CHALMERS
UNIVERSITY OF TECHNOLOGY

Early Detection of Rare Events: Predicting Battery Cell Deviations

Master's thesis in Complex Adaptive Systems

Jesper Boberg and Anders Segerlund

Department Of Physics

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Early Detection of Rare Events: Predicting Battery Cell Deviations

Jesper Boberg
Anders Segerlund

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2023

Early Detection of Rare Events:
Predicting Battery Cell Deviations
Jesper Boberg
Anders Segerlund

Supervisors: Herman Johnsson, Pehr Norström and Cecilia Dano, Volvo Cars
Examiner: Giovanni Volpe, Department of Physics, GU

Master's Thesis 2023
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Concept art showing the Volvo XC Coupe electrical car.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Early Detection of Rare Events:
Predicting Battery Cell Deviations
Jesper Boberg and Anders Segerlund
Department of Physics
Chalmers University of Technology

Abstract

Despite rigorous quality controls in battery cell production, the production process is still subject to quality deviations. These quality deviations; known as "rare events", initially act as passive quality deviations which may not affect the battery's performance. However, a passive quality deviation can transition into an active deviation that may give rise to behavioral issues in the battery cell at some point during its lifetime. An active quality deviation can cause the entire battery to misbehave and eventually fail. This thesis investigates the possibility of predicting these cell deviations in car batteries. Better predictions of these events would avoid expensive and troublesome car failures and enable preventive car maintenance to solve the problem.

In this report, different models are created and evaluated with the aim of preventing these deviations. The dataset is supplied by Volvo Cars and contains a large amount of data collected from Battery Electric Vehicles (BEVs), where the arguably largest challenge comes from the imbalance of the dataset. In addition to the modelling, the thesis includes a thorough data analysis with the aim of improving both the dataset itself and the data collection process at Volvo Cars.

These deviations occur extremely rarely, making a relatively large amount of false positives difficult to avoid. The results show that a simple time series model can catch these deviations relatively well but also brings along a large number of false positives. A neural network is able to improve this significantly, still being able to catch the majority of the deviations while producing a lot fewer false positives.

Keywords: Battery failures, Cell Deviation, Recurrent Neural Network, Time Series Analysis, Machine Learning, Multilayer Perceptron, Predictive Modelling, Physics, Car Batteries, Volvo Cars.

Acknowledgements

We want to thank our technical supervisor Herman Johnsson who has constantly been supporting us, answering any and all questions we have had. We have learnt a lot from you, thank you for that. We would also like to direct a warm thank you to our supervisors Pehr Norström and Cecilia Dano. Thank you for initiating this thesis, allowing us to do this project at Volvo Cars. Your continuous support and backing throughout the thesis has been great, you have made us feel very welcome. Furthermore, we want to thank Leopold Werberg for sharing some of his great expertise in batteries and especially for the help with the data labeling. Lastly, we want to thank Giovanni Volpe for agreeing to be the examiner of this thesis.

Anders Segerlund and Jesper Boberg
Gothenburg, 2022

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BEV	Battery Electric Vehicle
HEV	Hybrid Electric Vehicle
SoC	State of Charge
BMS	Battery Management System
MLP	Multilayer Perceptron
LSTM	Long Short-term Memory
RNN	Recurrent Neural Network
TIN	Temporary Identification Number

Contents

List of Figures

List of Tables

Report Structure

The structure of this thesis is chosen with the aim of presenting the work in a scientific, fair and understandable manner. This has resulted in a modified IMRaD-structure. This structure is briefly explained here to get a better overview and understanding of how to read this report.

Chapters 1-3 contain the introduction along with literature review sections. In large part this corresponds to the standard *Introduction*-part of the IMRaD-structure.

Chapters 4-5 aim to give a descriptive analysis of the dataset along with the preparatory work that is done before the machine learning models are created. This includes the chapter *Defining Battery Failures and Data Labeling* which account for how battery cell deviations are defined and furthermore how the data labeling is performed. This is necessary to have as background when continuing to the remaining chapters.

Chapters 6-9 follow the remaining parts of the IMRaD-structure. Chapter 6: *Methods*, describes the methods used in this thesis, both with data preprocessing, model setup and model evaluation. Chapter 7: *Results*, presents the results that are produced from the models before these results along with the methods and data are discussed in chapter 8: *Discussion*. Lastly, the thesis is concluded in chapter 9: *Conclusion*.

1
Introduction

1.1 Background

With increasing energy consumption, shortage of fossil fuels and an urgent need to drastically reduce the environmental impact of energy use, one significant challenge is to find substitutes to conventional energy systems that still yield equal performance with regards to e.g. reliability and durability.

Electrification of the transport sector represents one of the most critical transitions in today's industry, and the European Union has set up the goal to make it climate-neutral by 2050 [Fet20]. The transition is already in full progress and accelerating, which creates new challenges in order to keep up high-quality standards [EEA21]. New and rapid advances in battery technology have undoubtedly enabled the ability to substitute conventional drivelines with drivelines powered by electricity. On the other hand, the same advances also bring new challenges.

Due to their high energy density, power and efficiency, lithium-ion based batteries have emerged as the most reliable energy storage alternative [Wag+13]. With Li-ion battery cells being high capacity and series connected, limitations are imposed on the application of the Li-ion cells in a vehicle. Li-ion cells must undergo a series of quality control tests before being approved for assembly in a vehicle. A critical component in a li-ion cell is the separator between the anode and cathode. This separator serves as an isolation membrane and is essential for the safety and performance of the battery cell. Potential quality defects may cause undesirable customer-facing outcomes such as draining of the battery or unexpected shutdown. [Hof+21] states that despite rigorous quality testing within Li-ion cell production, quality control processes may be unable to detect minor but still relevant defects in the separator. Manufacturing variance may lead to nonuniform electrode coating, pinholes, particle-contamination in the separator or electrode misalignment, which may negatively impact the performance and manifest as a safety risk. Defects originating from the manufacturing process may lead to thermal runaway or short-circuit. According to [Kon+22], the failure rate of a Li-ion battery is estimated to be 0.025 - 0.1 PPM which is significantly less compared to the general industry standard for quality deviation. Yet, the Li-ion failure rate should be viewed with regards to the fact that there are hundreds of cells in a BEV, and there is a transformation ongoing to replace all production of cars with Internal Combustion Engines (ICE) with fully electric engines, meaning that in the coming decade, there will be a large number

of battery cells on the field with a potential failure. Further, a cell with inherent contamination might be fully functional for several years until the contamination gives rise to a pinhole in the separator and the emergence of a short circuit.

1.1.1 Battery Cell Failures

Battery cell failures on the field can be distinguished into two types of categories; deviations and degradation. According to [WA22], all types of batteries have a natural degradation and will wear out over time. As illustrated in figure 1.1, the normal type of degradation will increase towards the end of the battery's life.

Deviation, on the other hand, is referred to as a latent defect which most often was created during manufacturing and progressed into an active defect as the battery is used. [WA22] argues that deviations typically are found within one year of the battery's life. Therefore, failures found prior to this time point are very likely to be caused by deviations rather than degradation.

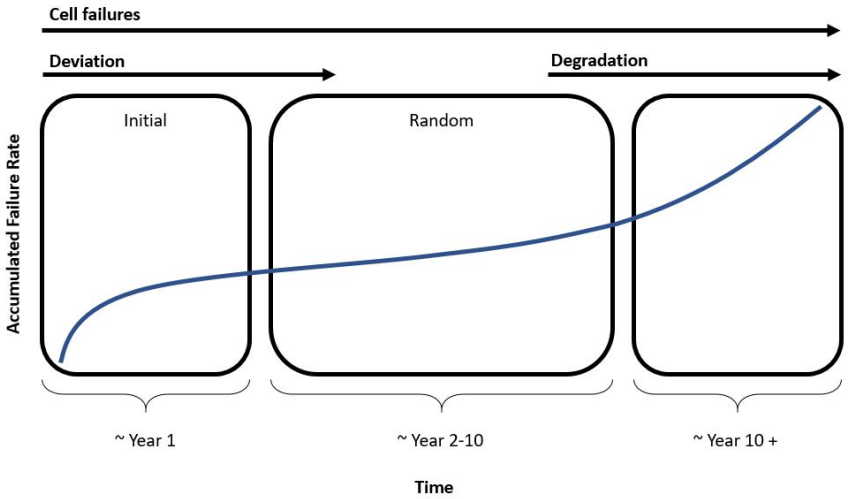


Figure 1.1: Mapping of deviation vs degradation [WA22].

Based on the outlined knowledge about quality deviations, the aim is to analyse how deviations may be reflected in the data. Although there is not a single root cause to a deviation, the report will view deviations as a result of a leak hole in the separator for limitation purpose and scope of the thesis. This is done to build a theoretical understanding of the phenomenon and its impact on the data.

1.1.2 Business Case

At Volvo Cars, the transformation to electric cars is transpiring very rapidly. The company is committed to becoming a fully electric car company by 2030 [22]. To keep the high quality associated with the brand, it is of uttermost importance that the cars run smoothly with as few disruptions as possible for the customer. Of equal importance is that on the rare occasion that a disruption occurs, it is handled

in a customer-friendly manner. In addition to enabling the best possible customer experience, the company's cost also has to be considered.

To reach this high standard in the context of battery failures, methods to predict these failures need to be evaluated with all aspects mentioned above in consideration. A failure needs to be predicted and handled before it actually occurs and causes problems for both the customer and Volvo Cars. The further ahead of a failure the prediction can be made, the better, however finding a failure more than a month before failure needs to be assessed as it might add relatively low value to the business case. Another critical assessment of a method for predicting failures is how many false positives are entailed with the actual true positives. False positives, meaning cars that the method predicts will fail but actually won't, cause unnecessary trouble for the customer along with high costs for the company. Finding a balance between not missing any failures and having as few false positives as possible, is not easy, and preferably the method used should be flexible in this aspect. As a guideline, the method should lean towards prioritizing not missing any failures. Most of the evaluation in this report prioritizes finding all failures and secondly minimizing the false positives. This enables the method to at least be used as a filter to find potential failures, which can then be further evaluated.

1.1.3 Hypothesis

Figure 1.2 represents a hypothesis based on the background. This figure illustrates how deviations in voltage or SoC may emerge once a passive fault transitions into an active fault. Given the assumption that a fully functional battery is put into use at time t_0 with a passive fault (e.g. a dust particle or contamination between the separator and the anode/cathode), at some point in time t_1, the passive fault becomes an active fault. At some point in time t_2, the deviation becomes "visible" and detectable to a machine learning model before a battery failure occurs at time t_3.

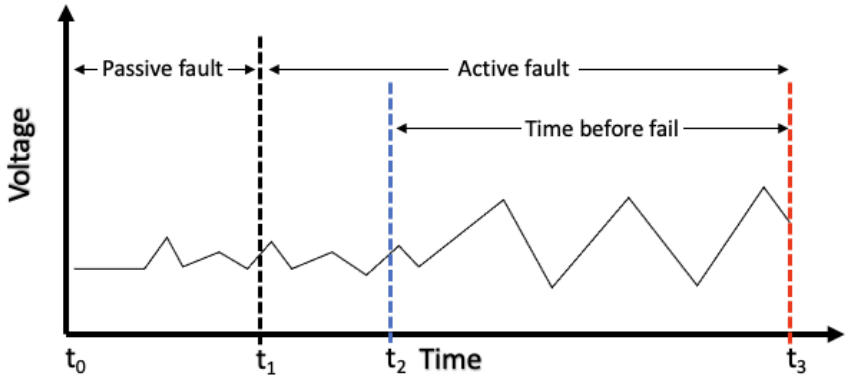


Figure 1.2: Illustration of expected cell-voltage behaviour given the assumption of emergence of leak-holes.

1.2 Aim

This thesis aims to investigate the possibility of predicting battery failures in BEVs. This will primarily be done by creating and comparing different machine learning models which attempt to predict these failures as early as possible. Hence the aim will mainly be to improve and evaluate the performance of these models. Moreover, different methods and models will be researched and evaluated in order to acquire knowledge about both the models themselves and their capability to solve the problem at hand.

In addition to the machine learning model creation, a data analysis will be performed. A large part of a project like this is understanding the data and getting the process from data collection to model evaluation to work as intended. The purpose of this is partially as groundwork for the predictive modelling but also as an evaluation of the dataset itself, which can then be used as a basis for Volvo Cars to improve their data collection process. The ability to predict a failure allows treating the problem preemptively, improving customer experience and generating value for the company. Furthermore, this type of modelling could hopefully be helpful in other areas as well.

1.3 Previous Literature

Spontaneous deviations, such as internal short circuits that sporadically occur during operation is, according to [Fen+18], an unsolved problem that hinders the widespread application of Li-ion batteries. As concluded by [WA22], there is no single root cause of a deviation in a battery cell, and as a consequence, previous literature related to the prediction of deviations encompass a wide range of various methods for detecting such deviations. For example, [Fen+18] presents a model-based fault diagnosis algorithm where voltage and temperature are transformed into an intrinsic electrochemical status reflecting the typical internal short-circuit features. This model-based method is also used by [Seo+20], who also incorporates internal resistance in the battery cell to detect deviations that could potentially stem from a manufacturing defect. Yet, the authors imply that degradation due to calendar aging or cycle aging will affect the practicability of the model.

A more general approach is taken by [Ard+20], which uses various types of machine learning models for fault detection, such as supervised, unsupervised and reinforcement learning. In contrast to the previously mentioned publications, this research does not relate fault detection to any particular root cause or take exact chemical processes into account. Rather the machine learning model uses the battery data to extract feature information and build a model to predict the remaining life of the battery.

1.4 Limitations

Even though the data is collected continuously from the Volvo Cars fleet, the model created will be based on a stationary dataset collected until April 2022. A possible extension to the model would be to have it be regularly updated as more data is collected, but no special consideration to this is taken in this report.

It should also be noted that the data used is from a specific car fleet which does not necessarily generalize to all BEVs in all regions, more on this in the data section 4.1.

1.5 Disclaimer

In order to preserve privacy and Volvo Cars' intellectual property, there are limitations to what can be shared in this report. All data throughout the project and this report are anonymous. Some of the aggregated quantities from the data will not be explicitly presented, some can be declared as an order of magnitude, whereas some will be completely left out. It should also be noted that the dataset used in this thesis is retrieved from a relatively small subset of the Volvo Cars BEV fleet. Therefore, the dataset and the ensuing analysis are not necessarily representative for all Volvo cars fleets, but the produced methods will likely be transferable.

2
Battery Literature

This chapter will introduce the reader to various cell battery types, their chemistry and how they operate. The focus will be on Lithium-ion (Li-ion) cells as this is the type of battery used in BEV/HEV cars, thus the scope of the thesis. This chapter about battery literature is based on the book *"Lithium-Ion Battery Failures in Consumer Electronics"* by [Aro+19]. It will serve as a light introduction to battery cell theory to provide a comprehensive understanding of the pros and cons of Li-ion cells and how they operate in general. Thus, competing battery chemistries will only briefly be addressed.

2.1 Lithium-Ion Cells

Lithium-ion technology has become indispensable due to its wide application ranging from portable consumer electronic devices and vehicles to large grid-based energy storage systems. Compared to competing cell chemistries such as lead acid, nickel metal hydride (NiMh), nickel cadmium (NiCd), Li-ion cells have a superior energy density and can store a greater amount of energy in a smaller and lighter package. Lithium is the third lightest element and additionally has one of the smallest ionic radius of any single charge ion, lithium-based cells allows having comparatively higher volumetric capacity and energy density, making them ideally suited for consumer electronic devices and electric vehicles.

2.2 Competing Cell Chemistries

Although the Li-ion cell is the undisputed selection for BEV/HEV cars, several other cell chemistries have their own significant advantages. To fully understand the advantages and disadvantages of the Li-ion cell, the most relevant competing cell chemistries will be briefly outlined below.

2.2.1 Lead Acid

Lead acid batteries represent the oldest recharging battery technology and are still widely used for e.g. vehicle startup and industrial applications. With a simple and inexpensive manufacturing process, lead acid batteries provide a significant advantage in terms of cost per watt-hour. Lead acid batteries consist of a dioxide cathode and an acid solution serving as the electrolyte, which tend to make the batteries bulkier and thus carry a very poor power-to-weight ratio. Lead acid batteries must

be charged slowly and stored in charged condition to prevent sulfation, making them inappropriate as batteries in BEV cars.

2.2.2 NiCd

Nickel-cadmium batteries (NiCd) are rechargeable batteries that use nickel hydroxide as a cathode, metallic cadmium as an anode and potassium hydroxide as the electrolyte. Compared to lead batteries, NiCd-batteries have longer life and require little maintenance. However, NiCd cells are more expensive than lead acid and also have a higher self-discharge rate and a lower cell voltage. The cell is also subject to a "memory effect", where the cell "remembers" the points in its discharge cycle where recharging began, and during subsequent use, a sudden voltage drop occurs at that point. To overcome memory effects in NiCd cells, it needs periodic full discharge. One advantage with NiCd cells is that they perform very good under rough conditions, and thus are suited for portable usage. They tolerate deep discharge and in contrast to Li-ion cells, they can be stored fully discharged.

2.2.3 NiMH

Nickel-metal hydride batteries (NiMH) are similar to NiCd cells where nickel hydroxide is used as the cathode. Still, hydrogen is the active element in a hydrogen absorbing anode instead of using cadmium as an anode. NiMH batteries have much higher energy density compared to NiCd cells, and can contain two or three times the capacity of NiCd batteries of the same size, although much less than Li-ion batteries. NiMH batteries were frequently used in the prior generation of electric and hybrid vehicles, but as of 2020 they have been superseded by Li-ion batteries.

2.3 Lithium-Ion Operation

The Li-ion technology is based on the intercalation principle shown in figure 2.1. Unlike a lead-acid battery cell, the electrolyte is not involved in a chemical reaction. Instead, the lithium ions intercalate and deintercalate reversibly into and out of the respective lattices chemistries. The anode and cathode serve as the host for the lithium-ions. They are separated by the porous film (separator), and in contrast to a lead acid cell, the electrolyte in a Li-ion cell consists of an organic solvent with a dissolved lithium salt.

According to [Aro+19], to increase the Li-ion cell's power capacity, it is desirable for the anode and cathode to have large geometric areas with high porosity to increase the reaction area.

2.3.1 Separator

The separator in a Li-ion is a permeable membrane between the battery's anode and cathode, forming a microporous layer. Usually produced by a polymeric material due to its chemical and electrochemical stability, along with its strong mechanical

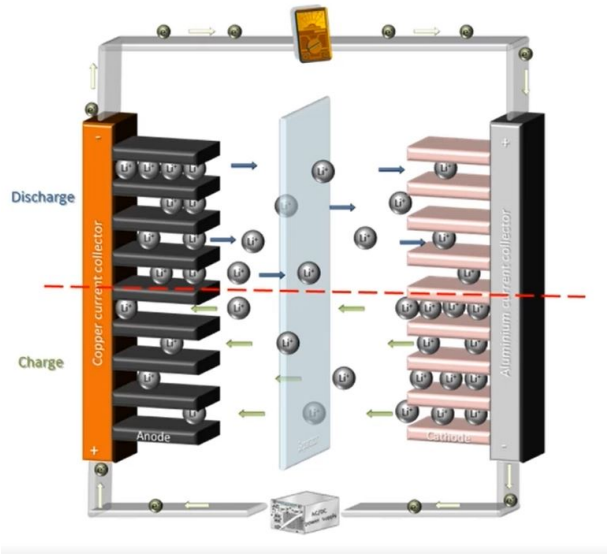


Figure 2.1: Intercalation concept for lithium-ion.

properties to withstand the high tension during battery construction. The separator is essential because the structure and properties considerably affect the battery performance, including energy and power density, cycle life and safety. The primary purpose is to act as a barrier between the electrodes to prevent internal short circuits, while also allowing the transport of ions. An illustration of a separator can be seen in figure 2.1.

