of the residual is taken (also referred to as L2) instead of the L1 method use previously [57]. Note that within the equation an expectile is predicted, which is referred to as $\hat{e}$ within the expression. Based on analysis conducted by the same authors it was found that expectile regression exhibits a better accuracy within the tails a distribution. The method however has not been applied yet within the field of battery uncertainty estimation. The loss function to determine the conditional expectiles is defined as:

$$\mathcal{L}\left(y, \hat{e}_\alpha \sim g(x)\right) = \sum_{i=1}^{N} w_{i,\alpha} \left(y_i - \hat{e}_{i,\alpha}\right)^2, \qquad \text{where: } w_{i,\alpha} = \begin{cases} 1-\alpha & y_i - \hat{e}_{i,\alpha} < 0 \\ \alpha & y_i - \hat{e}_{i,\alpha} \geq 0 \end{cases}. \tag{1.15}$$

1.7. Performance Evaluation

One of the most important parts of creating any model, is evaluating its performance, and most importantly its trustworthiness. This section is divided into two parts. First point prediction metrics are discussed in Section 1.7.1, afterwards uncertainty accuracy is discussed in Section 1.7.2.

Typically within deep learning it is common to divide the data set ($\mathcal{D}$), into a non overlapping training ($\mathcal{A}$), validation ($\mathcal{B}$) and testing ($\mathcal{C}$) set:

$$\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \mathcal{D}, \qquad \mathcal{A} \cap \mathcal{B} = \mathcal{A} \cap \mathcal{C} = \mathcal{B} \cap \mathcal{C} = \{\varnothing\}.$$

Here both the training and validation set are used heavily during development. While the test should ideally only be used for final reporting. As would be expected the methods within this section can be applied on any of the mentioned subsets. In this case the arbitrary or generic subset used below, is referred to as $\mathcal{X}$ [27].

1.7.1. Point prediction metrics

Below metrics commonly used within literature can be retrieved [29, 30, 34, 19, 27]. Here N refers to the amount of data points, and $\bar{y}$ describes the mean.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{1.16}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{1.18}$$

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2} \tag{1.17}$$

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{y} \tag{1.19}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{1.21}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1.20}$$

$$MAX = max\left(|y_i - \hat{y}_i|\right) \forall i \in \{1, ..., N\} \tag{1.22}$$

1.7.2. Uncertainty prediction metrics

In comparison to the point predictions metrics, the evaluation of uncertainty predictions generally requires a different procedure. An efficient manner to asses the accuracy of the predicted uncertainty is through the creation of calibration curves. The goal of the calibration curve is to present how successful the model predicts the confidence intervals, by providing insight into its over or under confidence [52].

To create the calibration chart, a three stage procedure was outlined in [52]. Within the report the authors main goal was to provide engineers with a framework and foundation through which uncertainty analysis could be integrated within their process. First the domain between 0 and 1, should be discretised into K equidistant confidence levels (expected or reference confidence):

$$C^K = [c_1 = 0, c_2, ..., c_{K-1}, c_K = 1].$$

For each of the mentioned points, a ”closed” confidence interval can then be constructed according to the generalised Equation 1.23:

$$CI_i^c = \left[F_i^{-1}\left(\frac{1-c}{2}\right), F_i^{-1}\left(\frac{1+c}{2}\right)\right] \qquad \forall c \in C^K, \forall i \in \mathcal{X}. \tag{1.23}$$

Note that this procedure is repeated for each prediction (i), and F denotes the cumulative distribution function (cdf). In certain conditions, it may not be feasible or meaning full to use the two sided confidence interval around the mean. For example in the cases of a lower bound, or left tail of a distribution prediction. For these cases the authors described the one-sided confidence interval (Equation 1.24) as a useful alternative:

$$CI_i^c = \left(-\infty, F_i^{-1}(c)\right] \qquad \forall c \in C^K, \forall i \in \mathcal{X}. \tag{1.24}$$

Note: In case the quantile or expectile regression method was utilised, and no distribution was reconstructed, these selected interval should be equivalent to the ones chosen for the output domain. Alternately methods which do not explicitly make predictions for each quantile, and thus form a distribution, can freely select a desired amount of interval. This latter mentioned procedure is the one most typically encountered within research.

Next utilising the models predicted uncertainty and the labels (validation/ test), the number of labels failing within the confidence interval, relative to the total amount of labels can be computed for each interval. This parameter will be referred to as $\hat{c}$, and express the models confidence relative to the original data, based on the models predict results ($\hat{y}$). Equation 1.25, represent the procedure in equation form:

$$\hat{c}_j = \frac{1}{N}\sum_{i=1}^{N}\delta_{y_i \in CI_i^c} \qquad \forall c \in C^K, \forall j \in \{1, ..., K\}. \tag{1.25}$$

Here N refers to the total amount of label data. In typical fashion this normally should be a subset of the data, different from the original training data. However it may also be possible to create a calibration plot based on the origan training data. $\delta_{y_i \in CI_i^c}$ is a binary value equal to 0, when the label is not part of the predict confidence interval, and 1 when it is.

The calibration curve can now be constructed by relating the expected or reference confidence (c) to the confidence observed within the model($\hat{c}$). This can be done for each of the predefined K confidence levels. Within Figure 1.18 the procedure is summarised and an example result is provided, constructed by the same authors.

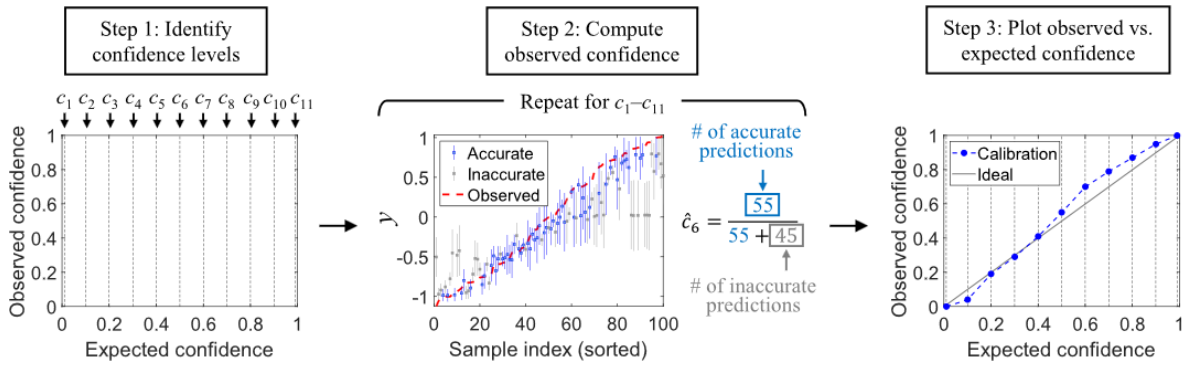


Figure 1.18: Three stage procedure to constructing calibration plots [52]

Within Figure 1.18 under-confidence can be recognised, when the chart is above the diagonal. This means that the observed confidence is broader that the expected confidence. For overconfident models, to opposite behaviour can be recognised.

Similarly to point prediction methods, similar metrics exist for uncertainty evaluation. The same authors, describing the procedure of constructing the calibration plot, described the expected calibration error (ECE). The ECE is based on the sum of the residuals between the expected and observed confidence, for each of the K confidence levels [52]. The ECE can be computed using:

$$ECE = \frac{1}{K}\sum_{i=1}^{K}|\hat{c}_i - c_i|. \tag{1.26}$$

The above presented metric relies on the creation of the calibration plot, such that the required elements can be retrieved for the evaluation of the ECE. Alternatively similar to the point prediction metrics, similar metrics exists for the evaluation of confidence intervals. These more ”simple” metrics are especially useful as a standardised approach for comparison. For example Equation 1.27 describes the sharpness metric, which is purely related to the predicted standard deviation [29],

$$Sh = \frac{1}{N}\sum_{i=1}^{N}\hat{\sigma}_i. \tag{1.27}$$

In [56] a range of prediction interval (PI) based metrics were presented. The formation of these metrics was vital for the creation of generic uncertainty aware deep learning model framework in [58]. Similarly to the quantile regression approach which was highlighted in Section 1.6.1 and used in [34], the loss function (here described as metric), was used to extract the models uncertainty. Firstly the prediction interval coverage probability (PICP) is equivalent to Equation 1.25, however typically within this cases a singular interval is chosen and presented. The metric for the width of the interval can be constructed using Equation 1.28, referred to as MPIW (mean width of the prediction interval). Here UB and LB refer to the upper and lower bound of the prediction, again using a predefined chosen interval,

$$MPIW = \frac{1}{N}\sum_{i=1}^{N}(UB_i - LB_i). \tag{1.28}$$

2

Formal Research Proposal

In this chapter a stand-alone formal version of the initial research proposal is provided. First in Section 2.1 a detailed introduction is provided into the topic of battery SOH estimation. Afterwards in Section 2.2 the main research questions are defined, followed by the methods and tools in Section 2.3. The proposal is concluded with a initial planning in Section 2.4 and a brief conclusion in Section 2.5. Additional planning information such as the work breakdown structure and Gantt chart are provided in Section 2.6, while the overall model pipeline can be viewed in Figure 2.1. A shortened version of this research proposal including only the research questions is included in Chapter 2.

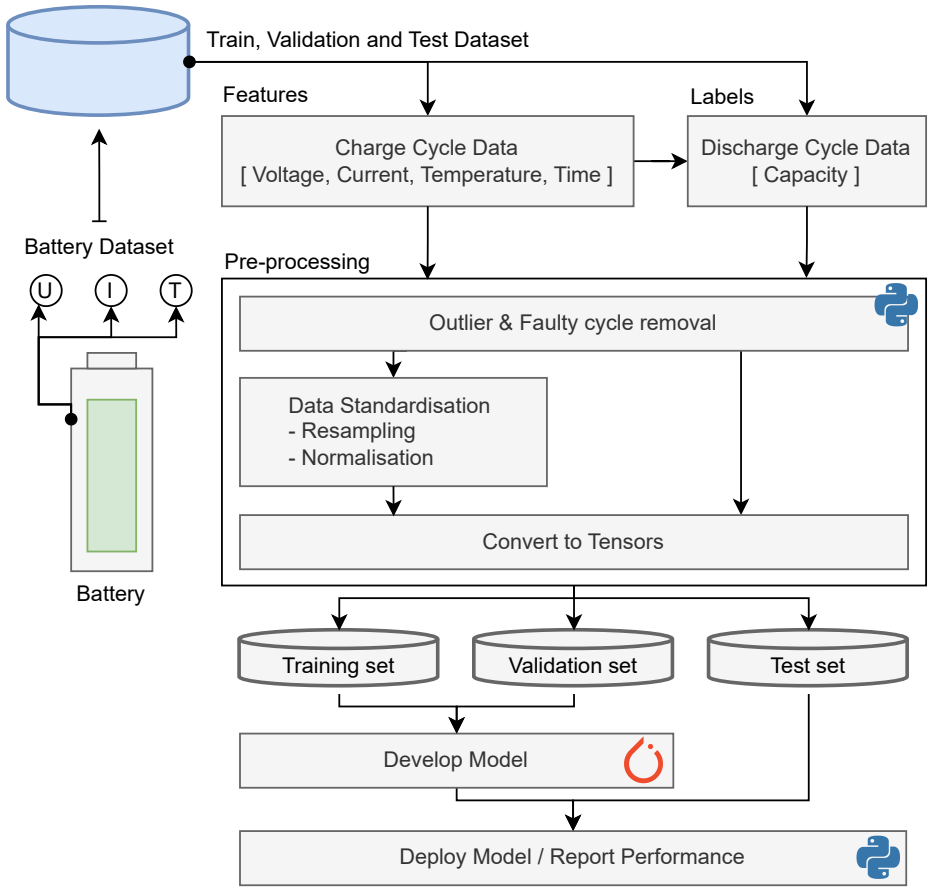


Figure 2.1: Proposed model pipeline for the battery state of health estimation model.

2.1. Introduction

Globally significant effort is being done to transition away from more traditionally used energy sources, such as oil and natural gas, to more sustainable production methods such as wind and solar. Numerous countries around the world have exhibited clear interests and are implementing targets to become more sustainable, and reduce their CO_2 emissions. This being formally coordinated by the climate targets set within the Paris agreements [1].

These efforts can be seen across various industries, where immense actions are being undertaken to electrify current practices and become more sustainable. The energy production industry has shown a large steady growth of both the rated solar and wind capacity [2]. While in other industries such as the transport industry, a transition to electrified drivetrain systems or battery electrical vehicles (BEV), can be recognised. In the automotive industry strong steady growth has been demonstrated. According to [3], this electrical vehicle market is projected to reach a 1 trillion (US) dollar market volume by the end of the decade. Although on a more limited scale, a similar desire to innovative exist in the aviation industry. Among many other interesting projects and ideas, strong interest has been developing into electric aircraft. With numerous companies such as *Airbus*, *Boeing*, *Embraer*, and *Elysian* exhibiting clear interests into the topic. However numerous challenges remain in place, delaying the transition. One of these challenges, especially relevant in a safety critical systems, such as the ones in aviation. Is the development of safe, reliable, and uncertainty aware battery management systems (BMS) [4, 5].

Regardless of their application and industry, Lithium-ion batteries have shown to be a popular candidate, due to their favourable cost and energy density figures [8]. The battery management system (BMS), plays an important role into both the safe control and operations of the batteries. The BMS, is in direct contact with the individual batteries in a battery pack. Through sensors it is able to receive information such as current (I), voltage (U), and temperature (T), which it then utilises to effectively and safely operate the battery pack. Another responsibility of the BMS is state estimation, providing the user and system with information about the batteries current health status [4]. When considering the battery prognostic side of the BMS, there are generally thee main analytical tasks, providing information into the general performance and health of the battery. These three metrics are defined as follows: state of charge (SOC), state of health (SOH), and remaining useful life (RUL). The SOC, is a measure for the amount of energy which is currently in the battery. While the SOH, refers to the total chargeable capacity relative to the rated capacity, taking into account battery degradation. Lastly the RUL refers to the amount of time, often described as the amount of charge-discharge cycles, until the batteries SOH drops below 80% of the rated capacity [8, 12, 16].

If we now focus on battery SOH estimation we observed that, significant effort has been devoted into the topic across a wide range of different domains and disciplines (battery electrical storage systems, electrical vehicle, etc...). Battery capacity or battery SOH is a property which cannot be measured directly, therefore numerous researchers and companies are contributing to the development of state-of-the-art estimation methods. Typically research contributions can be summarised into two main categories, experimental approaches and model based techniques [8, 16]. Experimental approaches are rarely practical during operation. On the other hand model driven approaches can produce highly accurate results, however they often rely on expensive and complex formulation, making it difficult to capture the intricate nature of a battery's internal chemical mechanisms [26]. As a results the model driven approaches relying on machine or deep learning have become popular, due to their ability to learn and extract deep features from measurements [34].

Battery SOH estimation is a multi-disciplinary research topic in which numerous disciplines are involved: Chemistry, Physics, Mathematics, and Informatics. Therefore it is important to state that the focus of the research performed will not be to develop a new deep learning framework, nor will it be to develop a novel electrochemical/ physical model to asses battery degradation. Significant research has been devoted into the development of SOH estimation methods through the use of machine learning. However these research papers often neglect or are unable to define the uncertainty their predictions [6, 5, 29]. This uncertainty quantification aspect is especially relevant for safety critical systems in aerospace. Since failure to provided accurate information on the capabilities and performance of any system can have enormous safety implications. Therefore the contributions of this research will be, the development and integration of existing techniques into a framework, suitable for SOH estimation. During the development process of said framework additional emphasis is put on lower bound estimation, and uncertainty quantification and evaluation. This allows for better understanding of the capabilities of the estimation model, considering application in the domain of aerospace.

The remaining sections in this proposal will threat the proposed research questions, in Section 2.2. The required methods, tools and results are introduced in Section 2.3 followed by the planning framework in Section 2.4. The report will be finalised with Section 2.5, in which the proposal conclusion will be provided.

2.2. Research Questions

As previously highlighted the research objective is to estimate the uncertainty of battery SOH predictions through the development of uncertainty-aware deep learning techniques. Therefore the primary research question can be defined as follows: *How can the uncertainty of lithium ion battery state of health be accurately predicted through the application of deep learning methods?*

The following sub-questions where created, to further assist the project thesis stated above.

1. How effective is a convolutional neural network (CNN) in predicting the capacity of a lithium-ion battery, in varying ambient conditions?
2. How can the accuracy of uncertainty predictions, made through the use of a CNN and expectile regression be evaluated?
3. How representative and accurate are the lower bound prediction interval of the predicted SOH, made through the use of a CNN and expectile regression?
4. How do the uncertainty predictions compare to other standard uncertainty quantification methods?

2.3. Method, Tools, and Expected Results

Review papers such as the one performed by researchers in [8] or [16] nicely highlight the advancements and the wide variety of methods which have been developed, both from a model and data driven point of view. More recently researchers have developed deep-learning methods, and achieved promising results, however lacked uncertainty quantifications. For example within [26], researchers proposed a deep convolutional neural network (CNN). Here features where recognised in automated manner by the model, relying solely on discretised current, voltage, and charge measurement taken during the full charging protocol. Alternatively the authors of [19], created a hybrid neural network architecture, similarly relying exclusively on measurements made during charging. In comparison to previous research, the authors proposed a CNN in combination with a long short term memory (LSTM), a type of recurrent neural network (RNN). Strong results were obtained by both authors, when evaluating their model on a dataset published by NASA (described in Table 2.2). Lastly [40] made use of only an LSTM within their architecture, for SOC predictions. With this paragraph we mainly want to highlight that a tremendous amount of work has been performed on the development of these deep learning models for engineering maintenance and prognostics. The models have become incredibly attractive within the field, due to their ability to extract high quality information from input data [33]. Therefore the research discussed here will serve as strong foundation for the research presented here.

Uncertainty quantification is an increasingly important aspect of deep learning. In recent years it has gained attention, with numerous high quality researcher papers being fully devoted to the topic. The methods proposed within these papers differ greatly in complexity, effectiveness, computation expense, and applicability. Therefore selecting the most applicable method is vital during the development of a framework [52]. Generally two types of uncertainty are mentioned, aleatoric and epistemic uncertainty. Where aleatoric refers to irreducible error, due to intrinsic uncertainty (e.g. noise) within approximation algorithms. Whereas epistemic or reducible error, refers to the lack of knowledge about the data [49, 27, 52]. We found that the approach presented in [49] was particularly attractive, since the two type of uncertainties are investigated separately and high quality results were presented. The authors proposed simultaneous quantile regression (SQR) and orthonormal certificates (OC), for the estimation of aleatoric uncertainty and epistemic uncertainty respectively. Integrating the previously outlined framework, with expectile regression highlighted in [57], the aim is to use a technique capable of delivering well calibrated and trustworthy uncertainty for battery SOH estimation.

To the best of my knowledge currently only the recent papers published by [34] and [45] have made use of quantile regression. However little analysis into lower bound of the predictions was performed, and epistemic uncertainty was not evaluated. The concept of expectile regression to my best belief has not been applied in the concept of battery SOH. Therefore through the use of the by literature inspired feature extractor, in combination with the uncertainty components, we aim to contribute to current research. The main development process can be highlighted as follows, and is described further below.

1. Data Gathering
2. Pre-processing
3. Model Implementation and Development
4. Model Training and Evaluation
5. Post-processing
6. Discussion of Results and Key Metrics

The most vital component to any machine or deep learning model, is acquiring a large amount of high quality data. In Table 2.2 we have summarised three main dataset sources, which we aim to consider in our research. Next the aim is to perform a high level analysis on the data, to get a general idea of the dataset characteristics. This step also includes the data cleaning and transformation, such that it can be used in Python (see Table 2.1). Next the main development phase is initiated, which includes the development and assessment of the model (step 3, 4). First the training and evaluation architecture of SQR and OC's are implemented. Afterwards the main model is trained and evaluating on their respective datasets. This step can be summarised as an iterative and dynamic step, in which different feature extractors are evaluated, to find the most suitable and effective configuration for the chosen 2-part uncertainty method. Finally the model performance is documented on a unique test, where both the point and uncertainty predictions are investigated trough specialised metrics.

Table 2.1: Python packages which will be utilised during the development of the model.

Python == 3.13.2

Package	**Description**	**Package**	**Description**
Numpy	Scientific library primary offering matrix and matrix operation support.	Pandas	Scientific library offering built-in data analysis functionality and support.
Matplotlib + Seaborn	Data visualisation library, in combination with the styling and statistical presets from seaborn.	Scikit-learn	Prebuild implementations of a range of standard machine learning implementations.
Pytorch	Tooling for the development of advanced deep learning models, offering built-in CUDA support.	Scipy	Scientific library offering pre build implementations of a range of algorithms.
Ipython	Python development through the use of Jupyter notebooks.		

Table 2.2: Required data and expected results.