According to [Aro+19], another important safety aspect of the separator is to shut down the conductivity, if the cell temperature rises above a certain level.

2.3.2 Cell Form Factors

Li-ion cells come in a variety of different form factors. Cylindrical cells represent the most well-recognized type of battery cell used by regular consumers, where the electrodes are rolled and packed into a cylindrical can. The three main form factors in battery electric vehicles, along with cylindrical batteries, are prismatic or pound cells. In pouch cells, the electrodes are stacked of separate components in a laminated architecture inside a pouch which can be seen in figure 2.2. Prismatic

Figure 2.2: Image of pouch cell.

cells are encased in aluminum or steel for stability. In contrast to pouch cells the electrodes are long continuous components encapsulated in a "jelly-rolled" manner as can be seen in figure 2.3.

Figure 2.3: Image of prismatic cell.

2.4 Battery Management System

Li-ion cells have to be operated within a specific, well-defined set of conditions in order to prevent an elevated risk of failure. Thus, all Li-ion cells include a battery management system (BMS) in order to prevent the battery cell from operating outside its rated specifications. In contrast to regular consumer electronics, where the BMS system may use only a single cell, the BMS system in a BEV/HEV car can operate hundreds of battery cells with relatively high voltage and current. Consequently, [Aro+19] states that the BMS system in a BEV/HEV car must be comparatively more sophisticated than those in regular consumer electronics.

Accordingly, the BMS systems typically provide the following functions.

- Ability to determine the battery's state of charge and state of health.
- Ability to control the charging and discharging of the cells to prevent the cells from operating outside their rated temperatures.
- Ability to balance the cells (e.g. by applying an artificial load to cells with higher state of charge value).
- Thermal management.
- Communication of battery status to a user interface.

2.5 Types of Deviations

A cell deviation is defined by comparing the deviating cell against the expected cell behaviour. In a battery pack with hundreds of cells, the analysis facilitates by having several cells to benchmark against. Below, four types of deviations are outlined that have been highlighted by [WA22].

2.5.1 Low Voltage / Self-Discharge / Soft Short

Low voltage deviation, also known as self-discharge or soft short deviation, is the most common failure mode for BEV cars according to [WA22]. It accounts for approx 98 percent of all battery cell deviations. Soft short is often called self-discharge, which might be a bit misleading since all cells have a natural self-discharge. But when referring to self-discharge as a deviation, it is referred to as a cell that is discharging at a higher rate than expected. This can be observed by the Battery Management System (BMS) as an imbalance in the battery pack, and the BMS system will trigger a fail code if the imbalance is greater than the normal SoC/Volt variance criterion caused by the expected tolerance spread.

There is rarely one single root cause of a soft short deviation, but it is often caused by some sort of leakage between the anode and cathode. One example of leakage between the anode and cathode can be due to a pinhole in the separator, which can be seen in figure 2.4. According to [WA22], pinholes are likely to result from a hidden fault from the production stage, such as dust particles, contamination or any kind of puncture in the separator during the production phase.

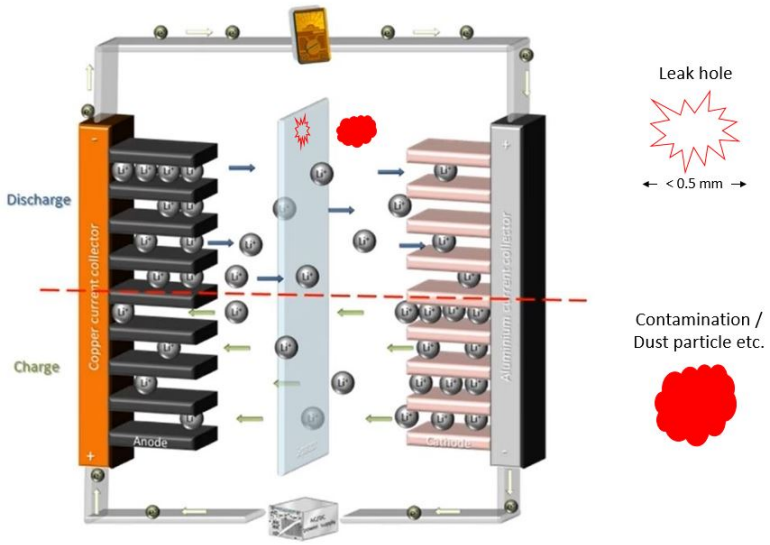


Figure 2.4: Example of leak hole in separator.

2.5.2 Low Capacity

Low capacity accounts for approximately one percent of all deviations and occur when a cell continuously has a lower capacity than the other cells in the battery pack. This phenomenon occurs when one cell has a higher voltage and SoC than the other cells in the pack. This might sound contradictory, but the phenomenon is often illustrated with water buckets. If the volume of the water bucket represents the cell capacity, a smaller bucket i.e. lower cell capacity, will be filled with water much faster than the other buckets.

Low capacity can arise if there is a pinhole for example in the pouch surrounding the electrodes. The pinhole is not severe in itself, instead the electrolyte may leak out, and moisture gets into the cell, causing the area close to the pinhole to swell as gases are created. This causes a drop in the adjacent cell capacity, and eventually, the cell dies as there can be no proper chemical activity in the cell that transfers ions. An example of a pinhole in the battery pouch can be see in figure 2.5.

Figure 2.5: Illustration of puncture in pouch cell.

The BMS system can also identify this, as the imbalance among the cells is greater than the variance criterion caused by the expected tolerance spread. As with soft short deviation, there is rarely one single root cause of low capacity, rather there can be several root causes.

2.5.2.1 Low Isolation

Low isolation accounts for approx one percent of the deviations, and is defined as a cell with a lower resistance value from the electrodes to the exterior of the cell. This deviation may occur if there is some physical defect caused by external damage that is causing a short circuit between the cell and for example the car chassis.

2.5.3 Spontaneous Internal Short Circuit / Hard Short

A spontaneous internal short circuit, also known as a hard short deviation, occurs if heat and gases are generated within the battery cell, causing swelling of the battery. This can propagate to neighbouring cells in the battery pack. Hard short deviations account for the most severe type of deviations, but occur extremely seldom.

3

Modeling Literature

This chapter explains and defines the models and evaluation metrics used in the report.

3.1 Model Selection

In a study by [Shi+17], time series analysis was conducted to predict anomalies in time series data. The authors argue that despite the high effectiveness of neural networks and other potential methods for predicting anomalies in time series data, they often require clear labels in order to build a trainable model. The authors further argue that more advanced models often become more effective when several features are incorporated into the model. The authors finally state that based on how a problem is defined, what counts as an anomaly can vary based on the data, and thus each problem may require its own model.

In the study [Ism+19], the authors show how neural networks have shown significant improvement and potential in the field of time series classification. Especially when working with multinomial time series, deep neural networks should be able to outperform simpler models. Being able to handle more features and effectively creating advanced decision boundaries, they are a good option when striving to improve model performance compared to simpler models.

3.2 Evaluation metrics

The evaluation metrics described below are defined in [Fat+08].

3.2.1 Confusion Matrix

A confusion matrix displays the distribution of true negatives, true positives, false negatives and false positives. Either in absolute terms or in percentages. An example of a confusion matrix can be seen in figure 3.1.

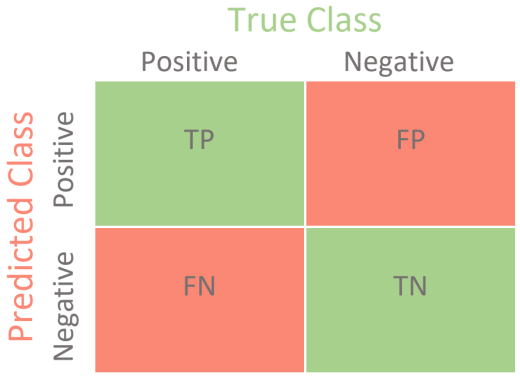


Figure 3.1: Confusion Matrix example, shows how True/False Positives/Negatives are displayed.

3.2.2 Sensitivity, Specificity and Precision

Sensitivity, also known as recall or true positive rate, is the fraction of true faults found by the model among all the actual faults.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \tag{3.1}$$

.
Specificity, also known as the true negative rate, computes the accuracy of the negative/healthy class:

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Positives+False Negatives}} \tag{3.2}$$

Precision measures the fraction of the true positives among all the model-classified positives:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \tag{3.3}$$

3.2.3 F-beta Score and Geometric Mean

The Geometric mean (G-mean), combines Sensitivity and Specificity as:

$$\text{G-mean} = \sqrt{\text{Sensitivity} \cdot \text{Specificity}} \tag{3.4}$$

$F_\beta-$score is a weighted mean of the precision and sensitivity with β as the weighting parameter.

$$F_\beta = (1+\beta^2)(\frac{\text{Precision} \cdot \text{Sensitivity}}{\beta^2 \cdot \text{Precision} + \text{Sensitivity}}) \tag{3.5}$$

3.3 Models

Many systems are thought to have a tipping point where a critical transition to a contrasting state can occur. Because a critical transition can occur unexpectedly,

there is a need to identify when a critical transition is approaching. The transition may be a new trend in the time series or a sudden deviation. This section will explore methods that can be used to identify the proximity of a system to a critical transition using a variety of so-called "early-warning" signals using time series analysis.

3.3.1 Rolling Median

A rolling analysis, or moving average of a time series model, is often used to assess the model's stability over time, according to [ZW03]. A rolling average allows assessing information over a window of time. When rolling median is applied in financial applications, a common technique is to evaluate the consistency of a time series. If the estimates over a rolling window change, the rolling median should capture instability. A weakness with rolling median is that small windows lead to more noise than the signal. The signal will be more accurate with a smaller window size but will pick up more noise compared to a larger window size and vice versa. Another issue with the rolling median is that it does not describe any future behaviour, all it does is describe trends in the data which have already occurred. The rolling median is calculated by sorting the values of the rolling window and taking the middle one. If there is an even amount of values in the rolling window, the mean of the two middle values is taken.

3.3.2 Exponential Weighted Moving Average (EWMA)

Along with the weaknesses of the rolling median, another problem with a moving average is that the trend is constrained to the same moving window size. These effects can be reduced by applying more weight to the more recent values. EWMA is widely used in technical analysis and volatility modeling by applying weights to the time series values. More weight is usually applied to more recent data, making them more relevant for future forecasts [Rad21].

Applications of EWMA have been widely used in anomaly detection. In a research by [Sun+06], an EWMA-based algorithm with a threshold-based schema is used in order to detect movement patterns in user data influenced by Markov dynamics.

3.3.3 Rolling Standard Deviation

Moving standard deviation is a statistical measurement that measures time series volatility. In contrast to the moving average, the rolling standard deviation does not make any prediction of the trend of the time series. The application of rolling standard deviations is not as widely used as SMA or EWMA, and is typically applied for detecting point anomalies in data, such as noise.

3.3.4 Multilayer Perceptron (MLP)

A Multilayer Perceptron is a common and quite basic type of neural network. It is a fully connected feedforward neural network that commonly uses backpropagation as a training method to optimize its weights. Each layer has an activation function, allowing the model to learn more complex structures in the data.

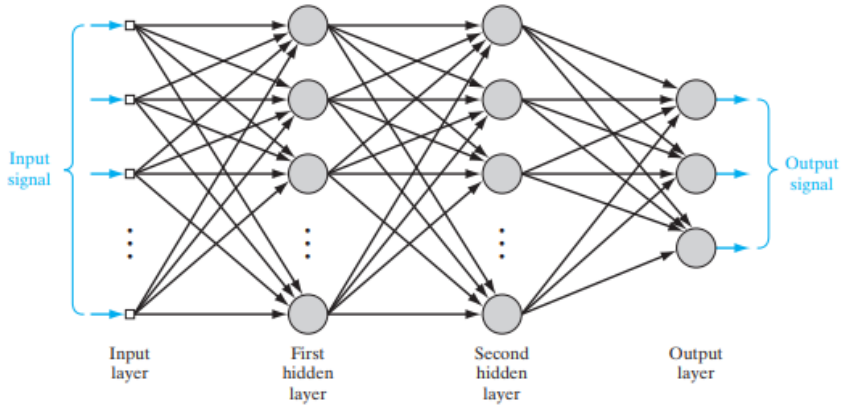


Figure 3.2: Architectural graph of a Multilayer perceptron with two hidden layers.

The number of layers and neurons in each layer is flexible and determines the depth of the network. The deeper the network is, the more advanced structures the model can learn. However it also comes with longer training times and having a too deep network will not benefit model performance. The output signal of the network can be followed by a decision boundary allowing the model to be used for binary classification [Hay09].

3.3.5 Long Short-Term Memory (LSTM)

An LSTM network is a type of Recurrent Neural Network (RNN), which unlike an MLP has cyclic feedback connections, allowing it to learn sequential data and not just single data points. A common structure for an LSTM unit is composed of a forget gate, input gate, output gate and cell state. A visualization of this can be seen in figure 3.3. [Ola15]
In each forward propagation, the values pass through all parts of the LSTM unit and perform a set of computations [Oku21].

Forget gate: This part of the unit controls what information to throw away and how much that should be remembered. The computation is shown in equation 3.6.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3.6}$$

where $f_t \in (0,1)^h$ is the forget gate's activation vector, $h_{t-1} \in (-1,1)^h$ is the hidden state vector, x_t is the input vector for the unit and W_f, b_f are the weights and biases for the forget gate. σ is the activation function.
Input gate: In the input gate, the cell state for the current input is added and decide

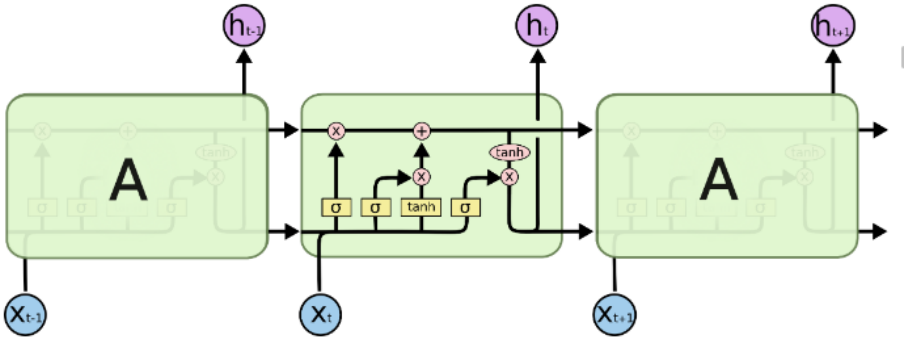


Figure 3.3: Architectural graph of a LSTM network.

how much should be added to the updated cell state as shown in equation 3.7.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C_t} &= tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \end{aligned} \tag{3.7}$$

where $i_t \in (0,1)^h$ is the input gate's activation vector, $\widetilde{C_t} \in (-1,1)^h$ is the cell input activation vector and W_f, W_C, b_f and b_C are the weights and biases for this gate.
Output gate: Determines how much of the current cell state should be carried on to the output. The computation is shown in equation 3.8.

$$\begin{aligned} i_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * tanh(C_t), \end{aligned} \tag{3.8}$$

where $o_t \in (0,1)^h$ is the output gate's activation vector, W_o, b_o are the weights and biases for the output gate and h_t is the new hidden state value.
Cell state: Lastly the current cell state is updated according to equation 3.9.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}, \tag{3.9}$$

where C_t is the new cell state vector, f_t the forget gates activation vector, C_{t-1} the old cell state vector, i_t the input gates activation vector and $\widetilde{C_t}$ the cell input activation vector [Oku21].

4

Descriptive Analysis of the Dataset

This project uses real-life BEV data collected from a Volvo Cars BEV fleet. The data is collected continuously as the cars are being used, providing a good opportunity to study not only a large number of cars but also individual battery cells as time series.

4.1 Data Overview and Basic Analysis

The dataset contains readouts from BEVs between the dates 2020-11-11 and 2022-04-24. Even though data is collected continuously and data from later dates is available, only readouts up to 2022-04-24 will be used. This keeps the dataset consistent and makes results comparable. The dataset contains approximately five million readouts from around 20,000 unique BEVs. It should be noted that this is merely a relatively small subset of the Volvo Cars BEV fleet. In figure 4.1, a histogram of the number of samples from each car is displayed.

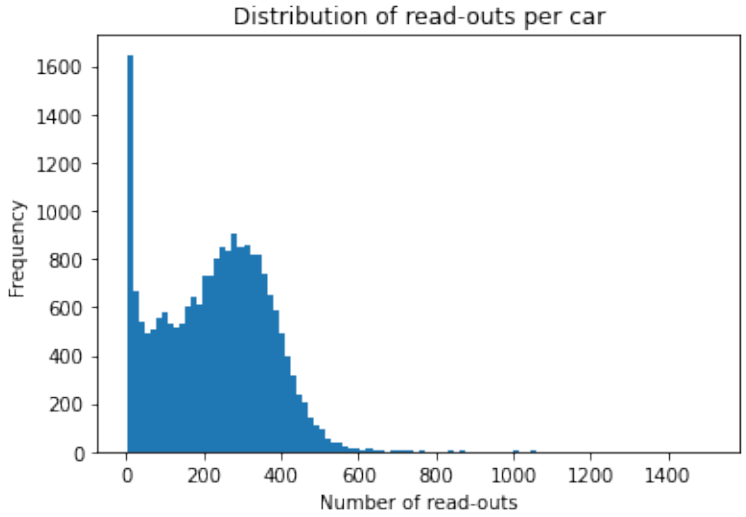


Figure 4.1: Histogram displaying the amount of read-outs for each car in the dataset. The majority of cars have generated at least a few hundred samples.

One factor to consider when looking at these types of datasets is how frequent and regular the read-outs from each car are. The average time between two read-outs

from a car is approximately 1.5 days. In figure 4.2, the histogram displaying the longest inactivity period for each car is shown.

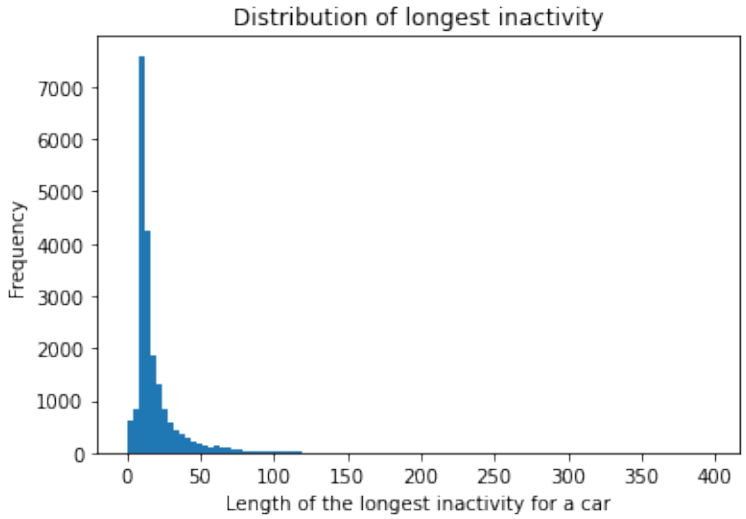


Figure 4.2: Histogram displaying the longest inactive time for each car. The large majority of the cars have a maximum inactivity time of 10-40 days.