Data_id	**Format**	**Information**	**Source**
NASA	.mat	Two datasets containing charge and discharge measurements (Capacity, Temperature, Voltage, & current) taken during a cyclic experiment on a single battery cell. A multiple amount of experiments can be retrieved within the data set, where conditions such as ambient temperature, and discharge procedure are altered. A constant current constant voltage charge procedure is utilised, and a constant current discharge procedure is used. Due to its popularity within literature this battery set was chosen, such that performance comparisons can be made [17].	[13, 14]
Toyota	.mat	Dataset containing charge and discharge measurements taken during a cyclic experiment on a single battery cell. A multiple amount of experiments can be retrieved within the data set, where the fast charging procedure is altered, at a constant ambient temperature. This dataset does not appear frequently within literature, therefore it would be valuable to investigate the models ability to deal with a different charge protocol [17].	[15]
SNL	.xslx	Dataset containing charge and discharge measurements taken during a cyclic experiment on a single battery cell. The same charge procedure as the NASA dataset is used, however now 3 different cell chemistries are tested. Additionally the discharge current, temperature and depth of discharge are varied. This dataset is useful to investigate if the model is able to learn and generalise well to a diverse set of operating conditions [17].	[12]
Results	N/A	The results of project will be twofold. Firstly a trained, validated and tested model will be achieved. Secondly based on the results a range of different analysis may be performed.	

2.4. Planning

The research phase is divided into 4 distinct phases, each having their own respective milestones. In Table 2.3 the key dates can be consulted, while within Figure 2.2, and Figure 2.3 the Gantt-chart and WBS can be viewed. For each of the phase an internal deadline, is set 1 week before the milestone. The main deliverables in the literature review phase (7 weeks) are the submission of the research proposal, literature review chapter and a presentation. In the first half of the first week of this period, time is made available to perform project project management and logistic related tasks. Afterwards the remaining time is scheduled for the actual literature review and the write-up. During the literature review research into the following 6 items will be performed: Market research, Battery types, Battery datasets, State of health assessment methods, Uncertainty quantification methods and Neural networks.

Research phase I (10 weeks), is nearly entirely devoted to model development (8 weeks). Here the model will be developed according to the previously highlighted steps, and initial results should be collected. It should however be highlighted that in the model development phase, first a ”standard” deep learning model will be created. Upon successful completion of the model, the expectile regression technique will be integrated. Furthermore an initial version of the methodology chapter should be completed at the end of this phase. Since the topic of deep learning is currently still new to me, the first 2 weeks of this period will be fully devoted to learning more about machine learning and deep learning. Additionally this time could also be utilised to implement feedback from the literature review phase.

Research phase II (14 weeks), will be a continuation of the previous phase, in which a 2 week break and 2 week contingency is added. The break is currently schedule from the July 7th to July 20th, while the 2 week contingency is added at the end of the phase. The initial schedule is to have the entire base code development aspect of the project finished before the break. After the break time is made available for model benchmarks, validation and verification (2 weeks) and the draft thesis/ deliverable will be prepared (4 weeks). Contrary to the regular 1 week internal deadline, in this cases the deadline is schedule 2 weeks before the end of phase. Here the first version of the thesis and a presentation should be made, furthermore a plagiarism report should also be submitted. The green light review is scheduled in the week of 21th of September.

Lastly the completion of the master thesis is currently scheduled for the week of 2nd November. Two critical deadlines here are the submission of an examination request, 4 weeks to the graduation. Followed by the submission of thesis documents (thesis, data source & model) on the TuDelft repository. For the graduation itself the presentation should be created and prepared. Furthermore any related documents and planning (reserving a room, finding a committee), should be done well in advance.

Table 2.3: Schedule, where key milestones are indicated

Phase	Start	End	Deliverable	Duration
Literature Study	17-02-2025	06-04-2025	Research proposal, Literature review chapter, Presentation	7 weeks
Research Phase I	07-04-2025	15-06-2025	Initial results, Methodology chapter, Presentation	10 weeks
Research Phase II	16-06-2025	21-09-2025	Thesis, Presentation, Plagiarism report	14 weeks
Research Dissemination	22-09-2025	02-11-2025	Examination request, Repository, Thesis, Presentation, Graduation	6 weeks
				Total: 37 weeks

2.5. Conclusion

Lithium-Ion batteries have seen an incredible increase in popularity, due to their favourable favourable cost and energy density figures. Currently they are being applied within a range of domains, offering a more sustainable alternative to traditional systems. One of the key problems however, is the uncertainty of their capacity which is dependent on degradation over time. Within this report the need for such uncertainty aware prediction methods was highlighted, and various state of the art research topics were presented. Afterwards a deep learning framework is proposed relying on expectile and quantile regression, to be able to make informed estimates of the uncertainty of the battery state of health.

2.6. Supplementary Planning Material

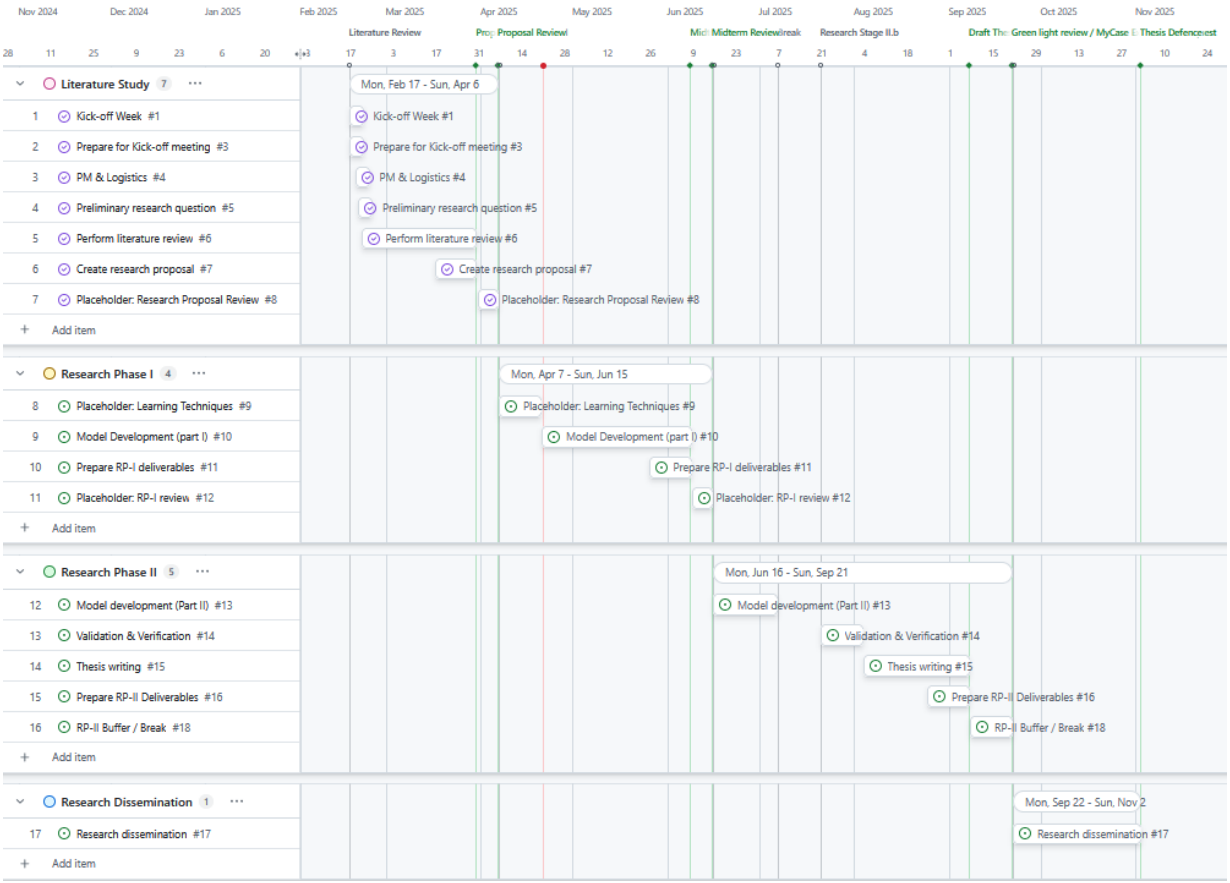


Figure 2.2: project Gantt chart

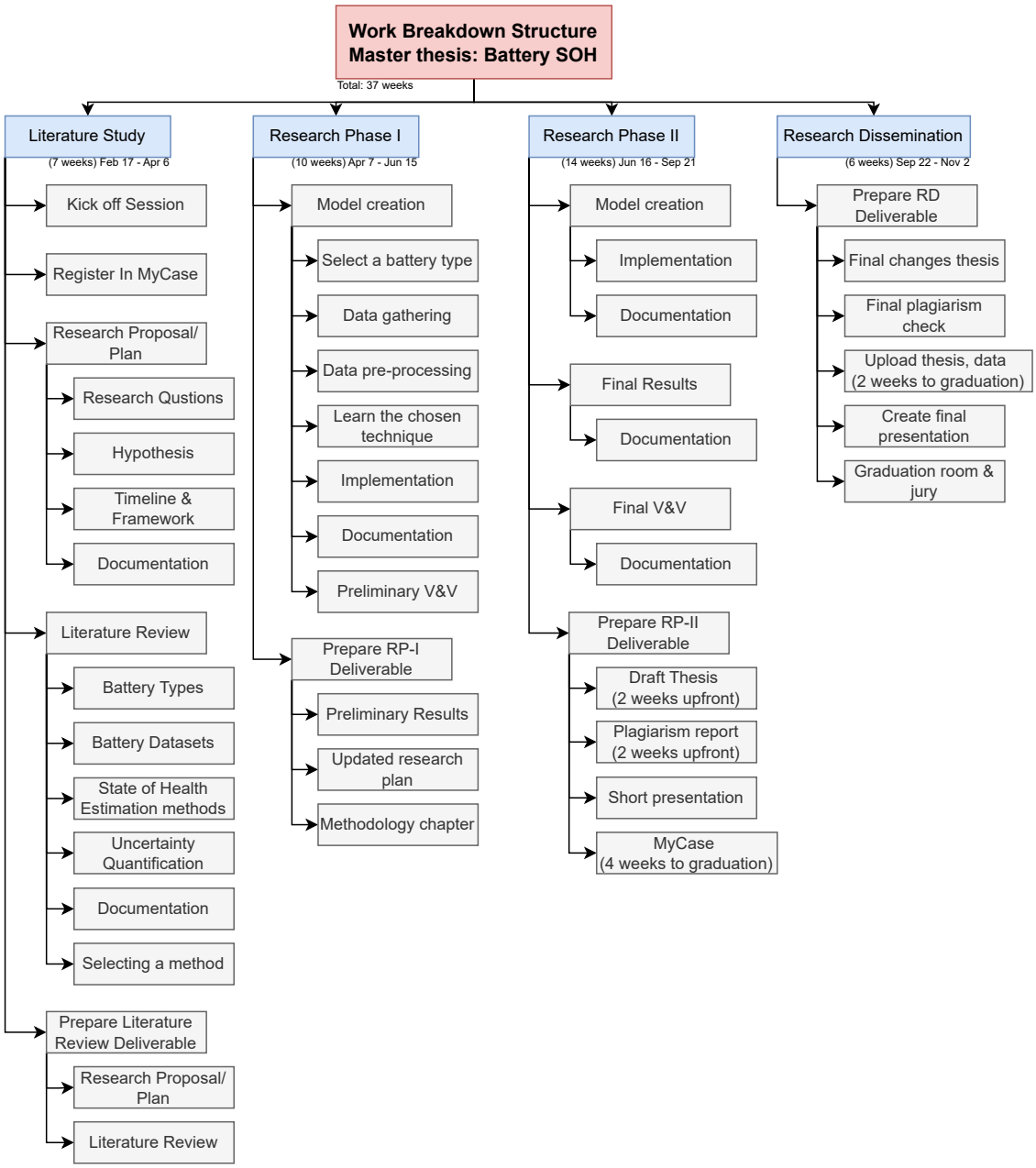


Figure 2.3: Project work breakdown structure

3

Research Questions

In the previous chapter, various topics such as battery SOH estimation methods and uncertainty quantification in deep learning where introduced. We observed that both machine and deep learning are attractive and applicable within the domain, with numerous researchers developing increasingly sophisticated techniques. However comparatively little effort and analysis has been performed in assessing the uncertainty of these deep learning models. This is an increasing important topic, especially when considering lithium-ion batteries as a potential sole power supply for aircraft. Previous research has mostly focussed on either developing deterministic point prediction models and visualising uncertainty through the use of classical machine learning. More recently a few research papers, although limited, have investigated the topic from a deep learning point of view through the development of novel estimation methods. However limited assessment of the achieved uncertainty was performed. Within this research we therefore aim to integrate existing SOH estimation techniques, with novel uncertainty estimation methods. Followed by a rigorous assessment of the produce of the produced uncertainty using the deep learning model.

The main research objective and research question can therefore be stated as follows:

Research Objective

The research objective is to better understand the lower bound of battery state of health estimations.

Research Question

How can the uncertainty of lithium ion battery state of health be accurately predicted through the application of deep learning methods?

In addition to the above mentioned research question, the following sub questions were constructed.

1. How effective is a convolutional neural network (CNN) in predicting the capacity of a lithium-ion battery, in varying ambient conditions?
2. How can the accuracy of uncertainty predictions, made through the use of a CNN and expectile regression be evaluated?
3. How representative and accurate are the lower bound prediction interval of the predicted SOH, made through the use of a CNN and expectile regression?
4. How do the uncertainty predictions compare to other standard uncertainty quantification methods?

References

[1] U. N. F. C. on Climate Change (UNFCCC). *Paris Agreement*. United Nations Treaty Collection. Adopted at COP 21, Paris, France, 12 December 2015. Entered into force on 4 November 2016. 2016. URL: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement.

[2] L. Clarke et al. "Energy Systems". In: *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by P. Shukla et al. Cambridge, UK and New York, NY, USA: Cambridge University Press, 2022. DOI: 10.1017/9781009157926.008.

[3] statista. *Electric Vehicles - Worldwide*. Online. Accessed: March 17, 2025. n.d. URL: https://www-statista-com.tudelft.idm.oclc.org/outlook/mmo/electric-vehicles/worldwide.

[4] H. Rahimi-Eichi et al. "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles". In: *IEEE Industrial Electronics Magazine* 7.2 (2013), pp. 4–16. DOI: 10.1109/MIE.2013.2250351.

[5] A. Thelen et al. "Probabilistic machine learning for battery health diagnostics and prognostics—review and perspectives". In: *npj Materials Sustainability* 2.1 (June 2024), p. 14. DOI: 10.1038/s44296-024-00011-1.

[6] F. Wang et al. "Explainability-driven model improvement for SOH estimation of lithium-ion battery". In: *Reliability Engineering & System Safety* 232 (2023), p. 109046. DOI: https://doi.org/10.1016/j.ress.2022.109046.

[7] M. Hannan et al. "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations". In: *Renewable and Sustainable Energy Reviews* 78 (2017), pp. 834–854. DOI: https://doi.org/10.1016/j.rser.2017.05.001.

[8] H. Tian et al. "A review of the state of health for lithium-ion batteries: Research status and suggestions". In: *Journal of Cleaner Production* 261 (2020), p. 120813. DOI: https://doi.org/10.1016/j.jclepro.2020.120813.

[9] A. Barré et al. "A review on lithium-ion battery ageing mechanisms and estimations for automotive applications". In: *Journal of Power Sources* 241 (2013), pp. 680–689. DOI: https://doi.org/10.1016/j.jpowsour.2013.05.040.

[10] Y. Li et al. "Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review". In: *Renewable and Sustainable Energy Reviews* 113 (2019), p. 109254. DOI: https://doi.org/10.1016/j.rser.2019.109254.

[11] X. Han et al. "A comparative study of commercial lithium ion battery cycle life in electrical vehicle: Aging mechanism identification". In: *Journal of Power Sources* 251 (2014), pp. 38–54. DOI: https://doi.org/10.1016/j.jpowsour.2013.11.029.

[12] Y. Preger et al. "Degradation of Commercial Lithium-Ion Cells as a Function of Chemistry and Cycling Conditions". In: *Journal of The Electrochemical Society* 167.12 (Jan. 2020), p. 120532. DOI: 10.1149/1945-7111/abae37.

[13] B. Saha et al. *Battery Data Set*. Tech. rep. NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA, 2007. URL: https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/.

[14] B. Bole et al. *Randomized Battery Usage Data Set*. Tech. rep. NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA, 2016. URL: https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/.

[15] K. A. Severson et al. “Data-driven prediction of battery cycle life before capacity degradation”. In: *Nature Energy* 4.5 (Mar. 2019), pp. 383–391. DOI: 10.1038/s41560-019-0356-8.

[16] R. Xiong et al. “Towards a smarter battery management system: A critical review on battery state of health monitoring methods”. In: *Journal of Power Sources* 405 (2018), pp. 18–29. DOI: https://doi.org/10.1016/j.jpowsour.2018.10.019.

[17] G. dos Reis et al. “Lithium-ion battery data and where to find it”. In: *Energy and AI* 5 (2021), p. 100081. DOI: https://doi.org/10.1016/j.egyai.2021.100081.

[18] International Energy Agency. *Global EV Outlook 2024*. Tech. rep. Licence: CC BY 4.0. IEA, Paris, 2024. URL: https://www.iea.org/reports/global-ev-outlook-2024.

[19] Y. Fan et al. “A novel deep learning framework for state of health estimation of lithium-ion battery”. In: *Journal of Energy Storage* 32 (2020), p. 101741. DOI: https://doi.org/10.1016/j.est.2020.101741.

[20] A. Eddahech et al. “Behavior and state-of-health monitoring of Li-ion batteries using impedance spectroscopy and recurrent neural networks”. In: *International Journal of Electrical Power & Energy Systems* 42.1 (2012), pp. 487–494. DOI: https://doi.org/10.1016/j.ijepes.2012.04.050.

[21] D.-I. Stroe et al. “Lithium-Ion Battery State-of-Health Estimation Using the Incremental Capacity Analysis Technique”. In: *IEEE Transactions on Industry Applications* 56.1 (2020), pp. 678–685. DOI: 10.1109/TIA.2019.2955396.

[22] C. She et al. “Battery State-of-Health Estimation Based on Incremental Capacity Analysis Method: Synthesizing From Cell-Level Test to Real-World Application”. In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* 11.1 (2023), pp. 214–223. DOI: 10.1109/JESTPE.2021.3112754.

[23] A. Guha et al. “State of Health Estimation of Lithium-Ion Batteries Using Capacity Fade and Internal Resistance Growth Models”. In: *IEEE Transactions on Transportation Electrification* 4.1 (2018), pp. 135–146. DOI: 10.1109/TTE.2017.2776558.

[24] W. Waag et al. “Adaptive estimation of the electromotive force of the lithium-ion battery after current interruption for an accurate state-of-charge and capacity determination”. In: *Applied Energy* 111 (2013), pp. 416–427. DOI: https://doi.org/10.1016/j.apenergy.2013.05.001.

[25] F. Naseri et al. “An Enhanced Equivalent Circuit Model With Real-Time Parameter Identification for Battery State-of-Charge Estimation”. In: *IEEE Transactions on Industrial Electronics* 69.4 (2022), pp. 3743–3751. DOI: 10.1109/TIE.2021.3071679.

[26] S. Shen et al. “A deep learning method for online capacity estimation of lithium-ion batteries”. In: *Journal of Energy Storage* 25 (2019), p. 100817. DOI: https://doi.org/10.1016/j.est.2019.100817.

[27] G. James et al. *An Introduction to Statistical Learning*. Cham: Springer International Publishing, Jan. 2023. DOI: 10.1007/978-3-031-38747-0.

[28] Y. Lei et al. “Machinery health prognostics: A systematic review from data acquisition to RUL prediction”. In: *Mechanical Systems and Signal Processing* 104 (2018), pp. 799–834. DOI: https://doi.org/10.1016/j.ymssp.2017.11.016.

[29] D. Roman et al. “Machine learning pipeline for battery state-of-health estimation”. In: *Nature Machine Intelligence* 3.5 (Apr. 2021), pp. 447–456. DOI: 10.1038/s42256-021-00312-3.

[30] X. Li et al. “State of health estimation for Li-Ion battery using incremental capacity analysis and Gaussian process regression”. In: *Energy* 190 (2020), p. 116467. DOI: https://doi.org/10.1016/j.energy.2019.116467.

[31] Q. Li et al. “State of health estimation of lithium-ion battery based on improved ant lion optimization and support vector regression”. In: *Journal of Energy Storage* 50 (2022), p. 104215. DOI: https://doi.org/10.1016/j.est.2022.104215.

[32] C. Hu et al. “Online estimation of lithium-ion battery capacity using sparse Bayesian learning”. In: *Journal of Power Sources* 289 (2015), pp. 105–113. DOI: https://doi.org/10.1016/j.jpowsour.2015.04.166.

[33] Y. LeCun et al. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: 10.1038/nature14539.

[34] Y. Zhang et al. “Reliability enhancement of state of health assessment model of lithium-ion battery considering the uncertainty with quantile distribution of deep features”. In: *Reliability Engineering & System Safety* 245 (2024), p. 110002. DOI: https://doi.org/10.1016/j.ress.2024.110002.

[35] J. Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85–117. DOI: https://doi.org/10.1016/j.neunet.2014.09.003.

[36] W. S. McCulloch et al. “A logical calculus of the ideas immanent in nervous activity”. In: *The Bulletin of Mathematical Biophysics* 5.4 (1943). Cited by: 12523, pp. 115–133. DOI: 10.1007/BF02478259.

[37] I. Goodfellow et al. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[38] R. Yamashita et al. “Convolutional neural networks: an overview and application in radiology”. In: *Insights into Imaging* 9.4 (June 2018), pp. 611–629. DOI: 10.1007/s13244-018-0639-9.