Another inactivity aspect, is the time between the factory completion date and the time when data collection starts. In figure 4.3, the time between the factory completion date and the first data read-out of each car is displayed as a histogram. It is evident that there are three peaks which the majority of the data is centered around.

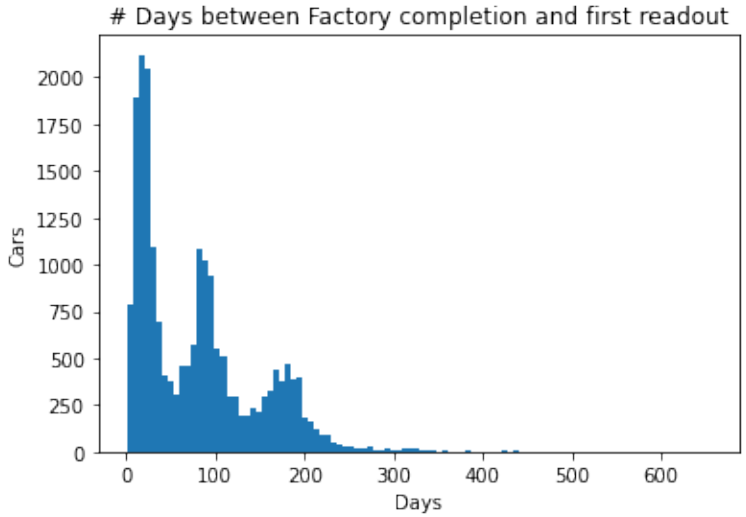


Figure 4.3: Histogram displaying the time between the completion date of the car and the time when data starts being gathered.

Feature	Description
State Of Charge (SoC)	List of 108 SoC values, each corresponding to a battery cell.
Voltage	List of 108 voltage values, each corresponding to a battery cell.
Timestamp	Date and time of day that the readout was collected.
TIN number	Anonymous identification number of the battery from which the readout is collected.
Sample ID	Identification number for the specific readout.
Diagnostic Tool Information	Information regarding which diagnostic configuration was used to make the readout.

Table 4.1: Table showing the data features available.

4.2 Data Features

Each data readout contains voltage and SoC levels measured at 108 places in the battery pack. These locations are onwards referred to as battery cells. Resultingly, each readout encloses 108 voltage values and 108 SoC values. In addition, readouts contain some additional information, these features are briefly described in table 4.1.

4.2.1 Delta Value

For both voltage and SoC, the difference between the largest and the smallest cell value of a readout is defined as the delta value of a battery at that specific time point. This gives a metric that indicates if the battery as a whole is performing as intended with regard to cell deviations. This is one of the main metrics that is used when investigating the health of a battery. If the delta value is high, the battery is most likely not working as intended. Figure 4.4, shows an example of the difference between a textbook example failing battery and a regular, most likely healthy battery at a specific time point.

4.2.2 Cell Deviation

In addition to the delta values, it is often of interest to look at which and how many cells are deviating from the others. Whereas delta values do not give any information regarding the number of cells that are deviating from the others, computing the cell deviation for each cell can give additional information on the battery behavior. The percentage cell deviation is defined as:

$$Cell\ deviation[\%] = 100 \cdot \frac{Cell\ Value - Median}{Median}, \tag{4.1}$$

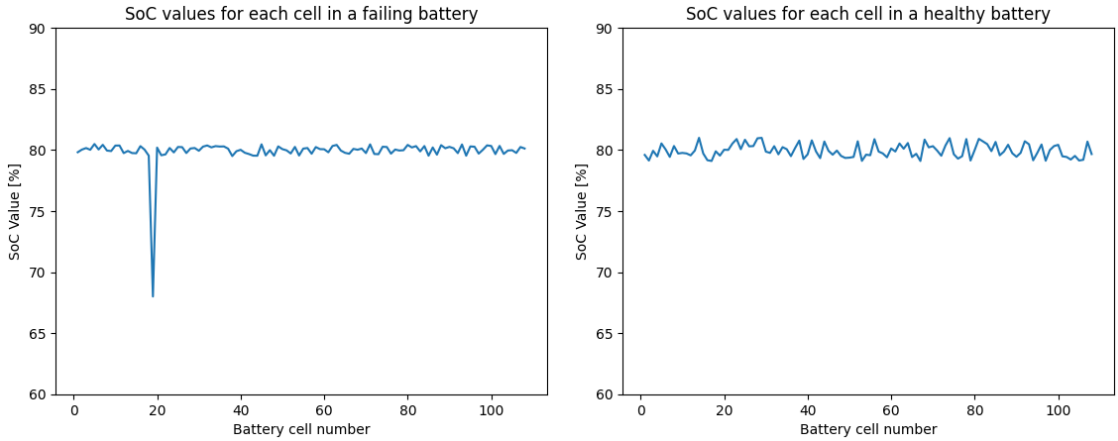


Figure 4.4: Figure displays an example of a clearly healthy vs unhealthy battery. The graphs show the SoC values for each cell of each battery at one specific time point. The left plot has a high delta value where as the right figure has a low delta value.

where *Cell Value* can be either voltage or SoC and *Median* is the corresponding median value. In some situations, the largest cell deviation of a battery is used as a metric and this will be very closely related to the delta value.

4.3 Battery Cell Deviation

In figures 4.5 and 4.6, the distribution of the battery SoC respectively voltage deviation is shown over all samples. The bulk of the samples have a very low deviation, supporting the prior knowledge about how rare these deviations are.

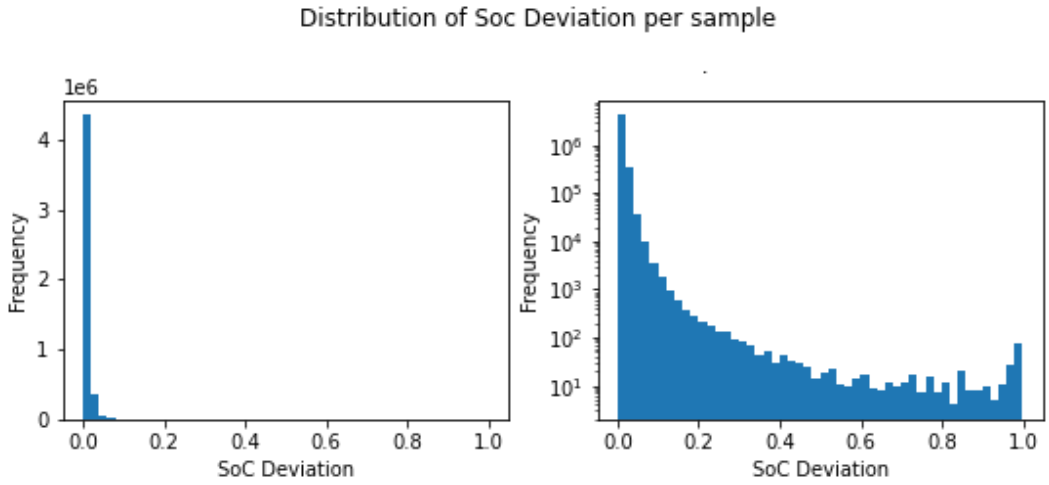


Figure 4.5: Histograms displaying the distribution of battery SoC deviations over all samples. A large majority of the samples have a very low deviation.

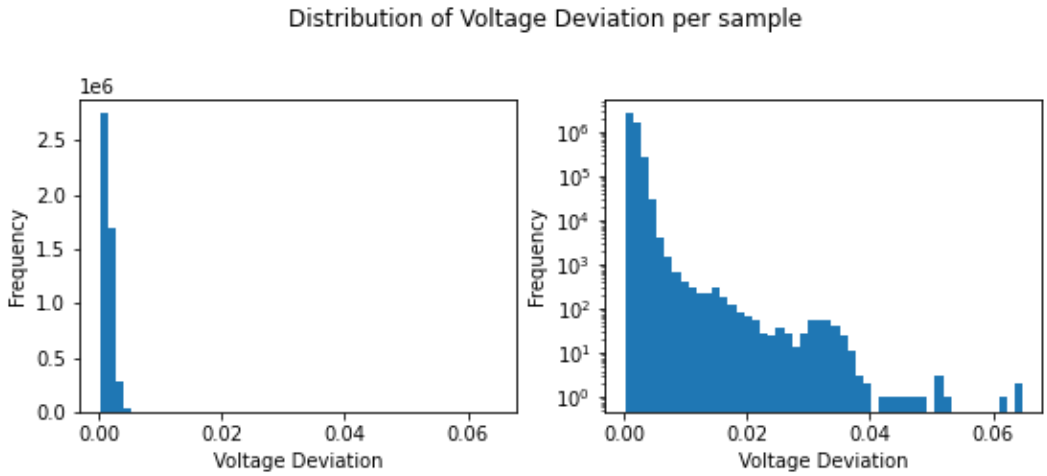


Figure 4.6: Histograms showing the distribution of the battery voltage deviation over all samples.

In figure 4.7, a histogram of the maximum deviation for each car is shown. That is, over all samples of a car's history, the sample with the highest deviation is picked. Here we can once again see that the majority of cars never reach high deviation levels.

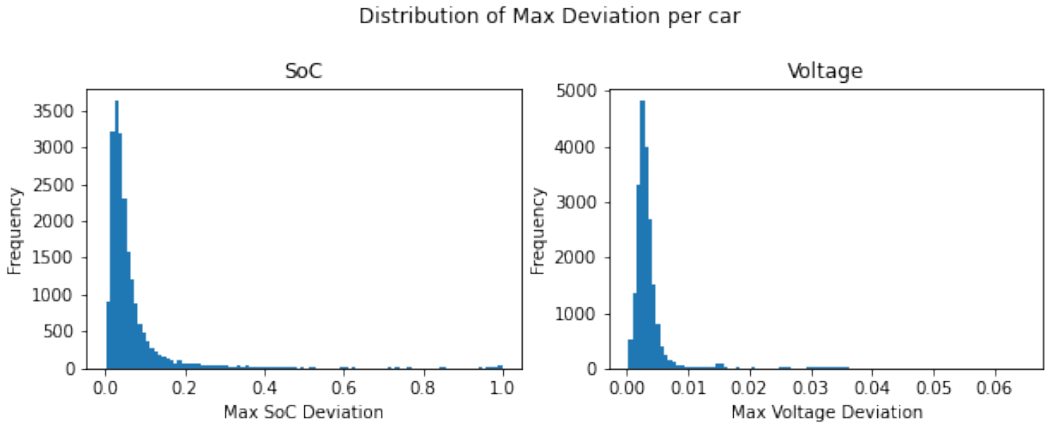


Figure 4.7: Histograms displaying maximum deviation for each car. The left figure is for the SoC where as the right figure is for the voltage.

In figure 4.8, the delta SoC is plotted against the delta voltage for each sample. On the diagonal, there is a clear bulk of samples, showing there is a close relation between the SoC and the voltage. However, there are also a significant amount of samples where the delta SoC is high while the delta voltage is low and the other way around. In figure 4.9, we can see the same density plot for the battery deviations. The same conclusion can be drawn from here.

In addition to the aggregated cell level metrics such as the maximum battery cell deviation, the cell values can also be inspected individually. The cell deviation for cell i is defined as the percentage deviation from the median of all cell values and is

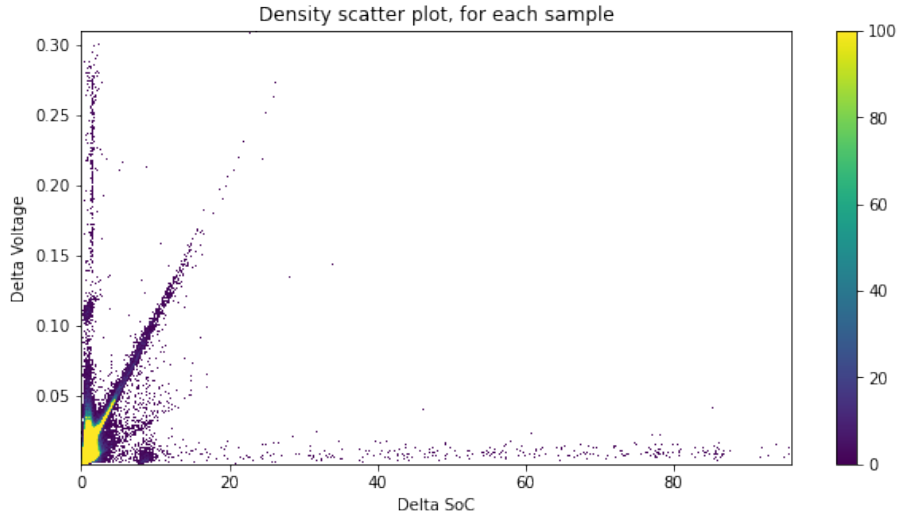


Figure 4.8: Density pair plot showing the relation between the delta SoC and delta voltage. The high density on the diagonal shows there is often a linear relation, but there are plenty of cases where this does not hold.

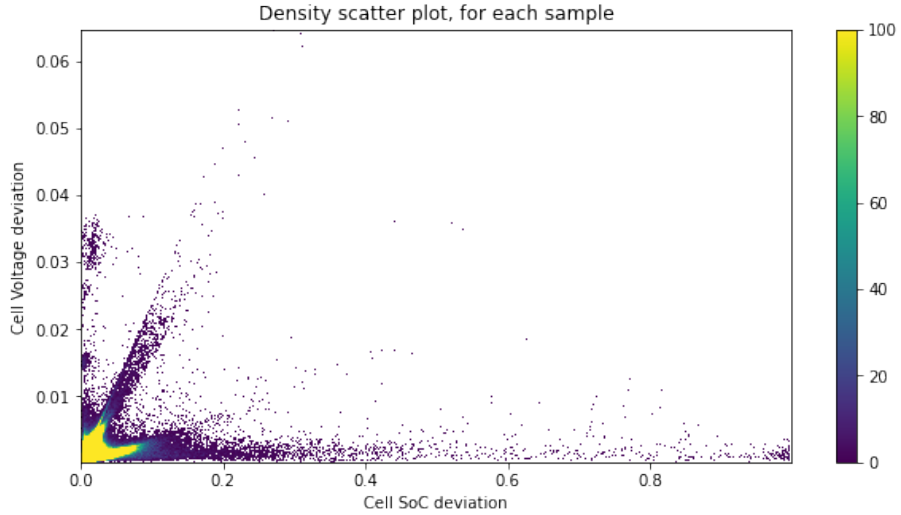


Figure 4.9: Density pair plot showing the relation between the SoC deviation and voltage deviation. There is a significant amount of samples on the diagonal suggesting a linear relation, however, a large number of samples do not have this relation.

computed using the formula shown in equation 4.1. In this project, this is applied to the SoC and voltage.

In figure 4.10, a sub-sample of cars with deviations is plotted, where the cell voltage deviation is plotted against time. A declining trend in the failing cells is observed. Thus the observation in figure 4.10 implies that the cell behaviour can be modelled with time series prediction algorithms.

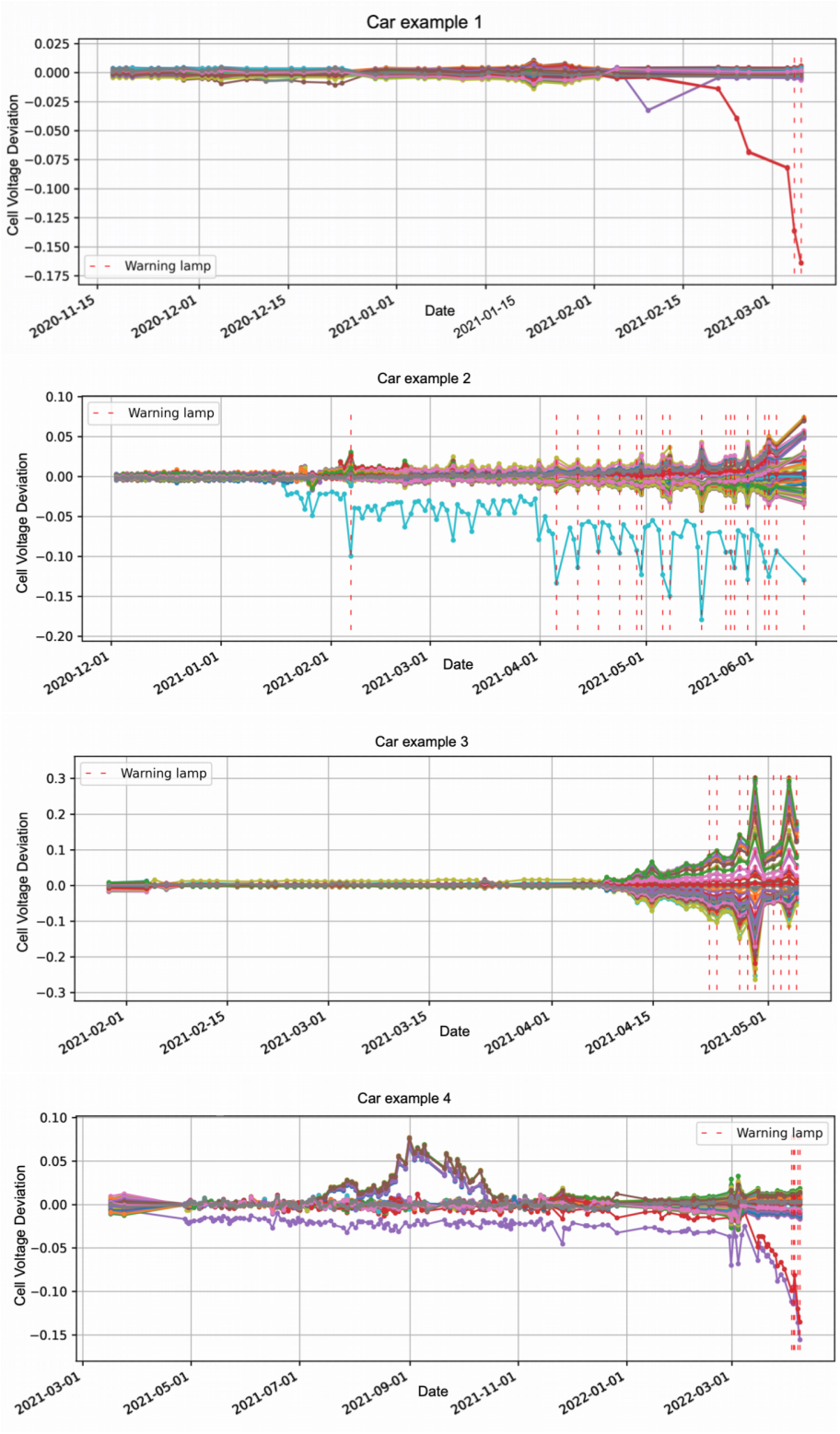


Figure 4.10: Examples of cell voltage deviation over time for cars with deviations.

4.4 Large SoC deviations

In figure 4.11, four examples where a battery has experienced high delta SoC at some time during its lifetime are displayed. In examples 1 and 3, the delta SoC seems to be somehow increasing over time, whereas in examples 2 and 4, the high delta SoC occurs suddenly without any visible prior indication.

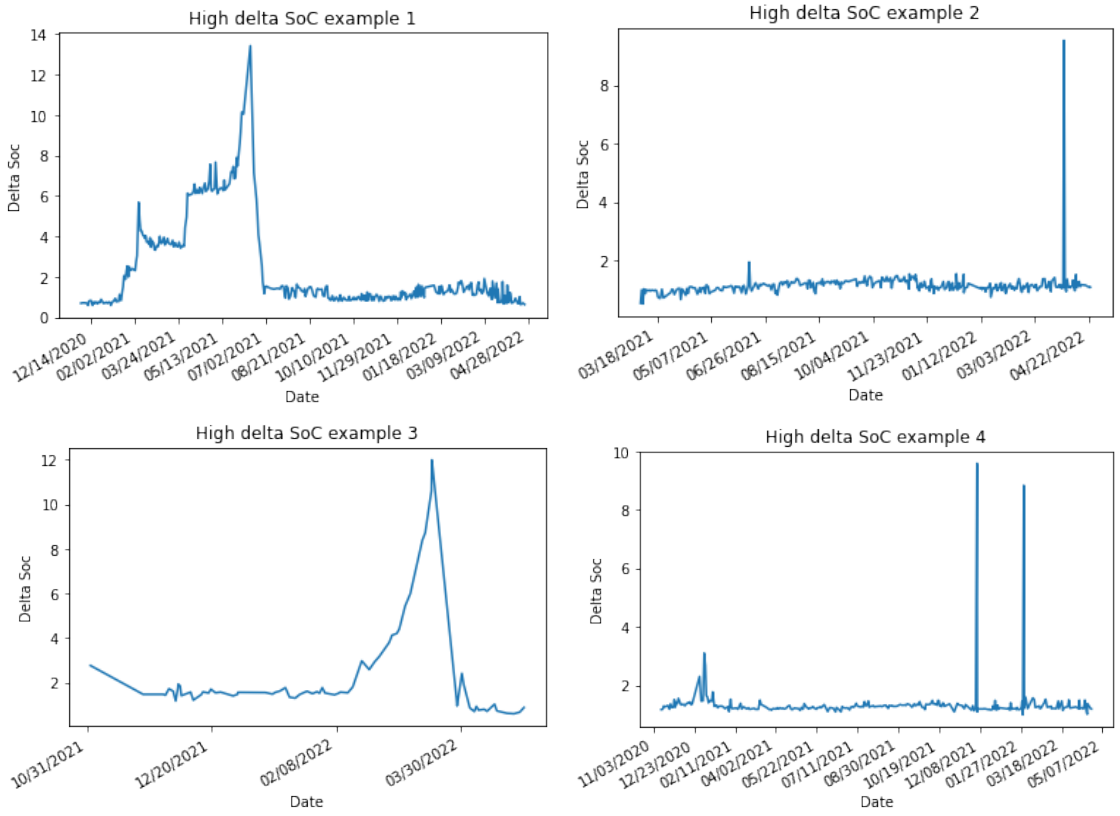


Figure 4.11: This figure shows four time series from cars that at some time point have had a high delta SoC.

5

Defining Battery Failures and Data Labeling

The dataset does not contain any specific information regarding which batteries have failed and been exchanged. Since this is what we want to predict, an important part of this work revolves around defining how to label the dataset. We decided to use a combination of a rule based approach combined with expert knowledge to label the data. First, all cars that never reach a delta SoC of 9% or higher are filtered out. This removes approximately 98% of all cars. The remaining cars are then examined and labeled manually in two steps. First, we removed the obvious samples ourselves and secondly, the final filtering was done with guidance from a Volvo battery expert. This procedure is visualized using a flowchart in figure 5.1.

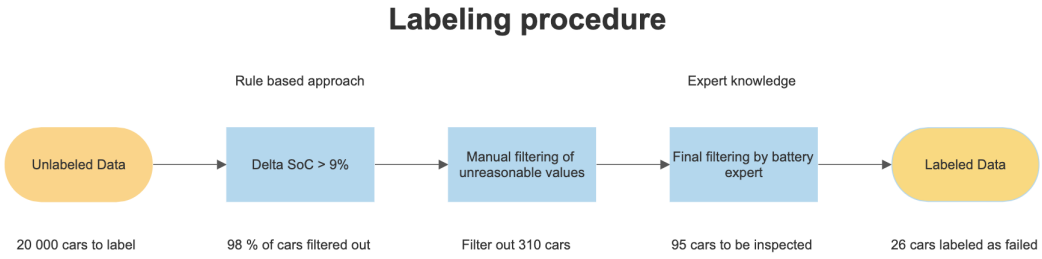


Figure 5.1: Flowchart of the labeling procedure.

An example of what is considered a true battery failure that we are trying to predict compared to a deviation that we are not concerned about can be seen in figure 5.2. An example of what is most likely a data error resulting in an unreasonably high delta SoC value is shown in figure 5.3, and this would be removed in the manual filtering of unreasonable values.

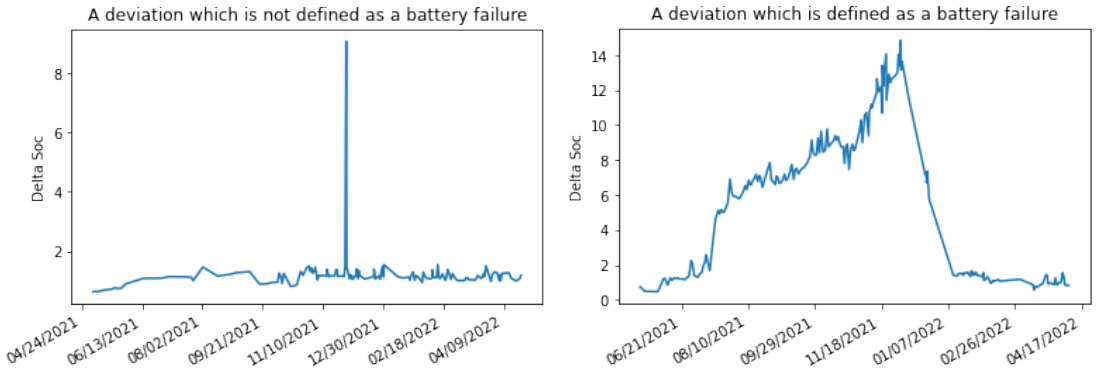


Figure 5.2: Figure displays a car that has a "false" deviation which does not lead to compared to a "true" deviation which leads to a battery failure.

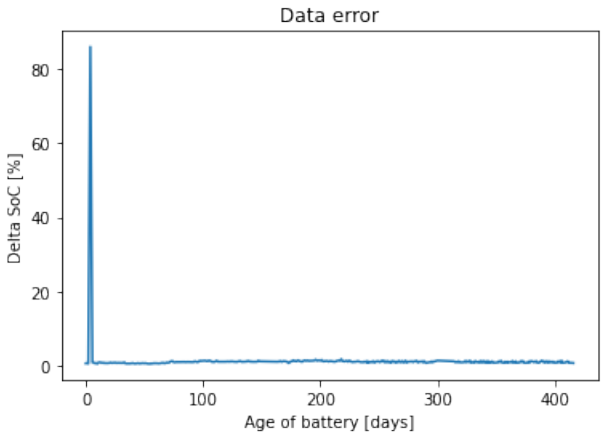


Figure 5.3: Example of a car which has an extreme outlying SoC value in a sample, most likely the result of a data error.

The labeling procedure results in 26 cars that are labeled as failed, some which with a very high confidence level can be defined as failures and some that ideally would require further analysis of the battery to determine confidently. Since there is no data on either when a battery fails or when the exact time of the warning light is activated, we have decided to define the time that the warning light is activated as the time when the SoC deviation reaches 9%. An example of this can be seen in figure 5.4.

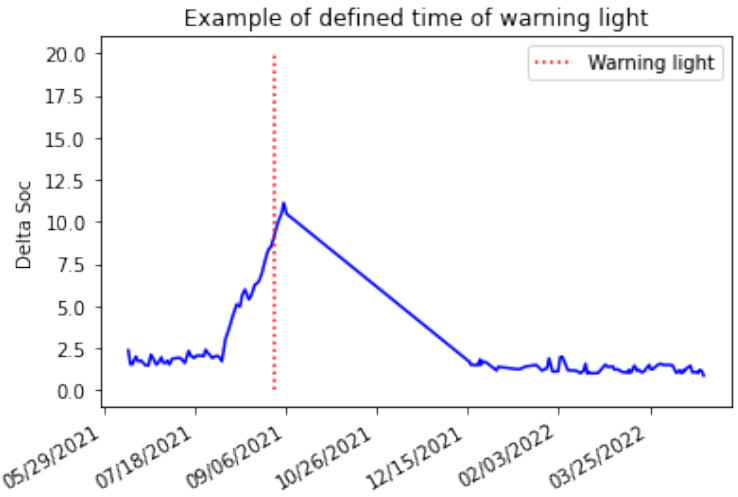


Figure 5.4: Example showing the warning light that is defined as the time when a failing battery reaches 9% delta SoC.

It should be noted that what in this report is defined as the warning light does not necessarily correspond to an actual warning light in the car. This is merely a defined pseudo-warning that a battery is considered to be diverging and about to fail. Since the model output was chosen as binary, all samples and time sequences belonging to a car that will eventually fail are labeled as failures.

6
Methods

The method chapter describes the workflow from data preprocessing up to model creation and evaluation. A significant amount of data preprocessing was done, and the processed data was used by all models. An overview of the steps in the project workflow is displayed as a flowchart in figure 6.1.

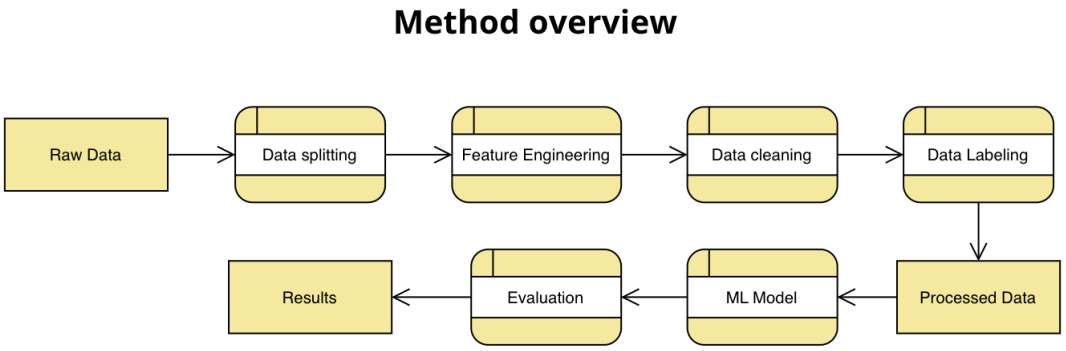


Figure 6.1: Flowchart outlining the setup used in this report.

6.1 Data Splitting

After the initial data analysis, the data was split into different sets. Since the dataset is so heavily imbalanced the data splits were stratified in order to keep the same proportion of failed cars in each split. The splitting was done as a three-part split, resulting in training, validation and test set. The function of the training set is to train the models. The validation set is used to evaluate the performance of the models both during and after training. These results were used for model optimization with regard to hyperparameter values, feature selection, sampling method etc. To receive useful results, whenever evaluation was done on the validation set, the entire validation set was used. As a result of the dataset imbalance, if a subset of the validation was chosen randomly, there would be a large risk that it would contain an inaccurate proportion of positive vs negative samples. The test set was hidden for a large part of the modelling process and was only used in the end to evaluate how well the chosen model/models perform. In table 6.1, the sizes of the splits are displayed.

Set	Portion of dataset [%]	# Failed cars	# Healthy cars
Training set	60	15	≈ 12000
Validation set	20	6	≈ 4000
Test set	20	5	≈ 4000

Table 6.1: Table showing the size of the data splits.

To avoid data leakage between the sets, the splits were done by cars rather than by individual samples. Hence there was no risk different time sequences or samples from the same car could be found in different splits.

6.2 Feature Engineering

The main features used in the models were the delta values and cell deviations for SoC and voltage, as described in section 4.2. The SoC- and voltage cell deviations were computed on a cell level. However, the interesting information which was used to see if a battery fails is in general the maximum of these deviations for each battery. The SoC- and Voltage deviation of a battery was therefore defined as the maximum corresponding cell deviation. This gives a value that is very closely related to the delta value of that battery. In addition to the deviations and delta values, features which describe how the cell deviation vary over the battery were aggregated. These are:

- Median SoC deviation.
- Median voltage deviation.
- Standard deviation of SoC deviation.
- Standard deviation of voltage deviation.
- Age of car.

6.3 Data Cleaning

The samples that contained the following irregularities were removed from the entire dataset.

1. Negative cell values: Some readouts contained negative voltage- or SoC values. This is not physically reasonable and these readouts were removed.
2. Missing values: Readouts, where some values are N/A, were removed.
3. Zero values: There was a small number of readouts where one or several cells had the value zero, while the rest of the values were considerably larger. When looking at the time series of these cells they were determined to be healthy and the readouts with zero values were classified as measurement errors and removed from the dataset.

6.4 Evaluation Metrics

A few different evaluation metrics were used to evaluate the models' performances. The aim of these metrics is to make the performance of models comparable and to give a fair representation of how well the models would perform in real life.

The main evaluation metrics that were used to evaluate how many cars or readouts were classified correctly are:

- Confusion matrix.
- Specificity (True Negative Rate).
- Sensitivity (True Positive Rate).
- Precision.

These metrics can also be combined into the $F_\beta - score$ and geometric mean. Descriptions of all these metrics can be found in section 3.2.

6.4.1 Evaluation by Read-out vs Evaluation by Car

The models evaluated in this report are intended to be used continuously in a car, once for each driving cycle. This means, an output declaring if the battery will fail is received at every driving cycle. Since the results are binary, the evaluation will revolve around seeing how often these results cohere with the true label. This entails two main ways of evaluating a model: evaluating for each readout/driving cycle or evaluating by the lifetime of a car. These ways of evaluating were used in different situations. During training, most models received feedback for each input, for this each sample needs to be evaluated individually. However, when the model has been trained and is evaluated on the validation or test set, either evaluation method can be used, whichever presents the model performance most accurately and fairly.

6.4.2 Time Before Warning Light

In addition to evaluating if the model correctly predicts if a car will fail or not, it is also important to evaluate how early this prediction is made. If a battery is correctly predicted to fail two weeks before it actually fails, this is a much more useful prediction than if the prediction is made two hours before failure.

The time of the warning light was defined as the time when a car that has been labeled as failing reaches a delta SoC level of 9% (see section 5). This definition does not correspond to an actual warning light in a Volvo car, but can be described as a pseudo-warning that the battery is about to fail. This time point is henceforth referred to as the time of warning light. The time before the warning light is therefore the difference between the time of the warning light and the timestamp of the readout which leads to a positive prediction from the model. These times were analyzed one by one, rather than aggregating the values.

6.5 Model Overview

The inference procedure of the model is intended to be the following: for every driving cycle of a car, a data readout is collected. The model uses this sample along with historical samples from the same car as input to evaluate if this car is at high risk of having a battery failure.

The problem is treated as a binary classification problem. That is, the model is answering the question: is this battery going to fail? A positive outcome means the model predicts the battery will fail and a negative outcome predicts that it won't.

6.6 Time Series Prediction

As the data is collected over time, it is natural to use a time series prediction algorithm to model the failures. The models that were used are based on a rolling window algorithm, all using a window size of 10. This means that the last 10 values were aggregated into one value according to one out of three algorithms: rolling median, exponential weighted moving average and rolling standard deviation (see 3.3). The binary classification then works by comparing this value to a threshold value, and if it exceeds the threshold the battery is classified as a failing battery.

As time series may exhibit different behaviours such as a sudden breakout of a new trend or a longer protracted declining trend, different threshold values might be able to capture different trend behaviour with different precision. To determine the optimized threshold value for the complete dataset, each algorithm was iterated over a range of threshold values. The threshold was then chosen based on which threshold value performs best with regard to the recall, precision, specificity and time before warning lamp.

6.7 Neural Networks

There are three main components of neural network models that were examined and evaluated to improve model performance. These are:

1. Network type: Multilayer Perceptron or Long short-term memory.
2. Sampling weights.
3. Feature selection.

All combinations of these main components were used separately as setups for a neural network which was trained as described in section 6.7.1, and then evaluated as described in section 6.4.

6.7.1 Network Setup

Below, the setup used for the neural networks is presented. The hyperparameters described were kept constant throughout the experiments performed, to keep the number of varying variables to a reasonable amount.

The length of the sequences that were used in the networks was set to 10. This was motivated by some testing along with visual analysis of failing sequences. A sequence length of 10 is deemed reasonable as it gives enough history to see trends while still keeping the amount of data noise to a minimum level.

The loss function that was used is binary cross entropy. This is a standard loss function for binary classification and is considered to be a good choice in this situation since the imbalance of the dataset is handled by the sampling. The optimizer algorithm that was used for training is Adam with a learning rate of 0.005.

6.7.2 Training

During training, a subset of the training data was used in each epoch. This subset contains all the positive samples along with a randomized subset of the negative samples, the amount of negative samples were defined by the sampling weights. At defined intervals of the training, the network was evaluated on both the entire training set and the entire validation set. The length of the training was 50 epochs.

6.7.3 Network Type

Two neural network types were examined, MLP and LSTM. A Multilayer Perceptron (MLP) is not necessarily designed to handle time series data, so when using the MLP the input data was flattened, giving a single input dimension of Number of Features $\cdot$ Sequence Length. In addition to the input and output layers, the MLP contains two fully connected hidden layers with 64 and 32 neurons. Each of these layers uses a ReLU activation function and the single output is pushed through a sigmoid function to receive an output value in the range $[0, 1]$.

The LSTM-model uses two hidden layers, each with 64 neurons. It also contains a classifying layer which is a fully connected layer with one output which as in the MLP uses a sigmoid activation function to receive the output value in range $[0, 1]$. Dropout is used as a regularization method with a dropout value of 0.75 to help prevent overfitting.

6.7.4 Sampling

To handle the imbalance of the dataset and ensure that the model was training effectively, a type of undersampling was used, (see figure 6.2).

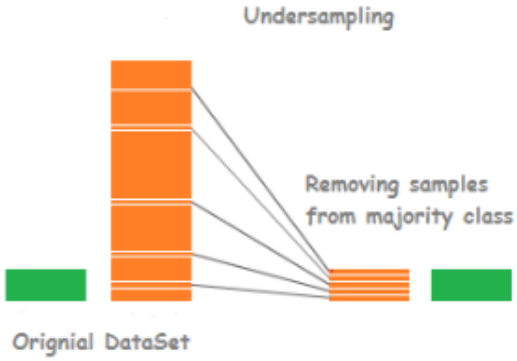


Figure 6.2: Visualisation of the undersampling method.

The undersampling was managed by a weight factor $w_{sampling}$, which defines the ratio between the number of negative and positive samples. The basic idea behind this weighting is that if the ratio is too large, the model will only favor correctly classifying the negative samples (majority class). However, if the ratio is too small, there may be a bias towards classifying samples as positive and create a huge amount of false positives. This ratio was evaluated for $w_{sampling} \in [1, 10, 100, 300]$.

6.7.5 Feature Selection

Three combinations of features were evaluated. As described in section 4.2, the main features that contain indications about the cell failures are the SoC- and voltage deviations on a battery level. These are defined as the respective maximum deviation on a cell level.