[39] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[40] E. Chemali et al. “Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries”. In: *IEEE Transactions on Industrial Electronics* 65.8 (2018), pp. 6730–6739. DOI: 10.1109/TIE.2017.2787586.

[41] S. Hochreiter et al. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[42] Y. Wu et al. *Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. arXiv: 1609.08144 [cs.CL]. URL: https://arxiv.org/abs/1609.08144.

[43] G. Van Houdt et al. “A review on the long short-term memory model”. In: *Artificial Intelligence Review* 53.8 (Dec. 2020), pp. 5929–5955. DOI: 10.1007/s10462-020-09838-1.

[44] W. Wang et al. “Residual Convolution Long Short-Term Memory Network for Machines Remaining Useful Life Prediction and Uncertainty Quantification”. In: *Journal of Dynamics, Monitoring and Diagnostics* 1.1 (Dec. 2021), pp. 2–8. DOI: 10.37965/jdmd.v2i2.43.

[45] C. Chen et al. “A Lithium-Ion Battery Degradation Prediction Model With Uncertainty Quantification for Its Predictive Maintenance”. In: *IEEE Transactions on Industrial Electronics* 71.4 (2024), pp. 3650–3659. DOI: 10.1109/TIE.2023.3274874.

[46] Q. Li et al. “Probabilistic neural network-based flexible estimation of lithium-ion battery capacity considering multidimensional charging habits”. In: *Energy* 294 (2024), p. 130881. DOI: https://doi.org/10.1016/j.energy.2024.130881.

[47] X. Glorot et al. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y. W. Teh et al. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[48] K. He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[49] N. Tagasovska et al. *Single-Model Uncertainties for Deep Learning*. 2019. arXiv: 1811.00908 [stat.ML]. URL: https://arxiv.org/abs/1811.00908.

[50] M. Abdar et al. “A review of uncertainty quantification in deep learning: Techniques, applications and challenges”. In: *Information Fusion* 76 (2021), pp. 243–297. DOI: https://doi.org/10.1016/j.inffus.2021.05.008.

[51] L. Basora et al. “A benchmark on uncertainty quantification for deep learning prognostics”. In: *Reliability Engineering & System Safety* 253 (2025), p. 110513. DOI: https://doi.org/10.1016/j.ress.2024.110513.

[52] V. Nemani et al. “Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial”. In: *Mechanical Systems and Signal Processing* 205 (2023), p. 110796. DOI: https://doi.org/10.1016/j.ymssp.2023.110796.

[53] N. Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), pp. 1929–1958.

[54] Y. Gal et al. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2016. arXiv: 1506.02142 [stat.ML]. URL: https://arxiv.org/abs/1506.02142.

[55] B. Lakshminarayanan et al. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. 2017. arXiv: 1612.01474 [stat.ML]. URL: https://arxiv.org/abs/1612.01474.

[56] A. Khosravi et al. “Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances”. In: *IEEE Transactions on Neural Networks* 22.9 (2011), pp. 1341–1356. DOI: 10.1109/TNN.2011.2162110.

[57] L. Waltrup et al. “Expectile and quantile regression—David and Goliath?” In: *Statistical Modelling* 15 (Oct. 2015), pp. 433–456. DOI: 10.1177/1471082X14561155.

[58] A. Khosravi et al. “Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals”. In: *IEEE Transactions on Neural Networks* 22.3 (2011), pp. 337–346. DOI: 10.1109/TNN.2010.2096824.

Part II

Scientific Article

Estimating Aleatoric and Epistemic Uncertainty of the Battery State of Health Using Simultaneous Quantile Regression and Orthonormal Certificates

J.S. Bogaert
Aerospace Engineering, Technical University of Delft

Battery state of health (SOH) estimation is one of the three main analytical tasks of a battery management system (BMS), when viewed from engineering maintenance and prognostics perspective. With the current global effort towards more suitable and greener processes, lithium-ion batteries have shown to be an important element in facilitating this transition. One industry where this can be noticed in particular is the transportation sector, where a strong shift towards battery electric vehicles (BEV) can be observed. Within the aviation sector, current research efforts include electrical flight. However, numerous challenges remain, that are typically observable within a safety critical domain such as aerospace. One these challenges includes the determination of uncertainty in battery SOH prediction. This would provide improved transparency on the capabilities and limitation of a model, when used as part of a battery system. Within this report we propose the use of a bidirectional gated recurrent unit (Bi-GRU) with learnable soft attention, to predict battery SOH based on charge measurements. Uncertainty analysis is enabled through the use of simultaneous quantile regression (SQR) and orthonormal certificate (OC), to be able to highlight and distinguish the aleatoric and epistemic uncertainty of the proposed model. We afterwards evaluate the model for point prediction accuracy using standard metrics, and evaluate the produced uncertainty using specialised test cases and calibration metrics. We achieved strong results using the proposed framework on a 2-phase fast charging dataset published by Toyota.

Nomenclature

a	=	Attention Weight
b	=	Learnable Biases
c	=	Confidence Level
$\hat{c}$	=	Observed Confidence
C	=	Capacity
e_u	=	Epistemic Uncertainty
h	=	Hidden State
h_c	=	Context Vector
I	=	Current
l_r	=	Learning Rate
N	=	#Points
t	=	Time
T	=	Temperature
V	=	Voltage
w	=	Weight
W	=	Learnable Weights
x	=	Predictor
y	=	Target
$\hat{y}$	=	Prediction
$\bar{y}$	=	Mean of the Target
α	=	Quantile
β	=	Momentum
δ	=	Binary Indicator
ϵ	=	Error
θ	=	Model Parameters
λ	=	Regularization Factor
μ	=	Mean
σ	=	Standard Deviation
$\mathcal{D}$	=	Dataset
$\mathcal{L}$	=	Loss Function
$\mathcal{U}$	=	Uniform Distribution

I. Introduction

Batteries are a crucial part of the current global effort to transition towards more sustainable processes and methods. Across numerous industries strong efforts can be noticed. Within the energy industry itself strong growth in both solar and wind capacity is observable [1]. While in the transport sector a push towards battery electric vehicles (BEV) is noticeable [2]. Within aviation a similar transition would be desired, however numerous challenges remain. Which include, various research and development related topics, and also include operation and business related questions, such as the implications of running an electrified fleet. If we focus on battery development, one of these challenges includes the development of uncertainty aware battery management systems (BMS) [3, 4]. From the battery prognostics side generally there are three main analytical estimation problems: estimation of state of charge (SOC), state of health (SOH), and remaining useful life (RUL). SOC is focussed around determining the amount of charge or capacity currently in the battery, while SOH is a measure for the total usable capacity in fully charged state (taking into account battery degradation). Lastly RUL describes the amount of time or cycles the battery can still undergo before end of life (EOL). The latter is also often defined as the time until the SOH drops below 80% (20% degradation) with respect to "original" battery [5–7]. In this report the focus will be put on developing an uncertainty aware battery SOH estimation method.

Battery SOH is a property, which in itself cannot be measured directly. Secondly, it is a property which varies with respect to time and operational environment, irregardless of whether or not the battery is being actively used. Leading causes for this degradation are related to the complex and dynamic behaviour of the "solid electrolyte interphase" (SEI) film [8–10]. As a results battery SOH estimation has been an active domain of research, where researchers have developed numerous different approaches to infer the capacity through highly specialised approaches. The general consensus among both reviewers and researcher, is that techniques can be divided into two general groups: experimental approaches and model based techniques [5, 6]. Experimental methods, include techniques such as incremental capacity analysis (ICA) [11], counting methods, and electro impedance spectroscopy (EIS) [12]. The key downside of these methods is that they often are not applicable during operation [5], additionally as observed within the respective papers, they rely either on specific (operating) conditions or highly specialised tools.

Model based approaches aim to solve the issue posed by experimental techniques, by offering an approach more applicable to a wide range of settings. As the name suggests, here a certain parameter (SOH or SOC) is inferred through the creation and tuning of "sophisticated" models. Two popular technique in this group are equivalent circuit models (ECM) and electrochemical models (EM), which both aim to mimic the internal characteristics and processes within a battery [5]. Alternatively, there are rapid advancements and numerous successful application of machine and deep learning (ML, DL) [13] in many different fields, ranging from engineering maintenance [14] to medicine [15]. It is therefore not surprising that the technique is also being heavily researched and applied within all three of the previously mentioned battery prognostics tasks (SOC, SOH, and RUL estimation).

Present work in battery SOH estimation, through the use of ML and DL, has to a large extent focussed on investigating the applicability of hand-crafted features or automatic feature extraction methods, for determining the battery SOH. From the ML perspective, researchers in [16] presented a relevance vector machine (RVM), achieving competitive results using 5 hand-crafted features, composed from sensor measurements taken during charging. Similarly [17] presented 4 different ML models, relying on a set of 30 features, to infer battery SOH. Although not strictly a downside, ML architectures require careful section of features, which are often complex to create, and may not translate well to other datasets. DL aims to solve this issue by letting the model "learn" which elements are important when making a prediction [13]. Researchers in [18], explored this exact topic. By presenting a convolutional neural network (CNN), relying solely on charge sensor data as input, to predict the battery SOH. Similarly in [19], a method combining a CNN, and a long short term memory (LSTM) model was presented. While in [20] a gated recurrent unit (GRU) was used instead of the LSTM. In all three reports highly accurate predictive performance was achieved. One key aspect, which is comparatively researched to a lesser extent, is the uncertainty related to the models themselves. Uncertainty is a topic of paramount importance, and this importance further increases when dealing with mission critical components [4, 14, 21]. Because failure of the system, or incorrect estimation of its capabilities, could have detrimental effect on safety, and thus forms a big risk. If we consider the potential application of batteries, as a primary source of energy for electric aircraft, this further suggests the need for well calibrated and accurate uncertainty estimation. Here well calibrated and accurate is described as the ability of the model, to provided uncertainty estimates which accurately represent the uncertainty observed within the data.

Although limited, some current work has investigated this topic of uncertainty. Within [22] a ResNet was proposed, for battery SOH estimation using the quantile Huber loss function. While researchers in [19], proposed a probabilistic output layer, together with a CNN, and LSTM model. Both examples employed an automatic feature extraction procedure, and where able to successfully predict uncertainty. The authors however performed limited assessment of the uncertainty and broad uncertainty was obtained. Referring back to [17], four high quality machine learning models where created, and afterwards assessed, for their point and uncertainty predictions. The authors developed three ML based models, namely: random forest (RF), guassian process regression (GPR), and Bayesian ridge regression (BRR). Furthermore a deep ensemble using a multi layer perceptron was proposed on the deep learning side. All of the above methods relied on features which were engineered based on the constant current, constant voltage (CC-CV) part of the charging curve. After rigorous evaluation op the models, using alpha-accuracy and calibration curves, promising results where achieved on four diverse datasets, using the deep ensemble method. The paper forms a strong baseline, and further motivated our research of using deep learning for battery SOH estimation.

With this paper we therefore aim to combine previous work, in the domain of battery SOH estimation and uncertainty modelling, with deep learning. By proposing a framework capable of predicting the uncertainty with battery SOH prediction, relying purely on charge data. In this manner the capabilities of a system can de derived before its mission is performed. The contributions of this report can be summarised as follows:

1) We present a deep learning model capable of performing automatic feature extraction on varying length time series. Afterwards as output the model provides both point prediction and uncertainty estimates for the SOH of lithium ion batteries during discharge.
2) We evaluate the created model both for aleatoric and epistemic uncertainty, relying purely on sensor measurements taken during charging. We explore the effectiveness of the model, on a fast charging dataset published by Toyota.
3) We perform a thorough evaluation of the produced uncertainty estimates, through metrics commonly used in the field of prognostics and maintenance.

The remaining section of this report are divided as follows. First in section II the used methodology is described. Here four key elements are treated: data prepossessing, models, uncertainty estimation and lastly model evaluation. Afterwards in section III, based on the methods highlighted a case study is performed, relating to battery state of health. The report is finalised with the conclusion and recommendations in section IV.

II. Methodology

In the following section, the methodology utilised to obtain the results highlighted in section III is presented. First in subsection II.A a generic set of pre-processing methods is described, required for the model described in subsection II.B. Afterwards in subsection II.C the main method to quantify uncertainty will be summarized. Lastly this section is concluded with methods to asses the accuracy of a given model in subsection II.D.

A. Problem Statement and Data Preprocessing

During the charging process of any battery, typically the following three parameters are measured both during charge, and discharge: Voltage (V), Current (I) and temperature (T). The manner in which these parameters vary depends on the charging procedure, environment, and the batteries system dynamics. Essentially the battery can be considered a system, with complex internal dynamics. The goal in this study, is to use these sensor measurement collected during the charge procedure, to then make an informed prediction on the usable capacity (C) measured during discharge in a battery. Formally this task can be written as: $C_{\text{Discharge}} = f(\{I, V, T\}_{\text{Charge}})$. If we now take a step back, from an operation point of view, this would provide the operator of a battery electric vehicle (BEV), with information regarding the capabilities (range), or status (age) of the BEV, before the BEV performs its mission.

Traditionally the most common approach to charging batteries, was through the use of the constant current, constant voltage charge procedure (CC-CV) [23]. Here charging is initiated at a fixed current, until the voltage reaches a set

threshold. Upon reaching this threshold, this current gradually decreases, such that a constant voltage is maintained and overcharging is avoided [3]. More recently there has been great interest into fast charging [24], which deviates slightly from this procedure. Within section III, reference charge cycles are provided for both charging protocols, in their respective subsection. However, regardless of the approach being used, these charge cycles are inherently different in length and characteristics. Therefore, this data should first be interpreted, converted, and standardised such that it can be used as part of a DL pipeline.

To conserve temporal information we found that sampling the original sensor measurements to a variable length (t) using a fixed sample rate (Δt), was preferred over sampling to a fixed length (using a variable sample rate). We then proceed to use nearest neighbour interpolation, to obtain a discrete representation of the time series. For efficient learning purpose, and to ensure the model focuses on patterns, rather than specific values, the data (denoted as x) is normalized to $[-1, 1]$. This is done for each charge cycle and measurement type independently using:

$$2\frac{x - \min(x)}{\max(x) - \min(x)} - 1. \tag{1}$$

Finally for each cycle in the dataset ($\mathcal{D}$) the following sample (feature, label pair) is obtained:

$$\text{Features } (x_i)\text{: } \begin{bmatrix} I_1 & I_2 & \dots & I_{t-1} & I_t \\ V_1 & V_2 & \dots & V_{t-1} & V_t \end{bmatrix} \in \mathbb{R}^{2\times t}, \qquad \text{Labels } (y_i)\text{: } \begin{bmatrix} C \end{bmatrix} \in \mathbb{R}, \qquad \text{Where } (x_i, y_i) \in \mathcal{D}.$$

1. Remarks on mini-batch creation using variable length input features.

The previously described sampling strategy samples an arbitrary length signal, into a variable length vector. Although this technique results in an accurate, and unbiased sampling of the original signal, it does have as downside that samples will vary in sequence length ($t \neq$ constant, $\quad \forall x_i \in \mathcal{D}$). This design choice has two main implications; (1) many DL architectures are created with assumptions that inputs are roughly similarly in dimension, and secondly (2) mini-batches during optimization, with for example SGD or Adam, require constant length sequences within each mini-batch [25]. (1) is easy solved with specialised model architectures and is treated in more detail in subsection II.B, while (2) can be solved through zero padding across each mini-batch. For the latter it is however critical that these zero elements are considered as padding and thus properly masked, to avoid that the model learns from these terms.

B. Baseline Models for State of Health Predictions

Above in subsection II.A, it became evident that we are dealing with a popular "many-to-one" or multivariate regression problem. To the best of our knowledge, when dealing with varying length time series in deep learning, you generally have two well researched base architectures. Namely recurrent neural networks (RNN) and temporal convolution networks (TCN) [26, 27]. Historically RNNs are the more "well-known" option, with ample resources being available. They are able to process time series, by effectively passing information back to themselves and thus created a form time awareness or memory [28]. Two popular developments of the RNN are, the long short term memory (LSTM) [29] and the gated recurrent unit (GRU) [30]. Both were created to simulate a better form of memory and limit issues such at the vanishing and exploding gradient problem [28].

As a starting point of the network we therefore make use of the GRU proposed by Cho et al. [30], which can be seen in Figure 1. The illustration of the GRU can be observed both as part of a sequence and by itself, to highlight the working principle. The reason we opt in favour of the GRU over the LSTM, is primarily due to it more lightweight construction. When data is not abundant, over-fitting may pose a significant concern and a reduction in model parameters can combat this. The remaining part of this subsection aims to introduce the mathematical concepts, required for the construction of the baseline model, which will be than modified in subsection II.C to model uncertainty.

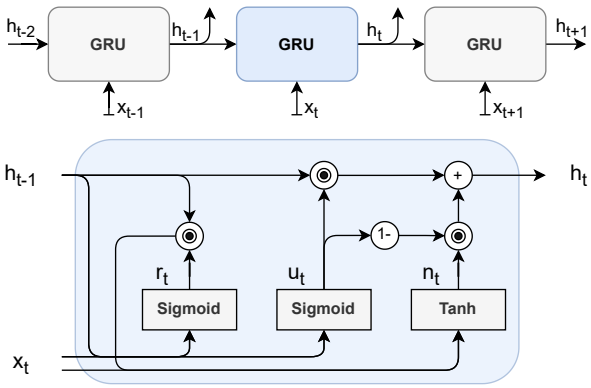


Fig. 1 Illustration of the internal cell architecture of a gated recurrent unit (GRU) and the integration into a sequence, inspired by an illustration from [31].

The GRU which is described here, is the formulation than can be retrieved from PyTorch [25] and can be visually observed in Figure 1. First the input data denoted by x_t, is provided to GRU in a sequential manner. The GRU itself is governed by 4 main equations, first based on the input (x_t) and previous hidden state (h_{t-1}), r_t and z_t can be computed:

$$r_t = \textbf{Sigmoid}\left(W_{xr}x_t + b_{xr} + W_{hr}h_{t-1} + b_{hr}\right), \tag{2}$$

$$u_t = \textbf{Sigmoid}\left(W_{xu}x_t + b_{xu} + W_{hu}h_{t-1} + b_{hu}\right). \tag{3}$$

As described in [28, 30], r_t is the "reset equation", while u_t is the "update equation". Both components control the flow of information within and thus between cells. Lastly the output or next hidden state is computed using the following two equations:

$$n_t = \textbf{Tanh}\left(W_{xn}x_t + b_{xn} + r_t \odot \left(W_{hn}h_{t-1} + b_{hn}\right)\right), \tag{4}$$

$$h_t = (1 - u_t) \odot n_t + u_t \odot h_{t-1}. \tag{5}$$

Here x_t is the input at a time step t, $W \in \mathbb{R}^{h' \times 2}$ and $b \in \mathbb{R}^{h'}$ are respective learnable weights and biases, $\odot$ refers to the element wise matrix multiplication, finally h_t refers to the hidden state or output of the GRU, at time step t and h' refers to the size of the hidden state. The subscripts for both weights (W) and biases (b), aligns with the general convention, where the first symbol aligns with the origin and the latter refers to the variable to which the new data is assigned. Practically this means that W_{xr}, is the weight matrix which is linked to the operation from x to r. The 4 equations described above, govern the information flow from the input x_t, to the output h_t, which than on its turn is used as in input hidden state for the next time step, in combination with the next time step. To conclude the GRU thus takes as input as time series: $\{x^{(1)}, ..., x^{(t)}\}$ and provides a series of hidden states as output: $\{h^{(1)}, ..., h^{(t)}\}$.

Although good accuracy can be achieved using the last hidden state ($h^{(t)}$) of the GRU component, using a (multilayered bi-direction) GRU, they may often get stuck in a sub optimal, local optimum. As expressed in [28] although GRU and LSTM have been a great improvement for sequence modelling in comparison to normal a RNN, due to the additional paths. They may still struggle to learn complex and long patterns. Adding a learnable attention mechanism (or "explicit memory" as referred to in [28]) seemed to greatly alleviate this issue for our case, on top of also providing an additional form of interpretability. We make use of a soft-attention mechanisms described in [32, 33] originally proposed by [34]. For the explanation below we therefore also make use of the same notation of those papers. The attention mechanisms described in these papers, has also successfully been applied for battery SOH

and RUL estimation using health indicators in [35]. The general architecture presented in that paper, greatly inspired the architecture which we present here. The soft attention mechanism works as follows. First the hidden state of each time step, is used to compute (alignment) scores at each time step;

$$\text{Score}\,(h^{(i)}) = v^T \mathbf{Tanh}\Big(Wh^{(i)} + b\Big), \qquad \forall i \in \{1, ..., t\}. \tag{6}$$

Here $v \in \mathbb{R}^{h'}$ is a learnable vector, $W \in \mathbb{R}^{h' \times h'}$ and $b \in \mathbb{R}^{h'}$ are learnable weights and biases, and $h^{(i)} \in \{h^{(1)}, ..., h^{(t)}\} \in \mathbb{R}^{h' \times t}$ represents the hidden state. We refer to the size of a hidden state at any time step ($h^{(i)}$), as h'. Next a weight for each time step, denoted as a_i (not to be confused with α which represent a conditional quantile), can be computed by taking the softmax of the alignment scores over the time dimension;

$$a_i = \text{Softmax}\left(Score(h^{(i)})\right) = \frac{e^{Score(h^{(i)})}}{\sum_{j=1}^{t} e^{Score(h^{(j)})}}, \qquad \forall i \in \{1, ..., t\}. \tag{7}$$

Resulting in a set of t weights, one for each time step, between 0 and 1, of which the sum is 1. These weights reflect the importance of a given hidden state vector, with respect to the other hidden states. Lastly the output, often described as the context h_c, can be computed by the weighted sum of each hidden state, using the respective weight (a_i);

$$h_c = \sum_{i=1}^{t} a_i h^{(i)}. \tag{8}$$

The context which is obtained, is of equal size to the original dimension of a hidden state ($h_c \in \mathbb{R}^{h'}$). However, in comparison to purely making use of the last hidden state, now a context is learned, which represents valuable information based on all hidden states. This vector can then be fed into a predictor with a non linear activation function (e.g. ReLu) when multiple layers are used, to obtain a prediction of the SOH. In literature this method of processing a (time) series into a fixed length vector, is often referred to as an "encoder". To conclude, intuitively soft attention thus provides the network with the ability to determine which hidden states may be important, to make a prediction. The full model can be found in Figure 6 at the beginning of section III.