The first feature selection contains the most basic and intuitive feature to use, the SoC deviation. This is closely related to the delta SoC which is what was used when defining the battery failures. The second feature selection is using both the SoC deviation as well as the voltage deviation. Lastly, the third feature selection uses the maximum deviation, median deviation and standard deviation of the cell deviations. All of these are for both SoC and voltage. In addition, the age of the car was added to each sample.

7
Results

7.1 Time Series Prediction Models

In figures 7.1 - 7.4, two types of time series analysis tools have been used to estimate when a warning lamp will be triggered. In example 1, shown in figures 7.1 and 7.2 a failing cell with a fast breakout is shown. The rolling median algorithm, predicts the warning light after the warning light is triggered. When using a rolling standard deviation, the algorithm predicts the warning lamp before it is triggered. The opposite situation appears in figures 7.3 and 7.4, when there is a slow breakout of the failing cell. This observation corresponds with the assumption that a rolling median is better to capture and predict trends while the rolling standard deviation is better for breakout anomalies. The indicators are triggered if the rolling median passes the threshold -0.03 and for rolling standard deviation it is triggered if the threshold passes 0.02.

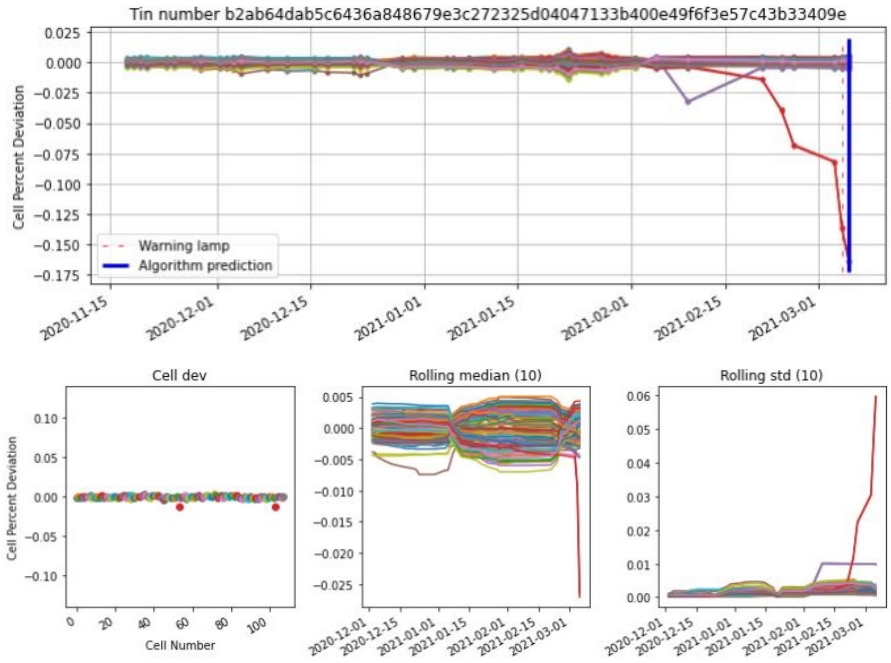


Figure 7.1: Example 1. Fast breakout (Rolling median) with bad performance on algorithm.

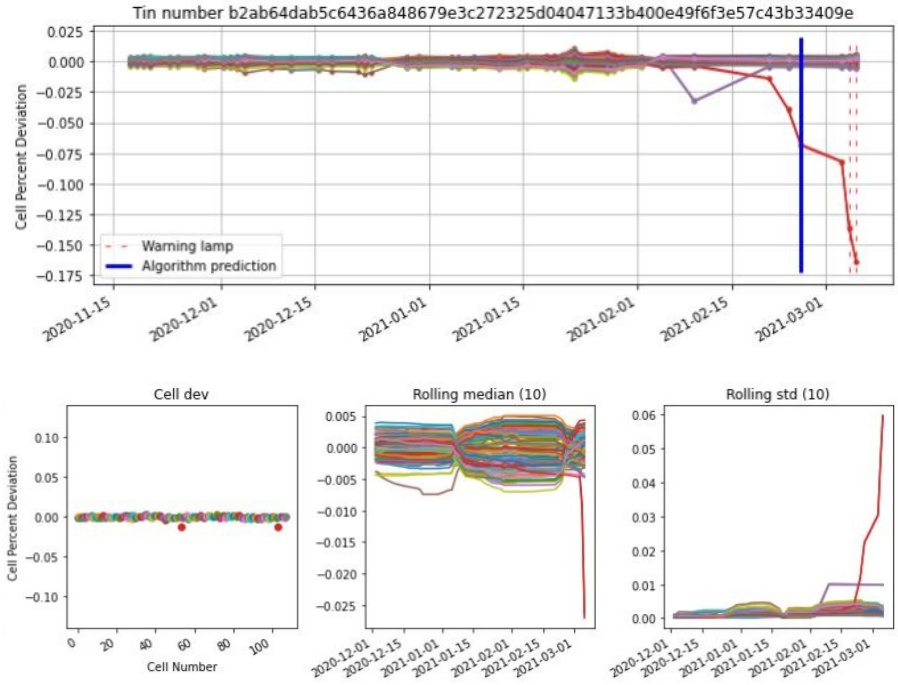


Figure 7.2: Example 1. Fast breakout (Rolling standard deviation) with good performance on algorithm.

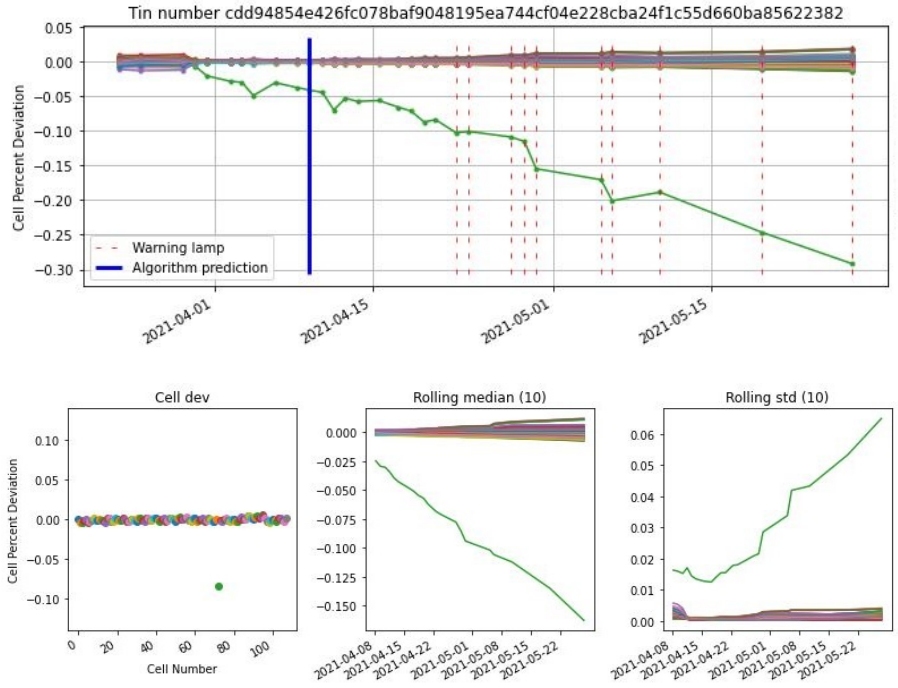


Figure 7.3: Example 2. Slow breakout (Rolling median) with good performance on algorithm.

7.1.1 Determining Thresholds

As described in section 6.6, the threshold value for each rolling window algorithm is determined by iterating over a range of threshold values and then evaluating

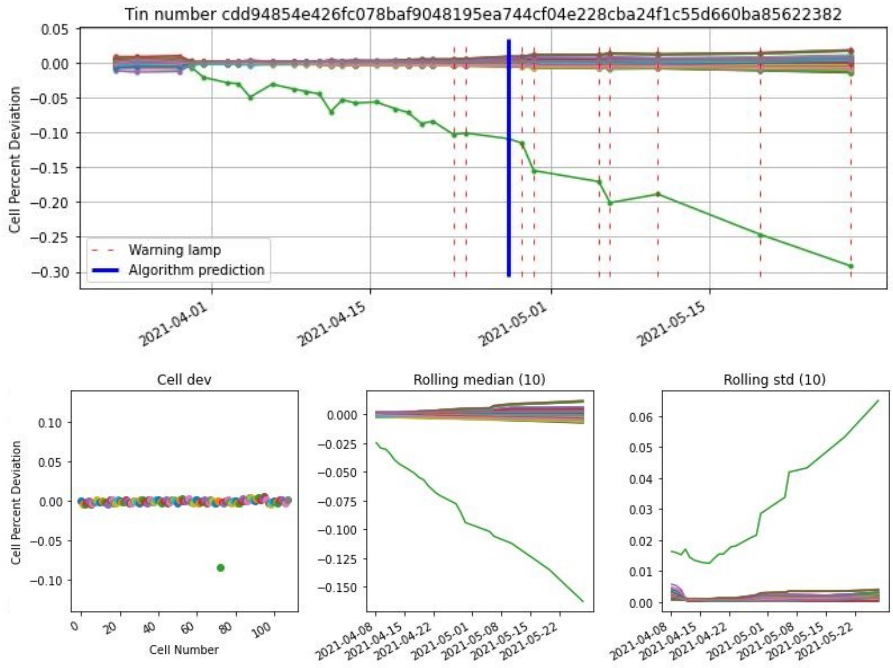


Figure 7.4: Example 2. Slow breakout (Rolling standard deviation) with bad performance on algorithm.

each threshold on the training set. The result is a threshold-based schema for each respective indicator (rolling median, rolling standard deviation and exponential moving average) which can be seen in figures 7.5-7.7, where the evaluation metrics recall, precision and specificity are plotted against the thresholds. The schema also includes *"Days to the first fail"* (denoted as *"Time to warning light"* in section 6.4.2) for the corresponding threshold values. The threshold value is chosen, prioritizing having a high specificity to reduce the false negatives while still finding close to all failures. The threshold values that are chosen for each method can be seen in table 7.1. In figure 7.5, the rolling median indicator gives a fairly good performance with nearly 100 percent precision and approximately 80 percent specificity and is the rolling window algorithm that performs best.

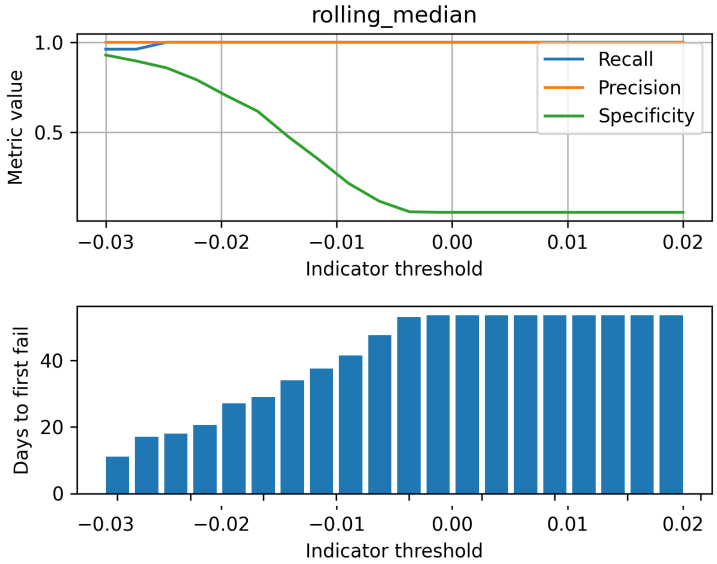


Figure 7.5: Evaluation of different threshold values for the rolling median algorithm.

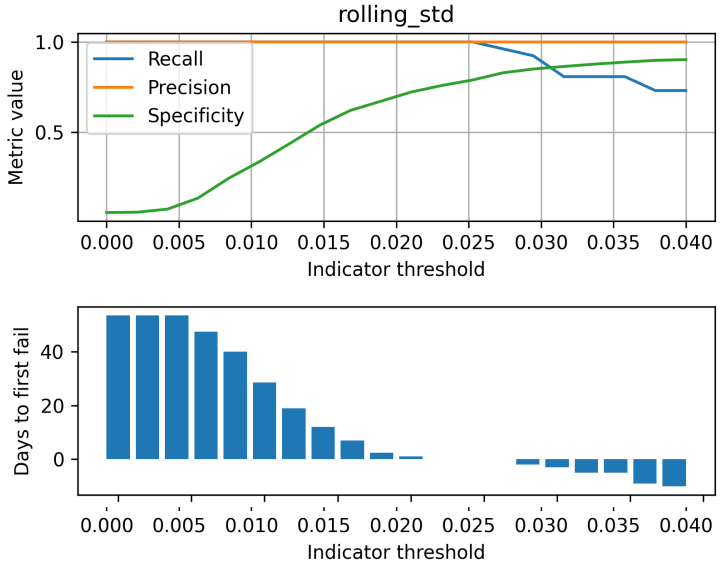


Figure 7.6: Evaluation of different threshold values for the rolling standard deviation algorithm.

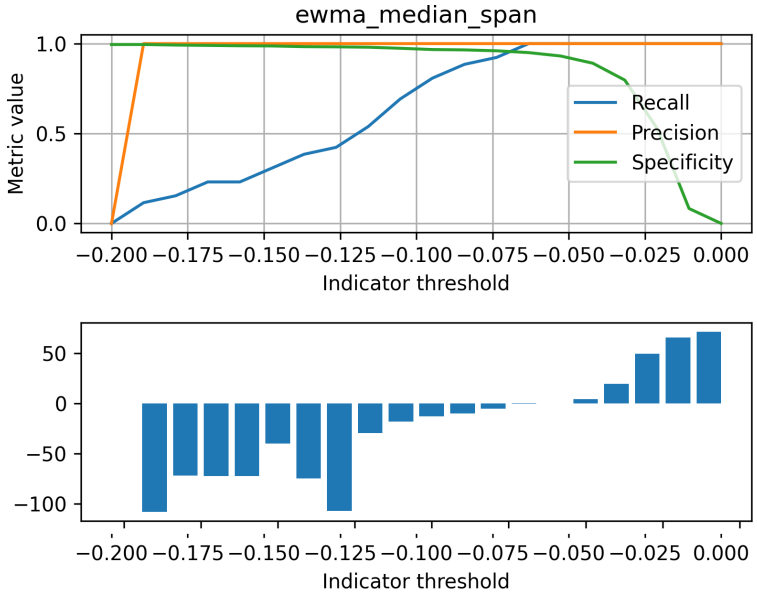


Figure 7.7: Evaluation of different threshold values for the EWMA algorithm.

Algorithm	Chosen threshold
Rolling Median	-0.03
Rolling Std	0.03
EWMA	-0.075

Table 7.1: Table showing the optimal thresholds chosen for each algorithm.

7.1.2 Evaluating Rolling Window Algorithms

The rolling window algorithms are first evaluated using the validation set producing the results seen in table 7.2.

Algorithm type	TP (Sensitivity)	TN (Specificity)
Rolling median	6 (100 %)	3150 (80.4 %)
Rolling Std	0 (0 %)	3905 (99.9 %)
EWMA median	4 (66.7 %)	3100 (79.5 %)

Table 7.2: Table showing rolling window algorithm results. All results shown are evaluated by car on the validation set.

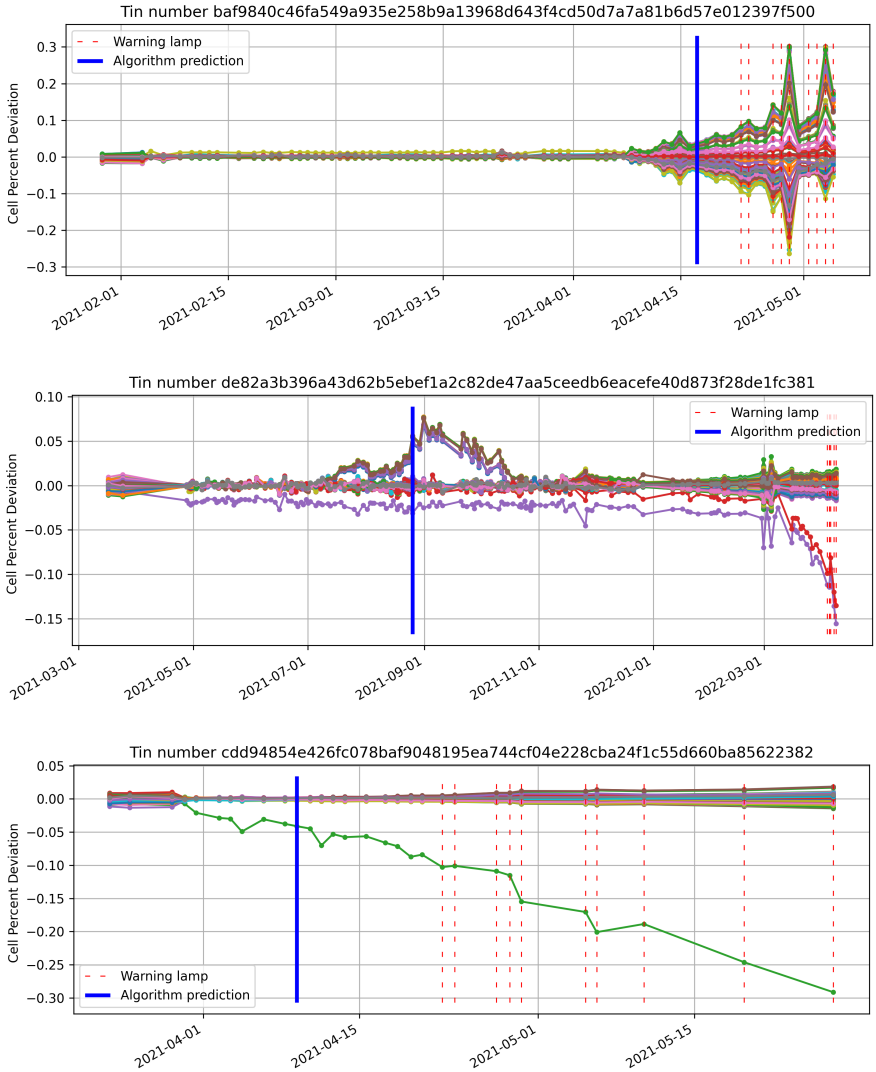
How early these predictions are made can be seen in table 7.3, where the time before warning light for each rolling window algorithm is presented for the failed cars in the validation set.

Model	Car 1	Car 2	Car 3	Car 4	Car 5	Car 6
Rolling Std	Not found	Not found	Not found	Not found	Not found	Not found
Rolling median	5 days	220 days	13 days	1 day	19 days	1 day
EWMA median	Not found	13 days	5 days	5 days	3 days	Not found

Table 7.3: Table comparing the results of the Time series prediction models. The results are from evaluation on the validation set.

7.1.3 The Best Rolling Window Algorithm

Based on the results presented in 7.1.2, the rolling window algorithm that performs the best is the rolling median algorithm (see table 7.2). In figure 7.8, the rolling median algorithm performance on the failed cars of the validation set is displayed. The blue vertical lines represent the time point where the algorithm predicts that a failure will occur at some point in the future. The dotted red lines are the defined time of warning lights, working as the ground truth.