C. Estimation of Uncertainty

Generally in the field of machine learning the two most employed types of uncertainty are aleatoric and epistemic uncertainty. Epistemic uncertainty, also frequently referred to as reducible uncertainty, is a type of error which can be attributed to the lack of knowledge about the data, model selection, and data quality. While on the other hand aleatoric uncertainty, also defined as irreducible error, arises from the principle that the problem is an approximation problem and there thus is intrinsic unpredictable uncertainty or noise engrained in the data [14, 36–39]. An easy manner to highlight both aspect was presented in James et al. [36], below the derivation is presented.

Given a random function f, the following relation holds: $y = f(x) + \epsilon$, where y is the target variable, x the predictor, and ϵ is a noise or error component engrained in the data. The function is estimated by means of function g: $\hat{y} = g(x)$, where $\hat{y}$ demotes the prediction. Now as highlighted in [36], the mean square error between the target and prediction can be evaluated. Note that independence between ϵ and $f(x) - g(x)$ is assumed and the expected value of ϵ is assumed to be 0.

$$
\begin{aligned}
\mathbb{E}\left[(y-\hat{y})^2\right] &= \mathbb{E}\left[(f(x)+\epsilon-g(x))^2\right] \\
&= \mathbb{E}\left[(f(x)-g(x))^2\right] - 2\mathbb{E}\left[\epsilon(f(x)-g(x))\right] + \mathbb{E}\left[\epsilon^2\right] \\
&= \mathbb{E}\left[(f(x)-g(x))^2\right] + var(\epsilon) + (\mathbb{E}\left[\epsilon\right])^2 \\
&= \underbrace{\mathbb{E}\left[(f(x)-g(x))^2\right]}_{\text{Epistemic}} + \underbrace{var(\epsilon)}_{\text{Aleatoric}}
\end{aligned}
$$

In a more practical setting, for the case of illustration, consider a generic fictitious lithium-ion battery, with an EOL (training threshold in Figure 2) of roughly 1500 cycles. We would now like to develop a model which is able to estimate the uncertainty in the capacity of said battery and we have ample training data available in said region (before EOL) to train a good model. The results of the model can be observed in Figure 2.

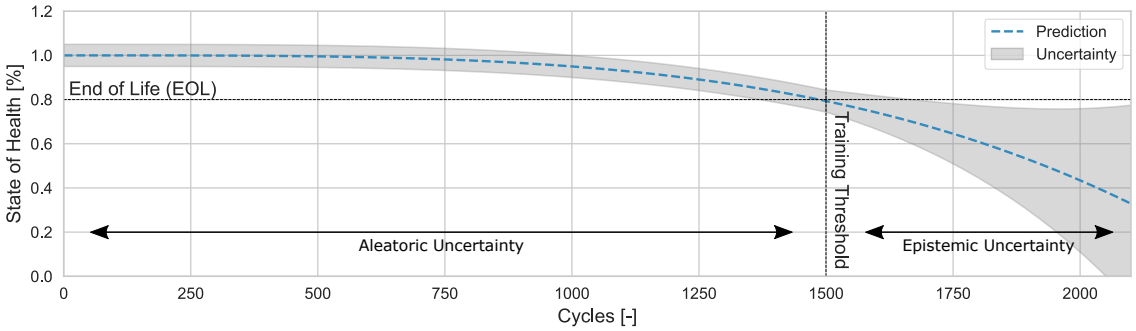


Fig. 2 An example of ideal uncertainty predictions for a generic fictitious lithium-ion battery.

For the first cycles up to EOL, a very thin region of uncertainty can be observed, around the prediction. Due to the large availability of data the model is able to make near prefect prediction in this region, however due to noise, their is still some uncertainty involved. This type of uncertainty is referred to as the previously mentioned aleatoric uncertainty. At EOL the most logical decision would be to retire said battery, in favour of a new battery, in the case we chose to operate it further. After EOL no training data was available to the model, therefore it would not be logical to assume that the model should generalise to said region. Within the figure, this is observed by the sharp growth in uncertainty, the further away the predictions are from the training distribution. In this region the model itself, is no longer a good approximation for the underlying degradation behaviour and predictions may greatly vary in accuracy. This second type of uncertainty is defined as defined as the epistemic uncertainty.

Often in literature numerous different approaches have been suggested to be able to get estimates for either the aleatoric, epistemic uncertainty or a combination of both. Here the methods differ on the basis of the assumptions being made and the overall complexity of the method. To be able to analyse the uncertainty in a batteries SOH we make use of divided approach, initially proposed by the researchers in Tagasovska and Lopez-Paz [37], where aleatoric and epistemic uncertainty are evaluated separate from each other, by means of Figure 3. The evaluation of the two uncertainty types in a separate manner, is achieved by first training a deep neural network, using a specialised loss function, to predict aleatoric uncertainty. The model weights and biases which are obtained are "frozen" and the intermediate output features of said network are used to train a more simple neural network, to estimate epistemic uncertainty.

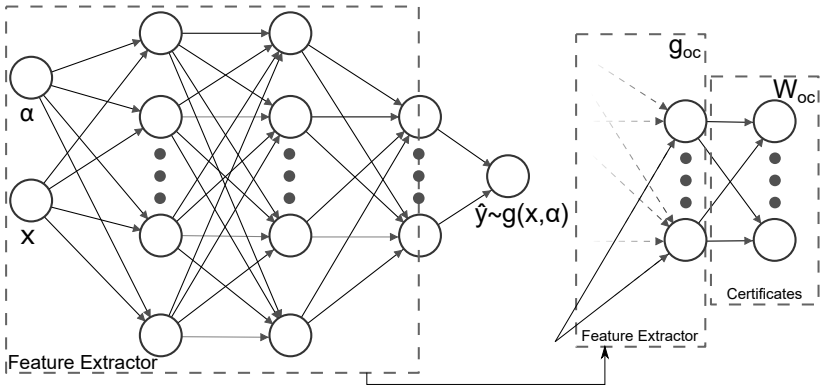


Fig. 3 On the left a simple feed-forward neural network used to predict the conditional quantiles of a dataset (Aleatoric uncertainty). On the right the network is observed in combination with the orthonormal certificates used for epistemic uncertainty predictions.

The reason we decide in favour of evaluating the uncertainty in a combined manner is based on the following reasoning. Firstly, as suggested by the authors in [37], both type of uncertainty aim to measure different things. Therefore a model which is specifically developed to identify both uncertainties separately, may be beneficial. From a user point of view it can be of valuable information to learn which type of uncertainty is prevalent. Additionally, although attractive methods such as: Monte Carlo dropout (MCD) [40, 41], evidential learning [42], ensemble methods (mean variance estimation (MVE) based [43] and interval based [44]), and Bayesian neural (BNN) networks exist (note: list is not exhaustive), they each come with their own unique set of shortcomings. For example a BNN is expensive, mathematically complex, and difficult to train. MVE behaves poorly to out of domain results, while ensemble methods and MCD can be inconvenient and expensive since they require multiple evaluations, either during training or inference [14, 45]. To our best belief and observations made in [14] all methods except MCD indirectly allow for individual analysis of both uncertainties. On the other hand, the 2-part method proposed in [37] comes with the big benefit that no assumptions are made on the distribution of the uncertainty, thus making it applicable to a wide variety of problems without the need to make distribution assumptions. To conclude, in general the uncertainty method may be considered a design choice, which can be chosen based on the area of application. Therefore if resources permit it, it could be valuable to explore various methods. Interested readers may find [14, 38, 39, 45] helpful during the search for an applicable technique.

1. Estimation of aleatoric uncertainty

At the base of the theory presented in [37] to estimate the conditional quantiles through the use of neural networks lies quantile regression. Quantile regression was initially proposed by [46], where through the use of an adapted L1 loss term, a regression could be performed around alternate locations (quantiles) in a dataset. Intuitively, a quantile can be interpreted as the amount of data, expressed as a percentage, that is below said value. For example the $\alpha = 0.5$ quantile ("mean"), is the location in the dataset, at which 50% of the data is smaller than the value indicated by the described quantile. We refer to both the original paper [46] and a more recent paper by the author [47] for the theoretical background of the method. In general terms however the method is concerned with solving [48],

$$\mathcal{L}\left(y, \hat{y}_\alpha \sim g(x,\alpha)\right) = \sum_{i=1}^{N} w_{i,\alpha} |y_i - \hat{y}_{i,\alpha}|, \qquad \text{where: } w_{i,\alpha} = \begin{cases} 1-\alpha & y_i - \hat{y}_{i,\alpha} < 0 \\ \alpha & y_i - \hat{y}_{i,\alpha} \geq 0 \end{cases}. \tag{9}$$

Equation 9 Is often referred to as the "quantile" or "pinball loss function". Here α is the desired quantile between 0 and 1 and is considered an additional input feature to the model. Furthermore, y is the target, $\hat{y}$ the prediction, w is a

weight depending on the residual and N the total number of data points. Similarly to traditional loss function such as the MSE (mean square error) or MAE (mean average error), the loss function aims to minimise the different between the truth and a models prediction. Contrary to them, it is able to provide aleatoric uncertainty estimates by providing a weight (α or $1-\alpha$) based on if the residual is positive or negative. Thus effectively balancing the residuals around zero, where the frequency of occurrence is based on the desired quantile.

One issue when solving Equation 9, is a phenomena referred to as "quantile crossing". This is a situation in which the predicted quantiles are not monotone, meaning that for example the mean ($\alpha = 0.5$) is located below the $\alpha = 0.3$ quantile [37, 48]. Numerically considering an example for battery capacity, this would indicate the mean capacity is 1.5Ah while the $\alpha = 0.3$ quantile is 1.7Ah (which is invalid). In attempt to solve this issue, numerous different novel approaches have been developed. For reference we briefly mention some approaches here. Within [48] a solution was proposed where first the expectiles of a distribution where derived using the L2 loss, and afterwards converted to more accurate quantiles. Alternatively [49] published a sequence of research papers into the topic of quantile regression using neural networks. Here both the Huber loss function was implemented over a discrete set of constant quantiles and additional constraints were implemented to avoid quantile crossing [49].

The approach which will be used in this study is referred to as simultaneous quantile regression (SQR) and was developed in [37]. Instead of solving Equation 9 independently for each desired quantile, the authors proposed to solve the same equation in a simultaneous manner using Equation 10. Here the quantile (α) is sampled from the uniform distribution $\mathcal{U}\,[0, 1]$. As a results the loss is performed over a variety of quantiles simultaneously within a mini-batch;

$$g^* = \arg\min_g \left[\frac{1}{N} \sum_{i=1}^{N} w_{i,\alpha} |y_i - \hat{y}_{i,\alpha}| \right], \quad \text{where: } w_{i,\alpha} = \begin{cases} 1-\alpha_i & y_i - \hat{y}_{i,\alpha} < 0 \\ \alpha_i & y_i - \hat{y}_{i,\alpha} \geq 0 \end{cases}, \alpha_i \sim \mathcal{U}\,[0,1]\,. \tag{10}$$

It is important to restate here that $\hat{y}_{i,\alpha} \sim g\,(x_i, \alpha; \theta)$ where θ are the model parameters. N refers to the total amount of data points in the training sets. y is the target, $\hat{y}$ the prediction, and α is a quantile, which is sampled from a uniform distribution between 0 and 1 for each training point. By optimising with respect to the equation above, the trained network (represented as g^*) will then be able to produce estimates for the requested conditional quantiles. The key benefit of this approach is that except for requiring an additional input, namely the desired quantile level, no further adjustments to general methods are required. They can be trained using the standard back propagation algorithm [13] and can afterwards produce a results for the desired quantile during inference.

2. Estimation of epistemic uncertainty

Equally important to the aleatoric uncertainty, epistemic uncertainty as previously described is mainly concerned with the uncertainty of the model. Epistemic uncertainty is a vital uncertainty component when dealing with neural networks, because it provides an indication about what the model knows and what it does not know. By looking for example purely at aleatoric uncertainty, one could falsely assume a models output to be reliable and good. However in reality when testing or deploying a model, it is evaluated on a dataset different from the one encountered during training. In this case it may falsely provide low uncertainty for a prediction or estimate based on what it encountered during training [37].

To be able to quantify epistemic uncertainty, orthonormal certificates (OC) are utilised. They were presented in the same research paper, in which SQR was presented, therefore the theory is again presented according to [37]. OCs can be considered a trainable and afterwards independently usable add-on layer to neural networks, which are capable of presenting if the main model has or has not learned about the data, during inference. The concept of has and has not learned, is accompanied by a low numeric value given low epistemic uncertainty (has learned) and a high value given high epistemic uncertainty (has not learned).

To be able make use of OCs, first it is assumed that a (deep) neural network has been fully trained and evaluated. If satisfactory performance has been obtained from an aleatoric point of view, the next step is to then fix or "freeze" all

weight and bias values of the model. To now evaluate for epistemic uncertainty, first the optimal model parameters (θ) of g^* are fixed or frozen. Next the last linear layer and activation function of the said model are discarded, and a learnable OC layer is added. OCs themselves are a weight matrix or single linear layer (without activation function and bias), where the input is of equal size to the penultimate layer of the main model (temporarily denoted as n). The output layer size is a tunable hyper-parameter (temporarily denoted as m). Again for a visualise interpretation we refer back to Figure 3, where both the extracted features and the certificates can be seen. The weights of the OC are learned through the minimization of the following equation, over the training set;

$$W_{OC}^* = \arg\min_{W_{OC}} \left[\frac{1}{N} \sum_{i=1}^{N} \underset{\alpha \sim \mathcal{U}[0,1]}{\mathcal{L}} \left(0, W_{OC} g_{oc}^*(x_i, \alpha; \theta)\right) + \lambda \left\| W_{OC} W_{OC}^T - I \right\|_1^2 \right], \qquad \text{where: } x_i \in x_{\text{train}}. \tag{11}$$

Here $\mathcal{L}$ is the quantile loss function described in Equation 9. The subscript highlights that the α quantile is sampled from a uniform distribution between 0 and 1, for each point. Lastly a penality is added based on the norm of the the weights of the OC multiplied by a factor λ, here I refers to the identity matrix. The addition of this term helps during optimisation, making it easier for the model to achieve a minima. Using this loss function and gradient descent, the OCs ($W_{OC} \in \mathbb{R}^{m \times n}$) will learn appropriate "weights" such that the learned feature space results in a zero value, while during inference adversarial samples are provided a non-zero value. For simplicity we refer to everything expect the last layer of the main model g as g_{oc}. Formally the relation between g and g_{oc} is thus defined as a "linear" or "fully connected" layer with ReLU activation function:

$$\hat{y} = g\left(x\right) = max\{0, W g_{oc}\left(x\right) + b\}.$$

Note that this should not be confused with a potential feature extractor before the quantile input and regression layer. The OCs are referred to as W_{OC}, highlighting that they in essence are learnable weight matrices. Now the epistemic uncertainty (e_u) can be computed by taking the mean across the output or certificate size (m), for any given point (x_i) in the dataset ($\mathcal{D}$);

$$e_u\left(x_i\right) = \frac{1}{m} \sum_{j=1}^{m} \left[\left(W_{OC}^* g_{oc}^*(x_i, 0.5; \theta) \right)^2 \right], \qquad \text{where: } x_i \in \mathcal{D}, W_{OC} \in \mathbb{R}^{m \times n}. \tag{12}$$

Through the use of Equation 12 an epistemic uncertainty estimate can be obtained for any point. However now the follow question remains: "how should the value provided by Equation 12 be interpreted?". Conceptually the general idea has been provided previously, anything close to zero is labelled as in distribution. While anything significantly deviating from a zero value is labelled as out-of-distribution (OOD). First as described in [37] the training distribution is logically assumed as (near) in-distribution. Then afterwards the threshold between in-distribution and out-of-distribution is defined as 95^{th} percentile of the epistemic uncertainty across the training set. If we now compute the epistemic uncertainty for a sample in the validation or test using Equation 12, we can now fully define if this points is defined in-distribution or out-of-distribution. In Appendix A we provide more information regarding the difficulty of selecting a feasible and accurate threshold.

Again Figure 3 presents an illustration of a model architecture to estimate both aleatoric and epistemic uncertainty using SQR and OCs. The case presented in the illustration is a simple multi layer perceptron, in the case studies discussed in section III the more complex model described in subsection II.B will be utilised. Here features will be extracted through the use of said model, which are then provided to the SQR an OC module. However the main theory presented in this section remains consistent.

D. Assessing the Models Predictive Accuracy

Now using the model we constructed above, the next vital task is to asses the predictive performance and accuracy of the model. The method we provided above is capable of delivering both point and uncertainty predictions, through

the use of conditional quantiles. Therefore the assessment of model accuracy in this section will be twofold. First a range of standard metrics, commonly used in the field of machine and deep learning will be highlighted. Afterwards we provide a set of evaluation metrics which are used in the field of engineering prognostics and uncertainty quantification. Especially the latter mentioned evaluation is critical because uncertainty quantification is not trivial and the quality of a models' uncertainty may vary greatly or generalise poorly.

1. Point prediction evaluation metrics

Assessment of model point prediction accuracy is the first crucial step to identify a models performance. Although the previously used loss function is already a good indication regarding the difference between the prediction ($\hat{y}$) and target (y), numerous other metrics exist, which are able to better capture if the used model is an accurate estimator for the target. We refer to the following general review paper, regarding various regression metrics for deep learning [50]. Note that in the equations below N refers to the amount of sample points, over which the metric is evaluated. Additionally all metrics with Table 1 excluding R^2, have a percentage based counterpart.

Table 1 Metrics to evaluate the point prediction accuracy.

Metric	Formula	Metric	Formula
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$	MAE	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$
MAX	$\max\left(\lvert y_i - \hat{y}_i\rvert\right) \forall i \in \{1, ..., N\}$	R^2	$1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$

The interpretation of RMSE, MAX and MAE are straightforward, they simply evaluate the difference between the prediction and target. The root mean square error (RMSE) evaluates larger differences more severely, while the mean average error (MAE) provides equal importance, MAX simply relates to the maximum error observed in the samples. Lastly R^2 refers to the coefficient of determination (or "R-squared"), which describes how well the model is able to describe the (variation of) target [36]. A value closer to 1 indicates that model is able to describe the variation of the target, while 0 indicates that is the model is not capable of describing the variation of the target. Here the model performance is similar to a model predicting the mean ($\bar{y}$) (note that a value below 0 indicates a "worse" relationship).

2. Uncertainty prediction evaluation metrics

Similarly to point predictive metrics, it is also possible to evaluate the predicted uncertainty. In literature this step is often forgotten, however its importance should not be neglected (as previously described). Although the process is slightly more lengthy in comparison, the uncertainty metrics provide great insight into the quality of the predicted uncertainty of a given model. Below the most significant evaluation metrics are introduced, which were highlighted and presented in [14, 51].

An efficient manner to investigate the accuracy of the uncertainty predictions is through the formation of a "calibration curve" [14]. The goal of calibration curves is to asses the relation between the predicted uncertainty and empirical or observable uncertainty in the data. Acknowledging this concept, the first criteria is to define a one dimensional grid of K preferable equidistant $1 - \alpha$ confidence levels;

$$C^K = [c_1 = 0, c_2, ..., c_{K-1}, c_K = 1] .$$

Now using the deep neural network trained using SQR, the values coupled to the aforementioned levels can be readily computed, using our optimised model g (remember: $\hat{y}_{c_k} = g^*(x, c; \theta)$). In more formal terms, the neural network functions as a discrete inverse cumulative distribution function (cdf), $F_X^{-1}(c_k)$ at each sample. Now given the safety critical context, we are interested in assessing the quality of the lower bound predictions and thus the confidence interval can be defined as:

$$CI^c = \left[F_X^{-1}(c), +\infty\right) \qquad \forall c \in C^K. \tag{13}$$

The above selected confidence interval depends on the research question, other research may benefit from a two-sided or one-sided upper bound configuration. By now observing the fraction of targets, which fall within their respective confidence interval, the observed confidence ($\hat{c}$) can be compared with reference confidence (c):

$$\hat{c} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{y}_i \in CI_i^c} \qquad \forall c \in C^K. \tag{14}$$

Here $\delta_{y_i \in CI_i^c}$ is a binary variable which is one when the i$^{\text{th}}$ prediction is part of the predicted confidence interval (of level c) and zero when it is not part of the confidence interval. Additionally comparing the observed confidence with the reference confidence, an additional pair of metrics can be constructed. The expected calibration error (ECE, Equation 15) [14] relates to the residuals, while the reliability score (RS, Equation 16) [51] is computed on the basis of the area formed between both functions.

$$ECE = \frac{1}{K}\sum_{i=1}^{K} |\hat{c}_i - c_i| \tag{15}$$

$$RS = \underbrace{\int_0^1 (\hat{c} - c)\, \delta_{\hat{c} \geq c} dc}_{RS_{\text{Above}}} + \underbrace{\int_0^1 (c - \hat{c})\, \delta_{\hat{c} < c} dc}_{RS_{\text{Below}}} \tag{16}$$

Similarly to Equation 14, δ is a binary variable equal to one, when the condition in subscript is true and zero when it is false. Therefore in Equation 16 the first term relates to the area above the ideal reference calibration, while the second term relates to the area under below it. A visual description of the calibration curve, in combination with the key components of ECE and RS is provided in Figure 4.