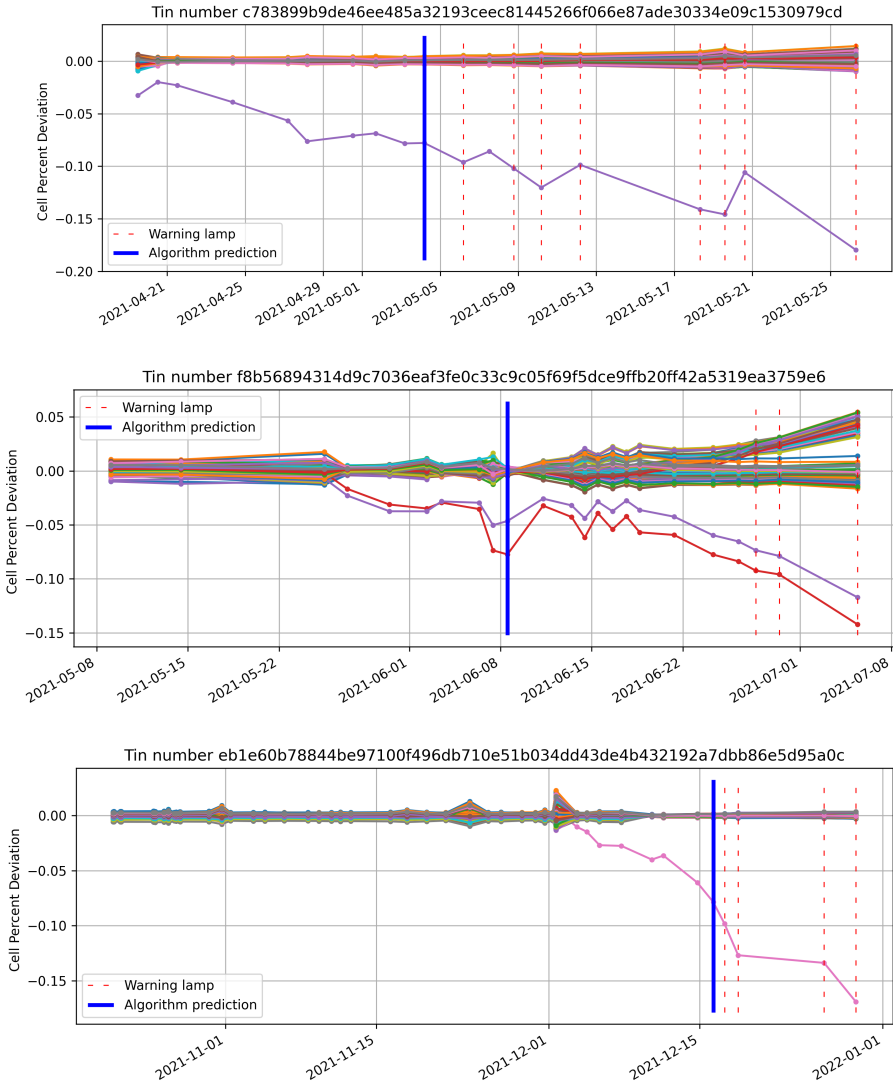


Figure 7.8: Result from evaluation on the validation set with optimised threshold-based schema, using a rolling median indicator.

In figure 7.9, the confusion matrix from the evaluation of the time series prediction model on the test set is shown. Three out of the five failing cars are detected ahead of the warning light giving a sensitivity of 60%, where as 3627 out of 4170 healthy cars were classified correctly resulting in a specificity of 87%.

In table 7.4, the time that the time series prediction model finds the failures in the test set ahead of the warning light is displayed.

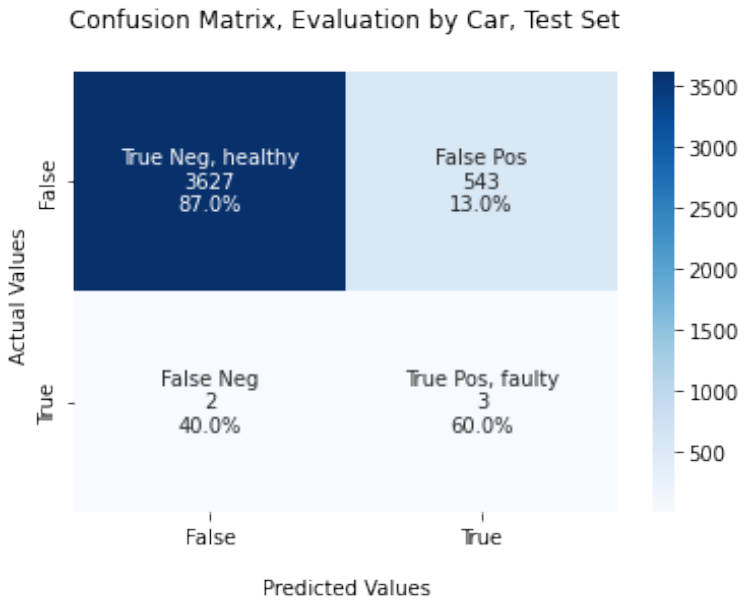


Figure 7.9: Confusion matrix displaying the results when evaluating the time series prediction model on each car of the test set.

Car number	Time before warning light
Car 1	64 days
Car 2	42 days
Car 3	2 days
Car 4	Not found
Car 5	Not found

Table 7.4: Table showing the time before failure that a time series prediction model manages to predict a car failure.

7.2 Evaluating the Main Components of Neural Networks

As described in the method section (see 6.7), the main components that have been varied are Network type, Sampling weights and Feature selection. In table 7.5, the number of true positives and true negatives along with the sensitivity (true Positive Rate) and specificity (true Negative Rate) are shown. These come from evaluating all cars in the validation set.

7.2.1 Network Type

In figure 7.10, the performances of the MLP and LSTM neural networks are compared for different sampling weights and feature selections. The results are fairly similar, but the MLP seems to be more stable and perform better for most setups.

Network type	$w_{sampling}$	Feature selection	TP (Sensitivity)	TN (Specificity)
MLP	1	SoC deviation	6 (100 %)	2555 (65.4 %)
LSTM	1	SoC deviation	6 (100 %)	2170 (55.6 %)
MLP	1	SoC- and Volt deviation	6 (100 %)	2271 (58.2 %)
LSTM	1	SoC- and Volt deviation	6 (100 %)	2590 (66.3 %)
MLP	1	Aggregated values	5 (83.3 %)	2705 (69.3 %)
LSTM	1	Aggregated values	6 (100 %)	3276 (83.9 %)
MLP	10	SoC deviation	6 (100 %)	3524 (90.2 %)
LSTM	10	SoC deviation	6 (100 %)	3562 (91.2 %)
MLP	10	SoC- and Volt deviation	6 (100 %)	3460 (88.6 %)
LSTM	10	SoC- and Volt deviation	6 (100 %)	3530 (90.4 %)
MLP	10	Aggregated values	6 (100 %)	1676 (42.9 %)
LSTM	10	Aggregated values	6 (100 %)	1078 (27.6 %)
MLP	100	SoC deviation	6 (100 %)	3855 (98.8 %)
LSTM	100	SoC deviation	5 (83.3 %)	3838 (98.3 %)
MLP	100	SoC- and Volt deviation	6 (100 %)	3850 (98.6 %)
LSTM	100	SoC- and Volt deviation	6 (100 %)	3844 (98.4 %)
MLP	100	Aggregated values	4 (66.7 %)	1650 (42.3 %)
LSTM	100	Aggregated values	6 (100 %)	621 (15.9 %)
MLP	300	SoC deviation	4 (66.7 %)	3900 (99.9 %)
LSTM	300	SoC deviation	3 (50 %)	3901 (99.9 %)
MLP	300	SoC- and Volt deviation	3 (50 %)	3903 (99.9 %)
LSTM	300	SoC- and Volt deviation	4 (66.7 %)	3880 (99.4 %)
MLP	300	Aggregated values	3 (50 %)	2998 (76.8 %)
LSTM	300	Aggregated values	6 (100 %)	427 (10.9 %)

Table 7.5: Table showing model results for all combinations of varied network components. All results shown are evaluated by car on the validation set.

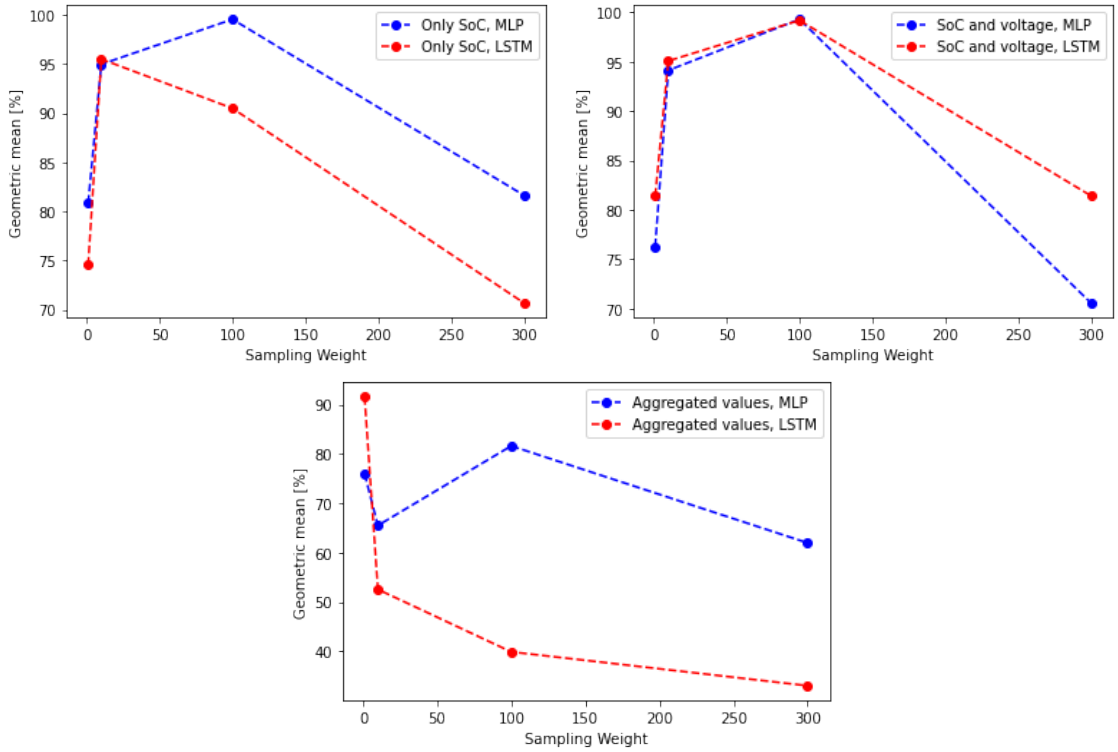


Figure 7.10: Comparing MLP and LSTM for different sampling weights and feature selections. Each graph shows the geometric mean as a function of the sampling weight for a specified feature selection.

7.2.2 Sampling

When looking at the results from table 7.5, it is noticeable that the component that has the largest impact on the model performance is the sampling weight. With a small value on $w_{sampling}$, there is a very large number of false positives. When $w_{sampling}$ is increasing, the number of false positives is reduced but with a higher risk of false negatives. This is clearly displayed in figure 7.11, where the sensitivity and specificity are visualized as functions of the sampling weights. By locking the other variables, setting the model type to LSTM and the selected feature as SoC deviation, the effect of varying $w_{sampling}$ is further investigated. In figure 7.12, the confusion matrices from evaluating each car in the validation set are displayed. In table 7.6, the time before failure for different sampling weights are shown.

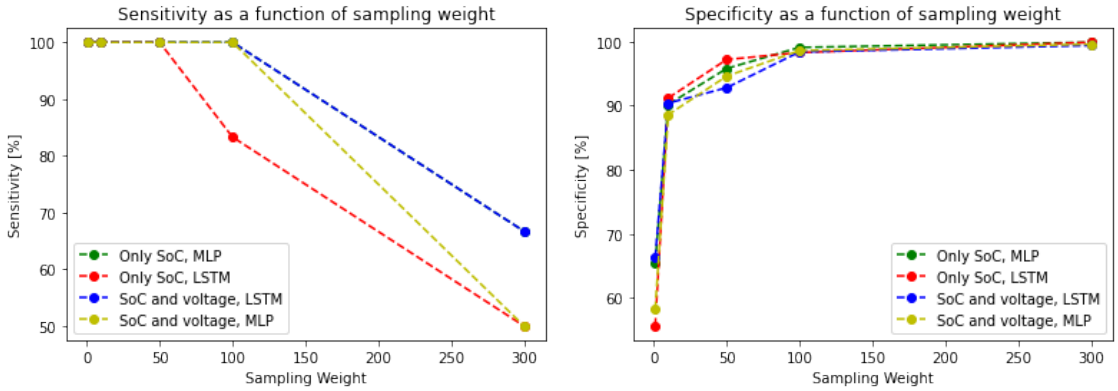


Figure 7.11: Figure shows the sensitivity and specificity as functions of the sampling weight. It is evident that the sensitivity decreases while specificity increases as the sampling weight increases.

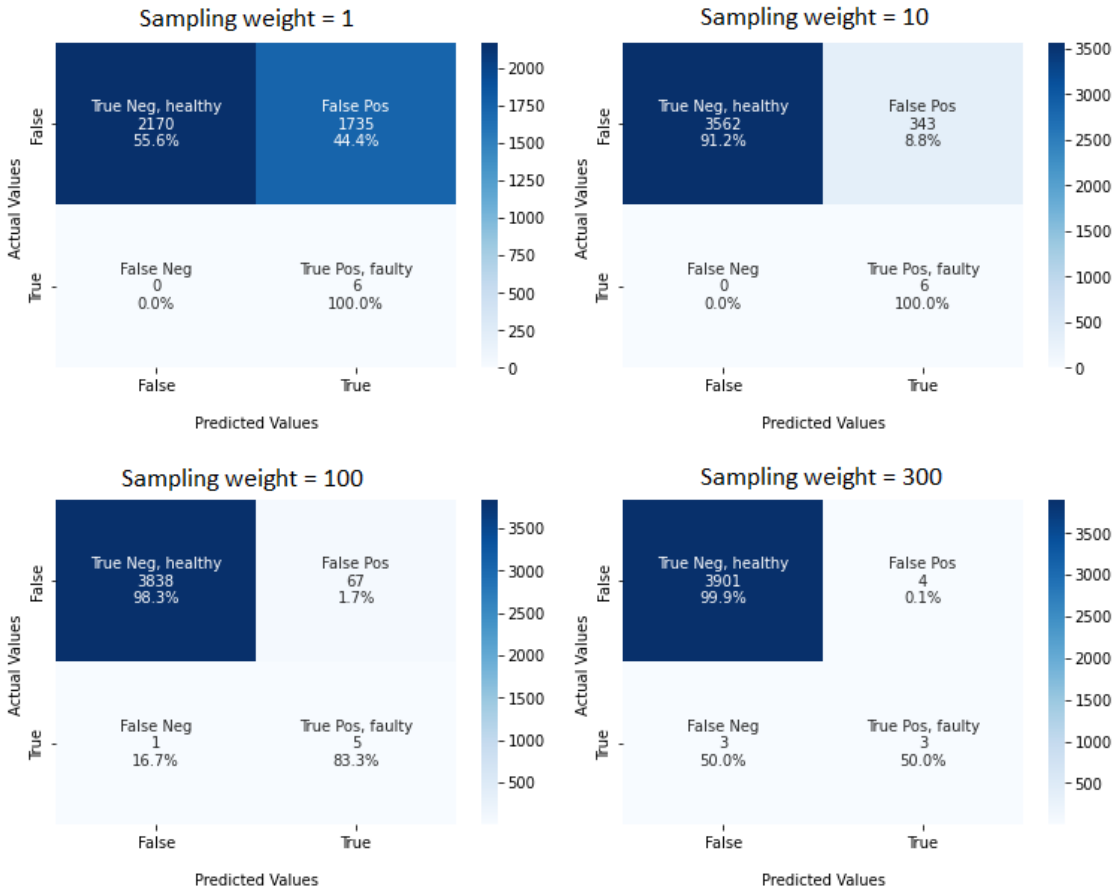


Figure 7.12: Figure shows the confusion matrices for different sampling weights. The confusion matrices are produced by evaluating each car in the validation set.

7.2.3 Feature Selection

In table 7.5, the most unpredictable results seem to be received by using aggregated values as features. The aggregated values consist of age, median deviation, max

	$w_{sampling} = 1$	$w_{sampling} = 10$	$w_{sampling} = 100$	$w_{sampling} = 300$
Car 1	58 days	12 days	2 days	Not found
Car 2	5 days	5 days	5 days	3 days
Car 3	14 days	14 days	9 days	1 day
Car 4	322 days	316 days	248 days	204 days
Car 5	10 days	6 days	Not found	Not found
Car 6	13 days	9 days	7 days	Not found

Table 7.6: Table showing the time before failure that an LSTM model trained with different sampling weights manages to predict a car failure.

deviation and standard deviation of the deviations for both voltage and SoC (see section 6.2). In figure 7.13, various evaluation metrics computed during training of an MLP model are shown. In figure 7.14, the evaluation metrics during training for the LSTM are displayed. The results from using only SoC deviation compared to using Voltage- and SoC deviations are very similar. In figure 7.15 we can see the confusion matrices for an MLP model with $w_{sampling} = 100$ for both these feature selections. Using this sampling weight and network type, the feature selection of only SoC deviation outperforms using both Voltage- and SoC deviations, 99.6% specificity compared to 98.6% specificity. Both evaluations find all the true positives giving a sensitivity of 100%. In figure 7.16, the same comparison for an LSTM model with the same sampling weight is made. The specificity is very close, but the model using only SoC deviation misses one of the failing cars giving a sensitivity of 83.3%.

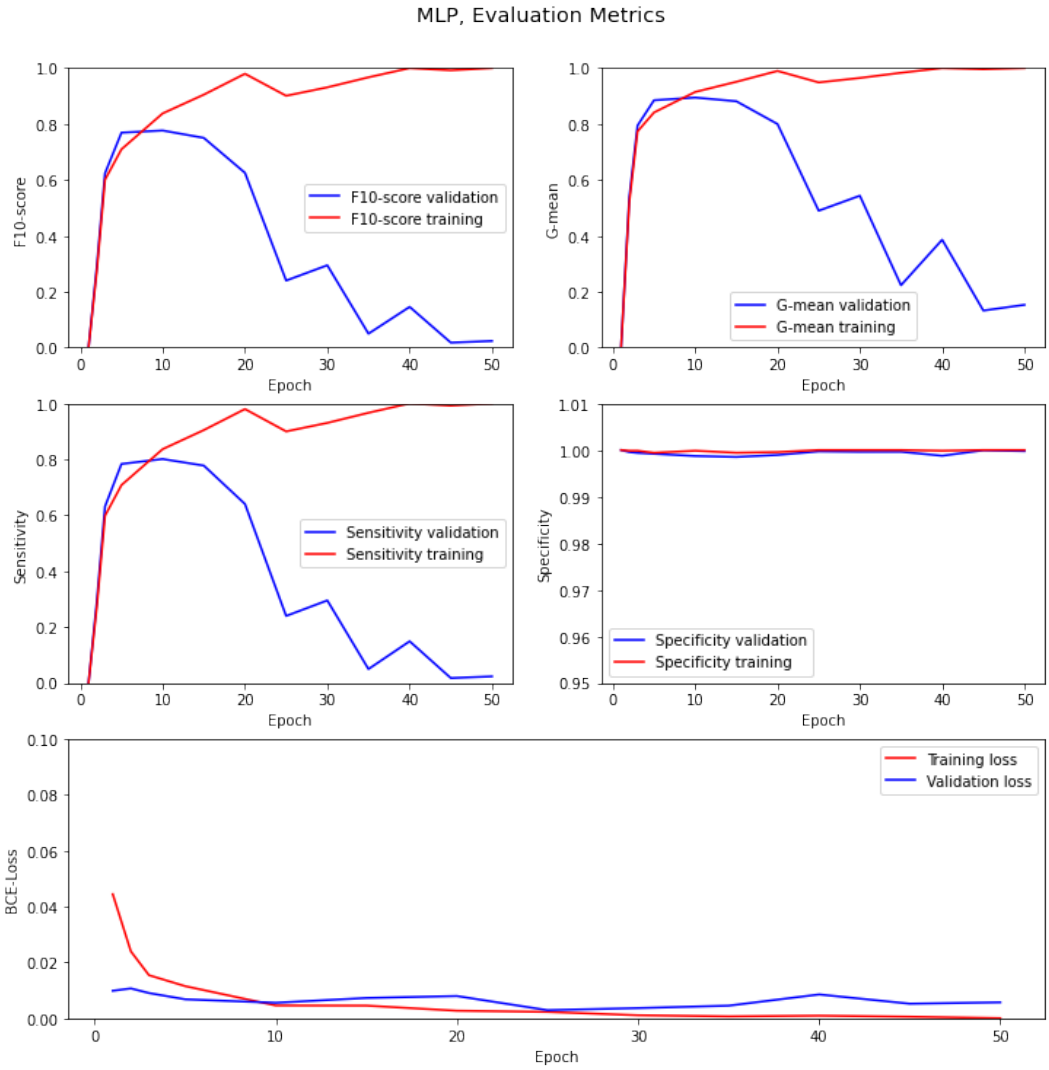


Figure 7.13: Evaluation metrics during training. MLP network, sampling weight = 100. We can see a type of overfitting where the sensitivity of the validation set decreases rapidly as the model overfits to the training set.

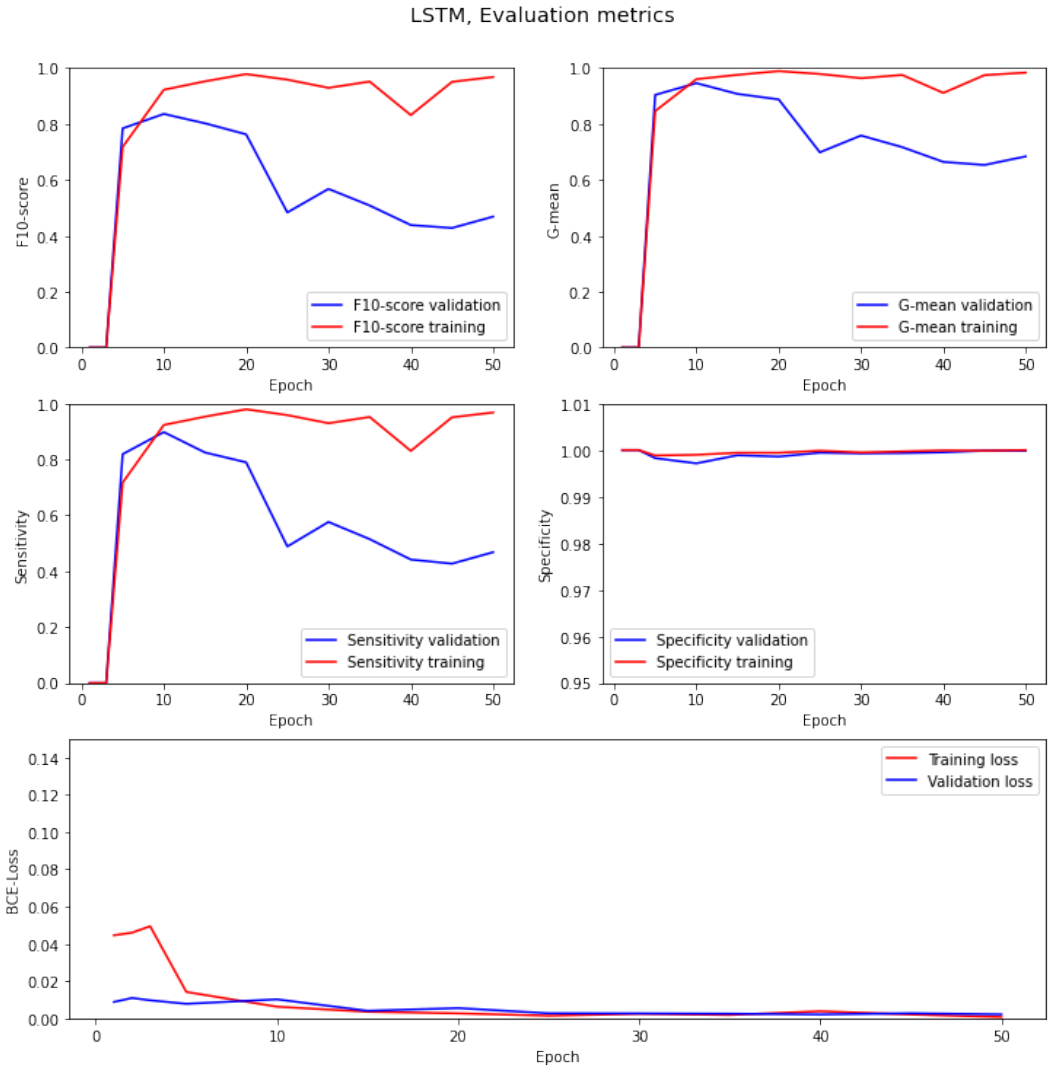


Figure 7.14: Evaluation metrics during training. LSTM network, sampling weight = 100. A type of overfitting is visible, where the sensitivity of the validation set decreases as the model overfits to the training set.

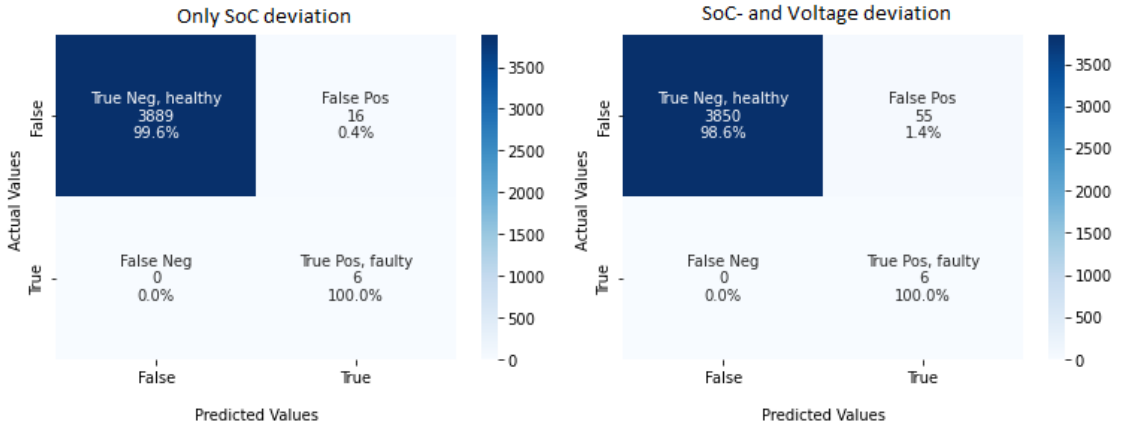


Figure 7.15: Figure shows the confusion matrices for different feature selections on MLP models. The confusion matrices are produced by evaluating for each car in the validation set.

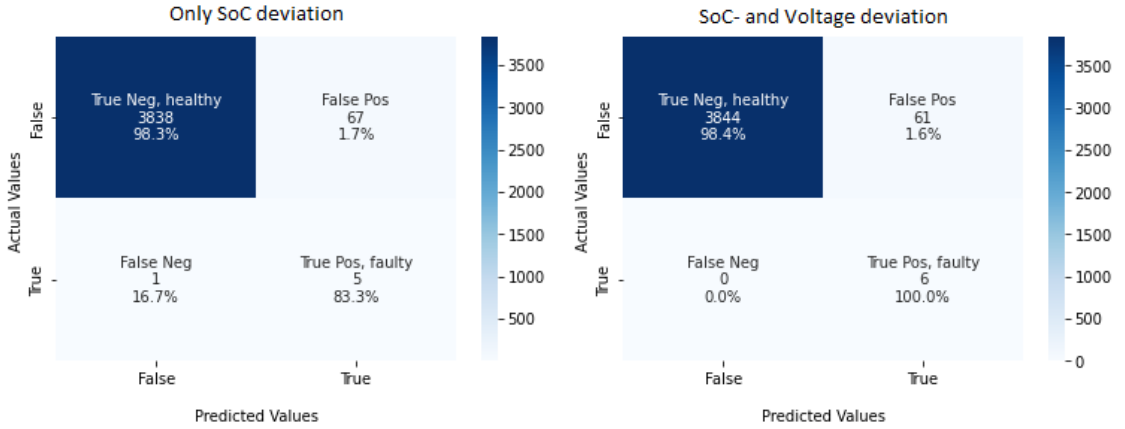


Figure 7.16: Figure shows the confusion matrices for different feature selections on LSTM models. The confusion matrices are produced by evaluating for each car in the validation set.

7.3 Comparison of Models

In table 7.7, the results from evaluating the time series prediction model and the neural networks on the validation set and test set are compared. Both the MLP and LSTM models are set up using a sampling weight of 100 and SoC- and Voltage deviation as input features. Both Neural Network models outperform the Time series prediction model in all metrics. The difference is small between the MLP and LSTM model, with the MLP model performing slightly better.
In table 7.8, the models are evaluated in terms of how early the model predicts the failures. This is evaluated on the test set.

Model	Sensitivity Test	Specificity Test	Sensitivity Validation	Specificity Validation
TS model	60 % (3 of 5)	87.0 %	60 % (5 of 6)	90.2 %
LSTM	80 % (4 of 5)	94.7 %	100 % (6 of 6)	98.4 %
MLP	80 % (4 of 5)	96.3 %	100 % (6 of 6)	98.6 %

Table 7.7: Table comparing the results of the Time Series Prediction model and the Neural Networks. The results are from evaluation on the validation set and test set.

Model	Car 1	Car 2	Car 3	Car 4	Car 5
Time Series Prediction	64 days	42 days	2 days	Not found	Not found
Multilayer Perceptron	299 days	34 days	5 days	Not found	2 days
Long Short-term memory	302 days	30 days	4 days	Not found	3 days

Table 7.8: Table comparing the results of the Time Series Prediction model and the Neural Networks. The results are from evaluation on the test set.

7.3.1 Best Performing Model

The best-performing model is determined to be an MLP model, using SoC- and Voltage deviation as input features with a sampling weight of 100. The model is evaluated on the test set and in figure 7.17, the confusion matrix when evaluating each car in the test set is shown. In figures 7.18 to 7.22, the SoC deviation as a function of time is shown for the failing cars in the test set. The model prediction is marked by the vertical blue lines whereas the time of the warning light/failure is marked with a red line.

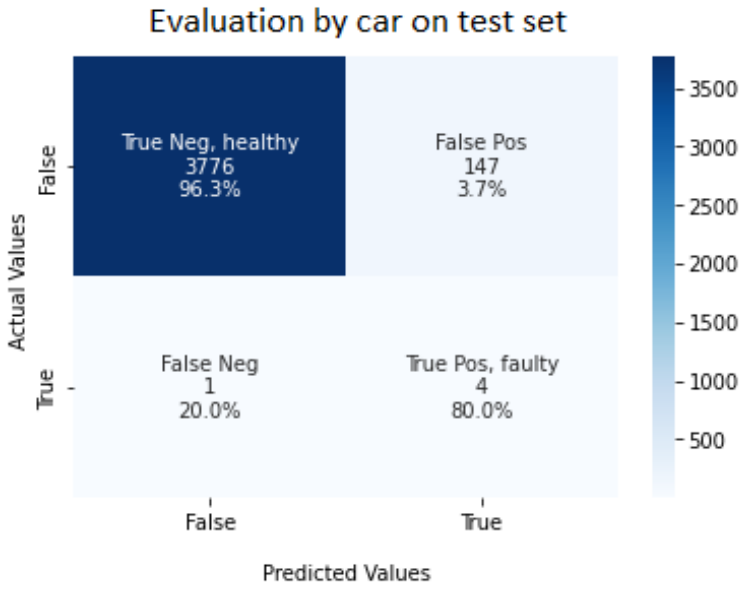


Figure 7.17: Confusion matrix displaying the results when evaluating the model on each car of the test set.

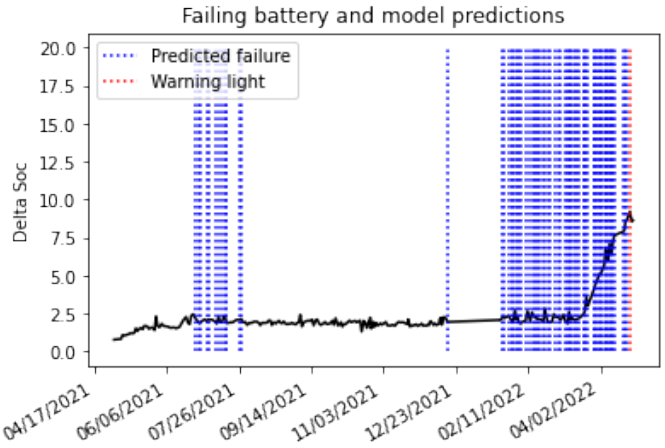


Figure 7.18: The model manages to predict the failure (blue lines), 299 days before the failure (red line).

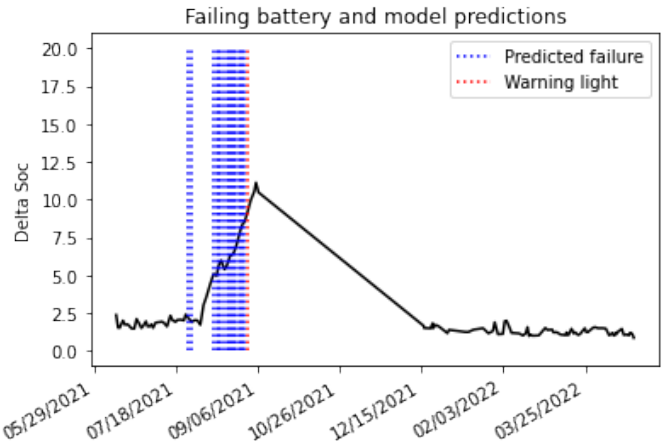


Figure 7.19: The model manages to predict the failure (blue lines), 34 days before the failure (red line).

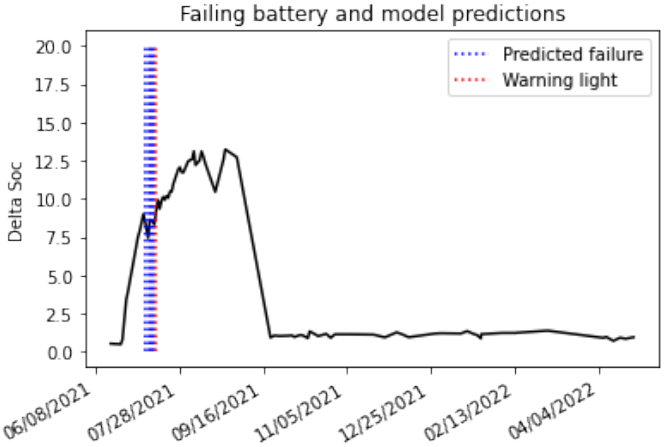


Figure 7.20: The model manages to predict the failure (blue lines), 5 days before the failure (red line).

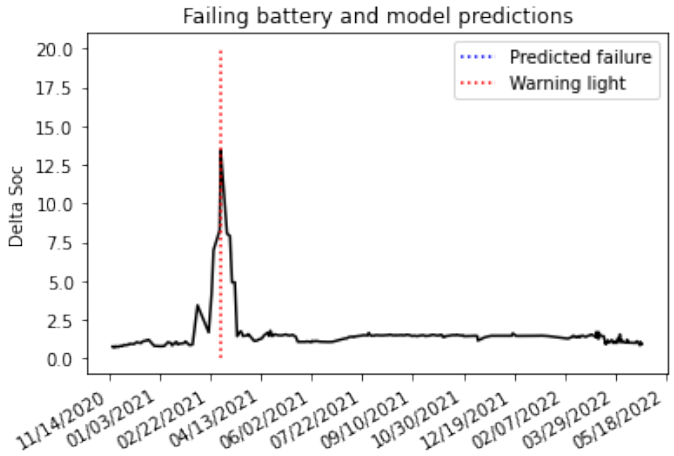


Figure 7.21: The model does not manage to predict the failure (blue lines), before the failure (red line).

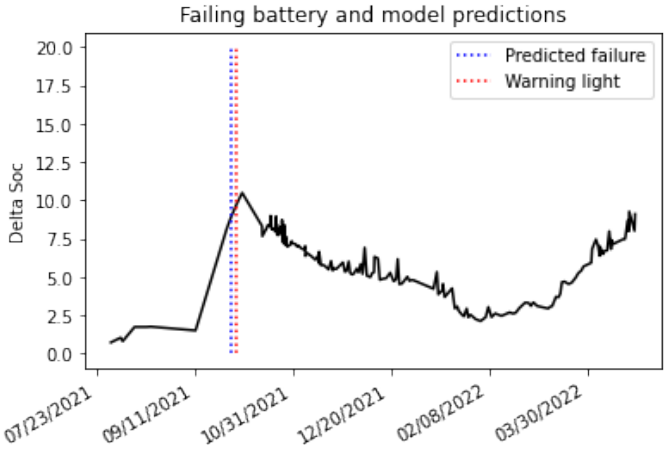


Figure 7.22: The model manages to predict the failure (blue lines), 2 days before the failure (red line).

8

Discussion

8.1 How to Measure the Performance of a Model

To determine if a model is performing as intended, evaluation metrics are used to evaluate the model's performance. The evaluation metrics need to both suit the problem, business case and dataset. The first decisive factor in this process was the dataset imbalance.

As described in the data section 4.1, the dataset is heavily imbalanced and therefor metrics such as accuracy are more or less useless. Instead, we focus on metrics that evaluate the classes separately and then potentially combine those scores. The sensitivity (True positive rate) measures how well the model finds the failing cars while the specificity (True negative rate) measures how well the negative class is classified. These are normally measured as percentages. However, since the dataset is so imbalanced and there are only a few positive samples in each data split, this makes the sensitivity very volatile to small changes, whereas it is not the same case for the specificity. This makes it harder to be confident in the results, especially when measuring sensitivity. With few positive samples to test the models on, there is a high risk of model overfitting, impacting the results, leading to the model not performing as expected when used in reality. Therefore, combined metrics from the sensitivity and specificity such as F-score and G-mean turned out to be less informative than expected. Examining the confusion matrix and reading out sensitivity and specificity separately, was in general the best method of evaluation.

The low amount of failed cars also meant that caution had to be taken when concluding the results received by testing different methods/parameters on the validation set. The risk becomes very high of overfitting to the validation set when the results are relying on so few samples. With the low amount of failed cars, combining sensitivity and specificity to a suitable score was not very easy, hence we in some evaluations needed to choose which to prioritize first.