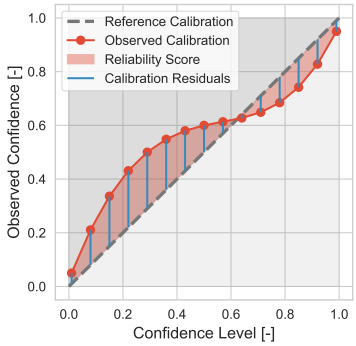


Fig. 4 Calibration curve in combination with the residuals required for the computation of the expected calibration error (ECE) and the areas for the reliability score (RS), expressed for a fictitious example.

In Figure 4, firstly the reference or ideal calibration curve can be observed. This line indicates the perfect condition in which the predicted confidence level, equals the observed confidence observed in the dataset. Secondly the observed calibration can be viewed, here the observed confidence is presented as a function of the reference confidence level. By now comparing the observed calibration with the ideal calibration, it becomes possible to make comments on the quality

of the predicted uncertainty. In general terms anything above the reference is consider under-confidence, while anything under the reference is interpreted as over-confidence. Ideally it would be desired that a model has ideal calibration, indicating a perfect match between the empirically observed confidence and the confidence level. However in reality this behaviour if often not achievable, in this case an under-confident model is preferred [14]. Lastly the residuals and the area between both calibration curves can be observed, required for the ECE and RS respectively. The metrics may be interpreted similar to other classic standard metrics, meaning a value closer to zero is preferred.

In the previous paragraph it was described that ideally perfect calibration is desired, however typically not feasible. Therefore under-confident behaviour is preferred, meaning the observed calibration is above the ideal calibration. To highlight the importance of under-confident behaviour, we present an example for fictitious batteries, at a certain cycle, with a capacity sampled from $\mathcal{N}\left(1.6, 0.1^2\right)$ in Figure 5. To the left the three probability density functions can be observed, the green line indicates the reference or neutral calibration. The blue dotted line simulates the distribution of the data leading to under-confident model, while the red dashed line represents the underlying distribution of the data leading to an over-confident model. To the right the calibration curves are depicted for each of the three distributions in comparison to the original sampled function. Three distinct points are highlighted in support of the example case below.

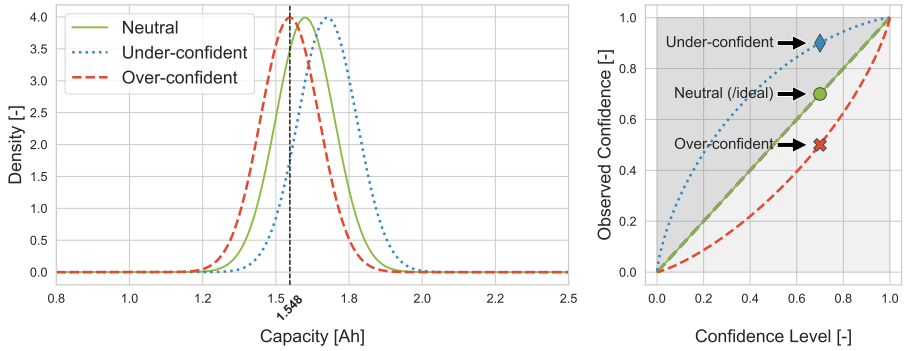


Fig. 5 Comparison of ideal, over and under-confident predictive uncertainty, applied to batteries with capacity sampled from $\mathcal{N}\left(1.6, 0.1^2\right)$.

Consider the case in which it would be desired to extract the $\alpha = 0.3$ quantile of a batteries capacity at a certain cycle, using a certain set of input features and using the aforementioned distribution (indicated by the green line in the figure). Through the Z-score, it can be determined that theoretically the battery should roughly have a capacity of $1.55Ah$. Since we are interested in determining the effectiveness of the model in predicting the lower bound, we construct the confidence interval as follows: $\left[F_X^{-1}\left(0.3\right), +\infty\right)$. Empirically it follows that this confidence interval should contain 70% of the number of points. In a model with neutral or ideal calibration, a perfect alignment between the observed confidence level and the (predicted) confidence level. If a model is found to exhibit over-confident behaviour (red dashed line), by constructing Figure 4. This would indicate that the model makes a prediction, which has a lower significance than expected. In practise this results in a lower bound prediction, which is higher than empirically desired. For example as previously stated, the model may provide the user with a capacity value of $1.55Ah$ for the conditional $\alpha = 0.3$ quantile, thus indicating 30% of the true values fall below said value and 70% above. If we then analyse the fraction of points which lay in the confidence interval using Equation 14. We may observe that interval only contains 50% of the total number of points, which is less than the expected 70%. As a result the probability that a value falls outside the interval indicated by the model, is higher than expected. It can readily be observed that this behaviour is highly undesirable, since the user may falsely be mislead to assume that the significance of the lower bound prediction of a battery its capacity, is higher than it actually is. Following the same procedure is can easily be argued that under-confident behaviour is preferred due to the inverse property, where the probability of the predicted value is higher than expected (indicated by blue dotted line in the figure). This conservative behaviour is opposite to an over-confident model and is of course desirable from a planning and operations point of view. In both cases it can thus be concluded that a discrepancy between the predicted and observed confidence exists. A particularly interesting phenomena is that both behaviours are not mutually exclusive, a model can be over-confident in one region and under-confident in another. This may occur when the data is more skewed or heavy tailed than the distribution predicted by the model.

III. Results and Discussion

In the following section we introduce a case study, to evaluate the model using the metrics presented in the previous sections. In subsection III.A we evaluate the model on a fast charging dataset published by Toyota. All datasets which were attempted in this report where found through a comprehensive study on data performed in [23] are open-access and thus freely accessible through their respective sources. A schematic representation of the model architecture which is used is observable in Figure 6.

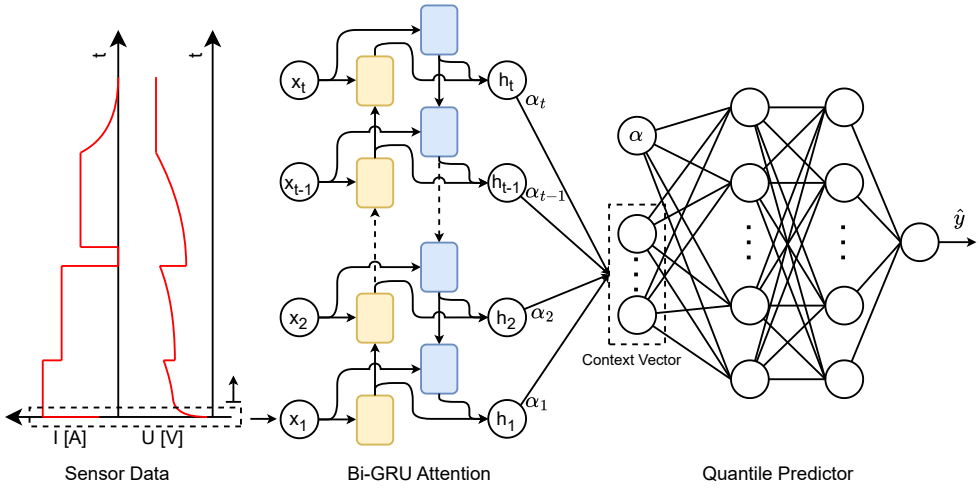


Fig. 6 Bidirectional gated recurrent unit, with quantile layer. Used to determine the battery SOH purely based on sensor measurements, in the case study.

A. Case study I: Toyota dataset

The dataset which we will explore in this section, is the fast charging dataset published by the Toyota research institute (TRI) [24] and is accessible through *[24]. Within this dataset researchers at TRI explored the influence of a diverse set of 2-phase fast charging procedures, on a batteries health. First we provide some information regarding the data itself, before moving to the results and discussion.

The dataset contains of 124, 1.1Ah phosphate based lithium ion batteries (LFP), divided over three batches[23]. Here each battery was charged using a "semi-unique" fast charging protocol, while it was discharged using the more traditional CC-CV protocol (see subsection II.A). We emphasises semi-unique since there is some overlap between batteries and their charging procedure. The cycling itself was repeated until EOL in a controlled environment with a temperature of 30°[24]. within Figure 7 an illustration is provided of both the current and the voltage measured during charging as function of the cycle count. These measurements will later function as the input to our model.

*https://data.matr.io/1/projects/5c48dd2bc625d700019f3204

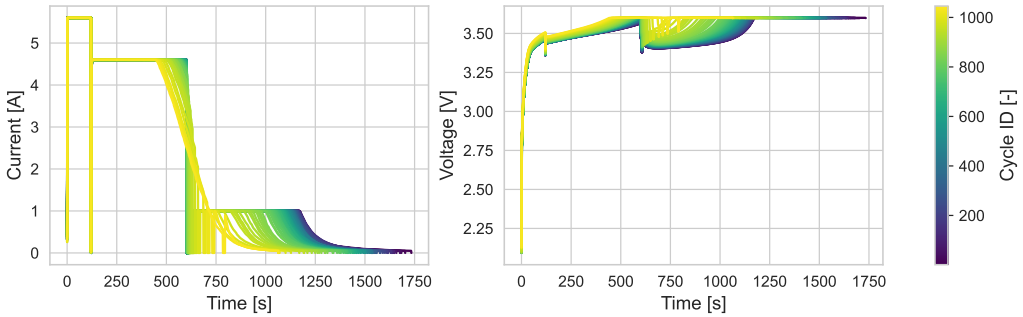


Fig. 7 Fast charging procedure used as part of the Toyota research dataset (5^{th} battery of the 3^{rd} batch).

Two interesting observations can be made in Figure 7. Firstly we can observe a version of the fast charging protocol which is being used. Here 2 "blocks" of primarily CC can be observed, where the magnitude is significantly higher than the CC part of a traditional CC-CV procedure. This 2-part, CC procedure is used up-to roughly 80% SOC, afterwards the regular CC-CV is used to fill the battery [24]. In this case the battery is first CC charged at $5.6A$, followed by a period of $4.6A$, upon reaching 80% SOC the battery is fully charged using CC-CV with a maximum of $1A$. The main idea of the fast charging protocol, is to as quickly as possible add current to the battery, while ensuring that the voltage limit is not reached. The second observation which can be made in Figure 7 is that both in the current and voltage measurements a clear variation and trend exist between cycles. Since the voltage and current are part of a regulated coupled system, we focus on the current. Firstly we observe that, as the battery ages, the time to fully charge the battery decreases. This can be explained by the reasoning that as the battery ages, its capacity decreases and thus the amount of current required to fill it also decreases. Additionally it can also be observed that the CC-CV part, also decreases in length. This again can be attributed to the batteries age and furthermore the voltage limit is being reached earlier during charging, thus reducing the CC part of the CC-CV. A similar phenomena occurs during fast charging, where the fast charging protocol is halted or interrupted due to the voltage threshold which is reached. The effect however is more drastic, especially near EOL, where after fast charging the battery is unable to reach $1A$ in the CC-CV part. The observation of this variation between cycles, forms the main motivation for determining the SOH purely based on sensor measurements. Additionally this variation in charge length, also led to the decision to select a model capable of analysing variable input length sequences. In this manner temporal information could be conserved, by avoiding the procedure of resampling to constant length, which we found to be unsuccessful for this dataset.

Now using the data presented above and the method shown in section II, we train a model using the hyper-parameters in Table 3. Most of the hyper-parameters were altered based on rough guidelines provided in [28], while others where found through grid search on a reduced dataset containing 450 samples per battery. The results are reported for the entire test dataset. Due to the amount of batteries available within the dataset, we limit ourselves to batch 3 (published in 2018) containing 46 cells. The train, validation, and test setup which was used can be retrieved in Table 2. For all figures we present a maximum of two batteries, with one achieving strong performance and the possible second case achieving less strong performance. The metrics themselves are evaluated on all cycles, to provide the most accurate assessment of the model its performance.

Table 2 Random data division used for training, evaluation and testing of the deep learning model, using the Toyota fast charging dataset.

Set	Keys
Train	B3_01, B3_02, B3_03, B3_04, B3_06, B3_08, B3_09, B3_12, B3_14, B3_15, B3_16, B3_17, B3_18, B3_19, B3_20, B3_21, B3_22, B3_25, B3_27, B3_31, B3_33, B3_44, B3_35, B3_36, B3_37, B3_38, B3_41, B3_42, B3_43, B3_44, B3_45, B3_46
Validation	B3_07, B3_10, B3_13, B3_24, B3_28, B3_39, B3_40
Test	B3_05, B3_11, B3_23, B3_06, B3_29, B3_30, B3_32

Table 3 Model Hyper-parameters for the Bi-GRU with soft attention configured for SQR and OC. The hyper-parameters were tuned and adapted for use with the Toyota dataset.

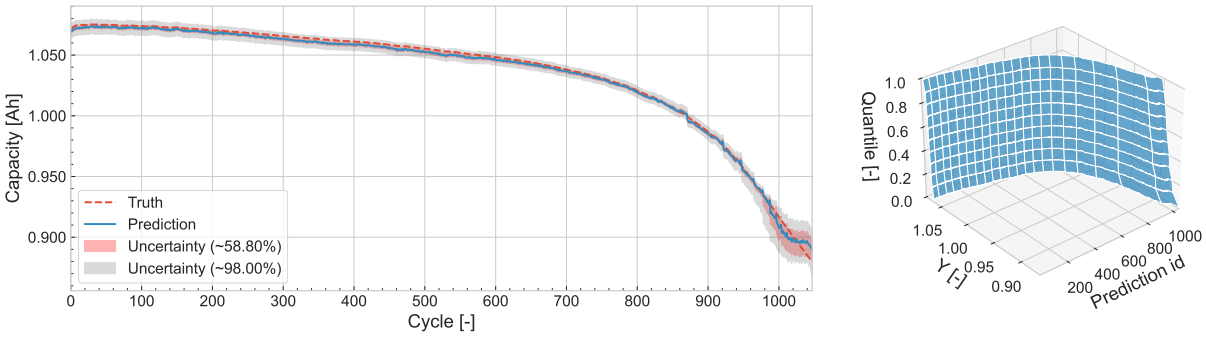
	Model				Optimiser		
	Hyper-parameter	*Value*	*Unit*		*Hyper-parameter*	*Value*	*Unit*
Model	Hidden state size	16	[-]	**Optimiser**	Optimiser	AdamW [52]	
	Attention size	32	[-]		Initial learning rate	1.0E-03	[-]
	Attention dropout	0.1	[-]		Weight decay	1.0E-05	[-]
					Mini-batch	64	[-]
	Quantile layer	32 (+1) - 128 - 128 - 1					
	Activation function	ReLU		**Scheduler**	Reduce l_r factor	0.2	[-]
					Reduce l_r patience	5	Epochs
Data	Δt	10	[s]		Reduce l_r sensitivity	1.0E-05	[-]
	Split	70-15-15	[%]		Early stop patience	10	Epochs
					Early stop sensitivity	1.0E-06	[-]

1. Evaluation of predicted aleatoric uncertainty

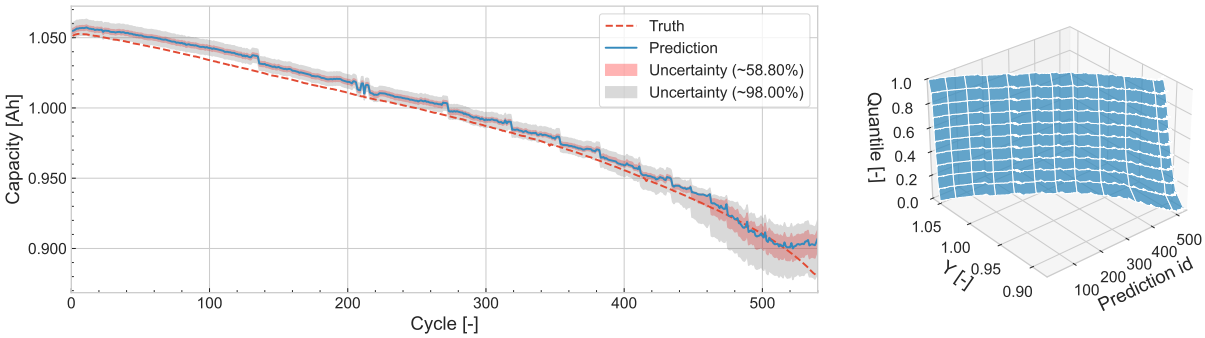
First we focus on the quality of the aleatoric predictions made by the model, evaluated on an unseen test set as seen in Table 2. In Figure 8 the aleatoric uncertainty determined using SQR and the BI-GRU with attention, is presented for battery B3_05 and B3_29. In Table 4 and Figure 9 the global metrics and calibration curve are presented

Table 4 Evaluation metrics for the Bi-GRU with attention tested on the Toyota dataset

Metric	MAE	$RMSE$	MAX	R^2	RS	RS_{Above}	RS_{Below}	ECE
Value	0.00231	0.00343	0.03108	0.99394	0.07887	0.07709	0.00178	0.07699



(a) Aleatoric uncertainty of the model predictions using the charging measurements of battery B3_05.



(b) Aleatoric uncertainty of the model predictions using the charging measurements of battery B3_29.

Fig. 8 Aleatoric uncertainty observable in the battery SOH model, determined using SQR.

For both batteries in Figure 8 it can be observed that the proposed Bi-GRU with attention is able to provide predictions which closely match the target, for almost the entire lifecycle of the battery. This is confirmed by the high R^2 and low MAE in Table 4, which are $0.00231Ah$ and 0.99394 respectively. The results obtained for battery B3_05 (Figure 8a) are considerably better that the ones obtained for B3_29 (Figure 8a). For battery B3_05 the model is able to consistently generalise for unseen data, closely matching the target variable for nearly 1000 cycles. In the predictions of B3_29 a consistent offset is observable between the target variable and the predictions. This offset is however limited in size, meaning a close match between the prediction and target remains. Furthermore a form of stepwise decay in the SOH estimation is observable, in comparison to the smooth degradation of the target variable. Furthermore the maximum observable error of $0.03108Ah$ ($\sim 3\%$ with respect $1.1Ah$), is found within this sample at EOL. This region consistently performed worse during all of our experiments, even for B3_05, predictions are worse in this region. Possible reasons for this behaviour and a possible concern of the model architecture are highlighted later.

With respect to the uncertainty, we observe thin confidence intervals and near monotone cumulative distribution functions. This highlights that randomly sampling α quantiles greatly reduced the formation of crossing quantiles. A second observation is that the uncertainty is narrow, which is the preferred behaviour from a planning point of view. Additionally when the predictive performance decreases, such as towards EOL, the uncertainty also increases, thus indicating more variability in the data from an aleatoric point of view. Within Figure 9 we compare the observed calibration with the reference calibration.

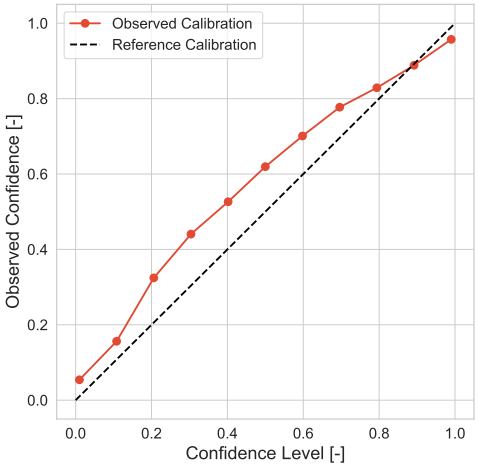


Fig. 9 Calibration curve comparing the observed calibration and reference calibration obtained for the Toyota dataset.

Unfortunately when assessing the calibration in Figure 9, it becomes immediately apparent that the model suffers from a light form of over-confidence at the high (1-α) confidence levels. This is further confirmed by the non-zero RS_{under} score in Table 4, thus indicting a negative discrepancy between the the reference calibration and the observed calibration. Particularly at the highest 0.99^{th} confidence level this effect is most noticeable, indicating that the predicted confidence levels contain a reduced amount of samples with respect to ideal conditions. As explained within subsection II.D, this behaviour is undesired, because of the possible dangerous implications. Intriguingly this over-confident behaviour rapidly changes to a desired under-confidence, from the 0.90^{th} confidence level onward. For the lowest confidence level, a near perfect calibration is observable. However from a usability point of view, generally such low confidence levels are hardly of interest, since we are interested in knowing the limits of our system with absolute certainty. If we purely consider for calibration, we do observe a strong calibration, with both the RS and ECE evaluating to low values.

Visualisation of attention weights is an additional nice property of using an attention mechanisms such as the one used within this research. This provides insight into which hidden states are weighted more heavily to make a prediction. In Figure 10 the attention map for the results provided in Figure 8a is provided, the input sensor measurements can be seen in Figure 7. In the attention map we observe that the model consistently allocates a higher weight to hidden states extracted near the middle to end of the sequence. Furthermore as the battery ages, the attention is being shifted backward in time, due to shorting of the input signal. Although we do not have insight into the exact meaning of the hidden states at each time step, we can make a general remark. Comparing the attention map to the input series, observable in Figure 7, we note that the model almost entirely neglects the hidden states directly coupled to the time steps linked to the fast charging part of the input sequence. As expressed in the introductory part of this section, during fast charging the battery is consistently brought to 80% SOC, albeit through different procedures. It can thus be considered that at the end of the fast charging procedure, before the 1A CC-CV, the batteries are in a similar charged state. The model has thus learned that the key features to battery SOH are mainly observable in the last CC-CV part of the charge sequence. This nicely aligns with [17], where features where created based solely on the CC-CV protocol. Furthermore, we may also observe that the region where the model achieved suboptimal performance, is characterised by a considerably different attention pattern. Here attention is shifted into the fast charging part.

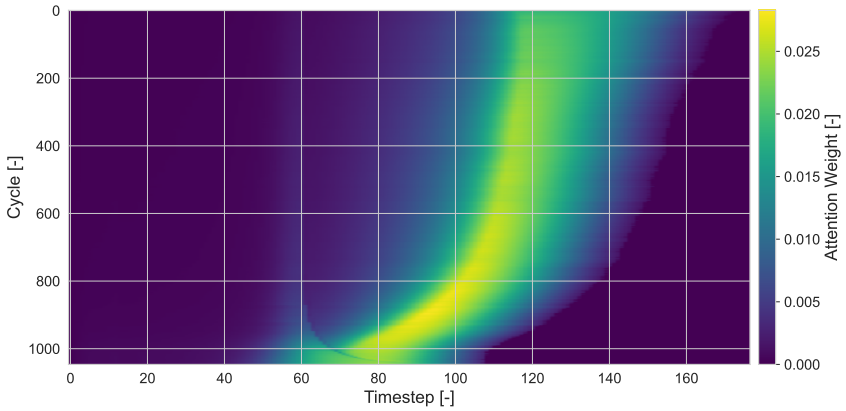


Fig. 10 Attention weights as a function of time for each cycle of battery B3_05

2. Discussion on behaviour near EOL

Regardless of the battery that was being analysed, it was observed that the model consistently had worse performance towards EOL. The region was also characterised by a broader confidence interval as observed in Figure 8. Occasional poor performance is to be expected, since a model cannot be assumed to generalise perfectly to all regions. However near EOL, consistent bad performance was found. Around EOL the SOH of a battery continues to decline, as observed by the "truth" within Figure 8. However if we consider the predictions which are made by the model, we observe an increasingly poor predictive quality. This quality only becomes worse, the closer the battery is to EOL, this is particularly observable in Figure 8b. Here instead of a downwards trend, the predictions are increasing as a function of

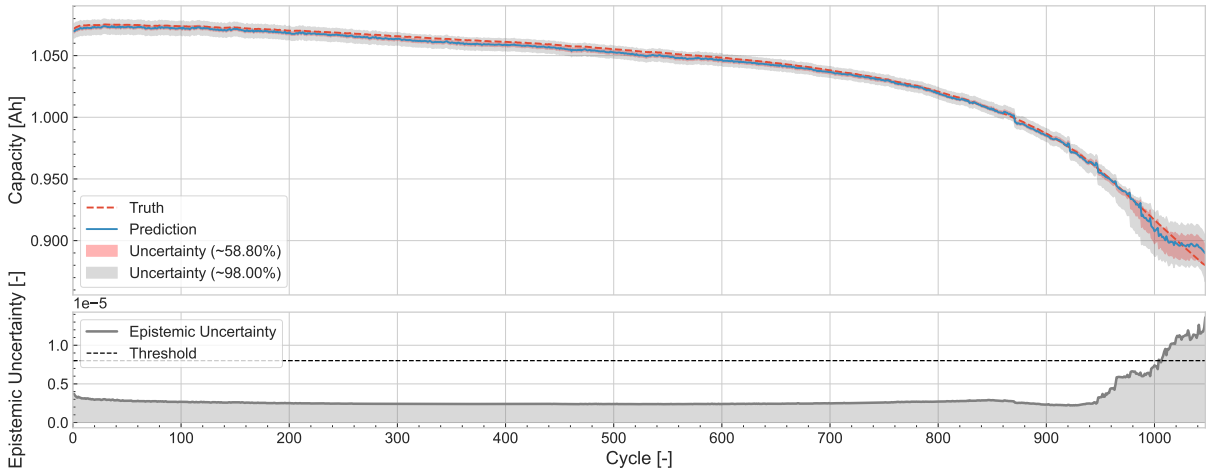
the cycle index. We believe that this can be attributed due to the two main reasons.

Firstly near EOL, a semi-unique phenomena occurs, where due to battery ageing the voltage threshold is reached earlier during the 2-phase fast charging procedure. As a result, it is no longer possible to charge at the elevated constant current and the amount of current which is being loaded gradually decreases (similarly as during the CC-CV protocol). Due to this observations it is only logical that the charge time also increases, instead of the traditional decrease in charge time as a function of battery age. Because this behaviour only occurs close to EOL, the amount of samples which behave accordingly in the datasets is comparatively limited. Empirically we found that on average 1.5 % (one outlier of 8.5%) of the samples per battery showed this behaviour. If we then consider that as a results the exposure to these samples in each mini-batch is limited, this could serve as the leading reasons for the poor model behaviour close to and at EOL.

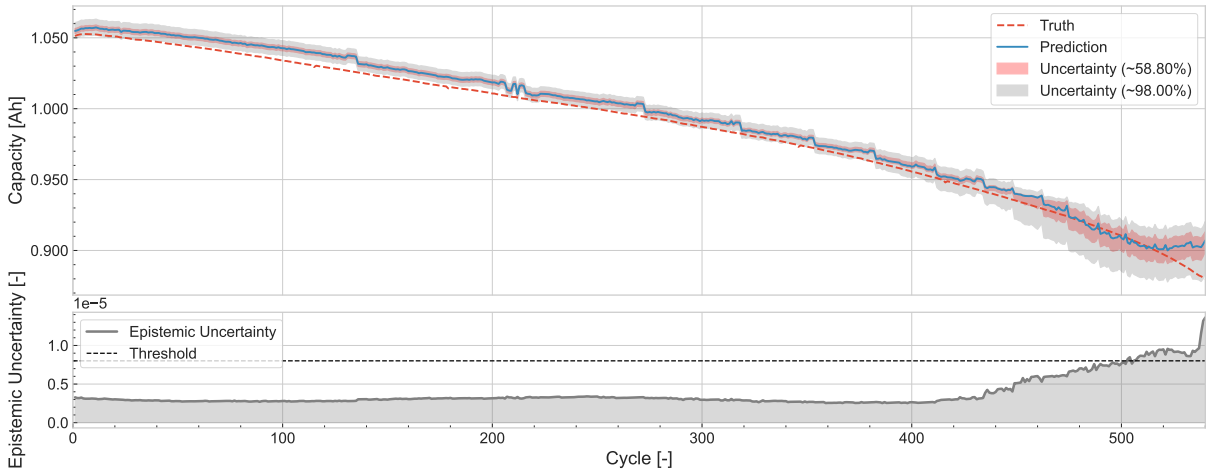
Another possible explanation for the behaviour at EOL, directly linked to the previous reasoning, is the model architecture. As discussed in section II we make use of a variable length input sequence, which is inserted into a GRU (Figure 1) to extract relevant features. Because shorter sequence generally mean a lower capacity, and longer sequence mean a higher capacity, it could be a possibility that the model may indirectly learn or model this behaviour, due to the sequential nature of the GRU cells. We believe that this has as effect that the predictions being proposed are to a large extent influenced by the length of the input sequence. This then explains the sudden increase in capacity at EOL, due to the increase in sequence length near EOL. If we focus on the attention weights observable in Figure 10, this indicates that the model is not purely looking at sequence length, however strongly influenced by it. We originally decided on variable length sampling out of necessity, since this allowed us to use a constant sample rate, thus preserving the temporal mapping between indexes between samples. We found that resampling to constant length, would significantly alter the signal and consistently provided poor and irregular predictive quality.

3. Evaluation of predicted epistemic uncertainty

Above, it was observed and described that the model has a particularly unique behaviour near EOL. Showing both an increase in uncertainty and a worsening point predictive accuracy. This behaviour could also be noticed when visualising the attention weights. Assessing purely from an aleatoric point of view provided little insight, except for the conclusion that the model is struggling to make reliable predictions within this region. We did however discuss that the model architecture or sampling strategy may play an important role in explaining this behaviour. Alternatively if we now asses for epistemic uncertainty, it could be that model was improperly trained for this region. In Figure 11 the epistemic uncertainty can be viewed for each SOH prediction, together with the set threshold (8.009E-06). For reference we include the distribution of epistemic uncertainty in the training set in Appendix A. Here we also discuss the difficulty on selecting a threshold when the training set contains adversarial or poorly learned samples.



(a) Epistemic uncertainty of the model predictions using the charging measurements of battery B3_05.



(b) Epistemic uncertainty of the model predictions using the charging measurements of battery B3_29.

Fig. 11 Epistemic uncertainty of the battery SOH model, determined using OCs, evaluated for two batteries.

The epistemic uncertainty was determined using OCs of size 128, trained using the ADAM [53] optimiser for 10 epochs, with a regularization factor (λ) of 1 and a learning rate (l_r) of 1.0E-3. In Figure 11 we can observe that for the initial cycles the epistemic uncertainty remains significantly below the threshold for both batteries. This is expected since previously it was also observed that the model is able to make strong predictions within this region. This thus highlights the epistemic uncertainty is able to correctly highlight that the extracted features, are within the learned feature space. The epistemic uncertainty remains low for nearly the whole lifecycle of the battery, only near EOL, a significant rise in epistemic uncertainty is observable. This highlights that within this region the model was unable to correctly generalise, due to variation between sample distributions or insufficient model capacity. Comparing the epistemic uncertainty of Figure 11a with Figure 11b, we can observe that the base uncertainty is slightly lower for B3_05 in comparison to B3_29. This can possibly be explained by the offset of the prediction in B3_29. More interestingly however, the epistemic uncertainty of battery B3_29 more quickly exceeds the threshold. This indicates that the model is struggling to generalise earlier in comparison to battery B3_05. Table 5 summarises the epistemic uncertainty observed on the entire test set. We include the fraction of points labelled as OOD, as well as the average epistemic uncertainty for each region. Here a region is defined as 0-10%, 10%-20%, ..., 90%-100% of the cycles of a battery its life. Similarly to the figure we observe that only near EOL samples are labelled as OOD, where 220 out of 606 cycles surpass the computed threshold. The average epistemic uncertainty in this region is 6.987E-06 which is below the threshold of 8.009E-06.

Table 5 Epistemic uncertainty of the Bi-GRU with attention when evaluated on the Toyota dataset. The epistemic uncertainty is visualised through the use of OCs, where the threshold is defined as: 8.009E-06.

Region	*0-10%*	10%-20%	20%-30%	30%-40%	40%-50%
Fraction of points OOD	0/626	0/626	0/626	0/626	0/626
Average u_e	3.000E-06	2.800E-06	2.733E-06	2.741E-06	2.765E-06
Region	*50%-60%*	60%-70%	70%-80%	80%-90%	90%-100%
Fraction of points OOD	0/626	0/626	0/626	0/626	220/606
Average u_e	2.759E-06	2.789E-06	2.960E-06	3.342E-06	6.987E-06

4. Evaluation of predicted epistemic uncertainty with mask enforced on the training set

We have now observed that the model is able to allocate higher epistemic values in a region where it was not able to provide the most applicable prediction. Because no explicit metrics were found to evaluate for epistemic uncertainty, we perform an additional experiments, to further investigate this topic. First consider the event where a model is not trained on data near EOL, but is afterwards deployed in a situation where batteries are near EOL. Practically this could occur when an manufacturer has build a model based on data up to a loss of 10% in capacity (90% of the original capacity is still available). However during operation, an operator decides to use the model up to a degradation of 30%. Ideally we would like to be able to distinguish the region where the model is applicable and where it is not, during inference. We therefore mask the last 40% of the cycling data, with respect to EOL, in the training set. It should be noted that the produced aleatoric uncertainty is different from the one observed, in the main experiment. For the purpose of this experiment we purely focus on epistemic uncertainty observable in Figure 12.

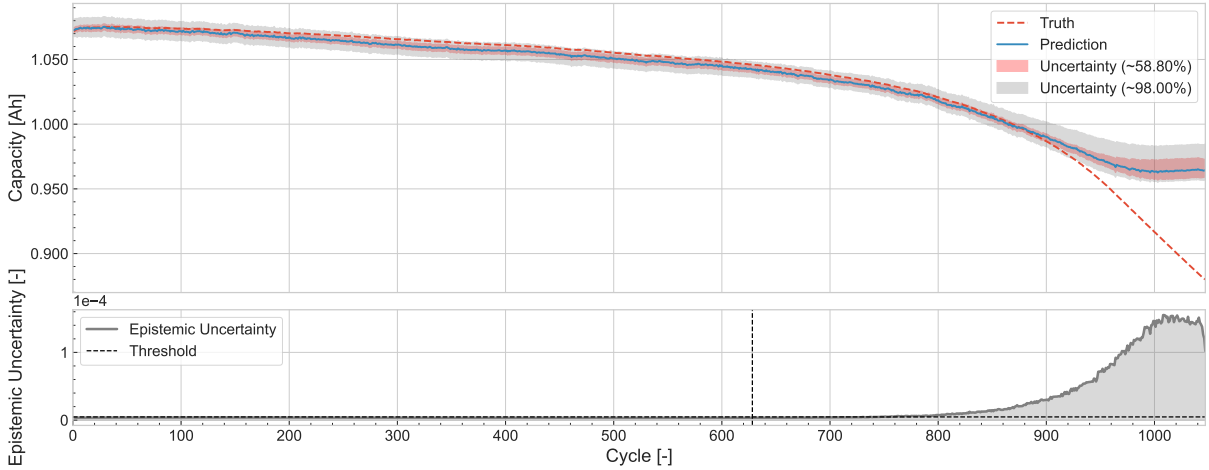


Fig. 12 Epistemic uncertainty of the model predictions using the charging measurements of battery B3_05, where the last 40% of the cycles where masked during training.

Consistent with the main results, we find that the model is able to provide epistemic uncertainty estimates which far surpass the values encountered in the training distribution, for regions where the model should not be expected to generalise. In this cases, this is clearly observable by the threshold between data which was provided to the model during training and the new life cycle data which was only provided during inference. What is surprising is that upon reaching the training threshold, the epistemic uncertainty remains low for a significant period, before sharply rising the further we move away from the training distribution. This highlight the model is able to generalise well to an unseen region, before alerting that its predictions may no longer be trusted.

Interestingly some additional observation can be made relating to the model its behaviour to the input sequence length, when reconsidering the masking experiment in Figure 12. If we observe points close to EOL, it can be observed that the epistemic uncertainty estimates slightly decrease, even though these points are moving further from the training domain. Although the epistemic uncertainty still evaluates in favour of OOD, the decreasing behaviour is in exact contrasts to what would be expected, since these data points are entirely different from the samples encountered during training. However as previously stated the cycles near EOL are characterised by an increases in sequence length, compared to the decreasing trend observed during most of the battery its lifecycle. We believe that this decreases, causes the sample to move closer to the training domain, thus leading to a decrease in epistemic uncertainty. We believe that this apparent heavy dependency on sequence length is a downside to the model in combination with the stochastic loss function making the model hard to optimise and sensitive to the selected hyper-parameters.

IV. Conclusion and Recommendations

Predicting battery state of health, is a difficult topic, with increasing relevance caused by the increasing popularity of battery electric vehicles (BEV). Due to various complex non-linear internal mechanism, over time the total usable capacity in a battery decreases. This is a huge disadvantage because systems rely heavily on said capacity to perform their mission. As a result processes need to be set in place to be able to provide accurate estimates on the total usable capacity in fully charged state, often referred to as battery state of health (SOH). Traditionally in data driven approaches such as machine learning, researchers would often design sophisticated features based on charge measurements. The features could then be used to train a model, which can afterwards be deployed to make informed predictions. In this report we present a data driven approach, relying solely on sensor data taken during charging, thus avoiding any sophisticated feature engineering process. Furthermore to assess the trustworthiness of our predictions we assess the models ability to represent both aleatoric and epistemic uncertainty. This is a topic which has been comparatively reached to a lesser extent, even though it is vital when considering the application of batteries in safety critical systems or domains.

Batteries are equipped with a range of sensors (voltage, current, temperature), which communicate information from the battery to the battery management system (BMS), which on its turn is responsible for safe control and system analysis. These sensor measurements taken during charging are inherently complex and have the key characteristic that they may vary in length. To preserve temporal information we first sample the original signal using a fixed sample rate. Next we provide the varying length input sequence to a bidirectional gated recurrent unit (Bi-GRU) with soft attention. This structure allows the model to learn and identify which parts of the input sequence are vital for making battery SOH predictions, by means of a context vector. Afterwards this discrete set of "features" or "states", are provided to a regressor with the desired quantile as additional input. We train the aforementioned architecture using simultaneous quantile regression (SQR), to investigate aleatoric uncertainty. Next epistemic uncertainty is estimated using orthonormal certificates (OCs), here an additional set of weights are trained to map adversarial data samples to non-zero value.

We evaluate the proposed Bi-GRU with soft attention on 1.1Ah (LFP) battery cells in a fast charging dataset published by the Toyota research institute. Focussing firstly on the predictive quality of the model we observed strong results, highlighted by the a low mean average error (0.002Ah) and low maximum error (0.034Ah). Furthermore the R^2 value between the target and the predictor is 0.994, highlighting that the model is able to successfully predict the degrading battery SOH. Next we found that the aleatoric uncertainty produced using our model, trained with SQR, was able to successfully and consistently provide thin confidence interval and near monotone cumulative distribution functions. The ability to model any distribution with monotonicity is a real benefit of the used method, in comparison to other techniques. Additionally through the construction of calibration curves, we found that lower bound predictions made by the model showed a minor sign of over-confidence at the high confidence levels. While exhibiting the preferred under-confident behaviour for confidence levels below the 0.9 confidence level. This miss-calibration is highlighted by the reliability score, which represent the area between the observed and reference calibration ($RS = 0.0789$, $RS_{\text{Below}} = 0.0771$, & $RS_{\text{Below}} = 0.0018$). Across all batteries a consistent deviation was observed between the prediction and truth near end-of-life (EOL), the high epistemic uncertainty in this region was successfully detected through the use of OCs.

To conclude, in this paper we presented the suitability of using SQR and OCs for reporting aleatoric and epistemic uncertainty in battery SOH estimation. However due to the stochastic loss function used in SQR we did encounter issues with model stability and hyper-parameter sensitivity during training. Additionally the selection of an adequate threshold for OCs remains an important issue, to be able reliably identify faulty samples. Future research is aimed at further analysing the issue of model stability, for example trough the integration of an alternate feature extractor (such as a temporal convolution neural network). Additionally developing deep learning models with uncertainty visualisation capabilities is not a trivial task, therefore during design it may be helpful to not separate both type of uncertainties. Therefore we are interested in investigating how the approach used in this paper compares to combined modelling techniques such as ensemble learning using mixture density networks. Lastly we are especially curious how the model architecture may perform in alternative tasks in the domain of batteries, such as state of charge estimation, alternate battery datasets, or remaining useful life forecasting.

A. Distribution of Epistemic Uncertainty

In the main body of this research paper, it was described that the threshold between in-distribution and out-of-distribution is defined as the 95$^{\text{th}}$ percentile across the epistemic uncertainty within the training set. When using this threshold we found that the OC model consistently determined predictions near EOL to be out-of-distribution. We further discussed that this out-of-distribution label aligned with observation we made regarding model behaviour in this region. The core difficulty however, lays in successfully determining a strong threshold value, which we were not able to find a good solution for. In this case the choice of 95$^{\text{th}}$ percentile threshold value can be considered to be rather arbitrary.

The OCs presented in [37] were most notably designed to detect adversarial or out of distribution samples. Meaning the samples should be significantly different from what is encountered during training. However if adversarial or irregular samples are encountered during training this does form an issue, since the threshold is set based on their existence. When selecting the 95$^{\text{th}}$ percentile across the training set, this assumes that 5% of the samples within the training set lay above said threshold. Thus by design 5% of the samples will be labelled as slightly or highly out-of-distribution (depending on the nature of the dataset). It is entirely realistic to assume that not all samples within the training set are learned well by the model (in our case the EOL behaviour). However it does immediately raise questions regarding why the threshold of 95 was chosen instead of 99 or even 90. We believed that 95 was the most sensible balance between the three values which were tested in [37], because it was sufficiently high to not falsely label points as out-of-distribution, while still being able to identify fault samples. Ideally a more appropriate method would be to set a threshold bound, using the procedure highlighted in [54]. Here by means of the Chebyshev inequality an upper bound probability can be given to a set threshold. As a results the selected threshold has a form of upper bound on the probability that a value is located above a threshold. In this case since we are dealing with strictly positive epistemic uncertainty values, the adaptation suggested by Cantelli is more applicable [55]:

$$P\left(|e_u\left(x_i\right) - \mu_{e,\text{train}}| \geq k\sigma_{e,\text{train}}\right) \leq \frac{1}{1+k^2} \qquad \text{where: } x_i \in \mathcal{D} \tag{17}$$

Here μ_e and σ_e are the mean and standard deviation of the epistemic uncertainty in the training set. e_u refers to the epistemic uncertainty of a to be classified point, computed using Equation 12. The integer variable k, can be selected based on the desired threshold between in an out of distribution. Lower k values, result in a higher probability that any given point is classified as out of distribution, due to the lower standard deviation which is accompanied by this decision. In Figure 13 a histogram of the epistemic uncertainty across the training set can be observed containing vertical lines for both the percentile based threshold and the standard deviation based threshold values. Here it can readily be observed that the threshold values based the standard deviation are quickly become quite large for a low k, as results the upper-bound probability suggested by Equation 17 provides little benefit, over the percentile based approach. Furthermore it will often fail to identify cases which fall outside the distribution.

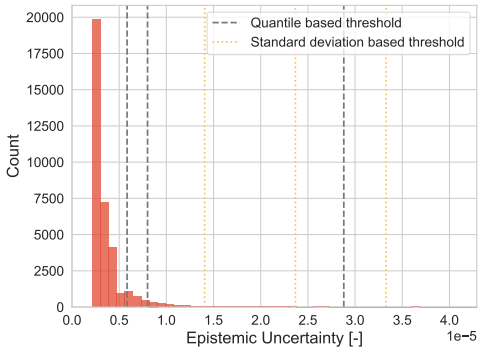


Fig. 13 Histogram depicting the epistemic uncertainty values encountered during training, using the Bi-GRU with attention trained using SQR and evaluated using the OC output layer.

B. Verification on Toy Example

Within this chapter the framework developed to verify the model presented within section II is provided. Within subsection B.A the models ability to estimate aleatoric uncertainty is demonstrated through the creation of a range of "toy examples". Afterwards within subsection B.B a similar procedure is utilised to highlight the implementation of the epistemic uncertainty and the models ability to provide estimates for it. Both subsection B.A and subsection B.B are heavily inspired by the toy example created and discussed by Tagasovska and Lopez-Paz [†], both subsections mainly served as an internal verification procedure, to test relevant aspects. As a small side note: all calibration curves within this section are assessed for right sided or lower bound confidence interval, similar to the main report.

A. Aleatoric uncertainty

Aleatoric (or irreducible) uncertainty describes randomness, noise, and general uncertainty engrained within the data. In most machine learning applications, this type of uncertainty is of great insight, since it provides information into the degree of uncertainty between the explanatory (x) and dependent (often also referred to as predicted or target) variable ($\hat{y}$). Referring back to the derivation made within section II, which for convenience is rewritten here:

$$\mathbb{E}\left[(y-\hat{y})^2\right] \Rightarrow \underbrace{\mathbb{E}\left[(f(x)-g(x))^2\right]}_{\text{Epistemic}} + \underbrace{var(\epsilon)}_{\text{Aleatoric}}. \tag{18}$$

The noise component (ϵ) becomes immediately visible [14, 36]. To be able to estimate this component, within the setting of battery SOH estimation, simultaneous quantile regression was utilised [37] to estimate the conditional quantiles. The original researchers highlighted that through the use of their technique, a diverse group of uncertainty behaviours could be estimated. The most relevant cases are highlighted below.

Table 6 Different types of uncertainty [37]

Type	Description
Heteroscedastic	Variance changes over time, particularly valuable for time-series forecasting.
Gaussian	Error follows a normal distribution.
Non-Guassian	Error follows another distribution such as a uniform or exponential distribution.

Although the original authors did provide a brief toy-example, highlighting the models functionality, they did not investigated the accuracy of the predictions being made. Therefore a simple experimental setup is created in which the cases described within Table 6 are observable, afterwards the predicted uncertainty is evaluated using methods described within subsection II.D. We evaluate the following 2 cases, where x is a discrete set of points (e.g. seconds), and y is the response. Within the first example noise (ϵ) is artificially added by sampling from a uniform distribution, while within the second case we sample from a normal distribution with varying standard deviation.

1) $y_i = 0 + \mathcal{U}(-0.5, 0.5), \qquad \forall i \in \{1, ..., N\}$
2) $y_i = \sin(x_i) + \mathcal{N}(0, \sigma = f(x)), \qquad \forall i \in \{1, ..., N\}$

Further implementation details include normalization of the explanatory variable, using minmax. While a standard train, test data split of 60 / 40 is utilised, sampled from the main set $(x_{test}, y_{test}) \wedge (x_{train}, y_{train}) \in \{(x, y)\}$. A simple standard feed-forward neural network, of structure 2-128-128-1 is used, with ReLU activation functions. The desired quantile is additional input (scaled with respect to the normalized input data). An illustration of this generic network is observable in Figure 3. The quantile loss function is used in combination with the Adam [53] optimiser

[†]Github repository of the original authors: `https://github.com/facebookresearch/SingleModelUncertainty/tree/master`

($l_r = 1e^{-3}$, weight decay $= 1e^{-5}, \beta_1 = 0.9, \beta_2 = 0.999$), and a mini batch size of 128. The total amount of epochs is varied per experiment, while for all experiment 21 quantiles between 0 and 1 are used to construct the calibration plot.

Case 1: Uniform noise The first case lays the focus on the model its ability to model uniform random noise. Using the function $y = 0 + \mathcal{U}(-0.5, 0.5)$, a set of 4000 points are sampled between $x_i = 0$ and $x_i = 20$. Dividing the dataset into a training and validation set, followed by min-max normalization with respect to the training set the following setup is created Figure 14. Additionally 100 epochs are utilised during training.

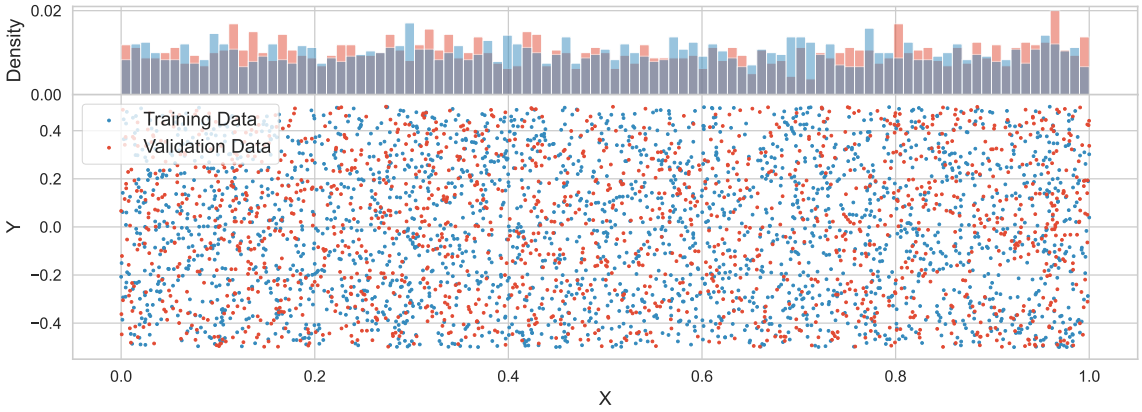


Fig. 14 Training and validation setup used within example case 1. Where y is independent of x and uniform random noise is added on top of the signal.

It can be observed that across the full data envelope, that both training and validation points are represented sufficiently well. After training, inference is done across a discrete set of quantiles between 0 and 1 based on the validation set. The results of this experiment can be visualised within Figure 15

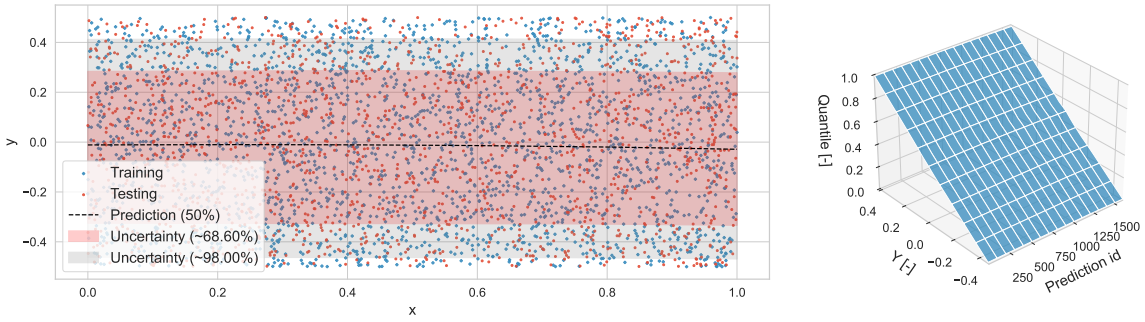


Fig. 15 Results of case 1, on the left the predictions can be viewed for a range of quantiles. Whereas on the right the cumulative distribution function is visualised based on the discrete set of evaluated quantiles, for each validation data point.

Within Figure 15 it may immediately be observed that the models point prediction of the mean is around zero. This observation coincides with the original sampled functions ($E[\mathcal{U}(-0.5, 0.5)] = 0$). We observe that the 98% confidence interval contains nearly all the data, both within the training and validation set. Next on the figure to the right, the cumulative distribution function (cdf) can be observed. Firstly the function created by a discrete set of quantiles, is monotone. This highlights that the method proposed by the authors, successfully reduces the phenomena of crossing quantiles. In fact for this example no crossing quantiles can be observed. Secondly the shape of the cdfs are in

accordance to a cdf of a uniform distribution, this coincides with the added noise component. The accuracy of the predicted uncertainty is further confirmed by the evaluation of the calibration curve and metrics (see subsection II.D for the relevant methods).

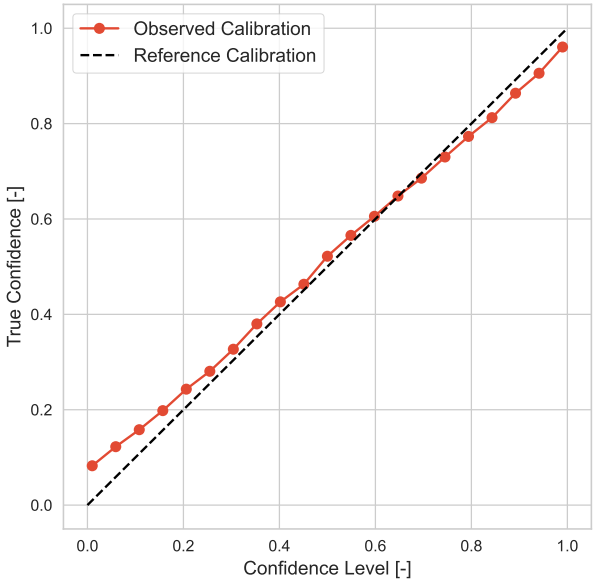


Fig. 16 Calibration curve comparing the expected confidence to the found confidence for case 1.

Figure 16 is a powerful manner to quickly evaluate the uncertainty of a given model. It can be observed that in general a strong match between the expected and found confidence can be observed. This is highlighted by the high R^2 and low ECE ($R^2 \approx 0.988$ and ECE ≈ 0.028). The RS score is approximately 0.027 (above: 0.019, below: -0.008), again indicating a strong and well calibrated uncertainty. The model does exhibit a slight form of overconfidence, towards the high $(1-\alpha)$ confidence levels. Indicating that the lower bound predictions for these high confidence levels are to high. As discussed within subsection II.D this behaviour is not ideal and can be dangerous, however the overconfidence is limited. Interestingly for the low $(1-\alpha)$ confidence levels, the opposite behaviour can be noticed. The model gradually transition from overconfidence, to neutral confidence, to underconfidence. It is however believed that an improved model and a model less prone to overfitting, could lead to improved performance (not in the scope of this simple verification example).

Case 2: Sinusoidal function with increasing variance In a similar fashion a sinusoidal wave is investigated, with the addition of normal noise, where variance/ standard deviation is sampled as a function of x. The following function is used within the experimental set up: $y_i = \sin(x_i) + \mathcal{N}\left(0, \sigma = 0.5 + 0.4 * \left(\frac{x_i}{10}\right)^2\right)$. Again 4000 samples are collected and 500 epochs are utilised during training. To avoid repetition, the focus is laid on the results, which are presented in Figure 17 and Figure 18.

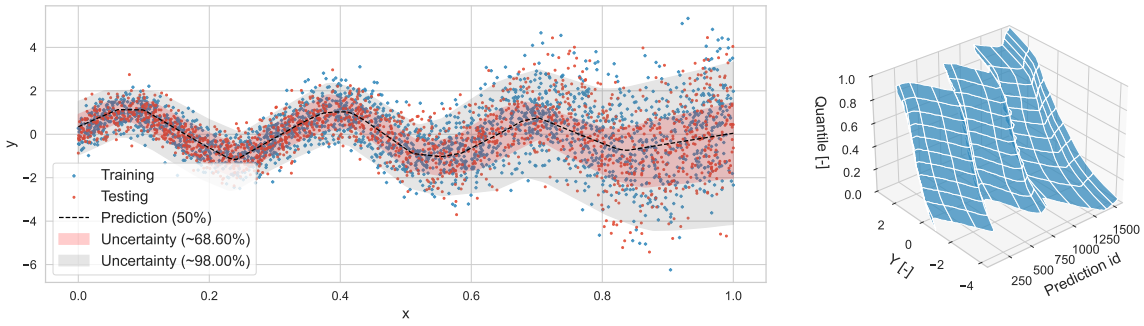


Fig. 17 Results of case 2, on the left the predictions can be viewed for a range of quantiles. Whereas on the right the distribution is visualised based on the discrete set of evaluated quantiles, for each validation data point.

Although the learning process is more lengthy and complex in comparison to case 1, satisfactory results are obtained. It is immediately observed that the neural net is able to learn and predict the sinusoidal nature of the training data. Furthermore the uncertainty of both the 68% and 98% confidence interval show an increase in size, with respect to an increase in x. The observation matches with the main logic behind the implemented experiment (heteroscedastic noise). The figure on the left highlights that SQR is able to provide monotone, heteroscedastic, normal/ Gaussian cdf. Now to evaluate the quality of the predicted uncertainty a calibration curve is constructed, which can be found in Figure 18.

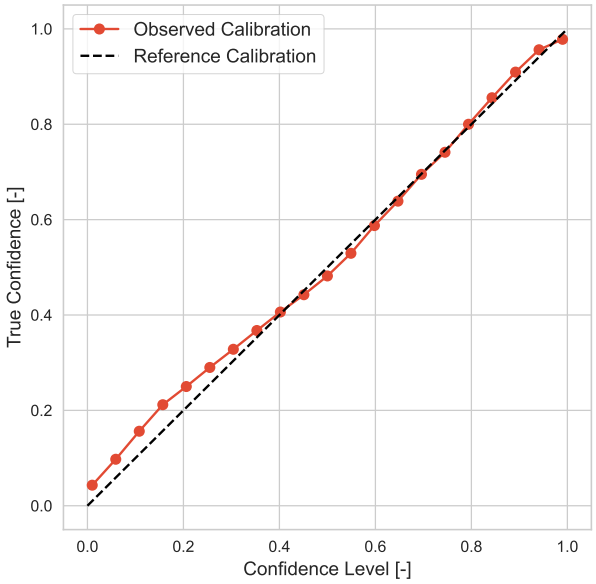


Fig. 18 Calibration curve comparing the expected confidence to the found confidence for case 2.

First a minor overconfidence is observed at highest $(1-\alpha)$ confidence level, similar but smaller than the overconfidence observed in comparison to the previous case. What is particularly interesting is that the model oscillates between over- and underconfident predictions throughout the entire centre band of the confidence levels, afterwards stabilising in the underconfident region for the low confidence levels. This could be a potential small indication of overfitting on the training set or poor generalization to the validation set. However, for the case of these fictitious examples no further analysis was performed, to potentially achieve a perfect match. In general strong calibration is observed, where the observed confidence is close to the reference confidence. This is highlighted by the metrics: $R^2 \approx 0.993$ and ECE $\approx$ 0.020, the RS score is 0.019 (above: 0.016, below: -0.003).

B. Epistemic uncertainty

Epistemic uncertainty is primary related to the uncertainty of a given model (see Equation 18) and thus how well it represent the underlying function. It becomes especially relevant when dealing with neural networks, due to the stochastic nature of the training procedure and their behaviour to adversarial or out of distribution samples. For the case of safety critical systems, this type of uncertainty becomes especially relevant. Because it gives an indication when the model itself is uncertain, or was not trained to make predictions for this data regime. To provide estimates for epistemic uncertainty we make use of orthonormal certificates, proposed by the same authors of SQR [14, 37].

Similar to cases highlighted for the aleatoric uncertainty, distinct cases can be constructed to force epistemic uncertainty within the model. To be able to focus on epistemic uncertainty, consider the functions implemented within subsection B.A. To estimate aleatoric uncertainty we sampled the training and validation set in a similar manner from the dataset. Now instead of ensuring the distribution between both sets are similar, we actively enforce a difference in distribution between the training and validation set. Here we make use of a mask, which hides certain data regions during training. This approach aims to simulate a behaviour, where during inference data is provided outside the training "envelope". Linking this idea to batteries, one can think of missing data points at EOL, or measurement errors (ideally we would of course like to identify such cases). This concept case is inline with the original idea of OCs within the context of epistemic uncertainty, highlighted within the research paper of the original authors[37]. Furthermore we hope that this example demonstrate the correct implementation and suitability of OCs to be able to provide insightful information regarding epistemic uncertainty within the context of battery SOH information. Note that the procedure we use here, is one of the many possibilities to test epistemic uncertainty.

Case 3: Masking of data Consider the case in which 4000 samples are made from the following distribution (equivalent to case 2): $y_i = \sin(x_i) + \mathcal{N}\left(0, \sigma = 0.5 + 0.4 * \left(\frac{x_i}{10}\right)^2\right)$. Before training the following mask is applied ($6.0 < x < 14.0 \wedge 17.5 < x$), which effectively removes the data within said region, while during inference said mask is not applied. 2000 epochs are utilised during training. Within Figure 19 one may observe the data setup which is used in combination with the results. Within the figure the aleatoric uncertainty can be observed in combination with the model its predictions, while the epistemic uncertainty of the model can be observed at the bottom. Based on the recommendations written within [37] we opt for SQR (see Equation 10) to train the certificates.

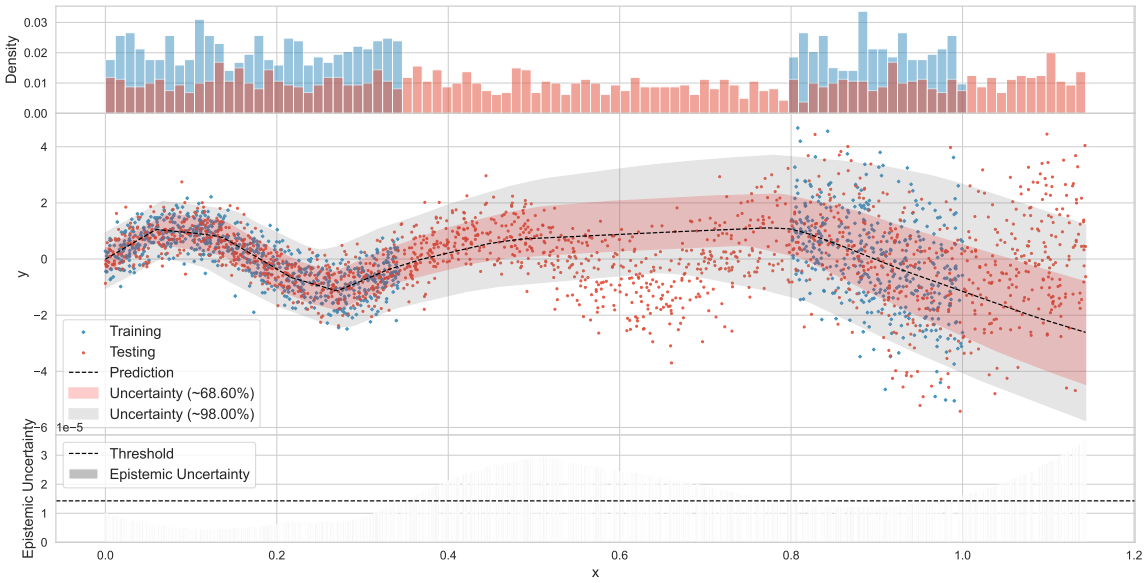


Fig. 19 Results of case 3, containing both the aleatoric and epistemic uncertainty.

Within Equation 10 the estimated epistemic uncertainty can be observed. Firstly it can be noticed that from an aleatoric point of view the model keeps providing estimates, although it does not comply with the underlying data. Two such cases can be observed within the figure, in both cases a rise in epistemic uncertainty can be observed within the bottom graph. The units of epistemic uncertainty should not be interpreted literally. The authors [37] recommend to evaluate points as out-of-distribution or adversarial when their epistemic uncertainty evaluates higher than the values encountered during training. The investigated three thresholds, namely the $\alpha = 0.9$, $\alpha = 0.95$, and $\alpha = 0.99$ quantile, for the case of this experiment we selected the $\alpha = 0.95$ quantile.

We must however note that the process of learning epistemic uncertainty was a bit more troublesome in comparison to the aleatoric uncertainty for the test cases. Firstly we noticed that the OCs where rather sensitive to the chosen hyper-parameters (certificate size, and epochs), loss function, and we achieved rather inconsistent results. Furthermore the OCs seem to struggle with masked locations close to the original training distribution. For significantly out of distribution points this behaviour was noticed to a less extent. The above presented cases was the best achieved results after numerous experiments.

Acknowledgments

We would like to declare that AI, more specifically GitHub copilot and Google AI studio, were utilised during the development process of both the code and its documentation. The tools were not utilised during the writing process of the thesis and the thesis paper.

References

[1] Clarke, L., Wei, Y.-M., De La Vega Navarro, A., Garg, A., Hahmann, A., Khennas, S., Azevedo, I., Löschel, A., Singh, A., Steg, L., Strbac, G., and Wada, K., "Energy Systems," *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, edited by P. Shukla, J. Skea, R. Slade, A. Al Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, and J. Malley, Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022. https://doi.org/10.1017/9781009157926.008.

[2] statista, "Electric Vehicles - Worldwide," Online, n.d. URL https://www-statista-com.tudelft.idm.oclc.org/outlook/mmo/electric-vehicles/worldwide, accessed: March 17, 2025.

[3] Rahimi-Eichi, H., Ojha, U., Baronti, F., and Chow, M.-Y., "Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles," *IEEE Industrial Electronics Magazine*, Vol. 7, No. 2, 2013, pp. 4–16. https://doi.org/10.1109/MIE.2013.2250351.

[4] Thelen, A., Huan, X., Paulson, N., Onori, S., Hu, Z., and Hu, C., "Probabilistic machine learning for battery health diagnostics and prognostics—review and perspectives," *npj Materials Sustainability*, Vol. 2, No. 1, 2024, p. 14. https://doi.org/10.1038/s44296-024-00011-1, URL https://doi.org/10.1038/s44296-024-00011-1.

[5] Xiong, R., Li, L., and Tian, J., "Towards a smarter battery management system: A critical review on battery state of health monitoring methods," *Journal of Power Sources*, Vol. 405, 2018, pp. 18–29. https://doi.org/https://doi.org/10.1016/j.jpowsour.2018.10.019, URL https://www.sciencedirect.com/science/article/pii/S037877531831111X.

[6] Tian, H., Qin, P., Li, K., and Zhao, Z., "A review of the state of health for lithium-ion batteries: Research status and suggestions," *Journal of Cleaner Production*, Vol. 261, 2020, p. 120813. https://doi.org/https://doi.org/10.1016/j.jclepro.2020.120813, URL https://www.sciencedirect.com/science/article/pii/S095965262030860X.

[7] Preger, Y., Barkholtz, H. M., Fresquez, A., Campbell, D. L., Juba, B. W., Romàn-Kustas, J., Ferreira, S. R., and Chalamala, B., "Degradation of Commercial Lithium-Ion Cells as a Function of Chemistry and Cycling Conditions," *Journal of The Electrochemical Society*, Vol. 167, No. 12, 2020, p. 120532. https://doi.org/10.1149/1945-7111/abae37, URL https://dx.doi.org/10.1149/1945-7111/abae37.

[8] Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., and Riu, D., "A review on lithium-ion battery ageing mechanisms and estimations for automotive applications," *Journal of Power Sources*, Vol. 241, 2013, pp. 680–689. https://doi.org/https://doi.org/10.1016/j.jpowsour.2013.05.040, URL https://www.sciencedirect.com/science/article/pii/S0378775313008185.

[9] Li, Y., Liu, K., Foley, A. M., Zülke, A., Berecibar, M., Nanini-Maury, E., Van Mierlo, J., and Hoster, H. E., "Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review," *Renewable and Sustainable Energy Reviews*, Vol. 113, 2019, p. 109254. https://doi.org/https://doi.org/10.1016/j.rser.2019.109254, URL https://www.sciencedirect.com/science/article/pii/S136403211930454X.

[10] Han, X., Ouyang, M., Lu, L., Li, J., Zheng, Y., and Li, Z., "A comparative study of commercial lithium ion battery cycle life in electrical vehicle: Aging mechanism identification," *Journal of Power Sources*, Vol. 251, 2014, pp. 38–54. https://doi.org/https://doi.org/10.1016/j.jpowsour.2013.11.029, URL https://www.sciencedirect.com/science/article/pii/S0378775313018569.

[11] Stroe, D.-I., and Schaltz, E., "Lithium-Ion Battery State-of-Health Estimation Using the Incremental Capacity Analysis Technique," *IEEE Transactions on Industry Applications*, Vol. 56, No. 1, 2020, pp. 678–685. https://doi.org/10.1109/TIA.2019.2955396.

[12] Eddahech, A., Briat, O., Bertrand, N., Delétage, J.-Y., and Vinassa, J.-M., "Behavior and state-of-health monitoring of Li-ion batteries using impedance spectroscopy and recurrent neural networks," *International Journal of Electrical Power & Energy Systems*, Vol. 42, No. 1, 2012, pp. 487–494. https://doi.org/https://doi.org/10.1016/j.ijepes.2012.04.050, URL https://www.sciencedirect.com/science/article/pii/S0142061512001779.

[13] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature*, Vol. 521, No. 7553, 2015, p. 436–444. https://doi.org/10.1038/nature14539.

[14] Nemani, V., Biggio, L., Huan, X., Hu, Z., Fink, O., Tran, A., Wang, Y., Zhang, X., and Hu, C., "Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial," *Mechanical Systems and Signal Processing*, Vol. 205, 2023, p. 110796. https://doi.org/https://doi.org/10.1016/j.ymssp.2023.110796, URL https://www.sciencedirect.com/science/article/pii/S0888327023007045.

[15] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K., "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, Vol. 9, No. 4, 2018, p. 611–629. https://doi.org/10.1007/s13244-018-0639-9.

[16] Hu, C., Jain, G., Schmidt, C., Strief, C., and Sullivan, M., "Online estimation of lithium-ion battery capacity using sparse Bayesian learning," *Journal of Power Sources*, Vol. 289, 2015, pp. 105–113. https://doi.org/https://doi.org/10.1016/j.jpowsour.2015.04.166, URL https://www.sciencedirect.com/science/article/pii/S0378775315008344.

[17] Roman, D., Saxena, S., Robu, V., Pecht, M., and Flynn, D., "Machine learning pipeline for battery state-of-health estimation," *Nature Machine Intelligence*, Vol. 3, No. 5, 2021, pp. 447–456. https://doi.org/10.1038/s42256-021-00312-3, URL https://doi.org/10.1038/s42256-021-00312-3.

[18] Shen, S., Sadoughi, M., Chen, X., Hong, M., and Hu, C., "A deep learning method for online capacity estimation of lithium-ion batteries," *Journal of Energy Storage*, Vol. 25, 2019, p. 100817. https://doi.org/https://doi.org/10.1016/j.est.2019.100817, URL https://www.sciencedirect.com/science/article/pii/S2352152X19302233.

[19] Li, Q., Zhong, J., Du, J., Yi, Y., Tian, J., Li, Y., Lai, C., Lu, T., and Xie, J., "Probabilistic neural network-based flexible estimation of lithium-ion battery capacity considering multidimensional charging habits," *Energy*, Vol. 294, 2024, p. 130881. https://doi.org/https://doi.org/10.1016/j.energy.2024.130881, URL https://www.sciencedirect.com/science/article/pii/S0360544224006534.

[20] Fan, Y., Xiao, F., Li, C., Yang, G., and Tang, X., "A novel deep learning framework for state of health estimation of lithium-ion battery," *Journal of Energy Storage*, Vol. 32, 2020, p. 101741. https://doi.org/https://doi.org/10.1016/j.est.2020.101741, URL https://www.sciencedirect.com/science/article/pii/S2352152X20315784.

[21] Wang, F., Zhao, Z., Zhai, Z., Shang, Z., Yan, R., and Chen, X., "Explainability-driven model improvement for SOH estimation of lithium-ion battery," *Reliability Engineering & System Safety*, Vol. 232, 2023, p. 109046. https://doi.org/https://doi.org/10.1016/j.ress.2022.109046, URL https://www.sciencedirect.com/science/article/pii/S0951832022006615.

[22] Zhang, Y., Zhang, M., Liu, C., Feng, Z., and Xu, Y., "Reliability enhancement of state of health assessment model of lithium-ion battery considering the uncertainty with quantile distribution of deep features," *Reliability Engineering & System Safety*, Vol. 245, 2024, p. 110002. https://doi.org/https://doi.org/10.1016/j.ress.2024.110002, URL https://www.sciencedirect.com/science/article/pii/S0951832024000772.

[23] dos Reis, G., Strange, C., Yadav, M., and Li, S., "Lithium-ion battery data and where to find it," *Energy and AI*, Vol. 5, 2021, p. 100081. https://doi.org/https://doi.org/10.1016/j.egyai.2021.100081, URL https://www.sciencedirect.com/science/article/pii/S2666546821000355.

[24] Severson, K. A., Attia, P. M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M. H., Aykol, M., Herring, P. K., Fraggedakis, D., Bazant, M. Z., Harris, S. J., Chueh, W. C., and Braatz, R. D., "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, Vol. 4, No. 5, 2019, p. 383–391. https://doi.org/10.1038/s41560-019-0356-8.

[25] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," , 2019. URL https://arxiv.org/abs/1912.01703.

[26] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D., "Temporal Convolutional Networks for Action Segmentation and Detection," , 2016. URL https://arxiv.org/abs/1611.05267.

[27] Bai, S., Kolter, J. Z., and Koltun, V., "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," , 2018. URL https://arxiv.org/abs/1803.01271.

[28] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016. http://www.deeplearningbook.org.

[29] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[30] Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y., "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," , 2014. URL https://arxiv.org/abs/1409.1259.

[31] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J., *Dive into Deep Learning*, Cambridge University Press, 2023. https://D2L.ai.

[32] Luong, M.-T., Pham, H., and Manning, C. D., "Effective Approaches to Attention-based Neural Machine Translation," , 2015. URL https://arxiv.org/abs/1508.04025.

[33] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E., "Hierarchical Attention Networks for Document Classification," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, edited by K. Knight, A. Nenkova, and O. Rambow, Association for Computational Linguistics, San Diego, California, 2016, pp. 1480–1489. https://doi.org/10.18653/v1/N16-1174, URL https://aclanthology.org/N16-1174/.

[34] Bahdanau, D., Cho, K., and Bengio, Y., "Neural Machine Translation by Jointly Learning to Align and Translate," , 2016. URL https://arxiv.org/abs/1409.0473.

[35] Zhu, Z., Yang, Q., Liu, X., and Gao, D., "Attention-based CNN-BiLSTM for SOH and RUL estimation of lithium-ion batteries," *Journal of Algorithms & Computational Technology*, Vol. 16, 2022. https://doi.org/10.1177/17483026221130598, URL https://journals.sagepub.com/doi/full/10.1177/17483026221130598.

[36] James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J., *An Introduction to Statistical Learning*, Springer International Publishing, Cham, 2023. https://doi.org/10.1007/978-3-031-38747-0.

[37] Tagasovska, N., and Lopez-Paz, D., "Single-Model Uncertainties for Deep Learning," , 2019. URL https://arxiv.org/abs/1811.00908.

[38] Basora, L., Viens, A., Chao, M. A., and Olive, X., "A benchmark on uncertainty quantification for deep learning prognostics," *Reliability Engineering & System Safety*, Vol. 253, 2025, p. 110513. https://doi.org/https://doi.org/10.1016/j.ress.2024.110513, URL https://www.sciencedirect.com/science/article/pii/S0951832024005854.

[39] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, Vol. 76, 2021, pp. 243–297. https://doi.org/https://doi.org/10.1016/j.inffus.2021.05.008, URL https://www.sciencedirect.com/science/article/pii/S1566253521001081.

[40] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, Vol. 15, No. 1, 2014, p. 1929–1958.

[41] Gal, Y., and Ghahramani, Z., "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," , 2016. URL https://arxiv.org/abs/1506.02142.

[42] Amini, A., Schwarting, W., Soleimany, A., and Rus, D., "Deep Evidential Regression," *CoRR*, Vol. abs/1910.02600, 2019. https://doi.org/https://doi.org/10.48550/arXiv.1910.02600, URL http://arxiv.org/abs/1910.02600.

[43] Lakshminarayanan, B., Pritzel, A., and Blundell, C., "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," , 2017. URL https://arxiv.org/abs/1612.01474.

[44] Pearce, T., Zaki, M., Brintrup, A., and Neely, A., "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach," , 2018. URL https://arxiv.org/abs/1802.07167.

[45] Khosravi, A., Nahavandi, S., Creighton, D., and Atiya, A. F., "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances," *IEEE Transactions on Neural Networks*, Vol. 22, No. 9, 2011, pp. 1341–1356. https://doi.org/10.1109/TNN.2011.2162110.

[46] Koenker, R., and Bassett, G., "Regression Quantiles," *Econometrica*, Vol. 46, No. 1, 1978, pp. 33–50. URL http://www.jstor.org/stable/1913643.

[47] Koenker, R., and Hallock, K. F., "Quantile Regression," *Journal of Economic Perspectives*, Vol. 15, No. 4, 2001, p. 143–156. https://doi.org/10.1257/jep.15.4.143, URL https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143.

[48] Waltrup, L., Otto-Sobotka, F., Kneib, T., and Kauermann, G., "Expectile and quantile regression—David and Goliath?" *Statistical Modelling*, Vol. 15, 2015, pp. 433–456. https://doi.org/10.1177/1471082X14561155.

[49] Cannon, A. J., "Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes," *Stochastic Environmental Research and Risk Assessment*, Vol. 32, No. 11, 2018, p. 3207–3225. https://doi.org/10.1007/s00477-018-1573-6.

[50] Terven, J., Cordova-Esparza, D.-M., Romero-González, J.-A., Ramírez-Pedraza, A., and Chávez-Urbiola, E. A., "A comprehensive survey of loss functions and metrics in deep learning," *Artificial Intelligence Review*, Vol. 58, No. 7, 2025. https://doi.org/10.1007/s10462-025-11198-7, URL http://dx.doi.org/10.1007/s10462-025-11198-7.

[51] De Pater, I., and Mitici, M., "Novel Metrics to Evaluate Probabilistic Remaining Useful Life Prognostics with Applications to Turbofan Engines," *PHM Society European Conference*, Vol. 7, No. 1, 2022, p. 96–109. https://doi.org/10.36001/phme.2022.v7i1.3320.

[52] Loshchilov, I., and Hutter, F., "Decoupled Weight Decay Regularization," , 2019. URL https://arxiv.org/abs/1711.05101.

[53] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," , 2017. URL https://arxiv.org/abs/1412.6980.

[54] de Pater, I., and Mitici, M., "Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder," *Engineering Applications of Artificial Intelligence*, Vol. 117, 2023, p. 105582. https://doi.org/https://doi.org/10.1016/j.engappai.2022.105582, URL https://www.sciencedirect.com/science/article/pii/S0952197622005723.

[55] Cantelli, F. P., "Sui confini della probabilità," *Atti del Congresso Internazionale dei Matematici*, Vol. 6, Nicola Zanichelli, 1928, pp. 47–59.

Part III

Reflection

5

Reflection on the Work Performed During the Thesis

With this chapter we would like to reflect on the work which was and was not performed in this thesis. We will provide an overview of the main contribution and core difficulties which were encountered and link this to the original research questions and objective. First in Section 5.1 we will discuss the original research questions and describe how they changed throughout the research period. Afterwards in Section 5.2 we will have a brief discussion on datasets.

5.1. Design of Research

The foundation of the research project is laid out during the literature review phase. During this phase knowledge on the domain is collected and a research plan is developed. Because of its importance we will first reflect on decisions which were made here, followed by changes which were made with regard to the original research proposal.

The main goal of a thesis was twofold, firstly it is an opportunity to develop yourself academically. Secondly the aim is to contribute to scientific literature by offering new insights into a certain knowledge or research gap. Because of the popularity of batteries in recent years, it is of no surprise that a tremendous amount of research has been performed across various disciplines. After reading a lot of papers on both state of health estimation methods and uncertainty quantification techniques we observed that comparatively little effort had been performed on integrating uncertainty aware deep learning techniques and battery state of health estimation methods. Since we had, no prior experience in deep learning, nor in batteries, this topic would definitely be very challenge, however at the same time it would provide us with the opportunity to learn something new, while still contributing to the field. Thus besides formulating a useful and applicable research question, we had to ensure the task remained feasible and realistic.

The original research objective defined in Chapter 3 was to better understand the uncertainty of battery state of health estimations. Similarly the main research questions was stated as follows: How can the uncertainty of lithium ion battery state of health be accurately predicted through the application of deep learning methods? As can be observed in the research paper, both the research objective and question remained the same to what we proposed originally. We successfully achieved this by presenting a new deep learning model taking charge data as input and providing a discharge capacity with confidence interval as output. We afterwards performed a detailed analysis of the quality of our predictions using a mixture of point prediction evaluation metrics, uncertainty evaluation procedures and specialised test cases. We conclude that the proposed model is able to reliably estimate both aleatoric and epistemic uncertainty, however a few challenges remain.

If we now focus on the sub-question which were developed, we observed that 3 out of 4 sub-questions were answered. Namely, we created a deep learning model and implemented a procedure to assess the effectiveness of its predictions. However if we focus on the methodology in the sub-questions, 2 changes were made. First, the original idea was to further improve an existing technique, by making use of expectile regression instead of quantile regression. The technique seemed very promising, since according to literature, improved calibration and monotonicity could be obtained. However the technique turned out be much more mathematical then initially expected and as results the idea was discarded. Instead we made use of a two part procedure referred to as simultaneous quantile regression and orthonormal certificates to be able to quantify both aleatoric and epistemic uncertainty.

The second change is related to the feature extractor which would be used in our research. Initially when creating our research design, we had developed a general idea of the current state of research in the domain of battery state

of health estimation. Here the typical approach was to convert the input time series into a fixed length sequence, which is then provided to the model. Unfortunately we found that this approach does not work for the fast charging procedure and thus we moved to variable length input sequences. In this cases the conventional convolutional neural network was no longer a valid option and as a result an approach compatible with variable length input sequences had to be selected. Therefore to conclude, the sub-questions were answered, however implementation details (methods) were altered in the process. We believe that the final method is a significant improvement to what was stated in the sub-questions originally.

Typically the last step of a deep learning process is to compare the approach used in our paper, with other approaches. In our thesis this step was defined as sub-question 4. During development we had encountered significant issues with model development as well as model stability. In our discussion we highlighted that the stochastic loss function could possibly explain this issue. On the other data standardisation (resampling and accurate normalisation) of the signals in the Toyota datasets was a particular challenge. More specifically, the varying sequence length, may also cause the model to fixate primarily on input length. However we found that proving this was particularly difficult, since experiments would contradict each other. Both aspects cause the development to take much longer than expected. The consequence of this was that we were unable to perform the task we initially intended to complete, even though most of the models for benchmarking have been developed. I believe that this is also an important lesson. We put a lot of effort into maximising the performance and improving stability of the model. However, in reality it would have been more valuable to formulate an answer to sub-question 4, by exploring an additional uncertainty estimation technique or dataset.

This thesis topic has primarily been a very enjoyable learning experience. Before I started with this topic, I knew nothing about deep learning and batteries, thus the topic was definitely overwhelming, ambitious and challenging. This project allowed me to explore numerous aspects related to data analysis from the beginning to the end. Topics which initially seemed a mystery to me have become much more clear and I thoroughly enjoyed the countless hours of reading into the topic. Throughout this thesis a series of mistakes were made, which would maybe not have happened if I had prior knowledge of the topic. Simply being able to deal with and process the sheer scale on information being available to us, has been a positive learning experience. But most importantly we learned from these mistakes, got better at processing academic information, explored a very fascinating topic, and presented an interesting methodology to contribute to our research objective.

5.2. Difficulty of Finding Appropriate Datasets

Deep learning is a method notorious for requiring a significant amount of high quality data to be able to train, and afterwards accurately validate and test a model. This is of course to be expected, since we expect a model to learn intricate process by itself. We observed that within the field of batteries a lot of high-quality data is available. These are being published by numerous research institutes such as NASA, Toyota research institute, CALCE, and Oxford. The issue however is that most of these datasets are limited in size, this issue only becomes more apparent when considering uncertainty quantification. Here many of the proposed evaluation metrics depend on the availability of large and diverse validation and tests sets. Initially we devoted a lot of our attention to the two datasets published by NASA, since these were popular datasets for machine learning and deep learning in literature. However, in hindsight we should have realised earlier that they were simply to small for our application. The key lesson learned here is that during the initial research design it is important to not only have a wide view of the available datasets, but also perform preliminary basic analysis of the datasets, to limit losing time during the main development phase.