In section 1.1.2, the business case of the project is described. There it is declared that even though it is important to reduce the number of false positives, the first and primary focus is to avoid false negatives. However, it is difficult to put a number on how much more important this is. Hence, in this project, we strived to catch all the positives first (during validation) and then choose the model that did this with the fewest amount of false positives. This was feasible due to the low amount of

positive samples.

What is missing from the metrics discussed above is the information regarding how early the model predicts the failures. As described in section 6.4.2, this is done through the metric named time before warning light. One of the main difficulties when evaluating the time before warning light, is the very limited number of cars that have failed in the dataset. There are only five respectively six failed cars in the validation and test set (see section 6.1), and these are the only ones that can be evaluated with regard to the time before failure.

The initial aim was to use one evaluation score which preferably was a scalar to make comparisons between models as easy and intuitive as possible. However, due to the reasons stated above, this was not deemed to be the best approach. Since the time before warning light is measured for so few samples, aggregating these values into a scalar such as median or mean becomes quite uninformative. Instead, displaying all the times from the failing cars in a table or other visualization was deemed more appropriate.

8.2 Data

The dataset used in this thesis is relatively young and contains quite a bit of errors and limitations. Most of the limitations come from the fact that this is real-life data collected from cars that are being used by real people. In addition to this, using this dataset to solve the problem of prediction battery deviations brings a lot of difficulties, the largest ones being caused by dataset imbalance and labeling.

8.2.1 Data Labeling

As described in section 5, the dataset does not contain information regarding which batteries have failed and been repaired or exchanged. This is one of the major critiques of the dataset with regard to solving the problem at hand. The models that are created will only be able to detect the type of deviations that we manage to find and label in the dataset. Therefore, the results are strongly dependent on how the data is labeled. Preferably, batteries that are taken into the workshop would be analyzed and information regarding which battery cell is malfunctioning would be made available. This would make it easier to ensure that the labeling coheres with reality.

It is also important to point out that if trying different labeling methods, it is not feasible to compare these to each other. Different labeling methods will create models trying to achieve different things and solve different problems. It is therefore very important to make sure that the labeling identifies the cases which align with the business case.

When labeling the data ourselves, two main methods were examined and used: Labeling by rule and Labeling by expert knowledge.

Labeling by rule in this context means setting a limit to how high the delta SoC or delta Voltage is allowed to be. If a battery passes this limit, we can define it as faulty and imagine a warning light would be activated in the car. In this project, this was used to define the so-called time of warning light for cars that have already been manually labeled as faulty. More specifically the time of warning light was defined as the first time the delta SoC value reaches 9%. As mentioned earlier, this does not necessarily correspond to an actual warning light in a Volvo car, but can be seen as a simplified version such as a warning light system or a pseudo-warning that the battery is failing.

A large issue with using only a delta SoC/Voltage limit is that there is a lot of unnatural noise in the data. There is a large number of samples that contain deviation spikes, which either can be caused by measurement errors in the data collection or by brief deviations which do not lead to battery failures and are not the deviations that are sought in this report. In figure 5.2 we can see the time series of SoC deviation for one battery which is the type of deviation that leads to a battery failure compared to a brief deviation or measurement error that does not.
Labeling by expert knowledge in this context means manually labeling every car based on previous knowledge regarding how these deviations behave. This is easiest to see by looking at the development of the delta SoC over time. If it is steadily increasing, it is a good indication that the battery is probably going to fail. The main problem with this approach is that it is a lot of work labeling the large amount of data that is available.

One important thing to notice is that for models that are trained and evaluated on samples that are subsequences of the entire car history, such as the neural networks, these subsequences would optimally need to be labeled individually. Since the model aims to give a binary result to the question: will this car battery ever fail? (see section 6.5). Then each subsequence of a car that has been defined as a failing car is labeled as a positive, regardless if the fault is active or not. In practice, this means that a large amount of the samples that are labeled as positives behave much more like negative samples, containing no sign of a deviation. This is main the reason why the results when looking at evaluation per sample appear much worse compared to when evaluating per car. This is considered to be one of the largest potentials for improvement when it comes to modelling these deviations.

8.2.2 Data Errors

The dataset contains some irregularities which are best explained as errors in the data collection process. Since these errors are connected to the data collection process which is being improved, these data samples have been completely removed from the dataset (including the test set). Given that these are determined to be errors rather than actual battery behaviors, they could be handled by adding a preprocessing step to the model and used directly in the car, but this is not considered in the scope of this thesis. One example of how this can look is seen in figure 5.3,

where the delta SoC is plotted against time. The car has one sample which deviates extremely, most likely due to some measurement error since this is not a realistic battery behavior.

Other errors that appeared in the data are N/A values and negative SoC values which were removed during the data cleaning. Making a more thorough analysis regarding the cause of these errors would be very useful but is not in the scope of this thesis.

8.2.3 Sample Frequency

In section 4.1, we look at how frequent and regular readouts from different cars are along with how much data we have from each car. In figure 4.1 we can see that the amount of data from each car varies quite a lot. There is a significant amount of cars that have less than 30 readouts collected, whereas the majority have a few hundred samples generated. In figure 4.2, we look at the longest inactivity period for each car and see that it also has some variation in it. The majority of all cars have a longest inactivity period of 20-30 days, but there is a significant number of cars that have been inactive for 50+ days as well. These variations in frequency and quantity of readouts from different cars is natural since this is real life data collected from customers with different driving needs. This is however something that should be taken into consideration when using this data for modelling.

In this thesis, the time sequences that have been used as model inputs are of fixed length. This length is defined as a number of readouts. This means that the time range of each sequence can vary significantly depending on how frequently that car is used. One way of getting a more constant time range for each time sequence could be to use some interpolation method aiming at one readout per time period rather than per driving cycle. However, this approach would be far from flawless either. In addition to the uncertainty that is introduced by interpolation algorithms, it is reasonable to think that an inactive car battery will behave differently during a constant time period compared to an active car. This comes down to the complexity of battery degradation/deviation where both calendar ageing and cycle ageing affect how the battery behaves.

Furthermore, in figure 4.3 the delay between the factory completion date and the first data readout is displayed as a histogram. We can see that this time varies a lot with three peaks containing most of the data. These peaks could potentially be different selling markets where the cars have to travel a various amount of time. Both the underlying reason for this and the effects of these varieties would be interesting to investigate, but is not in the scope of this thesis.

8.3 Time Series Prediction Models

Three rolling window algorithms were used and evaluated during this project, namely rolling median, exponential weighted moving average (EWMA) and rolling standard deviation. The results from these are shown in table 7.3.

The rolling standard deviation classifies nearly everything as negative. In the validation set, it does not manage to find any of the failing cars before the time of warning light and is therefore working very poorly. Seeing as the algorithm worked significantly better on the training set, it does not seem to generalize well to new data. This could stem from the relatively large variations in the data of the failed cars. It could also be the threshold that needs better tuning which could possibly give a better result.

The EWMA algorithm performed better than the rolling std but still gave quite poor results. Correctly classifying four out of six failed cars, while getting approximately 80% of the healthy cars correct is still not very good. The difference between this algorithm and the rolling median is that EWMA puts higher importance on the readouts that are closer in time to the current time. The risk with this in our dataset can be that outlying readouts, where a few sequential readouts can deviate a lot and then pass the threshold even if they should not. This is one potential reason why the algorithm performs worse than the rolling median.

The best rolling window algorithm by far was the rolling median algorithm which found all six failed cars of the validation set with a sensitivity of 80%. On the test set, the algorithm found three out of 5 failed cars but with a sensitivity of 87% (see 7.9). In figure 7.8 we can see that the algorithm finds increasing or decreasing trends that grow over a relatively long time. This supports the hypothesis that a cell failure is preceded by a declining trend in the cell voltage / SOC and is exactly what is expected of the rolling median. The reason for choosing the median rather than the mean is to reduce the effect of outliers and noise in the data. The type of unnatural deviations or measurement errors would have a much larger effect on the mean compared to the median. The rolling median does however not manage to find some of the failures that occur slightly more abruptly, which is an expected limitation when finding a threshold that does not bring along too many false positives. This aspect pinpoints the weakness with a time series analysis, since it can either be optimized to capture protracted trends or shorter breakout trends, but not both.

8.4 Performance of the Neural Networks

As described in section 6.7, the main parameters that were compared are the sampling weights, feature selection and neural network type.

8.4.1 Network Type

In figure 7.10, the MLP and the LSTM models are compared. In general, they perform quite similarly, however when using the aggregated values as features the LSTM performs very poorly. This could stem from several reasons such as network architecture, training algorithms, feature standardization etc. Since the aggregated values as feature selection in general perform worse compared to the other options, this is not investigated further.

When comparing MLP and LSTM for the other feature selections, the differences are very small. For some sampling weights, the MLP outperforms LSTM by a small margin whereas for some sampling weights, it is the other way around. The LSTM was expected to outperform the MLP since this is a problem with time sequences as input data. However, this is not really the case for the setups used. There is likely more potential in the LSTM models provided improvements in data cleaning, data augmentation, feature selection, network architecture etc. However, for the setups used here, the slightly simpler MLP models seem to work slightly better.

8.4.2 Sampling Weights

When looking at the results from table 7.5, it is noticeable that the component that has the largest impact by far on the model performance is the sampling weight. In figure 7.11, it is evident that with a small value on $w_{sampling}$, there is a very large amount of false positives. This is expected since the set of samples that the models are trained on has a relatively large number of positive samples compared to the negative ones. When $w_{sampling}$ is chosen larger, the model is much more restrictive with classifying samples as positive. This leads to a much smaller amount of false positives but at the risk of either missing failing cars or being very late in the prediction and classifying it correctly close to the actual failure.
In figure 7.12, we can see a very clear connection between the sampling weights and the number of samples that are classified as positives vs negatives. With $w_{sampling} = 1$ the specificity (True Negative Rate) is only 55.6%, when increasing the value to 10, it increases dramatically to 91.2%. When going up to $w_{sampling} = 100$, the specificity is as high as 98.3% however, the model now no longer finds all failed cars but misses one out of the six. With $w_{sampling} = 300$ we miss half of the failed car decreasing the sensitivity (True Positive Rate) to 50% but the specificity is 99.9%.

It is worth noting that when evaluating per sample compared to evaluating per car as in figure 7.12, we get a significantly lower sensitivity, that is we miss a much higher percentage of the failed samples. This stems from the fact that we label all sequences from a car that will have a battery failure as positive, all the way from

the first readout of that car up until the failure. This means there is a high risk that a lot of the samples show no detectable signs of failure, but are still labeled as failing samples.

By examining table 7.6, it is evident that the time before failure decreases as we increase the value of $w_{sampling}$. This is expected, as more predicted positive samples by the model (both true and false), give a higher chance of predicting a failure earlier.

8.4.3 Feature Selection

When looking at the results in figure 7.5, for models trained with aggregated features, it is evident that the model is overfitting the data to the training set. The sensitivity (True Positive Rate) along with the F-score of the training set, is increasing while the sensitivity and F-score of the validation set at a certain point start decreasing rapidly. More or less all models trained with aggregated features results in relatively poor performances. It is possible that these could be improved significantly by using methods such as early stopping, extra dropout, modified feature engineering and other types of standardizing/normalizing data features. Using other types of aggregated features could potentially also lead to a completely different result. However, since the results from using only SoC or Volt and SoC deviations are promising these methods are not investigated further in this thesis.

When comparing using only SoC deviation compared to both SoC and Voltage deviation the results are very close. Using only SoC deviation seems to be slightly superior in a few more cases, but for the optimal sampling weights, using both SoC and Voltage deviation works slightly better. Comparing the difference between using solely SoC data compared to Voltage data further would be interesting not only for modelling performance. It could give more insight into what affects these battery failures, and what features are relevant to study in this aspect. In figure 4.8, we can see that there is a connection between the delta voltage and SoC which the theory also confirms. However, there is plenty of samples where the quantities do not increase/decrease together leading us to believe that there is potentially more information to gather about these failures from using both quantities. This is one possible reason for the slightly better performance of models that use both SoC and Voltage as features.

It should be emphasized that this is a very small sample size and it is difficult to draw conclusions from this result. The results from using only SoC deviation compared to both Voltage- and SoC deviation are very similar.

8.5 Comparing TS and Neural Network Models

The rolling median model, while being the best rolling window algorithm does not reach anywhere near the performance needed for the use case of the model. This is quite expected, since the algorithm relies on a quite slow and monotonous expansion of the deviation, and the knowledge of the battery failures is that the battery behavior is more chaotic than this. This observation underlines the necessity of using a model that could capture multiple patterns (such as a Neural Network). On the other hand, a time series analysis offers can be a good benchmark to understand the data and the rise of failing cells to then create more advanced models.

The performances of the neural network models used in this report are significantly better than those of the time series prediction models. In table 7.7, the results of the models using their best setups are compared and both types of neural networks outperform the time series model by a significant amount. When analyzing how early the models predict the failures it is slightly closer, as seen in table 7.8, although the neural network models perform better in this metric as well.

The best MLP model reached a specificity of 98.6% and a sensitivity of 80% (4 of 5) on the test set. This can in many problems within machine learning and outlier detection be considered good results, and we consider it a good preliminary model. However, for the purpose of predicting these battery failures in real life, these results are not good enough. There are still too many false positives, mainly stemming from the problem being so imbalanced. The model also misses one of the test samples, making it problematic to rely on the model in real-life use. However, with improvements most importantly in the data processing and also in the modelling, we believe that a neural network model could perform to the level needed for this task.

One downside to using neural networks is that it can be difficult to understand on what grounds a model makes decisions. It can therefore be questioned if the decisions are well-grounded and can be relied upon. As it is hard to get a comprehensive understanding of their inner working after they have been trained, many ML and AI systems especially deep neural networks are essentially considered black boxes. Explanations and analysis of the model decision processes are particularly important when an ML or AI model is deployed in a decision support system. A high level of explainability helps to correctly predict a models behaviour, which is necessary to avoid mistakes and biases when creating the model. Further, high explainability facilitates a well-grounded understanding of the model which is essential for further improvements to address its shortcomings.

9
Conclusion

This thesis has revolved around developing and evaluating methods to predict battery cell failures in BEVs. This has been performed mainly by using two types of models, namely time series prediction models and neural networks. The results are decent but not good enough for the use case. The results display a relatively large number of false positives which for the most part stems from the problem being extremely imbalanced making these very difficult to avoid. The performance was improved significantly when using neural networks compared to time series prediction models, both with regard to detecting failures earlier and reducing the number of false positives.

A significant amount of time was spent working with the data, where data analysis, cleaning and labeling proved to be a large part of this work. In particular, the labeling of data and how a battery failure is defined, turned out to be some of the most influential and important aspects of the project. Improving the process and information regarding the data labeling is pinpointed as one of the most integral parts of improving the method of predicting battery failures. More data cleaning and investigation of the reason for anomalies in the data is also determined to be a huge improvement factor.

Many improvements could be made about the modelling, such as using different network architectures, improving the training procedure, optimizing hyperparameters further or using completely different, possibly more advanced model types. However, we believe that for this to yield significantly improved results, more work has to be done with the data first, which at this moment seems to be the main limiting factor. Additionally, due to the imbalance of the dataset, increasing the number of samples with failed batteries would likely be needed to improve results, either by collecting more data or with some type of synthetic data method.

Several expansions could be investigated as future work for this project. One alternative approach to this problem would be to test some type of unsupervised learning algorithm. This could be interesting since there is a lot of data available and the data labeling process is quite problematic. However, it does not remove the problem of actually defining what a failure is and would bring along other challenges. Additionally, experimenting with other types of networks such as an RNN with encoder-decoder could bring value. The primary future work that we suggest, is working with the data collection and data handling process. Investigating what is the reason behind some of the abnormalities found in the data and getting precise

information about failed batteries would enable a lot of improvements in this project.

Even if the models presented in this report did not reach the high performance standard that would be needed for them to be used in production, the results show a lot of promise that a neural network model could accomplish this. The results are deemed to be good and with several identified opportunities for improvement there is potential for such a model to be used in production.

Bibliography

[ZW03] Eric Zivot and Jiahui Wang. "Rolling analysis of time series". In: *Modeling Financial Time Series with S-Plus®*. Springer, 2003, pp. 299–346.

[Sun+06] Bo Sun et al. "Towards adaptive anomaly detection in cellular mobile networks". In: *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.* Vol. 2. IEEE. 2006, pp. 666–670.

[Fat+08] Mehrdad Fatourechi et al. "Comparison of Evaluation Metrics in Classification Applications with Imbalanced Datasets". In: *2008 Seventh International Conference on Machine Learning and Applications*. 2008, pp. 777–782. DOI: 10.1109/ICMLA.2008.34.

[Hay09] Simon S. Haykin. *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education, 2009.

[Wag+13] Ralf Wagner et al. "Current research trends and prospects among the various materials and designs used in lithium-based batteries". In: *Journal of Applied Electrochemistry* 43.5 (2013), pp. 481–496.

[Ola15] Christopher Olah. *Understanding LSTM networks*. 2015. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs (visited on 05/23/2022).

[Shi+17] Dominique T Shipmon et al. "Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data". In: *arXiv preprint arXiv:1708.03665* (2017).

[Fen+18] Xuning Feng et al. "Detecting the internal short circuit in large-format lithium-ion battery using model-based fault-diagnosis algorithm". In: *Journal of Energy Storage* 18 (2018), pp. 26–39.

[Aro+19] Ashish Arora et al. *Lithium-Ion Battery Failures in Consumer Electronics*. Artech House, 2019.

[Ism+19] Hassan Ismail Fawaz et al. "Deep Neural Network Ensembles for Time Series Classification". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–6. DOI: 10.1109/IJCNN.2019.8852316.

[Ard+20] Reza Rouhi Ardeshiri et al. "Machine learning approaches in battery management systems: State of the art: Remaining useful life and fault detection". In: *2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*. Vol. 1. IEEE. 2020, pp. 61–66.

[Fet20] C. Fetting. *The European Green Deal*. 2020.

[Seo+20] Minhwan Seo et al. "Online Detection of Soft Internal Short Circuit in Lithium-Ion Batteries at Various Standard Charging Ranges". In:

IEEE Access 8 (2020), pp. 70947–70959. DOI: 10.1109/ACCESS.2020.2987363.

[EEA21] European Environment Agency EEA. *New registrations of electric vehicles in Europe*. 2021. URL: https://www.eea.europa.eu/ims/new-registrations-of-electric-vehicles.

[Hof+21] Louisa Hoffmann et al. "High-Potential Test for Quality Control of Separator Defects in Battery Cell Production". In: *Batteries* 7.4 (2021), p. 64.

[Oku21] Hayrettin Okut. "Deep Learning: Long-Short Term Memory". In: June 2021.

[Rad21] Dario Radečić. *Time Series From Scratch — Exponentially Weighted Moving Averages (EWMA) Theory and Implementation*. 2021. URL: https://towardsdatascience.com/time-series-from-scratch-exponentially-weighted-moving-averages-ewma-theory-and-implementation-607661d574fe (visited on 05/23/2022).

[Kon+22] Lingxi Kong et al. "Evaluating the Manufacturing Quality of Lithium Ion Pouch Batteries". In: *Journal of The Electrochemical Society* 169.4 (2022), p. 040541.

[22] *Volvo Cars*. https://www.volvocars.com/intl/v/sustainability/highlights. Accessed: 2022-06-30. 2022.

[WA22] Leopold Werberg and Annika Ahlberg Tidblad. "Battery Cells: Anatomy amp; Deviations". 2022.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY