

**This paper was presented at the Chair of Production Engineering of  
E-Mobility Components (PEM) at RWTH Aachen University.**

## **Master Thesis**

Name: Kenny Yung

Matr.-No.: 416070

Topic: Prognosis of Battery Health from Historical  
Aging Data

Supervising assistant: Vera Rothmund, M.Sc.

1. Examiner: Prof. Dr.-Ing. Achim Kampker
2. Examiner: Prof. Dr.-Ing. Heiner Heimes

Aachen, the 25/04/2023

The content and result of this work is intended for internal use only. All copyrights are held by RWTH Aachen University. It is not allowed to pass on this work or parts of it to third parties without the explicit permission of the supervising chair.

## I Table of contents

I	Table of contents .....	i
II	List of Symbols and Abbreviations .....	iii
III	List of Figures.....	v
IV	List of Tables .....	vii
1	Introduction.....	9
1.1	Motivation of the Research .....	9
1.2	Research Question .....	10
1.3	Structure of the Thesis .....	10
2	State of the Art.....	12
2.1	Lithium-Ion Batteries .....	12
2.1.1	Types of Lithium-ion Battery .....	14
2.1.2	Lithium-ion Battery Applications .....	15
2.1.3	Battery Degradation .....	17
2.2	Previous Works in Battery Health Prognosis .....	19
2.2.1	Physics-based Modelling.....	19
2.2.2	Data-driven Modelling .....	20
2.3	Machine Learning Techniques .....	22
2.3.1	Linear Regression .....	22
2.3.2	Principal Component Analysis and Partial Least Squares Regression.....	25
2.3.3	Gaussian Process Regression .....	26
2.3.4	Support Vector Regression .....	28
2.3.5	Artificial Neural Network .....	29
2.4	Training and Evaluation .....	31
3	Data Driven Approaches .....	33
3.1	Data Sources .....	33
3.1.1	Dataset Alpha and Beta.....	33
3.1.2	Dataset Delta .....	34
3.2	Input Feature and Target Output.....	34
3.3	Models Selection.....	36
3.4	Hyperparameter .....	38

<b>4    Results .....</b>	<b>41</b>
4.1    Visualized Correlation .....	41
4.2    Hyperparameter Tuning .....	46
4.3    Performance Metrics.....	51
4.4    Cross Validation.....	55
4.5    Testing .....	56
<b>5    Discussion .....</b>	<b>63</b>
5.1    Performance .....	63
5.1.1    Accuracy .....	64
5.1.2    Scaling .....	65
5.1.3    Ease of Use.....	66
5.1.4    Models Verdict .....	66
5.2    Challenges.....	67
<b>6    Conclusion .....</b>	<b>68</b>
6.1    Summary .....	68
6.2    Potential Future Research .....	69
<b>V    Bibliography.....</b>	<b>70</b>
<b>VI   Appendix .....</b>	<b>77</b>

## II List of Symbols and Abbreviations

Symbol	Unit	Description
$\alpha$		Regularization factor
$\eta$		Learning rate

Abbreviation	Description	
A	Ampere	Measurement of electrical current
Ah	Ampere-hour	Measurement of battery capacity
ANN		Artificial Neural Network
BESS		Battery Energy Storage System
BMS		Battery Management System
CNN		Convolutional Neural Network
CV		Cross Validation
DOD		Depth of Discharge
EV		Electric Vehicles
GP		Gaussian Process
GPR		Gaussian Process Regression
GPU		Graphics Processing Unit
ICE		Internal Combustion Engine
Lasso		Least Absolute Shrinkage and Selection Operator
LCO		Lithium Cobalt Oxide
LCO		Lithium Cobalt Oxide
LFP		Lithium Iron Phosphate
LIB		Lithium-ion battery
LMO		Lithium Manganese Oxide
LTO		Lithium Titanate
MPLR		Multi-Layer Perceptron Regressor
NCA		Lithium Nickel Cobalt Aluminum Oxide
NMC		Lithium Nickel Manganese Cobalt Oxide
OCV		Open Circuit Voltage

<b>Symbol</b>	<b>Unit</b>	<b>Description</b>
OLS		Ordinary Least Squares
PCR		Principal Component Regression
PLS		Partial Least Squares
PV		Photo-voltaic
RBF		Radial Basis Function
Relu		Rectified Linear Unit
RMSE		Root Mean Square Error
RQ		Radial Quadratic
SEI		Solid Electrolyte Interphase
SOC	%	State of Charge
SOH	%	State of Health
STD		Standard Deviation
SVD		Singular Value Decomposition
SVR		Support Vector Regression
W	Watt	Measurement of electrical power in Joules/second
<b>Symbol</b>	<b>Unit</b>	<b>Description</b>
$a_e$	mm	Intervention range
$a_p$	mm	Depth of cut
$f_z$	mm	Feed per gear tooth
$z$		Number of gear tooth [Note: Sorting is alphabetical, regardless of upper and lower capitalization]
$\alpha$		Angle of Intrusion
$\phi$		Angle of rotation [Note: Greek letters are inserted after Latin letters.]
<b>Abbreviation</b>	<b>Description</b>	
DP	Polycrystalline diamond	

### III List of Figures

Figure 2.1: Overview of a lithium-ion battery construction.....	12
Figure 2.2: Overview of different cell form factors.....	14
Figure 2.3: Levels of battery pack constructions with prismatic cells form factor.....	14
Figure 2.4: Illustration of electric vehicle components and systems .....	16
Figure 2.5: Diagram of integrated residential solar battery storage system .....	17
Figure 2.6: Plot of linear, sublinear, and superlinear aging trajectories for LIB.....	19
Figure 2.7: Example of simple linear regression of one input variable .....	23
Figure 2.8: Plots of polynomials having various orders M, shown as red curves, fitted to a sample noise-added dataset of $\sin(x)$ .....	24
Figure 2.9: Illustration of PCA dimensionality reduction applied to a 2-dimensional input .....	26
Figure 2.10: Gaussian Process Regression model on a sample dataset .....	28
Figure 2.11: Differing support vectors and separating hyperplane of Support Vector Machines due to varying penalty parameters.....	29
Figure 2.12: Example of a neural network .....	30
Figure 2.13: Schematic of an ANN's neuron .....	30
Figure 2.14: Illustration of how the dataset is split into training, validation, and test sets .....	31
Figure 3.1: Anscombe's quartet dataset .....	37
Figure 4.1: Scatter plots of all input features from dataset Alpha to target output .....	42
Figure 4.2: Scatter plots of all input features from dataset Beta to target output .....	43
Figure 4.3: Scatter plots of all input features from dataset Delta to target output .....	44
Figure 4.4: Input coefficient magnitudes from dataset Alpha linear regression .....	45
Figure 4.5: Input coefficient magnitudes from dataset Beta linear regression .....	45
Figure 4.6: Input coefficient magnitudes from dataset Delta linear regression .....	46

---

Figure 4.7: Model prediction comparison from dataset Alpha test set .....	57
Figure 4.8: Model prediction comparison from dataset Beta test set.....	58
Figure 4.9: Model prediction comparison from dataset Delta test set.....	59

## IV List of Tables

Table 2.1: Common battery metrics.....	13
Table 2.2: Sample time-series data obtained from BMS .....	21
Table 3.1: Input features for predictive models extracted from field data .....	35
Table 3.2: Model hyperparameter options and each respective representation .....	39
Table 4.1: Hyperparameter tuning steps and intermediate results for Ridge.....	47
Table 4.2: Hyperparameter tuning steps and intermediate results for Polynomial.....	47
Table 4.3: Hyperparameter tuning steps and intermediate results for PCR.....	48
Table 4.4: Hyperparameter tuning steps and intermediate results for GPR .....	48
Table 4.5: Hyperparameter tuning steps and intermediate results for SVR .....	49
Table 4.6: Hyperparameter tuning steps and intermediate results for MLP.....	50
Table 4.7: Model hyperparameter options and optimal result choice for dataset Alpha.....	51
Table 4.8: Comparison of model candidates in RMSE across all datasets.....	52
Table 4.9: Comparison of model candidates in R2 score across all datasets.....	53
Table 4.10: Comparison of model candidates in runtime across all datasets .....	54
Table 4.11: Mean cross validation R2 scores of model candidates.....	55
Table 4.12: Standard deviation of cross validation R2 scores of model candidates .....	56
Table 4.13: Comparison of error in model output across all test sets.....	60
Table 4.14: Comparison of error in combined SOH across all test sets .....	61
Table 4.15: Comparison of error in final SOH difference across all test sets.....	61
Table 4.16: Average RMSE of model candidates for all test set .....	62
Table 6.1: Hyperparameter tuning steps and intermediate results for Lasso.....	77
Table 6.2: Hyperparameter tuning steps and intermediate results for ElasticNet .....	77

Table 6.3: Hyperparameter tuning steps and intermediate results for PCR.....	77
Table 6.4: Performance metrics of the model candidates in dataset Alpha .....	78
Table 6.5: Performance metrics of the model candidates in dataset Beta.....	78
Table 6.6: Performance metrics of the model candidates in dataset Delta.....	79
Table 6.7: Cross validation performance of model candidates in dataset Alpha.....	79
Table 6.8: Cross validation performance of model candidates in dataset Beta .....	80
Table 6.9: Cross validation performance of model candidates in dataset Delta .....	80

# 1 Introduction

This thesis explores the machine learning techniques that can be applied to the field of battery health prognosis to find the optimal forecasting algorithm. The model design is driven by various field data obtained from the real-world usage of different lithium-ion battery systems. This introductory chapter consist of three sections, outlining the motivation, the research question, and the structure of the thesis.

## 1.1 Motivation of the Research

In the 21<sup>st</sup> century, lots of attention has been put into transitioning the world to renewable energy. This is driven by the impact of climate change and the target to lower global greenhouse gas emissions, outlined in the Paris Agreement<sup>1</sup>, adopted by the United Nations Framework Convention on Climate Change (UNFCCC) including 196 countries. To achieve those goals, humanity is increasingly reliant on batteries as energy storage to achieve that goal. Lithium demand for battery applications increased by 20 % on average every year from year 2000 to 2010, and the same statistics increased by 107 % on average every year from year 2010 to 2020<sup>2</sup>. The cumulative demand of electric vehicles and energy storage went from just 0.5 GWh in 2010 to roughly 526 GWh in 2020<sup>3</sup>. The global lithium-ion battery market is expected to grow at a compound annual growth rate of 13.1 % from 44.5 billion US dollars to 135.1 billion US dollars.<sup>4</sup>

The longevity of lithium-ion batteries is still not fully understood despite being ubiquitous in our everyday lives. The purpose of battery health prognosis is to know the battery's useful life period for a given use case – or when its capacity and power delivery does not meet certain criteria. This is usually indicated by the “knee point”<sup>5</sup>, typically occurring when useful capacity is around 60-80 % nominal capacity, where the battery starts aging at a much quicker rate. Knowing batteries' longevity is important, as performance degradation and unexpected need to replace batteries will hamper new technologies such as electric vehicles (EV) and home energy storage to be widely adopted<sup>6</sup>.

There are a number of advantages in knowing a certain battery's useful life and longevity. In terms of safety benefits, knowing when a battery's knee point occurs can lower the risk of thermal runaway which is closely associated the mechanisms for degradation<sup>7</sup>. This will decrease the number of incidents where EVs and energy storage units catch on fire during operation<sup>8</sup>, such as the Chevrolet recall<sup>9</sup> that was caused by Bolt EVs catching on fire. In terms of

---

<sup>1</sup> cf. United Nations Framework Convention on Climate Change (The Paris Agreement) 2023

<sup>2</sup> cf. Bhutada, G. (Lithium Consumption Has Nearly Quadrupled Since 2010) 2022

<sup>3</sup> cf. Roper, W. (High Demand for Lithium-Ion Batteries) 2020

<sup>4</sup> cf. Market and Market (Lithium-ion Battery Market) 2022

<sup>5</sup> cf. Shah et al. (State of charge, remaining useful life and knee point estimation based on artificial intelligence and Machine learning in lithium-ion EV batteries: A comprehensive review) 2022

<sup>6</sup> cf. International Energy Agency (Global EV Outlook 2022) 2022.

<sup>7</sup> cf. Kong et al. (Li-Ion Battery Fire Hazards and Safety Strategies) 2018

<sup>8</sup> cf. Sun et al. (A Review of Battery Fires in Electric Vehicles) 2020

<sup>9</sup> cf. Chevrolet (Bolt EV and Bolt EUV Recall Information) 2022

consumers benefits, they can have better expectation of the battery lifespan and software that can automatically deploy or suggest ways of extending the battery's health. With regards to monetary benefits, less money would be needed to service warranties since manufacturers can preemptively repair defective modules before needing a warranty replacement.

As the lithium-ion battery market grows, demand for battery management software will surely follow. But there are many challenges with generating battery health prognosis methods. Lithium-ion batteries do not age at a linear rate that is consistent across the board<sup>10</sup>. Additionally, there are many factors that determine how quickly it ages that it is non-trivial to find which variables are indeed good indicators for health forecasting.

## 1.2 Research Question

This thesis intends to answer the following research question: *What is the optimal data-driven model for forecasting future battery health and aging behavior using machine learning methods?* The optimal model should have the highest accuracy and scalability without needing intensive resources such as utilizations of supercomputers. The model and algorithm should also be robust enough to be applied on different types of battery systems, including residential and mobility sectors. The goal is to find useful features from existing historical data on various battery systems to train a model that achieves under 1 % root mean square error (RMSE) in State of Health (SOH) prediction within an appropriate timeframe. To verify the results, this thesis includes tests of new unseen data or split existing data into training, verification, and test sets.

## 1.3 Structure of the Thesis

Following the introduction chapter, this thesis is divided into four main chapters. Chapter 2 presents the state of the art of battery technology and latest methods of data-driven machine learning (ML) techniques used for regression problems. The first section gives an overview of the types of lithium-ion batteries, how it is used in the electrification of the residential and industrial energy storage, and mobility sector, and the mechanism of health degradation. The following section discusses the basics of the various machine learning techniques explored in this research. The last section explains the methods of evaluating the models and the steps of separating training, test, and validation sets from the processed data.

Chapter 3 presents the in-depth details of the data driven approaches. The first section presents the sources of the data used in this thesis. The second section explains the process of extracting useful features from raw field data that feeds into the various machine learning models. There, backgrounds of different anonymized data sources are found. The third section introduces the best practices of model selection criteria. The fourth section explains the role of each model's hyperparameters.

---

<sup>10</sup> cf. Li (Random forest regression for online capacity estimation of lithium-ion batteries) 2018

Chapter 4 presents the results of the forecasting models. This chapter includes the visualizations of the input features, the process of tuning each model's hyperparameters, accuracy and cross validation metrics of the model candidates, and the test predictions of unknown input made using those trained models.

Chapter 5 discusses the findings from the results. The accuracy, scaling, and ease of use aspect of each model is discussed. Furthermore, the challenges encountered in this thesis is also presented.

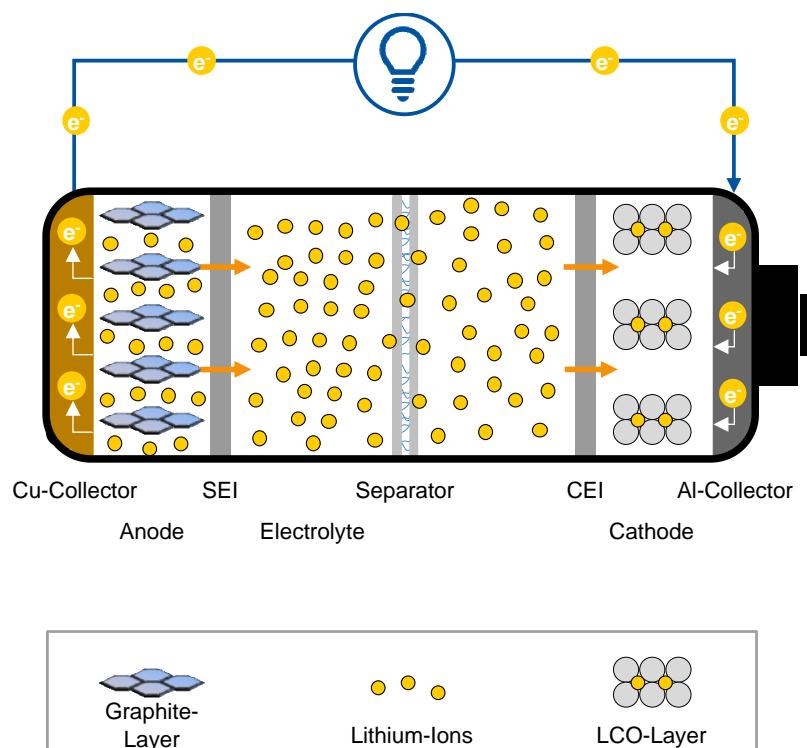
Lastly, chapter 6 summarizes the findings of this research into the appropriate machine learning models best used for the specific application. It also provides additional areas of research recommended for potential future steps into the topic.

## 2 State of the Art

A general overview on the technology relevant to the field of battery health prognosis, machine learning, and previous work done in the field is given in this overview chapter. It is divided into four sections. In the first section, the lithium-ion battery, its applications, and the modes of degradation are introduced. In the second section, the previous works in battery health prognosis, the two types of modelling, and the characteristics of field data are introduced. In the third section, the mathematical background on the machine learning techniques used for the regression problem are introduced. Finally in the fourth section, the training and evaluation method used for the forecasting models are introduced.

### 2.1 Lithium-Ion Batteries

Lithium-ion batteries (LIB) are a type of electrochemical energy storage system. They are so-called secondary batteries because of their rechargeability. Common applications for LIBs include portable electronic devices and electric vehicles. They are known for their high power and energy density, long cycle life, and low maintenance cost compared to other kinds of batteries such as lead-acid battery<sup>11</sup>. Lithium-ion batteries are favored over other types of batteries because of higher energy density, longer lifespan, and they do not suffer from the "memory effect" that can reduce the performance of other types of batteries. Figure 2.1 below shows the components within a lithium-ion battery.



**Figure 2.1: Overview of a lithium-ion battery construction**

<sup>11</sup> cf. Xiaoqiao Zeng et al. (Commercialization of LIBs) 2019, p. 1.

Lithium-ion cells consists of two electrodes (cathode and an anode), a separator that separates the cathode from the anode, and ion-conductive electrolytes that fills the pores of the electrodes and the remaining space inside. They work by moving lithium ions between the two electrodes during charging and discharging. When the battery is being charged, the lithium ions move from the cathode to the anode, and when the battery is discharging, the ions move in the opposite direction. This movement of ions creates a flow of electricity that deliver power for the designed application. Batteries in general are measured in the following metrics listed in Table 2.1.

**Table 2.1: Common battery metrics**

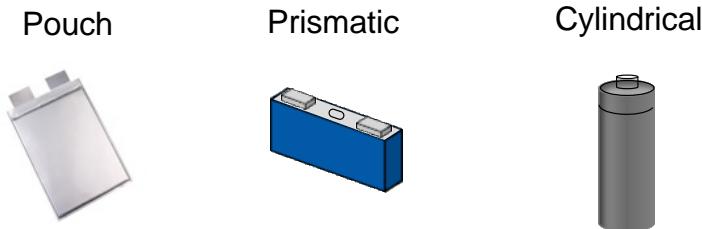
Parameter	Unit	Meaning
State of Charge (SOC)	%	Remaining charge capacity / total charge capacity
State of Health (SOH)	%	Available total capacity / designed capacity
Voltage	V	Electric potential
C-Rate	C (A/Ah)	Charging/discharging current normalized by capacity
Power P	W	Energy per time
Internal Resistance	Ohm	Opposition to the flow of current in battery resulting in the generation of heat
Capacity	Ah	Amount of available current for an hour
Energy Capacity	Wh	Amount of available energy
Energy Density by Weight	Wh/kg	Amount of energy per weight
Energy Density by Volume	Wh/L	Amount of energy per volume
Depth of Discharge (DOD)	%	Amount of energy discharged out of full capacity
Efficiency	unitless	Amount of useful energy available from battery over energy spent charging

SOC is the measure of the current available charge capacity, whereas SOH is the available capacity independent of the SOC. A battery with 90 % SOH would have 90 % of its energy storage capacity compared to when it is new.

C-rate is a measure of how quickly a battery is being charged or discharged (A/Ah). A battery discharging at 1 C would mean it would deplete its total capacity in one hour, and 0.5 C would mean depleting its total capacity in two hours.

### 2.1.1 Types of Lithium-ion Battery

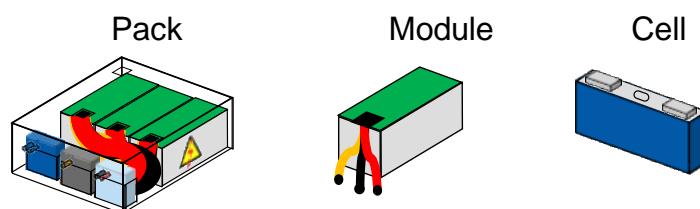
Lithium-ion batteries have many different sizes and form factors. Sizes range from tiny cells that fit inside one's ear to massive systems that fit inside a shipping container. Form factors, meaning shape, include cylindrical, prismatic or pouch, shown in Figure 2.2.



**Figure 2.2: Overview of different cell form factors**

The pouch form factor has a soft shell as the exterior casing which reduces weight and gives flexibility. However, they have the tendency to swell after certain cycles. Examples include modern smartphone batteries. The prismatic form factor contains large sheets of anodes, cathodes, and separators sandwiched into a rigid housing. They are cost-effective and suitable for space-constrained applications. They are typically used in energy storage and medical applications. The cylindrical form factor contains sheets of anodes, cathodes, and separators rolled into a cylindrical can. They have good mechanical stability and do not suffer from deformation. Examples include the standard 21700 and 18650 form factor cells used in automotive application.

In terms of architecture, battery systems of all sorts of capacity exist that are designed for their own specific applications. They can consist of individual cells but can also be connected in series or parallel into bigger groups to increase voltage, power, and capacity for the desired use case. Multiple cells can form into a module, and packs would consist of multiple modules. Figure 2.3 showcases the architecture of most battery systems.



**Figure 2.3: Levels of battery pack constructions with prismatic cells form factor<sup>12</sup>**

A battery module is an assembly consisting of multiple cells connected in series or parallel or combination of both along with the necessary wiring connection and exterior casing. A battery pack is the final shape of the battery system consisting of multiple modules, additional wiring connections, control systems such as the battery management system (BMS), and protection system such as the cooling system.

<sup>12</sup> cf. RWTH-Aachen Institute of Production Engineering of E-Mobility Components

In terms of chemistry, lithium-ion batteries can have different chemical makeup in the cathode and anode active material. They all have their own balance of advantages and disadvantages and are suitable for different applications. Common chemistry types include<sup>13</sup>:

1. Lithium Cobalt Oxide (LCO) – energy dense but low thermal stability
2. Lithium Manganese Oxide (LMO) – safest chemistry
3. Lithium Nickel Manganese Cobalt Oxide (NMC) – expensive but high performance
4. Lithium Iron Phosphate (LFP) – affordable, safe, and reliable
5. Lithium Nickel Cobalt Aluminum Oxide (NCA) – high energy density with long life
6. Lithium Titanate (LTO) – long life and fast charging

For the purpose of this research, it is the goal of the algorithm to be adaptable for all the different types of chemistry, form factor, architecture, size, and application. None of those factors should matter to the algorithm as long as the training data and prediction application is consistent for each specific model. Exemplary models used to predict SOH of a LFP chemistry home storage battery system with 10 kWh capacity would be trained with data from the same type of system with the same capacity and chemistry. Similarly, the same technique should yield similar accuracies predicting SOH of a NMC electric vehicle battery with 100 kWh capacity.

### 2.1.2 Lithium-ion Battery Applications

Lithium-ion batteries serve mainly as the energy storage as the world transitions away from fossil fuel towards renewable energy. In 2020, transportation and consumer electronics accounts for 54 % and 32 % of battery demand by application respectively<sup>14</sup>. However, it is projected that transportation and stationary applications will be the top two accounting for 85.7 % and 8 % respectively. Therefore, the transportation and energy sector where fossil fuel still currently dominates are the two applications focused in this thesis.

#### 2.1.2.1 E-Mobility Systems

The transportation sector is the main source of growth in electrification and battery use. Electric vehicles (EV) are quickly being adopted as the world transitions away from fossil fuel to reduce impact on climate change, in addition to the efficiency benefit becoming more apparent. EVs are more efficient at converting stored energy into motion compared to traditional internal combustion engine (ICE) vehicles, and they emit less greenhouse gas throughout their life cycle than its fossil fuel counterparts<sup>15</sup>. Many cities in the world have started to utilize electric ride-share scooters as an alternative cleaner mode of transportation, as worldwide users rose from 0.5 million in 2017 to 77.1 million in 2022 and is expected to grow to 134.7 million by 2027<sup>16</sup>. To illustrate this application, Figure 2.4 below shows the various components of an electric vehicles.

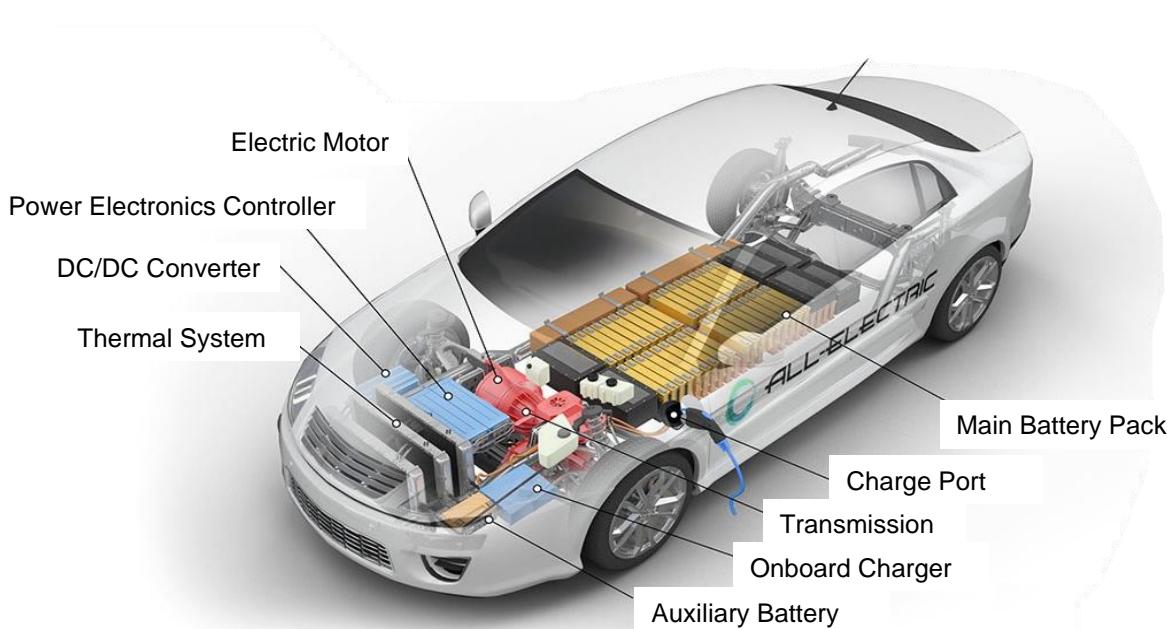
---

<sup>13</sup> cf. Heimes et al. (Lithium-Ion Battery Cell Production Process) 2018

<sup>14</sup> cf. Statista (Projected global battery demand from 2020 to 2030, by application) 2021

<sup>15</sup> cf. Hawkins (Comparative Environmental Life Cycle Assessment of Conventional and Electric Vehicles) 2013

<sup>16</sup> cf. Statista (E-Scooter-sharing – Worldwide) 2022



**Figure 2.4: Illustration of electric vehicle components and systems<sup>17</sup>**

### 2.1.2.2 Energy Storage Systems

The Battery Energy Storage System (BESS) in the home and industrial sector is the second main sources of growth in electrification and battery use<sup>18</sup>. As the world transitions into renewable energy, energy storage becomes necessary since the energy production fluctuates drastically. Solar power generation occurs only during daylight hours, and wind turbines produce power depending on the weather. Solar photovoltaic (PV) technology converts sunlight into electrical energy by using semiconducting materials that absorb light into electron particles<sup>19</sup>. PV panels on residential roofs along with local battery storage is a popular solution that is gaining in popularity as prices drop<sup>20</sup>. Similarly, utility-scaled BESS are used to store excess energy when energy production exceeds energy demand and conversely supplies energy to the grid when demand exceeds production.

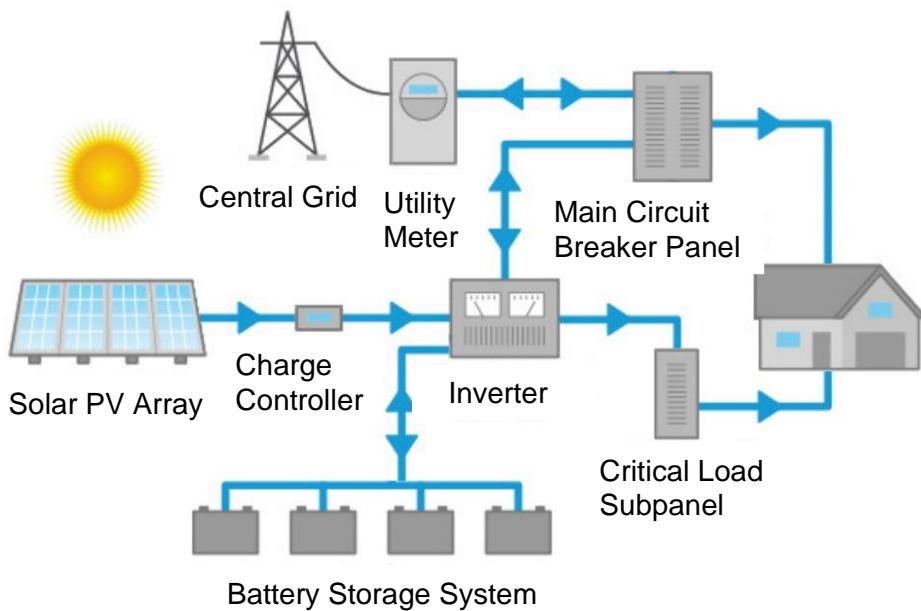
To illustrate this application, Figure 2.5 below shows how residential battery storage system is coupled with solar PV panels and all their various components.

<sup>17</sup> cf. U.S. Department of Energy (How Do All-Electric Cars Work)

<sup>18</sup> cf. Sterner et al. (Handbook of Energy Storage): 2019 p.53

<sup>19</sup> cf. U.S. Department of Energy (Solar Photovoltaic Cell Basics)

<sup>20</sup> cf. Sterner et al. (Handbook of Energy Storage): 2019 p.60



**Figure 2.5: Diagram of integrated residential solar battery storage system<sup>21</sup>**

### 2.1.3 Battery Degradation

As batteries get used, the available capacity decreases over time and use, and the internal resistance increases which leads to lower power capability. This phenomenon is called referred to battery degradation and is measured by State of Health in terms of capacity (SOH-C) or internal resistance (SOH-R). This thesis focuses on the capacity measurement of SOH because it is easier to obtain from field data, and LIB users are usually more concerned about loss of capacity over loss of performance. A battery's useful life is considered to be from 100 % SOH until around 80 % SOH for most applications<sup>22</sup>, at which the battery no longer meets its requirements of capacity and/or power performance. At that point, the system will require replacement of the battery in order to continue operation for its intended purpose.

There are two main modes of degradation<sup>23</sup>:

1. Cyclical aging – referring to the degradation due to cycling or usage of the battery.
2. Calendric aging – referring to the degradation due to age or time since manufactured date.

<sup>21</sup> cf. Solar Power Now (Solar Power Storage)

<sup>22</sup> cf. Shah et al. (State of charge, remaining useful life and knee point estimation based on artificial intelligence and Machine learning in lithium-ion EV batteries: A comprehensive review) 2022

<sup>23</sup> cf. Ecker, M .(Calendar and cycle life study of Li(NiMnCo)O<sub>2</sub>-based 18650 lithium-ion batteries) 2013

Physically, there are three degradation mechanisms that happens in the chemical level<sup>24</sup>:

1. Loss of lithium inventory – when lithium ions are consumed by parasitic reactions.
2. Loss of positive electrode material – when active mass of cathode is no longer available for insertion of lithium.
3. Loss of negative electrode material – when active mass of anode is no longer available for insertion of lithium.

The following battery conditions are known to cause accelerated battery degradation:

- Depth of Discharge (DOD) – correlating to the cyclical aging, the amount of usage and how deep the charge/discharge cycle is are factors in battery degradation. A study by Ning and Popov found a direct correlation to high cycle count causing capacity decrease<sup>25</sup>.
- Extreme temperature – battery temperature, depending on use conditions and season, is known to have a big impact on lithium-ion battery performance and degradation. They require a particular narrow temperature range to have optimal capacity and power delivery. A study by Jalkanen et al. found a much higher rate of capacity fade when charging at 45 °C and discharging at 65 °C as compared to room temperature<sup>26</sup>. Both extreme cold and extreme heat affect LIB's performance and cause accelerated degradation<sup>27</sup>.
- High voltage – extreme voltages and, in relation, State of Charge (SOC) are factors in battery degradation. Overvoltage of parasitic reaction and high end of charge voltage are both studied to have an adverse effect on battery health<sup>28</sup>.
- High current load – high charge and discharge rates contributes to higher rate of battery degradation. A study by Ning et al. found that there is a direct correlation for discharge rate vs capacity fade<sup>29</sup>.

As LIB degrades, a nonlinear degradation curve can occur at the knee point when "the capacity slowly declines throughout most of the battery's life, but begins to decrease rapidly in the latter stages"<sup>30</sup>. The beginning aging trajectories are often linear or sublinear<sup>31</sup> until the knee point, where it starts to age in a superlinear way. Figure 2.6 below shows the different aging trajectories possible for LIB.

---

<sup>24</sup> cf. Birk, C.R. (Degradation diagnostics for lithium ion cells) 2016

<sup>25</sup> cf. Ning et al. (Cycle Life Modeling of Lithium-Ion Batteries) 2004

<sup>26</sup> cf. Jalkanen et al. (Cycle aging of commercial NMC/graphite pouch cells at different temperatures) 2015

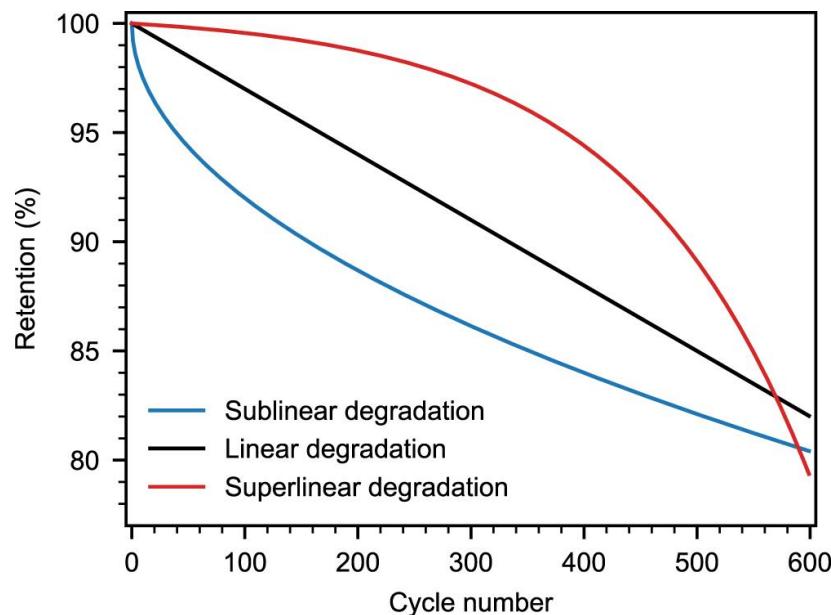
<sup>27</sup> cf. Vetter et al. (Ageing mechanisms in lithium-ion batteries) 2005

<sup>28</sup> cf. Ning et al. (Cycle Life Modeling of Lithium-Ion Batteries) 2004

<sup>29</sup> cf. Ning et al. (Capacity fade study of lithium-ion batteries cycled at high discharge rates) 2003

<sup>30</sup> cf. IEEE (Standard 485-2020) 2020

<sup>31</sup> cf. Attia et al. (Review—"Knees" in Lithium-Ion Battery Aging Trajectories) 2022



**Figure 2.6: Plot of linear, sublinear, and superlinear aging trajectories for LIB<sup>32</sup>**

Linear degradation means a constant rate aging. Sublinear degradation means a decreasing rate of aging as time passes. Superlinear degradation means an increasing rate of aging as time passes.

## 2.2 Previous Works in Battery Health Prognosis

Throughout academia, there has been a good amount of research on the topic of battery prognosis, which helps to establish the starting point of this thesis. The main types of prognosis methods are empirical models and theoretical models.

### 2.2.1 Physics-based Modelling

Physics-based models, also called theoretical models, uses known mechanical degradation modes as basis to parameterize the models. These models<sup>3334</sup>, describing degradation with electrochemical relationships, have the potential to be much more precise and cover wider range of conditions<sup>35</sup>. However, they rely on detailed physical information of the cell components and come at a much higher computation cost, making them challenging to apply in the real world.

One example of a physics-based model is Prada et al.'s electrochemical and thermal aging model<sup>36</sup> that established the capacity fade correlation to Solid Electrolyte Interphase (SEI) film growth at the negative electrode, cell impedance increases such as the SEI film resistance,

<sup>32</sup> cf. Attia et al. (Review—"Knees" in Lithium-Ion Battery Aging Trajectories) 2022

<sup>33</sup> cf. Purewal (Degradation of lithium ion batteries employing graphite negatives and nickel–cobalt–manganese oxide + spinel manganese oxide positives: Part 2, chemical–mechanical degradation model) 2014

<sup>34</sup> cf. Deshpande (Battery Cycle Life Prediction with Coupled Chemical Degradation and Fatigue Mechanics) 2012

<sup>35</sup> cf. Jin et al. (Comparison of Li-Ion Battery Degradation Models for System Design and Control Algorithm Development) 2017

<sup>36</sup> cf. Prada et al. (A Simplified Electrochemical and Thermal Aging Model of LiFePO<sub>4</sub>-Graphite Li-ion Batteries: Power and Capacity Fade Simulations) 2013

and the electrolyte mass transport resistance due to the mitigation of the negative electrode porosity.

### 2.2.2 Data-driven Modelling

Data-driven models, also called empirical models, uses historical data to form aging correlations<sup>37</sup>. Because measurements needed for physics-based models are difficult to obtain in the real world requiring costly instrumentations, data-driven models' advantage lies in using data from operation for calibration<sup>38</sup>. This enables a simpler way to incorporate online forecasting for fleets. Hence, this thesis intends to focus on data-driven models using ML methods. The models would extract battery conditions from field data that are known to cause accelerated battery degradation.

One example is Sohn et al.'s two stage deep learning model. A Convolutional Neural Network (CNN) based model "extracts temporal features across past and current cycles to sort out those that should be monitored closely for near-term failures, and then predict the number of cycles left to reach the knee-point for them."<sup>39</sup> The model extracts features from time-series data which "reflects dynamic changes in battery properties, resulting in improved prediction performance under realistic scenarios."<sup>40</sup> In addition to neural network approaches, Elastic Net regularized linear regression<sup>41</sup> and Gaussian Process Regression have also been used as ML approaches to health prognosis. Additional methods of estimating SOH states include Kalman Filter and Particle Filter<sup>42</sup>.

Field battery data in the industry consists of time-series data of various sensors collected by the battery management system (BMS), such as voltage, current, and temperature of the battery. They represent a snapshot of the battery's condition at each timestamp. Table 2.2 below shows a typical time-series data obtained from the BMS.

---

<sup>37</sup> cf. Xu et al. (Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment) 2018

<sup>38</sup> cf. Vermeer (A Comprehensive Review on the Characteristics and Modeling of Lithium-Ion Battery Aging) 2021

<sup>39</sup> cf. Sohn et al. (Two-stage deep learning for online prediction of knee-point in Li-ion battery capacity degradation) 2022

<sup>40</sup> cf. Sohn et al. (Two-stage deep learning for online prediction of knee-point in Li-ion battery capacity degradation) 2022

<sup>41</sup> cf. Severson et al. (Data-driven prediction of battery cycle life before capacity degradation): 2019

<sup>42</sup> cf. Xu et al. (Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment) 2017

**Table 2.2: Sample time-series data obtained from BMS**

<b>Time</b>	2018-03-28 10:07:00+0 0:00	2018-03-28 10:08:00+0 0:00	2018-03-28 10:09:00+0 0:00	2018-03-28 10:10:00+0 0:00	2018-03-28 10:11:00+0 0:00
<b>Voltage</b>	48.03	47.96	48.06	48.08	48.08
<b>Voltage_Cell_Min</b>	3.684	3.684	3.684	3.685	3.686
<b>Voltage_Cell_Max</b>	3.695	3.695	3.695	3.699	3.699
<b>Current</b>	17.23	17.03	17.03	17.68	17.92
<b>Current_Child_Min</b>	6	6	5.73	6	6
<b>Current_Child_Max</b>	6	6	5.64	6	6
<b>Current_Limit_Min</b>	45	45	45	45	45
<b>Current_Limit_Max</b>	50	50	49	50	50
<b>Tempera-ture_Cell_Min</b>	14.6	14.6	14.6	14.6	14.7
<b>Tempera-ture_Cell_Max</b>	15.1	15.1	15.2	15.2	15.2
<b>Remaining_Capa-city</b>	55.5	55.8	56	56.3	56.6
<b>Full_Charge_Capa-city</b>	142	142	141.9	142	142
<b>State_Of_Charge</b>	39	39	39	39.5	39.5
<b>State_Of_Health</b>	100	100	100	100	100
<b>Cycle_Count</b>	0	0	0	0	0
<b>Tempera-ture_Child_Min</b>	14.6	14.6	14.6	14.6	14.7
<b>Tempera-ture_Child_Max</b>	15.1	15.1	15.2	15.2	15.2
<b>Temperature</b>	14.85	14.85	14.9	14.9	14.95
<b>ID</b>	5.2441E+1 6	5.2441E+1 6	5.2441E+1 6	5.2441E+1 6	5.2441E+1 6

The most important measurements in field data are the voltage, current, and temperature measurements, as they pertain to the modes of health degradation mentioned in section 2.1.3. These data are logged by the device's BMS and are stored or transmitted online.

## 2.3 Machine Learning Techniques

The problem of predicting future SOH values with certain use case input is a regression problem. Regression is the statistical process for estimating the relationships between a dependent variable ('prediction' in this thesis) and one or more independent variables ('features' in this thesis)<sup>43</sup>. It is represented by the equation 2.1, with input  $\mathbf{X}$  and output  $\mathbf{Y}$  as a function of the input and the model parameter  $\beta$ , plus error term  $e$ :

$$Y_i = f(\mathbf{X}_i, \beta) + e_i \quad (2.1)$$

The process of generating a regression model involves training the model with  $N$  number of observations of the input and its corresponding output. Suppose we observe a real-valued input variable  $\mathbf{x} \equiv (x_1, \dots, x_N)^T$  and we wish to use this observation to predict the value of a real-valued target variable  $\mathbf{t} \equiv (t_1, \dots, t_N)^T$ . In this thesis, the target variable is the month-to-month change in SOH values, and the input variables are the features extracted from the data recorded by the battery management system (BMS). The process of extracting features is presented in the next section.

The goal of this thesis as stated previously is to explore the optimal ML models for the specific application of battery health prognosis. The prediction of future health status can then be generated by inputting the expected usage patterns of the past into the model. The types of regression models chosen as candidates are based upon both past academic research<sup>44</sup> and the latest methodologies taught in university courses. In the following sections, a general overview of each regression models explored is explained.

### 2.3.1 Linear Regression

The simplest model for regression is one that involves a linear combination of the input variables, shown in the following equation, with output  $y$  as a function of input  $x$  and weight  $w$ :

$$y(x, w) = w_0 + w_1 x_1 + \dots + w_i x_i \quad (2.2)$$

In matrix form, it can be represented in the form of basis function  $\phi(x)$  that could be any fixed linear or nonlinear function of the input variables:

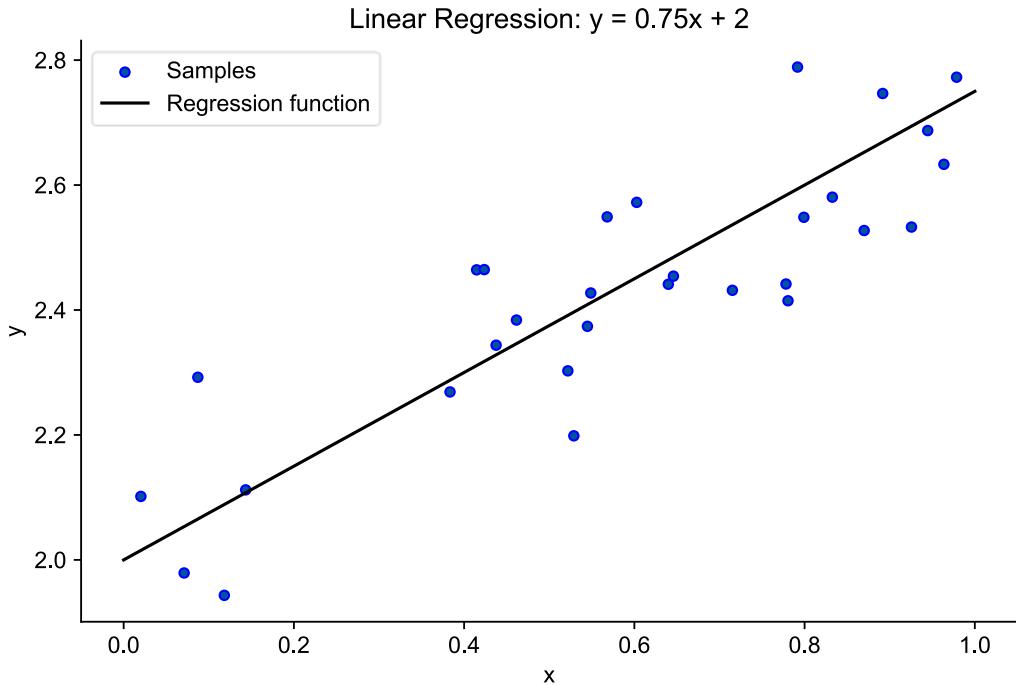
$$y(x, w) = \sum_{i=1}^{M-1} w_i \phi_i(x) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad (2.3)$$

To illustrate it visually, Figure 2.7 below demonstrates a sample 1-dimensional dataset that would be appropriate for a linear regression:

---

<sup>43</sup> cf. Bishop, C.M. (Patter Recognition and Machine Learning) 2006

<sup>44</sup> cf. Li et al. (Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review) 2019



**Figure 2.7: Example of simple linear regression of one input variable**

Linear regression fits a function with coefficients  $w = (w_1, \dots, w_p)$  to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Mathematically it solves a problem of the form called Ordinary Least Squares (OLS):

$$\min_w \|Xw - y\|_2^2 \quad (2.4)$$

### 2.3.1.1 Polynomial Regression

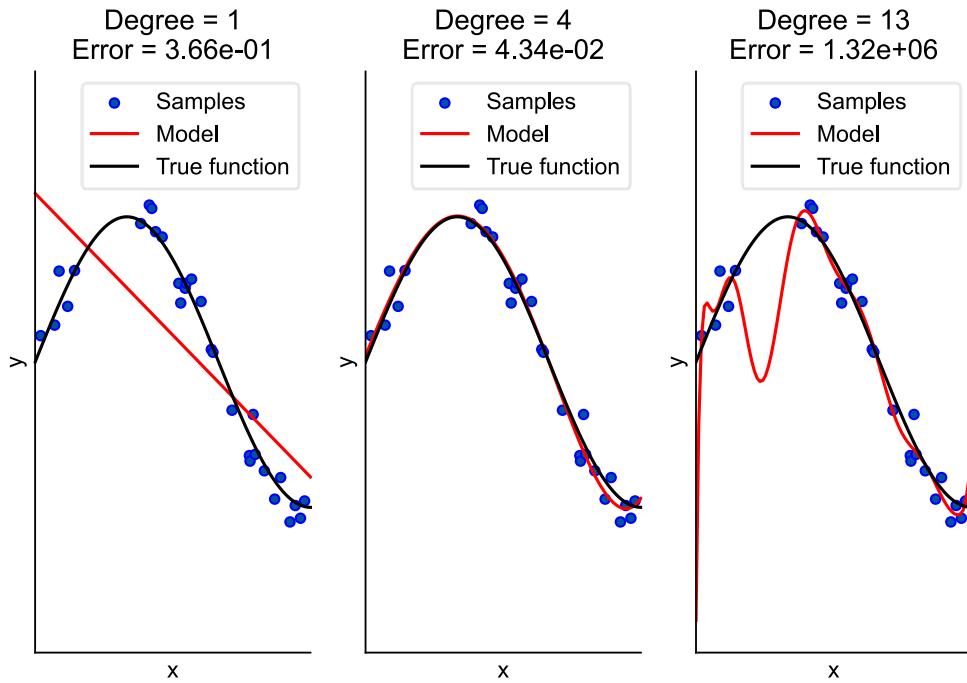
Polynomial regression can be equated to extending the linear regression with polynomial features as its basis function. It is similar to linear regression in that it models the behavior using a linear combination of the input variable, but it can also model a polynomial behavior by including the input variable raised to certain number of powers. It can be represented in the following equation:

$$y(x, w) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j \quad (2.5)$$

For a 2-dimensional problem, the expanded equation is as follows:

$$y(x, w) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 \quad (2.6)$$

The following figure illustrates visually the difference in the specifying the parameter for the model's order of power.



**Figure 2.8: Plots of polynomials having various orders  $M$ , shown as red curves, fitted to a sample noise-added dataset of  $\sin(x)$**

The black curve represents the ground truth of a  $\sin(x)$  function. The blue points represent the physical observation simulated by adding noise to the ground truth, and the red curve represents the polynomial regression model fitted to the training dataset consisting of the input ( $x$ ) variables and target ( $t$ ) variables of the observation in various polynomial degrees.  $M = 1$  represents a linear function acting in the same behavior as a linear regression which underfits the data,  $M = 4$  would be close to the optimal fit to the curve, while  $M = 15$  would be an overfitted model that would not yield accurate results between the training points.

### 2.3.1.2 Ridge Regression

Ridge regression is an extension of the simple linear regression that addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients<sup>45</sup>. The ridge coefficients minimize a penalized residual sum of squares represented with the following objective function:

$$\min_w \left\| Xw - y \right\|_2^2 + \alpha \left\| w \right\|_2^2 \quad (2.7)$$

The complexity parameter  $\alpha \geq 0$  controls the amount of shrinkage: the larger the value of  $\alpha$ , the greater the amount of shrinkage and thus the coefficients become more robust to collinearity.

<sup>45</sup> cf. Pedregosa, F. (Scikit-learn: Machine Learning in Python) 2011

### 2.3.1.3 Lasso

Lasso (Least Absolute Shrinkage and Selection Operator) is another extension of linear regression that estimates sparse coefficients<sup>46</sup>. It is useful in some contexts due to its tendency to prefer solutions with fewer non-zero coefficients, effectively reducing the number of features upon which the given solution is dependent. For this reason, Lasso and its variants are appropriate for the purpose of compressed sensing. It minimizes the least-squares penalty with a L1-norm of the coefficients as seen from the following objective function:

$$\min_w \frac{1}{2n_{\text{samples}}} \|Xw - y\|_2^2 + \alpha \|w\|_1 \quad (2.8)$$

### 2.3.1.4 Elastic-Net

ElasticNet is an extension of linear regression model trained with both L1 and L2-norm regularization of the coefficients<sup>47</sup>. This combination allows for learning a sparse model where few of the weights are non-zero like Lasso, while still maintaining the regularization properties of Ridge. It minimizes the following objective function with  $\alpha$  and  $\rho$  as regularization parameters:

$$\min_w \frac{1}{2n_{\text{samples}}} \|Xw - y\|_2^2 + \alpha\rho \|w\|_1 + \frac{\alpha(1-\rho)}{2} \|w\|_2^2 \quad (2.9)$$

## 2.3.2 Principal Component Analysis and Partial Least Squares Regression

Another strategy of improving predictive results is to lower the dimensionality of the input function. Principal Component Regression (PCR) and Partial Least Squares (PLS) Regression both belong to the decomposition family of estimators.

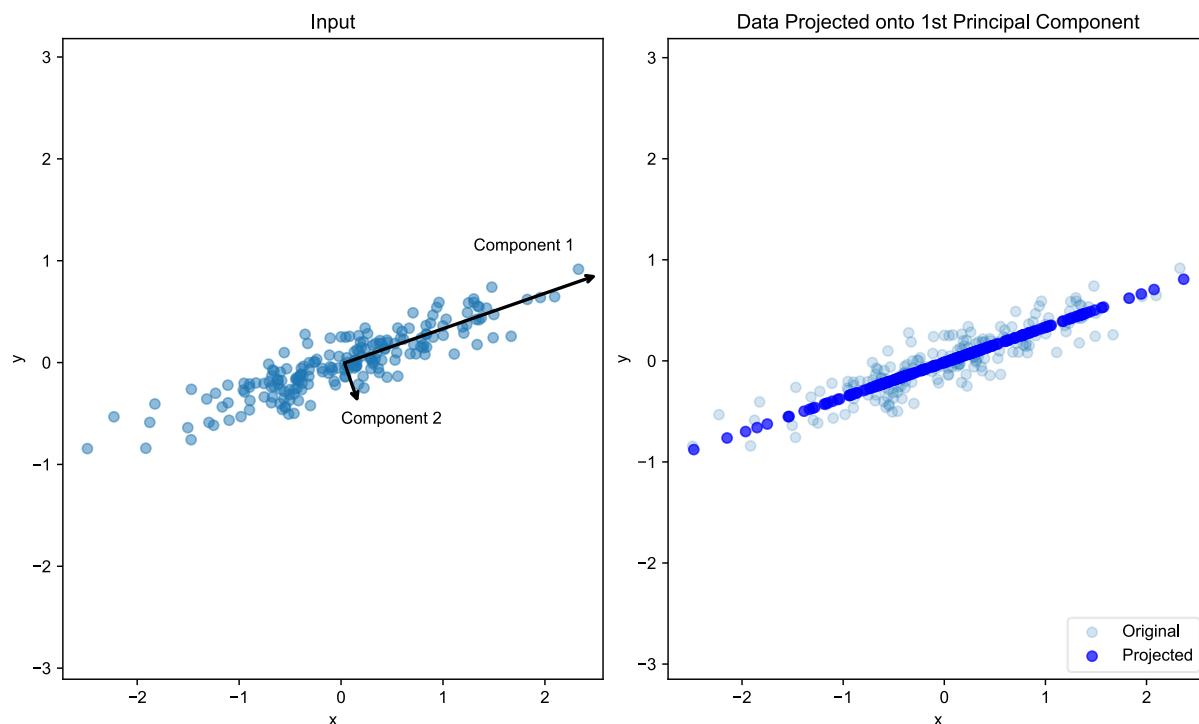
PCA is a linear dimensionality reduction technique that transforms a set of  $p$  correlated variables into a smaller  $k$  number of uncorrelated variables called principal components while retaining as much of the variation in the original dataset as possible. The first principal component is the direction from the data's center with the highest variance, and the subsequent principal components are the directions with the next highest variance. They are calculated by performing eigen-decomposition or singular value decomposition (SVD) and taking the highest  $k$  eigenvalues and corresponding eigenvectors of the covariance matrix<sup>48</sup>. The input will then be projected onto the principal components as the new input feature for regression. Figure 2.9 below illustrates an example of PCA being applied to a 2-dimension random generated input.

---

<sup>46</sup> cf. Pedregosa, F. (Scikit-learn: Machine Learning in Python) 2011

<sup>47</sup> cf. Pedregosa, F. (Scikit-learn: Machine Learning in Python) 2011

<sup>48</sup> cf. Abdi (Principal component analysis) 2010



**Figure 2.9: Illustration of PCA dimensionality reduction applied to a 2-dimensional input**

The left plot shows the resulting two principal components gained by PCA, where the direction of component 1 contains the biggest variance. The right plot shows the result of projecting the input data onto the direction of component 1 where a linear regression can be applied next.

Both PCR and PLS requires scaling of the input features because the components are highly sensitive to the relative ranges of the original features. It will attempt to capture the variance that occurred due to relative ranges, not the true variance between the features. PLS is similar to PCA in reducing dimensionality of the samples with the main difference being that PLS takes account of the target output for the supervised transformation<sup>49</sup>. It also attempts to find the multidimensional direction in the input space that explains the maximum variance direction in the output space. PCR decomposes input  $\mathbf{X}$  and output  $\mathbf{Y}$  by maximizing the covariance between their projections as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \quad (2.10)$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F}$$

$\mathbf{T}, \mathbf{U}$  = Projection of  $\mathbf{X}, \mathbf{Y}$

$\mathbf{P}, \mathbf{Q}$  = Orthogonal Loading Matrices

$\mathbf{E}, \mathbf{F}$  = Error terms

### 2.3.3 Gaussian Process Regression

Gaussian Process (GP) is a probabilistic method that gives a confidence for the predicted function. Gaussian Process Regression (GPR) is a supervised learning method designed to

<sup>49</sup> cf. Wold (PLS-regression: a basic tool of chemometrics) 2001

solve regression problems by fitting a function to the data using GP. GP can be seen as the joint distribution of (infinitely many) variables, each with multivariate normal (Gaussian) distribution, covering the continuous input domain, e.g., time or space. Instead of minimizing some objective function, GPR is based on the Bayesian framework, which computes the posterior distribution over the function values at any input location.

For a given set of training points, there are potentially infinitely many functions that fit the data. Gaussian processes offer a solution to this problem by assigning a probability to each of these functions. The mean of this probability distribution represents the most probable characterization of the data. Furthermore, this probabilistic approach allows the model to incorporate the confidence of the prediction into the regression result<sup>50</sup>.

The kernel, also called covariance function, is a crucial factor of GP that determines the shape of the prior and posterior. It expresses the similarity between any two given datapoints with the assumption that similar datapoints should have similar target values. The gaussian process fit automatically selects the best model parameters that maximize the log-marginal likelihood – which is the integral of the likelihood times the prior. Common kernels include the radial basis function (RBF) kernel and rational quadratic (RQ) kernel, represented by equations 2.11-12.

$$\text{RBF kernel} := k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right) \quad (2.11)$$

$d(\cdot, \cdot)$  = Euclidean distance

$l$  = length-scale parameter

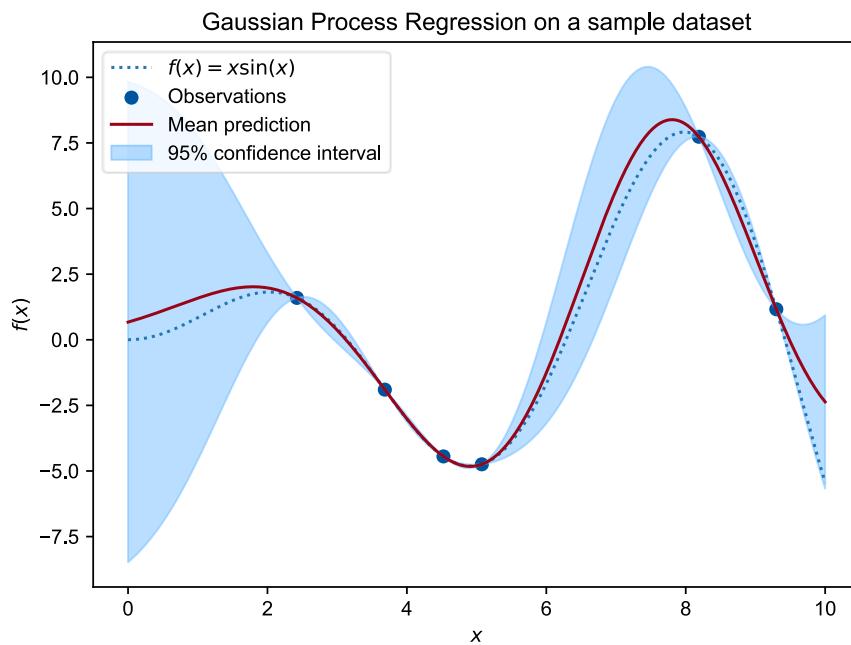
$$\text{RQ kernel} := k(x_i, x_j) = \left(1 + \frac{d(x_i, x_j)^2}{2\alpha l^2}\right)^{-\alpha} \quad (2.12)$$

$\alpha$  = scale mixture parameter

Figure 2.10 below illustrates a 2-dimensional dataset and the resulting fit and probability distribution using RBF kernel:

---

<sup>50</sup> cf. Rasmussen, C.E. (Gaussian Processes in Machine Learning) 2004



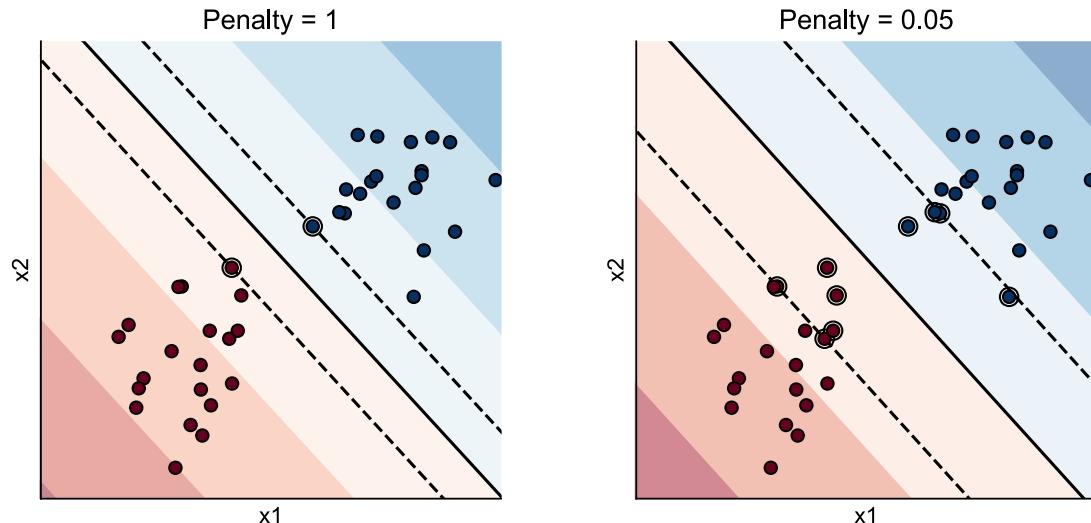
**Figure 2.10: Gaussian Process Regression model on a sample dataset**

The blue points represent the datapoints being fitted, the light blue area represent the confidence interval where no observation is available, and the red curve represents the mean of all predictive distribution which is the regression output.

### 2.3.4 Support Vector Regression

Support Vector Machines (SVM) is another supervised learning approach to predictive modelling for classification and regression. It is based on minimizing the empirical risk, which is the sum of the losses incurred when predicting the output variable for each data point in the training set. It works by separating the datapoints with a hyperplane that has the most separation from them, called the margin<sup>51</sup>. The resulting two parallel lines that are equidistant from the hyperplane touching the datapoints are called support vectors. Figure 2.11 below illustrates the resulting hyperplane for a sample classification dataset.

<sup>51</sup> cf. Khanna (Efficient Learning Machines) 2015



**Figure 2.11: Differing support vectors and separating hyperplane of Support Vector Machines due to varying penalty parameters**

The solid black line represents the separating hyperplane, the dotted line represents the support vectors, and the distance between the support vectors represents the varying sizes of margin. The margin and regularization factor can be tuned to balance the trade-off between the model accuracy and its generalization performance, changing the number of misclassified datapoints and the magnitude of associated penalty to the loss function. Seen from Figure 2.11The penalty is inversely proportional to the amount of misclassified datapoints, represented by the circled dots, allowed within the support vectors.

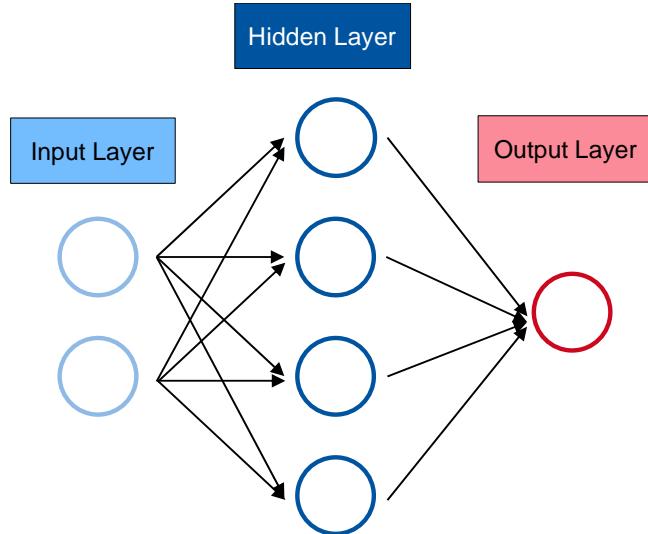
The regression extension of SVM, Support Vector Regression (SVR), fits the data by finding a separating hyperplane that encloses the input points within the margin<sup>52</sup>. SVR is able to handle nonlinear relationships between the input and output by using nonlinear kernel functions to transform the input features into a higher dimensional space. This allows SVR to find a nonlinear boundary that describes the datapoints and creates the best fit function for predicting the output variable. The kernel function is a measure of similarity between any two input points. Commonly used kernel functions include linear, polynomial, and radial basis function (RBF).

### 2.3.5 Artificial Neural Network

Artificial neural network (ANN) is a versatile and adaptive ML method that has been popular in many regression and classification applications. They are made up of components inspired by the biological nervous system. Multi-layer Perceptron Regressor (MLPR) is a feed-forward neural network, meaning the inputs are sent in the forward direction<sup>53</sup>. It can also contain hidden layers of various sizes which can make the model even denser. They include a system of the input layer of input features, hidden layer(s) of neurons, and the output layer, shown in Figure 2.12.

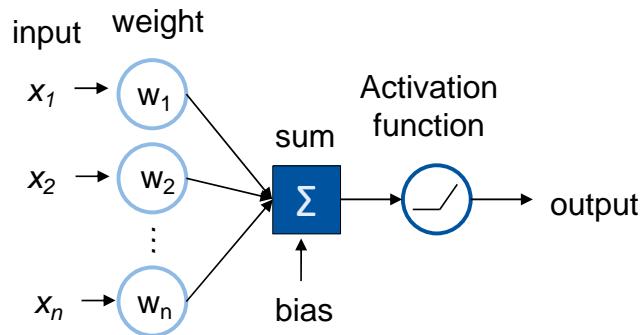
<sup>52</sup> cf. Zhang et al. (Chapter 7 - Support vector regression) 2020

<sup>53</sup> cf. Goodfellow et al. (Deep Learning) 2016, p. 165



**Figure 2.12: Example of a neural network**

The input and output layers correspond to the input and output of the regression model. Each connected neuron consists of weights for every neuron connected to it and its bias, shown in Figure 2.13.



**Figure 2.13: Schematic of an ANN's neuron**

There exist many activation functions that affects the behavior of an ANN. Some of the common choices are listed in equations 2.13-15 below:

$$\text{Sigmoid function } g(x) = \frac{1}{1 + e^{-x}} \quad (2.13)$$

$$\text{Hyperbolic tangent (tanh) function } g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.14)$$

$$\text{Rectified Linear Unit (ReLU) function } g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.15)$$

The training process of an ANN involves stochastic gradient descent to obtain the learned weights and biases of the model, given by equation 2.16. It utilizes a series of back propagation to compute the gradient of a loss function with respect to the weights of the ANN one layer at a time, iterating backwards from the last layer to the first<sup>54</sup>. The resulting gradient is added to

<sup>54</sup> cf. Goodfellow et al. (Deep Learning) 2016

the existing weights, and the process repeats until convergence. For regression, MLP uses the Mean Square Error loss function, given by equation 2.17:

$$\mathbf{W}^{i+1} = \mathbf{W}^i - \epsilon \nabla Loss_{\mathbf{W}}^i \quad (2.16)$$

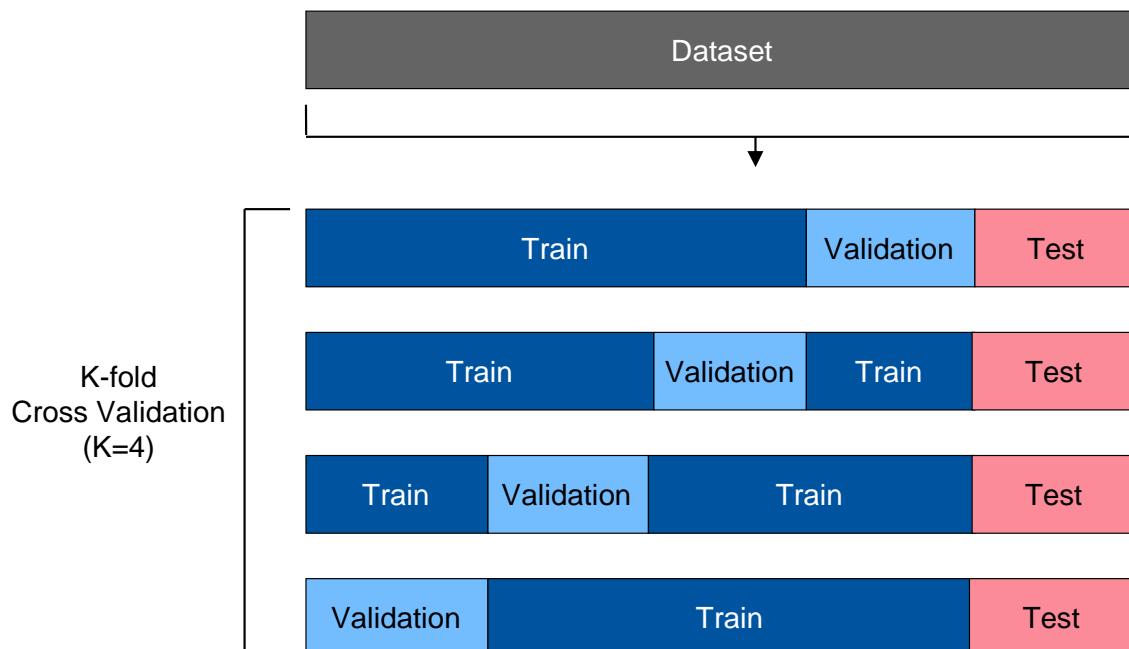
$\mathbf{W}^i$  = Weights at i-th iteration

$\epsilon$  = learning rate

$$Loss(\hat{\mathbf{y}}, \mathbf{y}, \mathbf{W}) = \frac{1}{2n} \sum_{i=0}^n \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2 + \frac{\alpha}{2n} \|\mathbf{W}\|_2^2 \quad (2.17)$$

## 2.4 Training and Evaluation

In the field of data science, it is crucial to ensure unbiased results by establishing the training, evaluation, and verification set. It is standard practice to randomize the data points and have separate portions of the available data used to train the model and evaluate how the trained model performs. This way the evaluation is done with data unseen by the model. Then the test set can be sourced from new unseen data or split from the existing data. Figure 2.14 below illustrates how a dataset from this research is split:



**Figure 2.14: Illustration of how the dataset is split into training, validation, and test sets**

K-fold cross validation is an evaluation method used in machine learning to learn how a model performs with unseen data<sup>55</sup>. Each dataset is shuffled and split into k number of samples where 1 is used as the test set while the rest is used as the training set. Then the other subsamples will act as the test set in the subsequent iteration through the whole dataset. In this research,

<sup>55</sup> cf. Kelleher et al. (Fundamentals of Machine Learning for Predictive Data Analytics) 2015, p.469

the test set is held out in terms of battery system and k-fold cross validation is used within the training and validation set in terms of individual data points.

To evaluate a model's accuracy performance, two metrics are used – root mean square error (RMSE) and coefficient of determinant (R2 score). The RMSE and R2 scores are computed using the validation set after each model is trained using the training set. All datasets are sampled randomly to divide itself into 80 % for the training set and 20 % for the verification set.

RMSE measures the average difference of predicted and true value and is calculated with equation 2.18:

$$\text{RMSE}(y, \hat{y}) = \sqrt{\left( \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2 \right)} \quad (2.18)$$

$y_i$  = true value

$\hat{y}_i$  = predicted value

R2 score represents the proportion of variance of output that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance<sup>56</sup>. It is calculated with equation 2.19:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.19)$$

$y_i$  = true value

$\hat{y}_i$  = predicted value

$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$  = true value

---

<sup>56</sup> cf. Sci-kit Learn

## 3 Data Driven Approaches

This chapter presents the algorithm of forecasting battery health using the machine learning techniques explained in section 2.2, the origin of the data used in this research, and the criteria of choosing the optimal model. The following sections goes over the process of obtaining the input and output feature for the predictive models, the process of training, evaluating, testing, and tuning the models.

The prognosis algorithm consists of the following steps:

1. Process all available raw field data to obtain the specific monthly input feature  $\mathbf{x}$  and output  $\mathbf{y}$  for the predictive models
2. Split processed data into training, validation, and test set
3. Train model(s) with the training set and determine the best configuration with evaluation set
4. Test performance of trained models with test set or new unseen data
5. Deploy model to obtain health forecast and update model over time as new data arrive by processing monthly field data with the same procedure as the first step

### 3.1 Data Sources

The data required for predictive modeling used in this research are obtained from three different sources. This thesis was written as part of a partnership with a company that specializes in battery analytics. The company provided part of the data utilized in this research sourced from their customers. To ensure robustness of the prognosis algorithm, multiple types of lithium-ion battery system are included in this research: two from residential sector and one from lab testing. With the goal of utilizing field data to train models for health forecasting, the results from those datasets will take precedent over the lab data, which serves as a verification.

#### 3.1.1 Dataset Alpha and Beta

The first two datasets, referred to as dataset *Alpha* and *Beta*, are both residential energy storage systems from different manufacturers. They utilize lithium-ion batteries to store excess electricity produced by the solar photovoltaic panels during the day and power the home when the sun is down. There are each 200 systems within the dataset chosen as the combined training and test data. They were filtered out from the total data pool to account for errors in health estimations and unusual behaviors such as capacity changes due to upgrades or warranty services.

The open circuit voltage (OCV) curve for a LIB which indicates the battery capacity based on voltage level has a different behavior depending on the charge and discharge rate as well as how much health degradation it has. Therefore, it is unreliable to solely rely on BMS data for SOC and SOH. The estimated SOH value for each system is calculated once per month using company proprietary algorithm for the data sourced from them. The time resolution for the timeseries data is 5 seconds.

### 3.1.2 Dataset Delta

To verify the accuracy and robustness of the algorithm, a dataset of lab produced battery aging from a joint Stanford / Massachusetts Institute of Technology research<sup>57</sup>, referred to as dataset *Delta*, is used as another result metric. 124 commercial lithium iron phosphate/graphite cells are cycled under fast charging conditions until a significant decrease in the measured capacity.

The estimated battery health status for each test run is calculated by total discharged energy over charged energy, and the time resolution for the timeseries data is roughly 10 seconds. The quality of data here is drastically different compared to field datasets, as they are in a control lab environment with consistent experiments. The results from the models are expected to be much better in accuracy but serves as a supporting argument for choosing certain models.

## 3.2 Input Feature and Target Output

Because of its relatively small time-resolution, field data is dimensionally too large to be used directly as features for regression. Therefore, data pre-processing is necessary to condense the high-dimensional, sparse field data into small-dimensional, easier to compute data.

On one hand, compressing raw data will lose details. However, the goal of light weight processing requirements exists for this research so that the algorithm can be utilized in a much broader range of application. A lower processing requirement will also enable a much higher throughput of results when managing a fleet of systems, as industrial users often do.

The regression models all take the same input variable and produces the same output variable. Output  $y$  is change in SOH, denoted by  $\Delta Q$ . Table 3.1 below lists all the variables that makes up input  $x$ :

---

<sup>57</sup> cf. Severson et al. (Data-driven prediction of battery cycle life before capacity degradation): 2019

**Table 3.1: Input features for predictive models extracted from field data**

Features	Variable name	Unit
Age	age	Month
Season	season	Unitless (normalized)
Temperature Mean	temp_mean	°C
Temperature 98th quantile	temp_98q	°C
Temperature 2nd quantile	temp_2q	°C
Min Cell Temperature	temp_cell_min	°C
Max Cell Temperature	temp_cell_max	°C
Temperature Spread	temp_spread	°C
Voltage Mean	volt_mean	Volt
Voltage 98th quantile	volt_98q	Volt
Voltage 2nd quantile	volt_2q	Volt
Time spent over Voltage Threshold	volt_over	hour
Current Mean	curr_mean	Ampere
Current 98th quantile	curr_98q	Ampere
Current 2nd quantile	curr_2q	Ampere
Positive Current Mean	curr_mean_chg	Ampere
Negative Current Mean	curr_mean_dsc	Ampere
Current Mean during charge	curr_use_chg	Ampere
Current Mean during discharge	curr_use_dsc	Ampere
Power Mean during charge	power_charge_mean	Watt
Power Mean during discharge	power_discharge_mean	Watt
Depth of Discharge Total	dod	Cycle
Depth of Discharge Rate	dod/h	Cycle/hour
Cell Drift	cell_drift	Volt
Total Energy Use	energy_total	Joules

The input features chosen is based on the known modes and conditions of degradation explained in section 2.1.3 as well as previous research mentioned in section 2.2.2. Age variable is chosen based on the calendric degradation mode. Season variable transforms the time of month into a standardized numerical value according to the cold and hot seasons using the sine function. Depth of discharge variables are chosen based on the cyclical degradation mode. All temperature statistical variables are chosen based on extreme temperature related degradation mode. The same is also chosen for extreme voltage and current related degradation mode. Current and power variables are split into charge and discharge categories to separate the two use cases. The 98<sup>th</sup> and 2<sup>nd</sup> quantile statistics for temperature, voltage, and current are used to extract the time spent at either extreme excluding outliers. For battery packs, additional cell related statistics are used if available to add feature detail.

All features extracted from raw field data are standardized by removing the mean and scaling to unit variance with the equation 3.1:

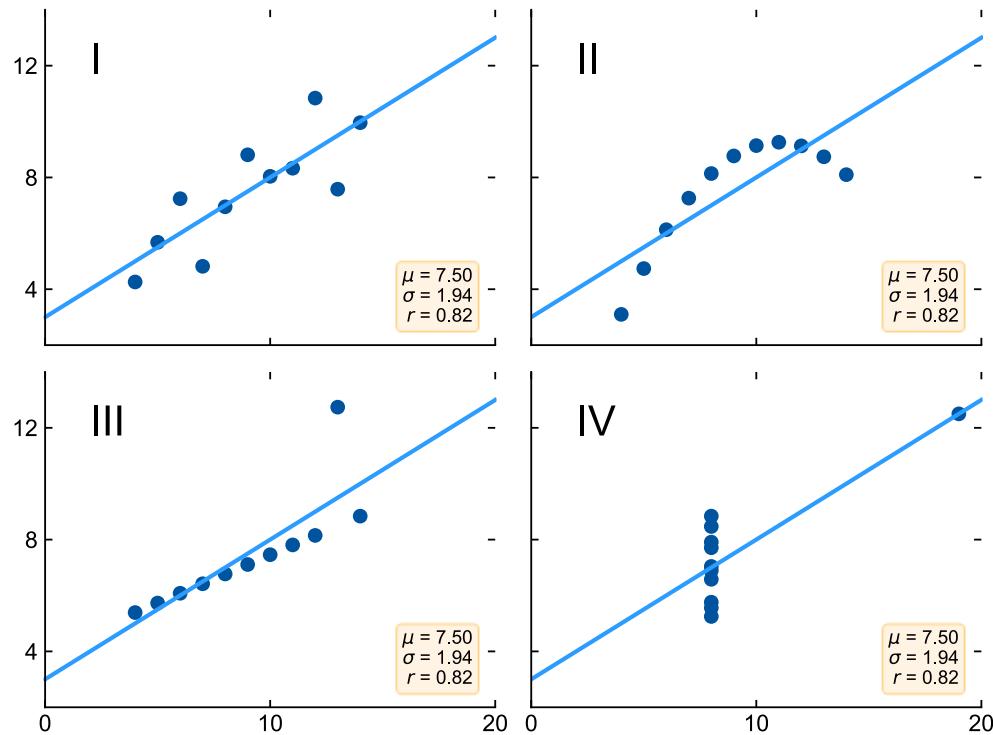
$$z = \frac{x - u}{s} \quad (3.1)$$

where  $u$  is the mean of the training samples and  $s$  is the standard deviation of the training samples. This way, the need to normalize input for certain models because of their sensitivity to feature magnitudes can be eliminated.

Output  $y$ , change in SOH ( $\Delta$ SOH), is also calculated using the values obtained from the partner company's proprietary algorithm. The output is also smoothed with a running average of 3 months to eliminate big swings in SOH caused by noise.

### 3.3 Models Selection

Choosing the optimal regression model is no trivial problem, as numbers can often lie if one looks at it solely. Anscombe's quartet dataset illustrates this problem of only looking at metrics as the decision maker. This dataset has the unique property where all 4 sets of datapoints has identical descriptive statistics such as mean, variance, and the resulting linear regression line, yet are very different in nature.



**Figure 3.1: Anscombe's quartet dataset<sup>58</sup>**

As seen from Figure 3.1 above, the first dataset (top left) has a linear trend with noise where the regression line is appropriate. The second dataset graph (top right) appears clearly to be a parabolic relationship and not linear. The third dataset (bottom left) has a strong linear trend, but the regression line is skewed by a single outlier, which exerts enough influence to lower the correlation coefficient. The fourth dataset (bottom right) is similar to the third dataset where it exemplifies when one high-leverage point is enough to produce a high correlation coefficient, even though the other data points do not indicate any relationship between the variables. The Anscombe's quartet demonstrates the importance of identifying properties of the specific dataset, as well as the influence of outliers.

Another factor when choosing the optimal ML model is the ability to explain to stakeholders how and why certain predictions are made. There is often a tradeoff between accuracy versus interpretability. Classification of “white-box” versus “black-box” models is used to convey a model’s interpretability<sup>59</sup>. White-box models are ones that are easy to explain and interpret such as linear regression and decision trees, whereas black-box models are more mysterious in the relationship between its input and output such as neural networks. Black-box models usually have better accuracy, but the inner workings, such as ANN’s weight for each neuron, are harder to interpret. White-box models has high interpretability but sacrifices capability to

<sup>58</sup> cf. Anscombe, F.J. (Graphs in Statistical Analysis) 1973

<sup>59</sup> cf. Loyola-González, O. (Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View) 2019

model data complexities. In this thesis, the linear models, PCR, PLS, as well as GPR are considered closer to white-box in the spectrum, and SVR and MLPR are considered closer to black-box models. Ultimately, it is important to consider all aspects of the models and datasets to determine the suitable type of regression model for each specific application to prevent undesirable results.

### 3.4 Hyperparameter

Hyperparameters are parameters defining a model's intrinsic property that are not directly learned within models from training data. Because of the wide range of ML models, hyperparameters here includes all model configuration choices. The same type of model with varying hyperparameters can behave drastically different, hence the goal of tuning hyperparameter is to tailor and adapt the model to the specific use-case and dataset. The process of tuning hyperparameters involves researching each mathematical technique and testing to see if the candidates perform as expected. There are a few ways to approach this process.

Exhaustive Grid Search generates all possible combination of hyperparameters from a specified list of values for every parameter. This strategy tests out all combination using brute force, but can miss potentially better parameter values not listed by the user.

Randomized Parameter Optimization implements a randomized search over the parameters, where each setting is sampled from a distribution of possible values. This strategy allows a budget to be chosen independent of each parameter. This strategy is useful to obtain the magnitude of the optimal parameter value before fine tuning it.

Sklearn's *GridSearchCV*, considered an exhaustive grid search approach, is used in this thesis to test combinations of hyperparameter options to obtain the best options. When a set of hyperparameters are found within the specified group, further investigation is conducted according to the insight gained from the first round of testing. The wrong direction of search is known when the results are worse than before. In the end, the hyperparameters with the highest accuracy performance is chosen for comparison of each model type. It is observed that black-box models explored in this thesis contains more hyperparameters to tune than white-box models. Table 3.2 below lists the relevant hyperparameters for each regression models explored in this thesis and their explanation.

**Table 3.2: Model hyperparameter options and each respective representation**

Model	Hyperparameter	Representation
Linear	N/A	N/A
Ridge	Alpha (Regularizer)	Specified float
Lasso	Alpha (Regularizer)	Specified positive float
ElasticNet	Alpha (Regularizer)	Specified positive float
	L1 Ratio	Specified float
Polynomial	Degree	Specified positive integers
Principle Component PCR (PCR)	Number of Components Kept	Specified integer, MLE
Partial Least Squares (PLS)	Number of Components Kept	Specified integer
Support Vector Regression (SVR)	Kernel	Linear, Poly, RBF, RQ, Sigmoid, Pre-computed
	C (Regularization)	Specified positive float
	Epsilon	Specified positive float
Neural Network (MLPR)	Hidden Layer Sizes	Array of shape (Number of neurons each layer)
	Neuron Activation	Identity, Logistic, tanh, Relu
	Batch Size	Auto, Specified Integer
	Learning Rate	Constant, Inverse Scaling, Adaptive
Gaussian Process Regression (GPR)	Kernel	Constant Kernel, Radial Basis Function (RBF), Matern, Radial Quadratic (RQ), Exp-Sine-Squared, Dot-Product

OLS Linear Regression does not have any applicable hyperparameter selection. Ridge and Lasso extensions of the linear regression only has the alpha parameter to tune, representing the L1 and L2 regularization constant for the minimization function respectively. RMSE and R2 results are recorded using the validation set using the default value of 1.0, then a smaller value and larger value for alpha is used to train the model again to see which one yields an improvement. Once the direction for tuning is known, this step is repeated until the optimal value for alpha is found. These steps are generally repeated for all model candidates.

ElasticNet's hyperparameter includes alpha and the ratio for L1 weight, which determines the size of both L1 and L2 regularization weight. The L1 ratio is tuned first and then the alpha value second. An initial value of 0.5 is used for both weights as a starting point.

For polynomial regression, a starting degree of two is chosen to see if there is an improvement over linear regression which is equivalent to a degree of one. Subsequent number of integers are then tested to gain insights about the trend.

For PCR and PLS, a starting number of two is chosen to see if the input features can be projected into a lower dimension for higher accuracies.

For GPR, all available kernel options are tested to see which one achieves the best results. The constant and dot-product kernels are tested in combinations of other kernels. The RBF kernel is chosen as the starting point because of its smoothness and universal adaptability.

For SVR, a linear kernel is chosen as the initial kernel because of the insight found from polynomial regression. A small value of epsilon is chosen to have a larger margin separating hyperplane because of the amount of noise in the data, which would enable more datapoints to incur no penalty associated in the training loss function.

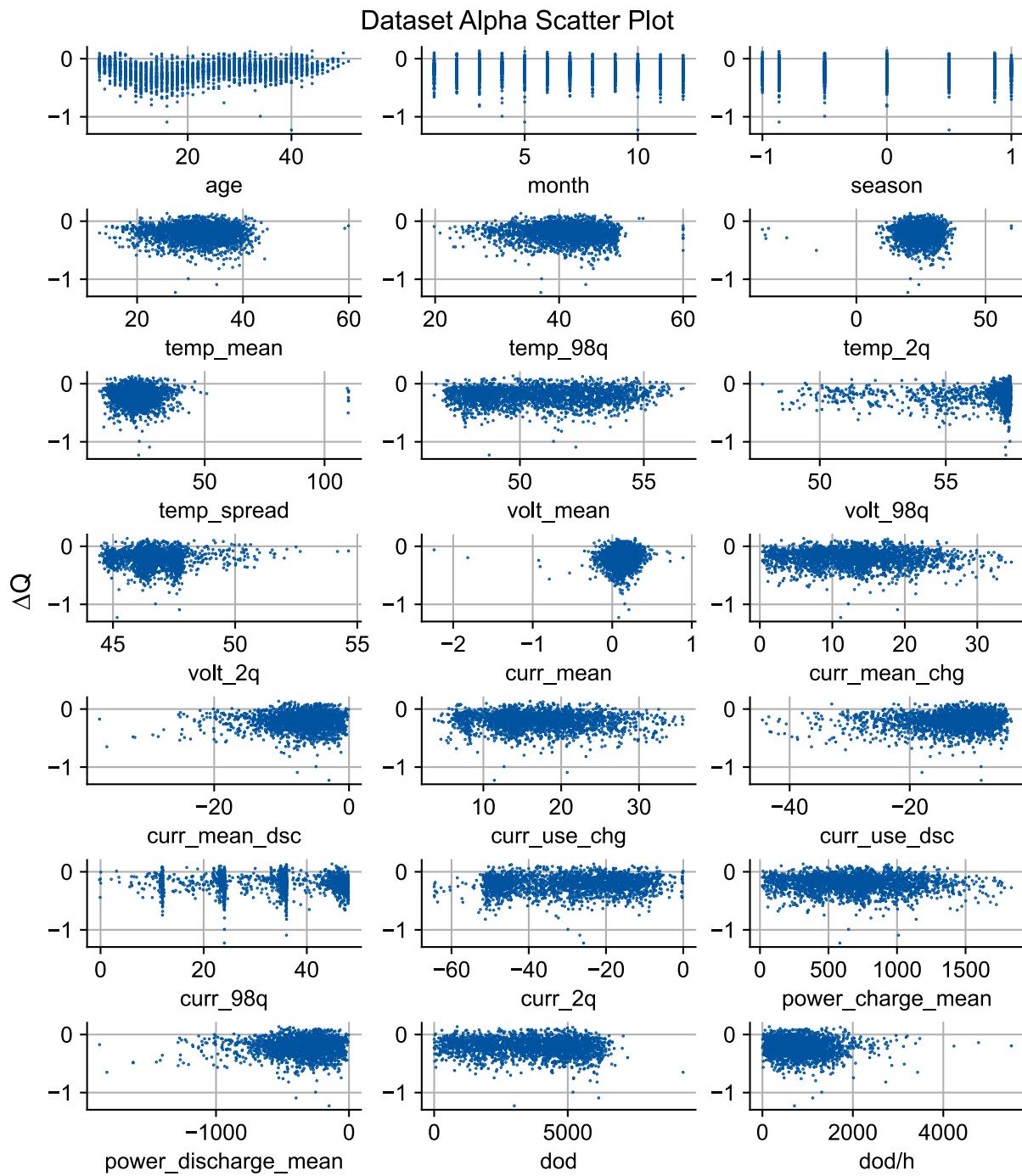
Since there is no right answer to how many layers and neurons are the most suitable, a single layer of 50 neurons is chosen for MLPR to be the starting point. Tanh is chosen as the starting activation function because of its nonlinearity in both positive and negative spectrums.

## 4 Results

The results presented in this chapter are obtained using the datasets and models trained with the input features mentioned in section 3.2. The models utilize the available open-source Python package Scikit-learn (*Sklearn*) which is commonly used in both academia and industry. In addition, other common Python packages such as *Pandas*, *NumPy*, and *Matplotlib* are used in the algorithm and visualizations.

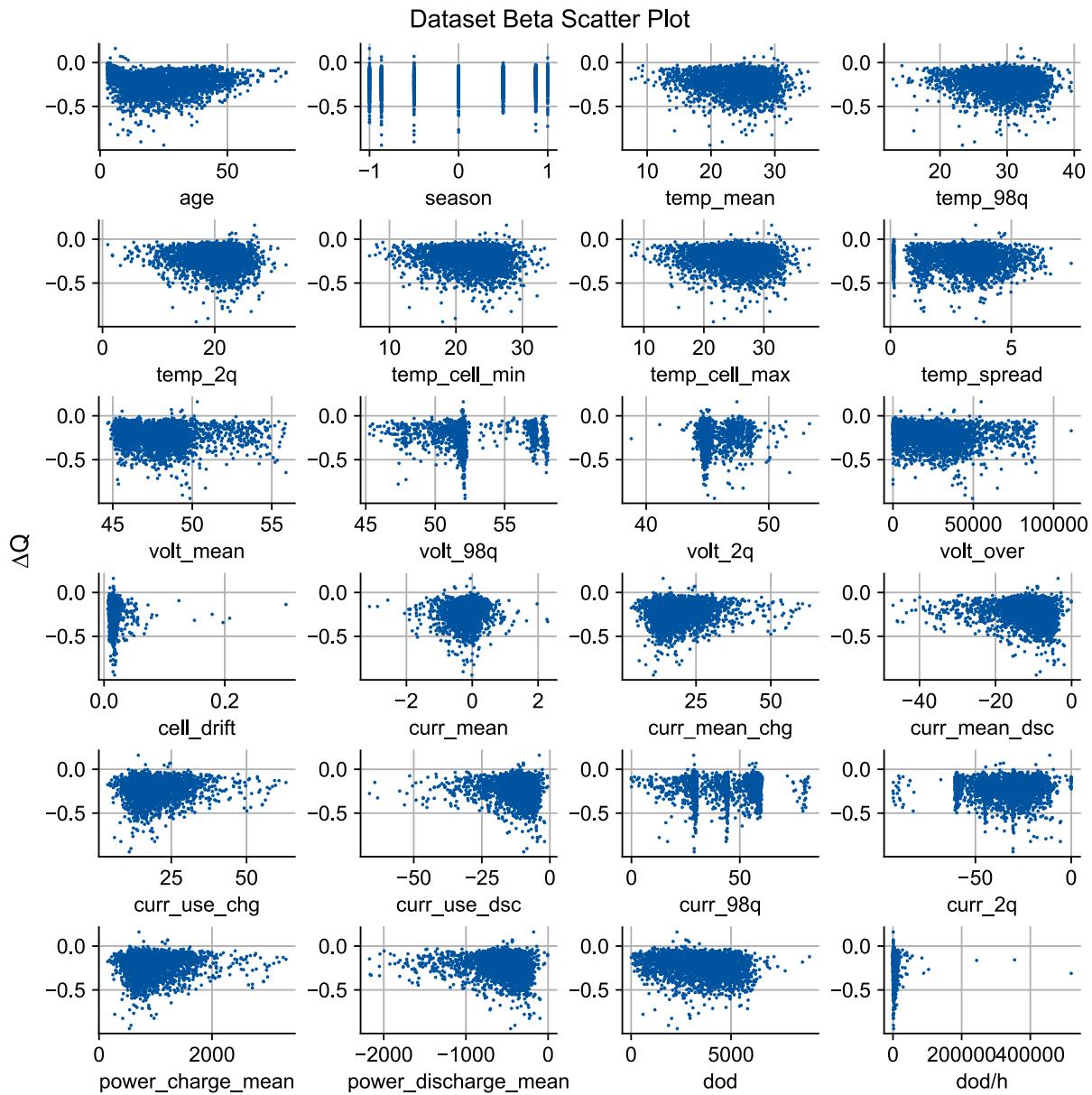
### 4.1 Visualized Correlation

The datasets are visualized to assist in having a general sense of the correlation behavior. Scatter plots of all input features to target output is shown as well as the corresponding coefficient magnitudes for each dataset. The following figures shows the correlation between the input features and the target output,  $\Delta\text{SOH}$ .



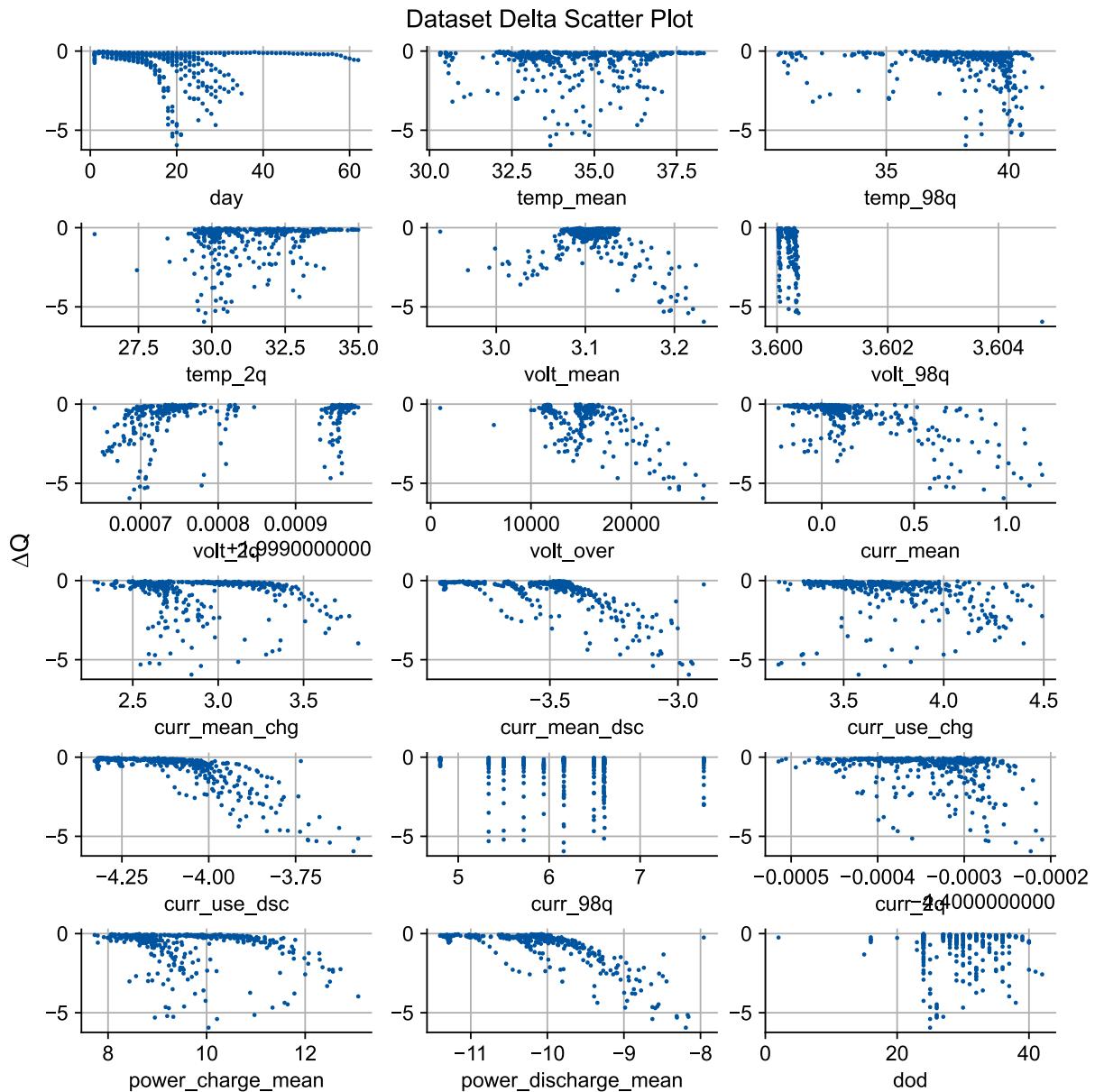
**Figure 4.1: Scatter plots of all input features from dataset Alpha to target output**

Most of the plots in Figure 4.1 exhibits a flat trend with a few exceptions. The age plot indicates a slight increase in  $\Delta SOH$  during first 12 months of age, with the biggest drop in SOH occurring within the first 12-20 months of age, followed by a gradual decrease in  $\Delta SOH$  as age increases. There seem to have no correlation in the time of year to drop in SOH. The 2<sup>nd</sup> quantile temperature and mean current plots exhibits a circular cluster. The 98<sup>th</sup> quantile current plot has four distinct clusters, indicating consistent upper limits among the datapoints. All the rest of the input features plots are clustered in circular or flat elliptical shapes.



**Figure 4.2: Scatter plots of all input features from dataset Beta to target output**

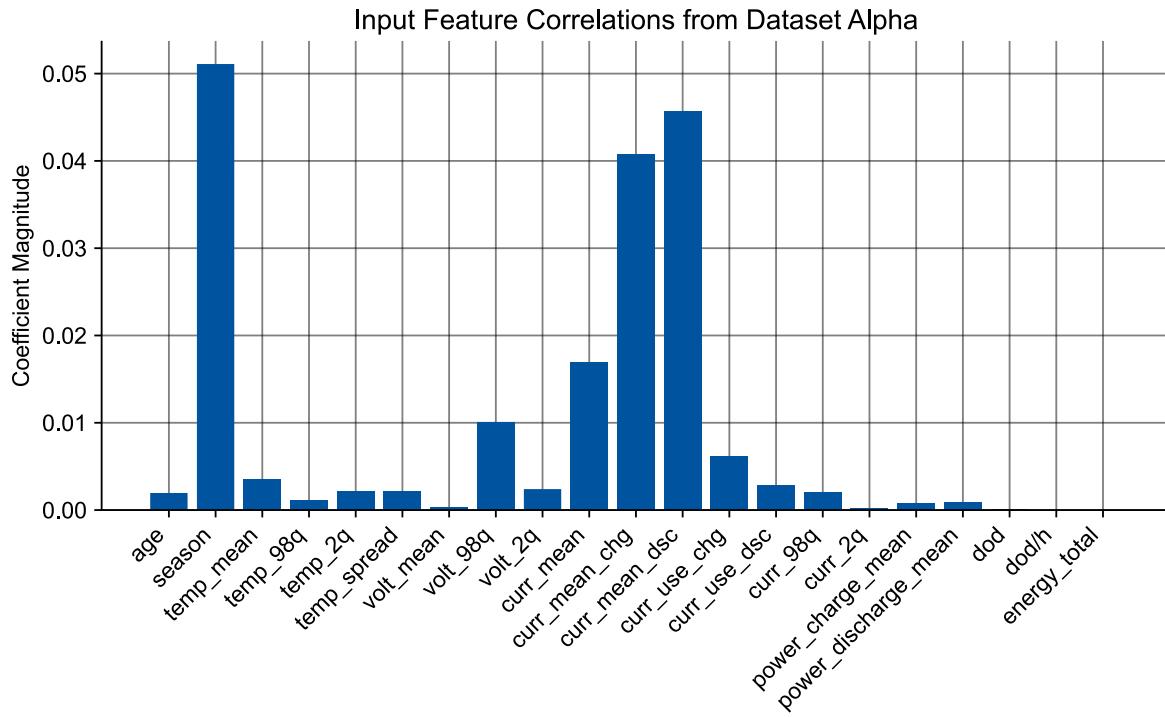
Visually, the scatter plots for dataset Beta seen in Figure 4.2 is similar to ones for dataset Alpha with slight differences. The age plot has a similar behavior with a sharp initial increase in  $\Delta\text{SOH}$  then proceeds to decrease slowly through time. The biggest of the drop in SOH occurs in the first 5-40 months of age which is a wider window of range than dataset Alpha. The rest of the plots also contains mostly flat elliptical shaped clusters. In the charge and discharge current and power plots (labeled curr\_use\_chg, curr\_use\_dsc, power\_charge\_mean, and power discharge\_mean), higher  $\Delta\text{SOH}$  did appear concentrated in lower current and power magnitudes. However, there are just as many sample points with low current magnitudes and low  $\Delta\text{SOH}$ , and that correlation does not agree with known degradation conditions.



**Figure 4.3: Scatter plots of all input features from dataset Delta to target output**

Seen in Figure 4.3, the lab data is a lot cleaner with much less noise existing. There are clear trends in multiple plots in Figure 4.3. High degradation is associated with high duration spent at high voltage state (labeled volt\_over), high mean current (labeled curr\_mean), high charging power (labeled power\_charge\_mean), low discharging current (labeled curr\_mean\_dsc), and low discharging power (labeled power\_discharge\_mean). This is aligned with known degradation conditions.

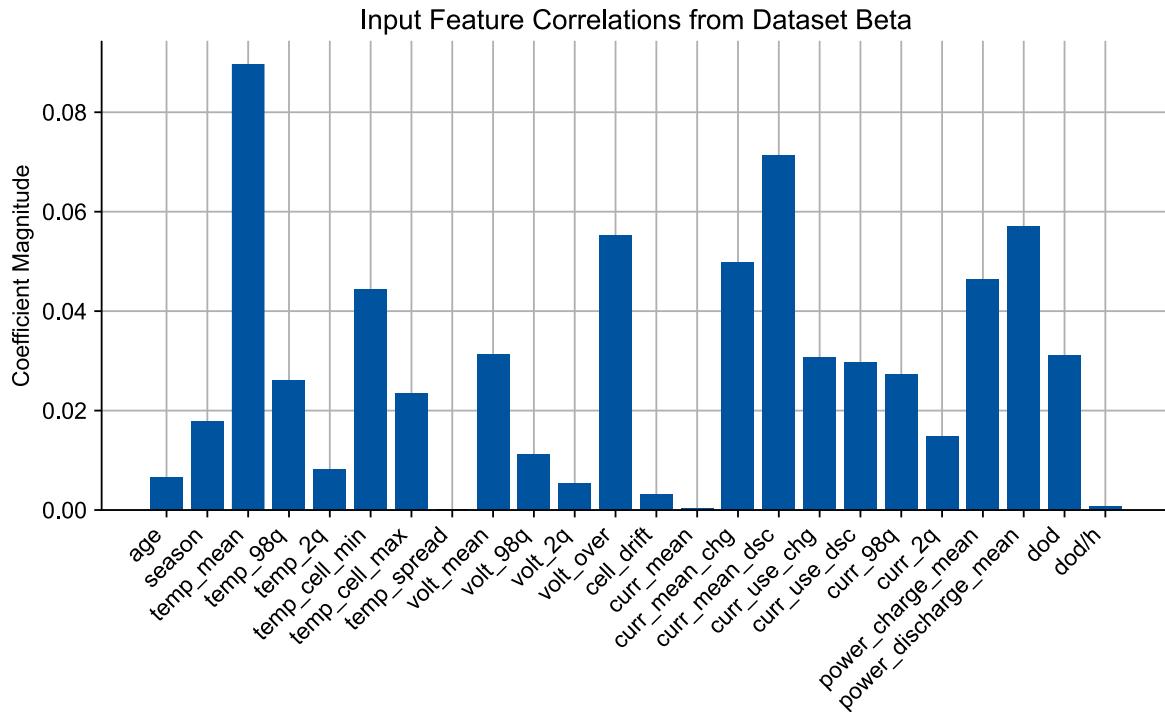
In addition to input scatter plots, another way to gain insight into the correlation between input features and target output is to analyze the simple linear regression's coefficients for each input feature. The larger the coefficient magnitude is, the bigger the correlation is between the corresponding input feature and the output. The lower the coefficient magnitude is, the lower the correlation. Figure 4.4 shows the bar plot of the input coefficient magnitude obtained from the linear regression for dataset Alpha.



**Figure 4.4: Input coefficient magnitudes from dataset Alpha linear regression**

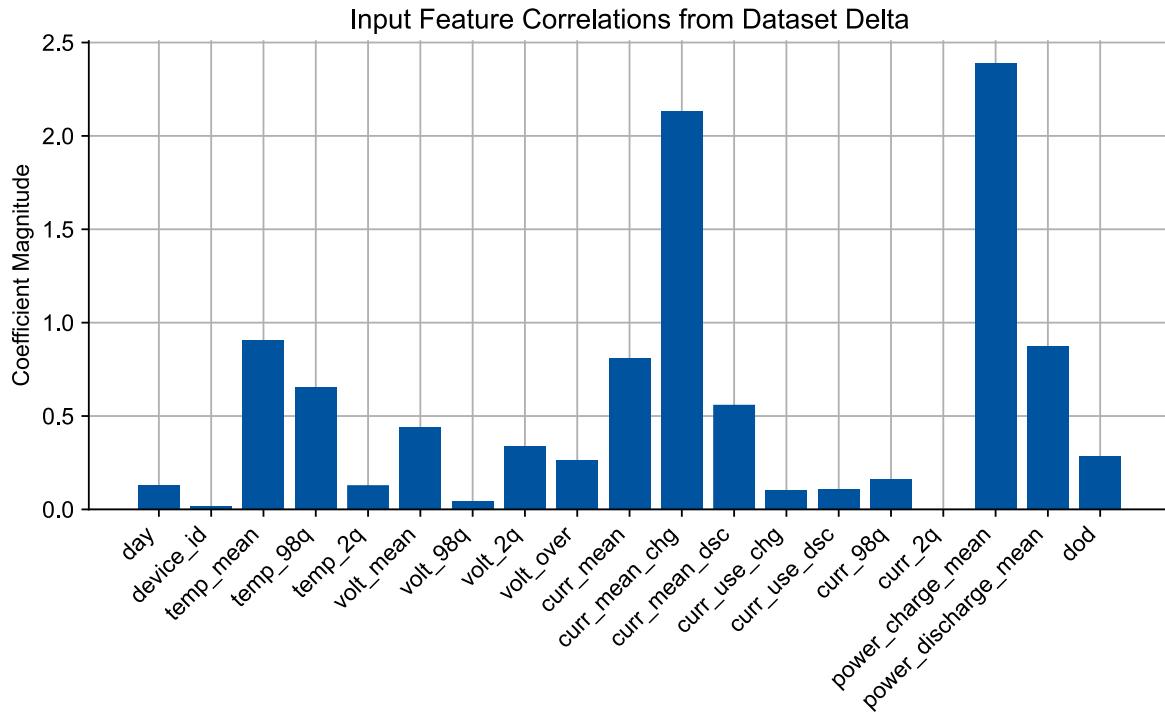
According to Figure 4.4, the input features with the highest correlation are season, mean discharging current, and mean charging current in descending order. The input features with the lowest correlation are mean voltage, 2<sup>nd</sup> quantile current, depth of discharge, and total energy.

Figure 4.5 shows the bar plot of the input coefficient magnitude obtained from the linear regression for dataset Beta.



**Figure 4.5: Input coefficient magnitudes from dataset Beta linear regression**

According to Figure 4.5, the input features with the highest correlation are mean temperature, time spent at high voltage, mean discharging current, and mean discharging power in descending order. The input features with the lowest correlation are temperature spread, mean current, and depth of discharge rate. Figure 4.6 shows the bar plot of the input coefficient magnitude obtained from the linear regression for dataset Beta.



**Figure 4.6: Input coefficient magnitudes from dataset Delta linear regression**

According to Figure 4.6, the input features with the highest correlation are mean charging current, mean charging power, and mean temperature in descending order. The input features with the lowest correlation are mean voltage, 2<sup>nd</sup> quantile current, depth of discharge, and total.

In summary, the highest coefficients across the datasets are unique, but the features with the highest correlation are all related to temperature, current, power, and duration spent over voltage threshold. This confirms the prior knowledge about the conditions that leads to higher battery degradation.

## 4.2 Hyperparameter Tuning

Tuning each model's hyperparameter takes both prior knowledge about the dataset as well as the intrinsic meaning of each hyperparameter. The following tables shows the procedure of choosing the optimal hyperparameter with intermediate accuracy results for dataset Delta.

**Table 4.1: Hyperparameter tuning steps and intermediate results for Ridge**

Parameter	Choice	RMSE	R2 score
Alpha	1	0.3197	0.9149
Alpha	0.5	0.3166	0.9166
Alpha	0	0.3062	0.922

There is no applicable hyperparameter selection for OLS linear regression. Table 4.1 above shows the accuracy result from various hyperparameter of Ridge regression. For all extension of linear regression including Ridge, Lasso, and ElasticNet, the alpha regularization parameter specifying the constant for the weight in the minimization function has low sensitivity to their accuracy performance, meaning drastically different values yields similar results. The difference between the alpha choices is less than 0.02 in RMSE and R2 score. According to Table 4.1, any positive nonzero regularization parameter caused the models to have worse accuracy performance, and any negative regularization parameter causes results to diverge. The optimal choice of zero for alpha essentially turns it into the ordinary least squares linear regression, evident by their identical RMSE. Tables for Lasso and ElasticNet tuning steps are similar to Ridge and are located in appendix A.1.

**Table 4.2: Hyperparameter tuning steps and intermediate results for Polynomial**

Parameter	Choice	RMSE	R2 score
Degree	2	0.5401	0.7573
Degree	3	0.4962	0.7951
Degree	4	0.4769	0.8108
Degree	5	0.6668	0.6301

Table 4.2 above shows the accuracy result from various hyperparameter of Polynomial regression. According to Table 4.2 for polynomial regression, the accuracy increases as number of degrees increases until four, where accuracy starts to decrease. The function degree of four yields the best accuracy performance for dataset Delta, with degree of three being the next best parameter. There is a big drop in performance going from the optimal degree of 4 to 5, and even more for the number of degrees above.

**Table 4.3: Hyperparameter tuning steps and intermediate results for PCR**

Parameter	Choice	RMSE	R2 score
Number of components	2	0.7625	0.5162
Number of components	5	0.6322	0.6675
Number of components	12	0.3864	0.8758
Number of components	19	0.3062	0.9220

Table 4.3 above shows the accuracy result from various hyperparameter of PCR. Both PCR and PLS regressions attempt to reduce dimensionality by specifying the number of components to keep in the function. According to Table 4.3, the best accuracy performance for CR is achieved with the full dimension of the input features, rendering them no improvement to linear regression. Any number of components to keep less than the full dimension yields worse accuracy performance than linear regression, and the accuracy difference between lowest and highest number of components is substantial. Table for PLS tuning steps is identical to PCR and is located in appendix A.1.

**Table 4.4: Hyperparameter tuning steps and intermediate results for GPR**

Parameter	Choice	RMSE	R2 score
Kernel	Radial Basis Function (RBF)	0.3178	0.916
Kernel	Matern	0.224	0.9583
Kernel	Rational Quadratic (RQ)	0.2316	0.9554

Table 4.4 above shows the accuracy result from various hyperparameter of GPR. With the only hyperparameter being the kernel, GPR is fairly straight-forward in tuning. All individual kernels and combination of one another are tested. According to Table 4.4, the Matern kernel performs the best along with the RQ kernel in close second with the help of a regularization parameter to ensure a positive definite matrix.

**Table 4.5: Hyperparameter tuning steps and intermediate results for SVR**

Parameter	Choice	RMSE	R2 score
Kernel	RBF	0.4422	0.8373
Epsilon	0.5		
C	1		
Kernel	Linear	0.3432	0.902
Epsilon	0.5		
C	1		
Kernel	Linear	0.3348	0.9067
Epsilon	0.2		
C	8		
Kernel	RBF	0.2735	0.9377
Epsilon	0.2		
C	5		
Kernel	RBF	0.2167	0.9609
Epsilon	0.03		
C	8		

Table 4.5 above shows the accuracy result from various hyperparameter of SVR. SVR's hyperparameters include the kernel type, regularization parameter C, and the margin epsilon. The RBF kernel was able to outperform the linear kernel. Results for epsilon and C parameter agrees with the intuition of noisy data, with a small epsilon and large C being the optimal hyperparameter values. However, the accuracy results will start to decrease once the epsilon becomes too small and C becomes too big.

**Table 4.6: Hyperparameter tuning steps and intermediate results for MLP**

Parameter	Choice	RMSE	R2 score
Hidden Layer Sizes	(50, 50)	0.2694	0.9396
Activation Function	tanh		
Alpha	1		
Learning Rate	Constant		
Hidden Layer Sizes	(50, 50)	0.2418	0.9514
Activation Function	relu		
Alpha	1		
Learning Rate	Constant		
Hidden Layer Sizes	(50, 50)	0.226	0.9575
Activation Function	relu		
Alpha	0.1		
Learning Rate	Constant		
Hidden Layer Sizes	(50, 50, 50, 50)	0.1873	0.9708
Activation Function	relu		
Alpha	0.1		
Learning Rate	Constant		
Hidden Layer Sizes	(200, 200, 100, 100)	0.2007	0.9665
Activation Function	relu		
Alpha	0.07		
Learning Rate	Constant		

Table 4.6 above shows the accuracy result from various hyperparameter of MLP. MLP possesses far more hyperparameters to tune than all other model candidates. Hyperparameters include the hidden layer sizes of the neural network, neuron activation function, training batch size, and learning rate. The hidden layer sizes specify how many neurons and layers the network has besides the input and output layer. The number of layers and the number of neurons each layer is far from trivial to determine because there is an infinite number of options. The optimal parameter that yields the highest accuracy is four layers with two 200-neuron layers followed by two 100-neuron layers. The neuron activation function is used for each neuron during forward propagation, with the relu function being the optimal.

There is slight variation in optimal hyperparameter choices between the three datasets, but the difference is minimal. Table 4.7 summarizes the relevant hyperparameters for each regression model discussed above and the optimal choice for this dataset Alpha.

**Table 4.7: Model hyperparameter options and optimal result choice for dataset Alpha**

Model	Hyperparameter	Result
Linear	N/A	N/A
Ridge	Alpha (Regularizer)	0
Lasso	Alpha (Regularizer)	0
ElasticNet	Alpha (Regularizer)	0
	L1 Ratio	0.5
Polynomial	Degree	2
Principle Component PCR (PCR)	Number of Components Kept	22 (full dimension)
Partial Least Squares (PLS)	Number of Components Kept	22 (full dimension)
Support Vector Regression (SVR)	Kernel	RBF
	C (Regularization)	4.0
	Epsilon	0.01
Neural Network (MLPR)	Hidden Layer Sizes	(200, 200, 100, 100)
	Neuron Activation	Relu
	Batch Size	Auto
	Learning Rate	Constant
Gaussian Process Regression (GPR)	Kernel	RQ

### 4.3 Performance Metrics

This section shows the performance metrics of all ten model candidates by which they are evaluated. The root mean square error (RMSE) and coefficient of determinant (R2 score) indicate accuracy performance, and the training runtime indicates complexity of the models. RMSE is measured in  $\Delta$ SOH [%], R2 score is unitless, and runtime is measured in seconds [s].

The following tables list each performance metrics for each model candidates across all three datasets, their ranking within each dataset, and their average ranking. Tables grouped by dataset instead of models are listed in appendix A.2 for reference. The idea model would have the lowest RMSE, biggest R2 score, and the lowest runtime.

**Table 4.8: Comparison of model candidates in RMSE across all datasets**

Model	RMSE (Alpha) [%SOH]	Rank (Al- pha)	RMSE (Beta) [%SOH]	Rank (Beta)	RMSE (Delta) [%SOH]	Rank (Delta)	Average Rank
Linear	0.1382	7	0.1189	5	0.3062	5	5.7
Ridge	0.1382	7	0.1212	7	0.3166	9	7.7
Lasso	0.1383	9	0.1233	9	0.3090	7	8.3
ElasticNet	0.1383	9	0.1233	9	0.3090	7	8.3
Polynomial	0.1337	4	0.1119	4	0.5401	10	6.0
GPR	0.1107	2	0.0968	1	0.2316	3	2.0
SVR	0.1189	3	0.1038	3	0.2167	2	2.7
PCR	0.1382	7	0.1231	8	0.3062	5	6.3
PLS	0.1381	5	0.1193	6	0.3062	5	5.3
MLPR	0.1058	1	0.1005	2	0.1897	1	1.3

Table 4.8 above shows the RMSE each model achieved using the optimal hyperparameters for each dataset. On average, MLPR achieved the best accuracy with the lowest error, followed by GPR and then SVR. All three have relatively similar results within a few tenths of %SOH. Ridge, Lasso, ElasticNet, Polynomial, and PCR all performed worse than linear regression with higher errors.

**Table 4.9: Comparison of model candidates in R2 score across all datasets**

Model	R2 (Al-pha)	Rank (Al-pha)	R2 (Beta)	Rank (Beta)	R2 (Delta)	Rank (Delta)	Average Rank
Linear	0.0540	8	0.0705	5	0.9220	4	5.7
Ridge	0.0540	7	0.0343	7	0.9166	9	7.7
Lasso	0.0527	9	0.0000	9	0.9205	7	8.3
ElasticNet	0.0527	9	0.0000	9	0.9205	7	8.3
Polynomial	0.1142	4	0.1765	4	0.7573	10	6.0
GPR	0.3924	2	0.3840	1	0.9554	3	2.0
SVR	0.2989	3	0.2905	3	0.9609	2	2.7
PCR	0.0541	6	0.0027	8	0.9220	5	6.3
PLS	0.0550	5	0.0637	6	0.9220	5	5.3
MLPR	0.4452	1	0.3350	2	0.9701	1	1.3

Table 4.9 above shows the R2 score each model achieved using the optimal hyperparameters for each dataset. The average R2 ranking for each model is identical to the average RMSE rankings. As such, MLPR, GPR, and SVR also ranked top three in accuracy. R2 scores for field data is noticeably lower than ones for lab data.

**Table 4.10: Comparison of model candidates in runtime across all datasets**

Model	Runtime (Alpha) [s]	Rank (Al- pha)	Runtime (Beta) [s]	Rank (Beta)	Runtime (Delta) [s]	Rank (Delta)	Average Rank
Linear	0.0269	3	0.0643	6	0.0040	4	4.3
Ridge	0.0018	1	0.0062	2	0.0008	1	1.3
Lasso	0.0957	7	0.0069	3	0.0054	7	5.7
ElasticNet	0.0660	5	0.0016	1	0.0049	5	3.7
Polynomial	0.0660	5	0.6408	8	0.0049	5	6.0
GPR	263.85	10	350.20	10	0.2306	9	9.7
SVR	4.0557	8	0.0539	5	0.0105	8	7.0
PCR	0.0355	4	0.0976	7	0.0023	2	4.3
PLS	0.0104	2	0.0192	4	0.0025	3	3.0
MLPR	23.541	9	30.640	9	3.1652	10	9.3

Table 4.10 above shows the runtime each model achieved using the optimal hyperparameters for each dataset. GPR and MLPR consistently obtained the longest and second longest runtimes respectively. Ridge regression consistently obtained the shortest runtime. SVR's longer runtime for dataset Alpha can be considered an outlier considering the much shorter runtimes for the other datasets. The rest of the models including Linear, Lasso, ElasticNet, Polynomial, PCR, and PLS all achieved runtimes that are within tenths of hundredths of a second from each other. This is well within the margin of the computer's timing precision.

For the linear models, it is observed that the specialized regularization regressions (Ridge, Lasso, and ElasticNet) do not outperform the ordinary least squares regression in accuracy. They did have a slightly lower runtime, but it is negligible and within margin of error given the magnitude in the hundredths of a second. PCR and PLS regression, in similar manner, both performed worse than the linear regression benchmark while having similar runtimes. For this reason, these extensions to the linear regression models are omitted in the subsequent sections during comparison. Polynomial regression yields mix results ranging from slightly better accuracy in field data to much worse in lab data while having similar or better runtimes as linear regression.

## 4.4 Cross Validation

Cross Validation (CV), as explained in section 2.4, indicates how well each model generalizes across the dataset. A model that gives good results in one individual fold of cross validation but not in others would not be an ideal model even if the result from that specific fold outperforms all other models. The two metrics measuring robustness are the mean R2 score over all CV folds (CV score) and the standard deviation of the R2 scores across all folds (STD), both unitless. An ideal model should have high CV score and low standard deviation. High CV score indicates robustness and consistency across all ranges of intended data with lower chance of overfitting. Low standard deviation indicates low variance across all folds. *Sklearn's cross\_valide* function used to perform k-fold cross validation with k = 4. The following tables show the CV score and standard deviation of each candidate models across all three datasets. Additional CV tables grouped by dataset instead of models are attached in appendix A.3 for reference.

**Table 4.11: Mean cross validation R2 scores of model candidates**

Model	CV (AI-pha)	Rank (Alpha)	CV (Beta)	Rank (Beta)	CV (Delta)	Rank (Delta)	Average Rank
Linear	0.0644	4	0.0730	4	0.6726	3	3.7
Ridge	0.0644	3	0.0728	6	0.8922	1	3.3
Lasso	0.0639	6	0.0730	2	0.6464	6	4.7
ElasticNet	0.0639	6	0.0730	2	0.6464	6	4.7
Polynomial	0.1290	1	0.0873	1	0.1646	8	3.3
GPR	0.0644	4	0.0730	4	0.7882	2	3.3
SVR	-0.0061	9	0.0444	9	-0.1285	9	9.0
PCR	0.0637	8	0.0724	7	0.6726	5	6.7
PLS	0.0644	2	0.0716	8	0.6726	4	4.7
MLPR	-79319	10	-59489	10	-33225014	10	10.0

On average, GPR, Ridge, and Polynomial regression achieved the highest mean CV R2 scores. Polynomial regression ranks first in the two field datasets but not as well in the lab dataset. MLPR performed the worst with the lowest mean CV R2 scores, with SVR at second to last place. In addition, Lasso, ElasticNet, PCR, and PLS all performed worse than linear regression with lower average CV R2 scores.

**Table 4.12: Standard deviation of cross validation R2 scores of model candidates**

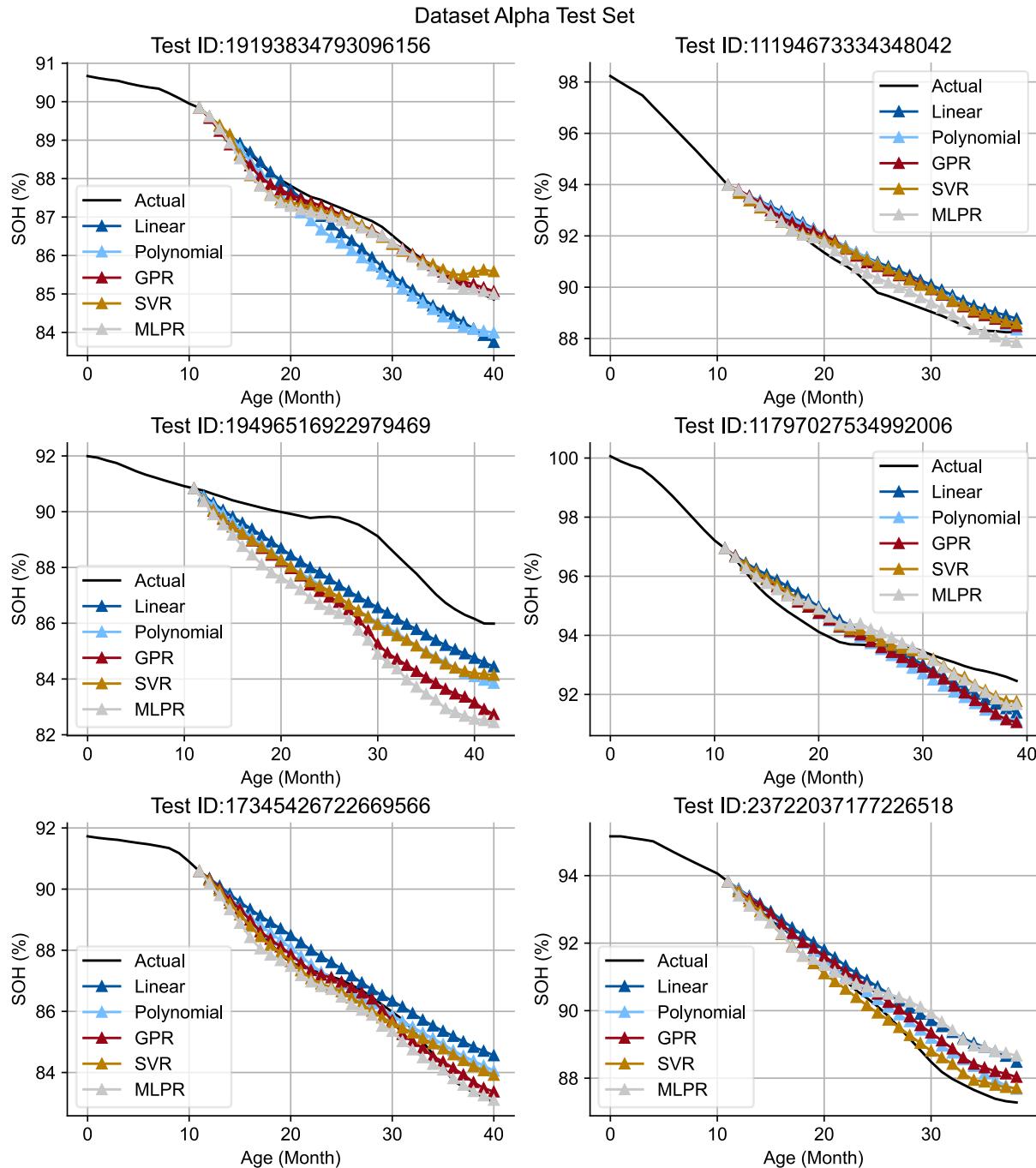
Model	STD (Al-pha)	Rank (Alpha)	STD (Beta)	Rank (Beta)	STD (Delta)	Rank (Delta)	Average Rank
Linear	0.0135	4	0.0043	1	0.4450	4	3.0
Ridge	0.0135	4	0.0044	3	0.0473	3	3.3
Lasso	0.0144	7	0.0045	4	0.5086	7	6.0
ElasticNet	0.0144	7	0.0045	4	0.5086	7	6.0
Polynomial	0.0394	9	0.1689	9	0.8184	9	9.0
GPR	0.0135	4	0.0043	1	0.0425	2	2.3
SVR	0.0066	1	0.0063	8	0.0321	1	3.3
PCR	0.0137	6	0.0046	6	0.4450	6	6.0
PLS	0.0133	2	0.0056	7	0.4450	4	4.3
MLPR	82609	10	59472	10	22824499	10	10.0

On average, GPR has the least variance across all folds, and it is also the only model that outperforms linear regression at second place. SVR has the lowest variance in two out of three datasets, but the result from dataset Beta skewed its average ranking. Ridge regression follows closely behind linear regression with minimal differences. MLPR ranks last again with the highest variance.

Overall, SVR, GPR, and Ridge regression performs the best in generalizing the data. MLPR consistently ranks last in CV score and STD with more than five magnitudes higher in CV scores and standard deviation. The linear models generally performed well, and GPR also ranks in the top four in all datasets. All other models' cross validation results are within small margins of difference from linear regression in the order of two decimal points.

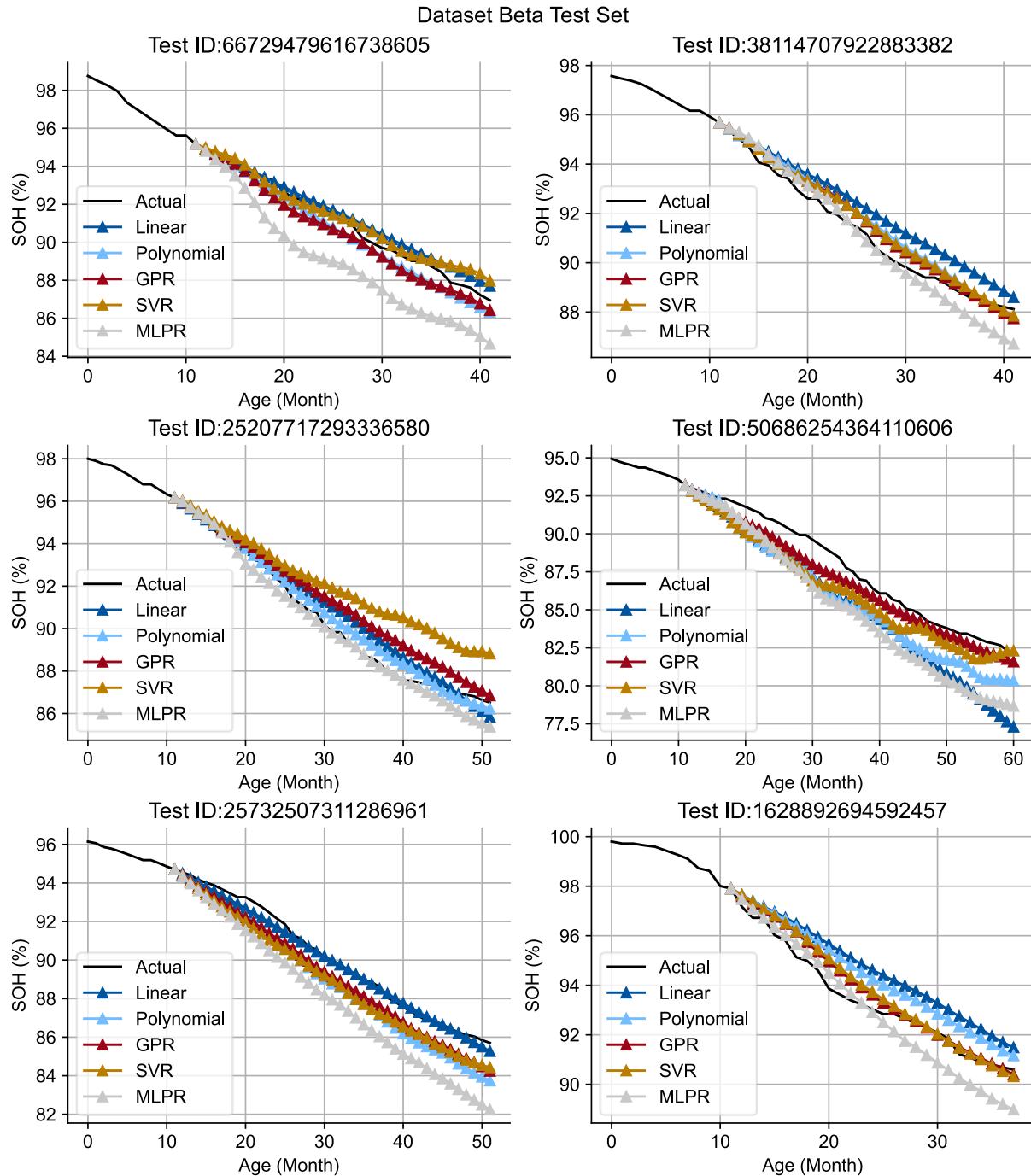
## 4.5 Testing

After training the forecasting models and verifying their performance metrics and properties, they are tested on unseen data to simulate their performance in a real-world deployment scenario. For each dataset, the period starting from 12 time-units onwards for each system's  $\Delta$ SOH are assumed to be unknown. Each time-unit's aggregated statistics are used as input to predict the change in SOH ( $\Delta$ SOH) for the next month. All predicted  $\Delta$ SOH is then appended together to create the predicted SOH through time. The results for each model are plotted in the figures below, and their respective error metrics are calculated and shown in the tables that follow.



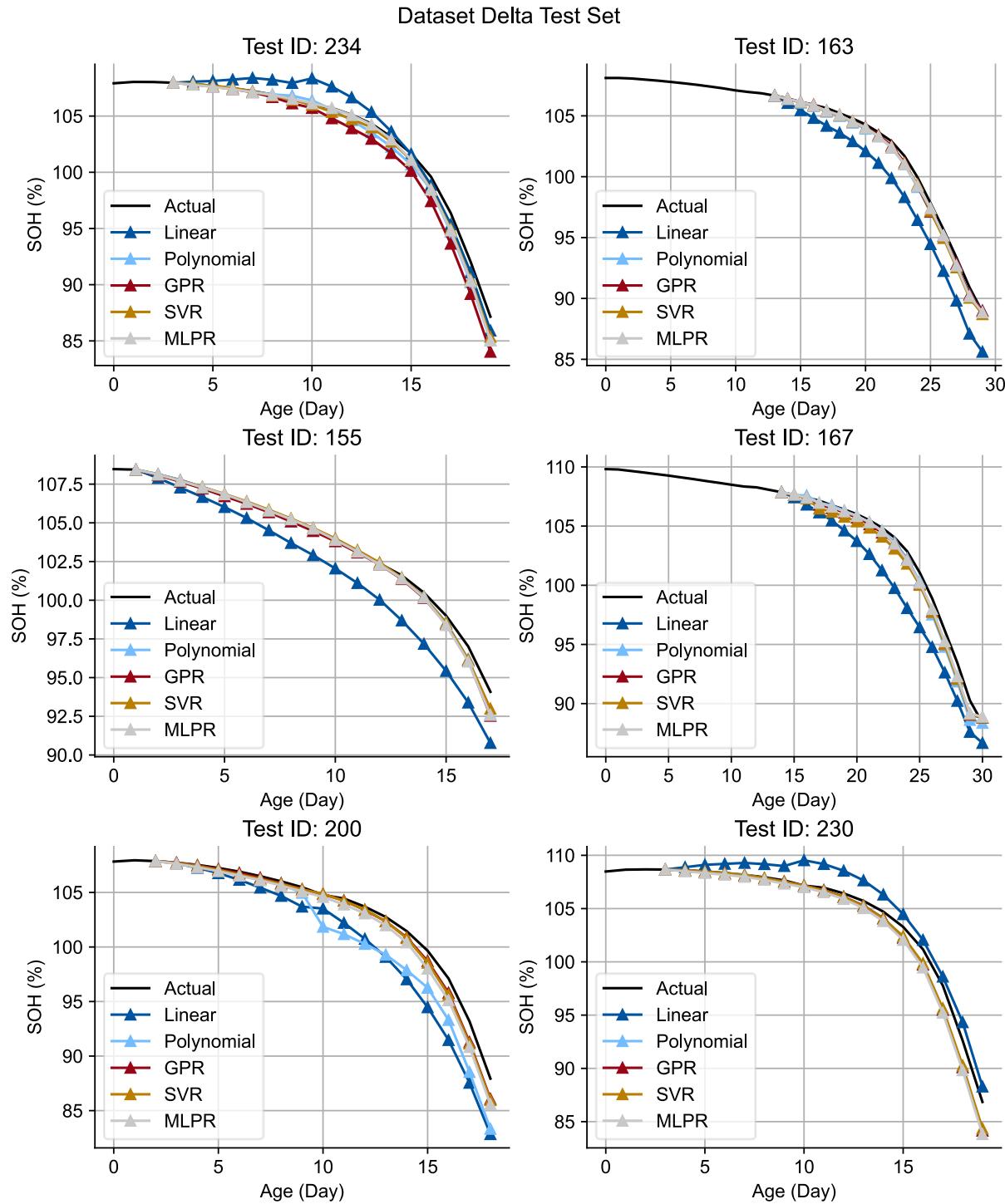
**Figure 4.7: Model prediction comparison from dataset Alpha test set**

Figure 4.7 showcases the model candidates' prediction of six systems from the test set. Visually, it is apparent that nonlinear models outperform the linear models. For all the test set plots for Alpha, linear regression predicted a mostly constant health degradation. The polynomial regression was able to capture some nonlinearity, seen in the bottom two plots of where the predicted curves follow the true value better than linear regression. SVR, GPR, and MLPR's predicted curve is much more accurate, and can reflect the subtle rate of changes in  $\Delta\text{SOH}$ . In the top left plot, all three models follow the true value precisely. In the middle left plot, the predicted results for GPR and MLPR both have a sharper drop in SOH around the 27-month mark, correlating to the similar drop in rate of  $\Delta\text{SOH}$  in reality. In the other plots, all three models performed in similar manner.



**Figure 4.8: Model prediction comparison from dataset Beta test set**

Seen in Figure 4.8, results from dataset Beta testing looks similar to ones from dataset Alpha. The gaps between predictions and actual values are smaller, which confirms the fact that models here produce a lower RMSE overall than Alpha. All models besides linear regression are able to capture the nonlinearity in each system through time with varying levels of accuracy. Linear regression and MLPR are the less accurate than the rest. The differences in final SOH for MLPR are noticeably large in the top left, bottom left, and bottom right plots. Although linear regression performed the worst in the middle right plot, it performed acceptably on the other plots.



**Figure 4.9: Model prediction comparison from dataset Delta test set**

According to Figure 4.9, the models trained on the lab data looks significantly more accurate than ones trained with field data despite the higher RMSE. All model candidates predicted curves that resembles the actual SOH values with varying levels of accuracies. Linear regression appears to be the least accurate visually. Polynomial regression produced fairly accurate results considering it ranks last in RMSE and R2 scores. In all the plots, GPR, SVR, and MLPR predicted extremely similar outputs that are within small margins of each other. All three follows closely to the actual SOH values in almost the same manner. The only exception is the top left plot where GPR performs slightly worse than SVR and MLPR.

The error metrics for all prediction of systems in the test set are recorded in the following tables. They are divided into RMSE for  $\Delta\text{SOH}$  prediction each time-unit, RMSE for combined SOH prediction, and the average final difference in SOH between prediction and actual value.

**Table 4.13: Comparison of error in model output across all test sets**

Model	Output RMSE (Alpha) [%SOH]	Rank (Alpha)	Output RMSE (Beta) [%SOH]	Rank (Beta)	Output RMSE (Delta) [%SOH]	Rank (Delta)	Average Rank
Linear	0.1064	1	0.1056	2	0.3495	5	2.7
Polynomial	0.1155	4	0.1122	3	0.1528	4	3.7
GPR	0.1110	2	0.1052	1	0.0631	1	1.3
SVR	0.1138	3	0.1222	4	0.0709	2	3.0
MLPR	0.1272	5	0.1287	5	0.0727	3	4.3

Table 4.13 shows the RMSE for each model's  $\Delta\text{SOH}$  prediction. Linear regression performs well for unseen field data but performs the worst in unseen lab data. GPR performed the best out of all datasets. MLPR performs poorly in unseen field data but performs well for unseen lab data. SVR performs well consistently across all unseen data. Overall, polynomial ranks second to last place.

**Table 4.14: Comparison of error in combined SOH across all test sets**

Model	Combined RMSE (Alpha) [%SOH]	Rank (Alpha)	Combined RMSE (Beta) [%SOH]	Rank (Beta)	Combined RMSE (Delta) [%SOH]	Rank (Delta)	Average Rank
Linear	0.7150	3	0.8020	2	1.2862	5	3.3
Polyno-mial	0.7652	4	0.8589	3	0.6397	2	3.0
GPR	0.6949	2	0.7704	1	0.8548	4	2.3
SVR	0.6809	1	0.9599	4	0.6100	1	2.0
MLPR	0.7658	5	1.0314	5	0.7401	3	4.3

Table 4.14 shows the RMSE for each model's combined SOH prediction. On average, SVR performs the best when analyzing the total output instead of individual monthly predictions. GPR follows closely in second place, performing well in field data. MLPR again ranks last in unseen field data and third in unseen lab data. Overall, SVR and GPR have best performances, polynomial has acceptable performance, and MLPR was not able to outperform linear regression baseline.

**Table 4.15: Comparison of error in final SOH difference across all test sets**

Model	Final $\Delta$ SOH (Alpha) [%SOH]	Rank (Alpha)	Final $\Delta$ SOH (Beta) [%SOH]	Rank (Beta)	Final $\Delta$ SOH (Delta) [%SOH]	Rank (Delta)	Average Rank
Linear	1.0448	3	1.1630	2	1.6271	2	2.3
Polyno-mial	1.0764	5	1.2241	3	1.6700	3	3.7
GPR	1.0475	4	1.0506	1	2.1105	5	3.3
SVR	0.9032	1	1.3837	4	1.5480	1	2.0
MLPR	0.9419	2	1.6487	5	1.8577	4	3.7

Table 4.15 shows the average final difference in SOH between model prediction and actual value. In this test metric, the relative difference across all the models is lower than the other test metrics. SVR is the only model that was able to outperform the linear regression baseline. GPR ranks third on average after linear regression, and MLPR and Polynomial ranks last overall.

**Table 4.16: Average RMSE of model candidates for all test set**

Model	Field Average RMSE [%SOH]	Field Rank	Improvement over Linear [%]	Lab Average RMSE [%SOH]	Lab Rank	Improvement over Linear [%]
Linear	0.4323	2	0.00	0.8178	5	0.00
Polynomial	0.4630	3	-7.10	0.3963	2	51.54
GPR	0.4204	1	2.75	0.4590	4	43.88
SVR	0.4692	4	-8.55	0.3404	1	58.37
MLPR	0.5133	5	-18.74	0.4064	3	50.31

Table 4.16 summarizes the results for the model candidates across all test sets separated by field and lab data, taking account of both  $\Delta$ SOH and combined SOH predictions. Holistically, SVR performed the best across all datasets. However, when considering only field data (Alpha and Beta), GPR performed the best and is the only model that outperform linear regression baseline with less than 3 % improvements. This result does not reflect the true performance because there are a few outliers that skewed the average of SVR performance, seen in Figure 4.8. When considering only lab data, all model candidates are able to outperform linear regression baseline with over 40 % improvements.

## 5 Discussion

In this chapter, the results of the thesis work are discussed. In the first section, the performance and scoring criteria of each candidate model including accuracy, scaling, and adaptability is discussed. In the second section, the challenges encountered throughout the thesis work are presented and discussed.

### 5.1 Performance

The nature of noise existing in all field data explored in this thesis is apparent when looking at the correlation plots in section 4.1. Figure 4.1 and Figure 4.2 representing the scatter plots for each input feature to the target output (change in SOH) demonstrates the high magnitude of noise and the lack of correlation when examining input features individually. This is supported by the fact that most input scatter plots are elliptical or circular clusters. The correlation for field data is not strong visually to give confidence in the input features chosen. However, simply looking at the data two-dimensionally does not allow the full representation to be observed. When the input is projected to the full dimensional space, there might exist a much more obvious correlation that is impossible to visualize. Unfortunately, it is difficult to visualize the full input vector's correlation containing more than 20 dimensions. There is an improvement of noise level from the lab data seen in Figure 4.3, with high degradation correlation observed for high duration spent at high voltage state, high mean current, high charging power, low discharging current, and low discharging power. This observation is aligned with known degradation conditions.

One observation about the coefficients from linear regression is the discrepancies between the various datasets. The reason could be many sources including noise and quality of sensors. However, the advantage of having many input features is that the chosen model can determine the right input feature depending on the correlation in the training set and apply a higher weight on that variable. One input feature might have low correlation on one dataset but a high correlation on another dataset.

As mentioned in section 2.1.3, battery health is assumed in the battery analytic industry to age at a linear rate before the knee point. The basic linear regression is utilized as a benchmark given its wide use as the standard trend fitting method, its straight-forward calculation, and transparent white-box model with high interpretability. Just because battery health does not age in a linear way does not mean linear regression is not useful. Because the target output is  $\Delta\text{SOH}$  and not directly SOH, a linear behavior in each time-unit versus input feature can still generate a nonlinear aging trend. This is demonstrated in Figure 4.9 with test set plots of dataset Delta, where health predictions of linear regression are clearly nonlinear. This enables the ability to analyze input coefficients as indicators for input correlation. It is observed that the top three input features with the highest coefficient magnitude are different across all datasets. This is expected given the amount of noise and lack of correlation in the input feature scatter plots. However, it is still seen that temperature, voltage, and current all play an important role in determining  $\Delta\text{SOH}$  even though the exact coefficients are inconsistent across different datasets. Therefore, linear regression is evident as a good baseline model for comparison.

### 5.1.1 Accuracy

The extensions of linear regression are the natural next steps in investigating the relationship between input and output. It is found that the specialized regularization in objective function for Ridge, Lasso, and ElasticNet regressors does not improve accuracies compared to ordinary linear regression. There is a negative correlation between regularization strength alpha and accuracy performance for all three models mentioned. It is not until alpha is set to zero when it essentially becomes an ordinary linear regression that the models match the performance. Therefore, they are excluded in the remainder of the model comparisons.

Polynomial regression is used to investigate whether there is a nonlinear or exponential relationship between the input and output. Polynomial regression with a degree of one is equivalent to ordinary linear regression, so degrees of two and above are explored to see if there is a quadratic relationship between input and output. For the two field datasets, polynomial regression with degree two achieved better accuracy than linear regression, and all degrees above two starts to diverge from base truth, with exponential growth in RMSE as the degree increases. For the lab dataset, degree of four yields the best results. This fact indicates that nonlinearity in aging exists in the datasets. However, the difference between linear regression and polynomial regression is very minimal, and it comes with a cost of performing worse than linear regression on unseen test data. On clean data with much less noise, Polynomial demonstrates that it can capture the data's nonlinearity with much better performance in the lab dataset.

Principal Component Regression (PCR) and Partial Least Squares (PLS) regression attempts to improve accuracy performance by lowering the dimensionality of the input space, but they failed to produce any meaningful improvement. During hyperparameter tuning, it is found that the lower the number of components specified, the higher the RMSE and lower the R2 score is. Because PCR's transformation is purely unsupervised, meaning that no information about the targets is used, it may perform poorly in datasets where the target is strongly correlated with directions that have low variance. PLS attempts to capture the effect of the direction with the lowest variance by making use of target information during the transformation before applying linear regression but failed to produce any improvements. This behavior demonstrates that battery degradation depends on more than just a few input features. Instead of projecting the data onto a lower dimension space, the opposite is needed to gain accuracy improvements.

Gaussian Process Regression (GPR), Support Vector Regression (SVR), and Multi-Layer Perceptron Regression (MLPR) all achieved top three in accuracy performance during both training and testing across all datasets. Their accuracy performances are very closely in match. When strictly looking at training and validation metrics, MLPR ranks first overall in RMSE and R2 score, GPR ranked second overall, and SVR ranked third overall. All three are able to predict nonlinear degradation rates that reflects the actual SOH values. However, when accounting for unseen data, GPR has the most consistent results that generalized the data appropriately, whereas MLPR and SVR can easily overfit the training data. This is observed by the fact that GPR has better results in cross validation and test set than the other two. It has comparatively higher average scores across all folds and smaller standard deviations. GPR's

unique feature of providing probabilistic output gives it an advantage over SVR and MLP, whose high training accuracy comes at the cost of poor performance in unseen data.

### 5.1.2 Scaling

Among the viable models that achieve better accuracy, MLP and Polynomial possess the best properties for scaling. SVR is considered acceptable in scaling, and GPR is considered the worst among the model candidates. Although the measured runtime is not scientifically accurate since they do not account for CPU's other background tasks, it nevertheless gives a good approximation of how well each model scale with data size.

Linear models have the lowest runtimes and scales the best relative to data size with a training time complexity of  $O(N)$ . It takes an average of 11x longer from small (Delta) to large (Alpha and Beta) dataset. Because of their low computational cost, prediction requires only a single dot product once the model is trained. The downside is the need to use linear algebra to perform matrix inverse which could involve an extremely small determinant. This step of the computation has the potential to become unstable as input dimension increases.

Polynomial regression scales well between dataset sizes, but it scales poorly as the degree parameter increases. It takes an average of 72x longer from small to large dataset. Since the optimal degree for this application is small, the runtime penalty of increasing dataset size is still minimal compared to the rest of the model candidates. The major drawback of polynomial regression observed is the strict requirement of the function's input domain, as inputs that are outside the bounds of input range will quickly diverge. This phenomenon is demonstrated in the overfitting plot on the right in Figure 2.8. The model will only be accurate if the input is within the input domain of the training data. Therefore, it requires sufficient data points to cover the entire range of input domain.

GPR requires significantly higher runtimes for bigger dataset than all other models to the order of three or more magnitudes, taking an average of 1331x longer from small to large dataset. Training and predicting time are measured in multiple minutes compared to less than a second. This is due to its training time complexity of  $O(N^3)$ .

MLP scales very well with a training time complexity of  $O(N^2)$ , taking an average of 8x longer from small to large dataset. It has a higher runtime proportional to the number of layers and neurons chosen. Multi-Layer Perceptron type neural networks are inherently computationally heavy in training and running, but the computation can be easily parallelized with the use of graphics processing units (GPU). The runtime results demonstrates that it scales better than GPR when comparing datasets of smaller and bigger sizes.

SVR slots in between GPR and MLP in terms of scaling, taking an average of 194x longer from small to large dataset. With a training time complexity of  $O(N^2)$ , SVR possess the same theoretical scaling properties as MLP. However in terms of measured runtimes, SVR is much faster in all datasets, especially when the layer and neuron sizes increase for MLP.

### 5.1.3 Ease of Use

Besides ordinary least squares linear regression, there is no model that can be applied to any dataset without tuning hyperparameters. The amount of hyperparameter that requires tuning is correlated to the interpretability of each model. Ones that are classified as black-box models have considerably more hyperparameters to tune than ones that are classified as white-box models.

Among the viable models that achieved better accuracy, GPR is the easiest to adapt without the risk of overfitting. With only the kernel to choose and optional regularization parameter, GPR takes the least amount of effort to implement.

Since MLPR is a black-box type model, the process of tuning hyperparameter for ANN is by far the most effort out of all the model candidates. It has a tendency to unknowingly overfit the dataset during the tuning step without warning. This behavior is observed when measuring performance for the test sets. SVR and MLPR models with hyperparameters that optimizes accuracy in the training and verification set is found to have poor performance in the test set. The opposite is also true – models with hyperparameter choices that yields good performance in the test set produces worse accuracy in the training and verification set.

Another drawback of MLPR is the sensitivity to initialization of neurons. Because the initialization is randomly assigned each run in Sklearn's model, a different model is generated every run with varying performance metrics. This behavior is observed across all datasets with different sizes, with a more pronounced effect in the smaller dataset Delta. This phenomenon is reflected by the extreme mean cross validation score and standard deviation that are six to eight orders of magnitude higher than all other models. The wide swing in R2 scores during k-fold cross validation is due to its stochastic nature of random weight initializations. This makes MLPR very sensitive and inconsistent. A repeatable output can be achieved by specifying the hyperparameter for random seed, but there is no way of knowing if that specific model is the optimal one or contains unwanted behaviors.

SVR has less hyperparameters to tune than MLPR, so its tuning effort is higher than GPR but lower than MLPR. Considered a black-box type model as well, SVR has the same tendency of overfitting as MLPR but with less severity. It is observed that tuning hyperparameters epsilon for error margin and C for error size tolerance according to the training and verification set yield lower than expected results in the test set.

### 5.1.4 Models Verdict

Linear models including Ridge, Lasso, and ElasticNet are considered useful models with properties of high interpretability and low computational cost. However, they are also the least accurate models.

PCR and PLS regressor is not considered an optimal battery health forecasting model due to poor accuracies. The best accuracy performance is only achieved with specifying the full input dimension as the number of components, where they would match the baseline linear regression's accuracies. This behavior is observed in all datasets.

Polynomial regressor is not considered an optimal battery health forecasting model due to its minimal improvements in accuracy and potential to produce diverging outputs when the input is outside of its training domain.

GPR, SVR, and MLP are all considered optimal battery health forecasting models. All three models outperform linear regression with various strengths and shortcomings. GPR is the easiest to implement but does not scale well with large datasets. MLP has the best accuracy but the most difficult to implement and has a high risk of overfitting. SVR has the fastest runtime but the worst accuracy out of the three and has a medium risk of overfitting.

## 5.2 Challenges

The level of noise existing in all field data obtained for this thesis is a major challenge in creating the optimal predictive model. This is reflected in both the significantly lower R<sup>2</sup> scores as well as larger deviation from ground truth during testing for the models trained with the field datasets versus the lab dataset. The cause of noise can be originated from several sources. Since each system is unique and not in a controlled environment, there is an inherent variation in use environment. There exist many possible variations even the same model of battery storage system is analyzed, as no system will ever have the same electrical production and demand in a single given day. It is also common to have missing data in field data obtained from manufacturers. It is also common to have gaps in data within certain period of time if the particular system is offline or getting serviced.

Another major challenge in the thesis is the variation in knee point behaviors. There is a lack of consistent training data points with the expected knee point in degradation for the field datasets in order for the trained models to predict when the knee point will occur. The degrading behavior is drastically different from one system to another compared to lab data with controlled environment. This demonstrates that there are many factors in determining how fast a battery degrades in the real world. However, this also poses the challenge of adapting a general model for all the possible degrading paths.

Capacity estimation is also a challenge in terms of data accuracy. The SOC and SOH values from BMS is historically not accurate. This is the reason voltage is chosen as an input feature instead of SOC. The open circuit voltage (OCV) curve for a LIB which indicates the capacity at different voltage level has a different curve depending on the charge and discharge rate as well as how much health degradation it has. Therefore, it is unreliable to solely rely on BMS data for SOC and SOH. For this reason, SOH for field data obtained in this thesis is calculated using the partner company's proprietary algorithm.

## 6 Conclusion

In this concluding chapter, the process and findings of this thesis are summarized. In addition, the outlook on future research are presented.

### 6.1 Summary

In the rapidly growing market of lithium-ion battery, there is a need for better battery management tools to increase longevity and safety of various energy applications. This thesis aims to search for the optimal data-driven model for the application of battery health prognosis with the best balance of accuracy performance, robustness, and runtime.

The dataset used for training and testing the models are sourced from three sources, mentioned in section 3.1. Two are from the residential home storage sector, and one is from a published lab testing result. A total of six types of machine learning regression techniques discussed in section 2.3 makes up the candidate models being explored – Linear Regression, Polynomial Regression, Principal Component Regression (PCR), Gaussian Process Regression (GPR), Support Vector Regression (SVR), and Multi-Layer Perceptron Regression (MLPR).

The same algorithm outlined in chapter 3 is applied to each dataset, where selected input features, explained in section 3.2, are extracted from each specific time period. The input features are then separated into training, verification, and test sets. The training set is used to train the individual models and select their optimal hyperparameter, discussed in section 3.4. The best configuration for each dataset is then used for comparison and evaluation. The performance criteria for evaluating the models include root mean squared error (RMSE), coefficient of determination (R<sup>2</sup> score), training and operational runtime, and cross validation deviation. Each model candidate's performance metric is listed in section 4.3. Finally, each model's behavior is tested by applying the models on unseen data from test set to confirm visually, found in section 4.5.

This thesis concludes that Support Vector Regression (SVR), Multi-Layer Perceptron Regression (MLPR), and Gaussian Process Regression (GPR) are all good model choices for battery health prognosis application with various unique drawbacks for each model. All three models are able to outperform linear regression as the baseline for both field data and lab data. All three have superior accuracies among all ML models explored in this thesis. SVR possesses the best balance of all performance metrics, ranking third in accuracy, first in runtime and second out of three in cross-validation. MLPR possesses good attributes in accuracy and scalability but does poorly in adaptability, ranking first in accuracy, last in runtime, and last in cross-validation. GPR on the other hand possesses good accuracy and ease of use but has poor scalability, ranking second in accuracy, last in runtime, and first out of three in cross-validation. With enough expertise and time, ANN type models have the biggest potential to outperform all other types of ML models. Hence it is the preferred choice for all state-of-the-art predictive models. However, MLPR's small advantage in accuracy over SVR in this specific application is not worth the extra time and effort required for this specific application.

Which model is considered the optimal would depend on the user's priority. GPR is the ideal choice for applications that values ease of implementation and generalization over large fleet sizes over the added computation time. MLPR is the ideal choice for applications that value accuracy with enough expertise and resources to optimize the model. SVR is the ideal choice for the best balance of all criteria in terms of accuracy, computational cost, and ease of use.

## 6.2 Potential Future Research

There exist areas for improvement in this thesis and other strategies and methods for data-driven prognosis in battery health.

One potential for future research is to deal with the amount of noise in field data. Machine learning can certainly be applied to identify corrupt or missing data within the time-series data. With a cleaner training data, the models can predict future health values more accurately. More advanced models that can identify abnormalities in input data would be a higher goal to achieve to deal with the persistent noise in field data. Higher effort in data cleaning would also benefit the effect of data noise in accuracy. One idea is down-sampling the time-series data resolution from seconds to minutes or hours before extracting features.

One way to improve the runtime for Gaussian Process Regression is to explore the options of Sparse Variational GPR and using GPU accelerated GPR. Sparse Variational GPR can be used to approximate priors<sup>60</sup>, reducing the complexity of the model. GPUs can also be utilized to train GPR models in parallel. The *GPy*<sup>61</sup> and *GPyTorch*<sup>62</sup> Python packages include tools to implement the above-mentioned GPR extensions in a faster way.

Another area to be improved is the hyperparameter optimization. Since the approach to tuning hyperparameters is trial and error, there is always a possibility that the optimal result found is only a local maxima. The hidden layer sizes of a MLPR, for example, has infinite number of combinations to test. The global maxima may exist with a completely different set of hyperparameters. Further ML methods can be utilized to improve the process of finding the optimal hyperparameters for all the types of models explored in this thesis. Naturally, this steps into a whole separate topic in the ML field.

The area that can have the most impact for most amount of people is the integration of battery health prognosis and cloud computing into every BMS of LIB systems. Connecting every device to the cloud enables data-driven models with high accuracy and robustness and can enable all the additional features of safety and battery health management. This kind of built-in predictive model within every BMS in the fleet will continuously be improved and updated as new data gets uploaded to the cloud. There is a huge potential in terms of additional value-adding services and features. These include health forecasting under various scenarios, use pattern recommendations to the customer, and early warning for warranty service or potentially dangerous situations.

---

<sup>60</sup> cf. Burt et al. (Rates of Convergence for Sparse Variational Gaussian Process Regression) 2019

<sup>61</sup> cf. SheffieldML (*GPy* - A Gaussian Process (GP) framework in Python) 2020

<sup>62</sup> cf. Gardner et al. (*GPyTorch*: blackbox matrix-matrix Gaussian process inference with GPU acceleration) 2018

## V Bibliography

**GPy - A Gaussian Process (GP) framework in Python.** <https://gpy.readthedocs.io/>. Accessed: 11.04.2023

**IEEE Recommended Practice for Sizing Lead-Acid Batteries for Stationary Applications.** Piscataway, NJ, USA

**2017 American Control Conference (ACC):** 2017 American Control Conference (ACC). 5/24/2017 - 5/26/2017: IEEE 2017.

**(Machine Learning):** Machine Learning: Elsevier, 2020

**Abdi, H; Williams, L. J. (Principal component analysis):** Principal component analysis. In: WIREs Comp Stat, Vol. 2, 2010, Iss. 4, pp.433–459

**Anscombe, F. J. (Graphs in Statistical Analysis):** Graphs in Statistical Analysis. In: The American Statistician, Vol. 27, 1973, Iss. 1, pp.17–21

**Attia, P. M; Bills, A; Brosa Planella, F; Dechent, P; dos Reis, G; Dubarry, M; Gasper, P; Gilchrist, R; Greenbank, S; Howey, D; Liu, O; Khoo, E; Preger, Y; Soni, A; Sripad, S; Stefanopoulou, A. G; Sulzer, V. (Review—“Knees” in Lithium-Ion Battery Aging Trajectories):** Review—“Knees” in Lithium-Ion Battery Aging Trajectories. In: J. Electrochem. Soc., Vol. 169, 2022, Iss. 6, p. 60517f.

**Awad, M; Khanna, R. (Efficient learning machines):** Efficient learning machines. Theories, concepts, and applications for engineers and system Designers / Mariette Awad, Rahul Khanna (The expert's voice in machine learning). New York: Apress Open, 2015

**Bhutada, G. (Lithium Consumption Has Nearly Quadrupled Since 2010):** Lithium Consumption Has Nearly Quadrupled Since 2010. <https://elements.visualcapitalist.com/lithium-consumption-has-nearly-quadrupled-since-2010/>. Accessed: 11.04.2023

**Birk, C. R; Roberts, M. R; McTurk, E; Bruce, P. G; Howey, D. A. (Degradation diagnostics for lithium ion cells):** Degradation diagnostics for lithium ion cells. In: Journal of Power Sources, Vol. 341, 2017, pp.373–386

**Bishop, C. M. (Pattern Recognition and Machine Learning):** Pattern Recognition and Machine Learning (Information Science and Statistics). Softcover reprint of the

original 1st edition 2006 (corrected at 8th printing 2009) ed. New York, NY: Springer New York, 2016

**Bisong, E. (Building machine learning and deep learning models on Google Cloud Platform):** Building machine learning and deep learning models on Google Cloud Platform. A comprehensive guide for beginners / Ekaba Bisong. New York: Apress, 2019

**Bousquet, O; Luxburg, U. von; Rätsch, G. (Ed.) (Advanced Lectures on Machine Learning):** Advanced Lectures on Machine Learning. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004

**Burt, D. R; Rasmussen, C. E; van der Wilk, M. (Rates of Convergence for Sparse Variational Gaussian Process Regression):** Rates of Convergence for Sparse Variational Gaussian Process Regression: Apollo - University of Cambridge Repository, 2019

**Deshpande, R; Verbrugge, M; Cheng, Y.-T; Wang, J; Liu, P. (Battery Cycle Life Prediction with Coupled Chemical Degradation and Fatigue Mechanics):** Battery Cycle Life Prediction with Coupled Chemical Degradation and Fatigue Mechanics. In: J. Electrochem. Soc., Vol. 159, 2012, Iss. 10, A1730-A1738

**Ecker, M; Nieto, N; Käbitz, S; Schmalstieg, J; Blanke, H; Warnecke, A; Sauer, D. U. (Calendar and cycle life study of Li(NiMnCo)O<sub>2</sub>-based 18650 lithium-ion batteries):** Calendar and cycle life study of Li(NiMnCo)O<sub>2</sub>-based 18650 lithium-ion batteries. In: Journal of Power Sources, Vol. 248, 2014, pp.839–851

**Edge, J. S; O'Kane, S; Prosser, R; Kirkaldy, N. D; Patel, A. N; Hales, A; Ghosh, A; Ai, W; Chen, J; Yang, J; Li, S; Pang, M.-C; Bravo Diaz, L; Tomaszewska, A; Marzook, M. W; Radhakrishnan, K. N; Wang, H; Patel, Y; Wu, B; Offer, G. J. (Lithium ion battery degradation: what you need to know):** Lithium ion battery degradation: what you need to know. In: Physical chemistry chemical physics PCCP, Vol. 23, 2021, Iss. 14, pp.8200–8221

**Gardner, J. R; Pleiss, G; Bindel, D; Weinberger, K. Q; Wilson, A. G. (GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration):** GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration: arXiv, 2018

**Ge, H; Aoki, T; Ikeda, N; Suga, S; Isobe, T; Li, Z; Tabuchi, Y; Zhang, J. (Investigating Lithium Plating in Lithium-Ion Batteries at Low Temperatures Using Electrochemical Model with NMR Assisted Parameterization):** Investigating

Lithium Plating in Lithium-Ion Batteries at Low Temperatures Using Electrochemical Model with NMR Assisted Parameterization. In: J. Electrochem. Soc., Vol. 164, 2017, Iss. 6, A1050-A1060

**Goodfellow, I; Bengio, Y; Courville, A. (Deep learning):** Deep learning (Adaptive computation and machine learning). Cambridge, Massachusetts: The MIT Press, 2016

**Greenbank, S. (Data-driven battery state of health diagnostics and prognostics):** Dissertation University of Oxford. Oxford, 2022

**Hawkins, T. R; Singh, B; Majeau-Bettez, G; Strømman, A. H. (Comparative Environmental Life Cycle Assessment of Conventional and Electric Vehicles):** Comparative Environmental Life Cycle Assessment of Conventional and Electric Vehicles. In: Journal of Industrial Ecology, Vol. 17, 2013, Iss. 1, pp.53–64

**Heimes, H. H. (Lithium-ion battery cell production process):** Lithium-ion battery cell production process. Aachen, Frankfurt am Main: PEM der RWTH Aachen University; DVMA, December 2018

**International Energy Agency (Global EV Outlook 2022):** Global EV Outlook 2022.  
<https://www.iea.org/reports/global-ev-outlook-2022>. Accessed: 11.04.2023

**Jalkanen, K; Karppinen, J; Skogström, L; Laurila, T; Nisula, M; Vuorilehto, K. (Cycle aging of commercial NMC/graphite pouch cells at different temperatures):** Cycle aging of commercial NMC/graphite pouch cells at different temperatures. In: Applied Energy, Vol. 154, 2015, pp.160–172

**Jin, X; Vora, A. P; Hoshing, V; Saha, T; Shaver, G. M; Wasynczuk, O; Varigonda, S. (Comparison of Li-ion battery degradation models for system design and control algorithm development):** Comparison of Li-ion battery degradation models for system design and control algorithm development: 2017 American Control Conference (ACC). 2017 American Control Conference (ACC), 5/24/2017 - 5/26/2017: IEEE 2017, pp.74–79

**Kelleher, J. D; Mac Namee, B; D'Arcy, A. (Fundamentals of machine learning for predictive data analytics):** Fundamentals of machine learning for predictive data analytics. Algorithms, worked examples, and case studies / John D. Kelleher, Brian Mac Namee, Aoife D'Arcy. Cambridge, Massachusetts: The MIT Press, 2015

- Kong, L; Li, C; Jiang, J; Pecht, M. (Li-Ion Battery Fire Hazards and Safety Strategies):**  
Li-Ion Battery Fire Hazards and Safety Strategies. In: Energies, Vol. 11, 2018, Iss. 9, p. 2191f.
- Li, Y; Liu, K; Foley, A. M; Zülke, A; Berecibar, M; Nanini-Maury, E; van Mierlo, J; Hoster, H. E. (Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review):** Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. In: Renewable and Sustainable Energy Reviews, Vol. 113, 2019, p. 109254f.
- Li, Y; Zou, C; Berecibar, M; Nanini-Maury, E; Chan, J. C.-W; van den Bossche, P; van Mierlo, J; Omar, N. (Random forest regression for online capacity estimation of lithium-ion batteries):** Random forest regression for online capacity estimation of lithium-ion batteries. In: Applied Energy, Vol. 232, 2018, pp.197–210
- Loyola-Gonzalez, O. (Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View):** Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. In: IEEE Access, Vol. 7, 2019, pp.154096–154113
- Markets and Markets (Lithium-ion Battery Market):** Lithium-ion Battery Market.  
<https://www.marketsandmarkets.com/Market-Reports/lithium-ion-battery-market-49714593.html>. Accessed: 11.04.2023
- Ning, G; Haran, B; Popov, B. N. (Capacity fade study of lithium-ion batteries cycled at high discharge rates):** Capacity fade study of lithium-ion batteries cycled at high discharge rates. In: Journal of Power Sources, Vol. 117, 2003, Iss. 1-2Iss. , pp.160–169
- Ning, G; Popov, B. N. (Cycle Life Modeling of Lithium-Ion Batteries):** Cycle Life Modeling of Lithium-Ion Batteries. In: J. Electrochem. Soc., Vol. 151, 2004, Iss. 10, A1584
- Pedregosa, F; Varoquaux, G; Gramfort, A; Michel, V; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. (Scikit-learn: Machine Learning in Python):** Scikit-learn: Machine Learning in Python. In: Journal of Machine Learning Research, Vol. 12, 2011

- Prada, E; Di Domenico, D; Creff, Y; Bernard, J; Sauvant-Moynot, V; Huet, F. (Simplified Electrochemical and Thermal Model of LiFePO<sub>4</sub> -Graphite Li-Ion Batteries for Fast Charge Applications):** Simplified Electrochemical and Thermal Model of LiFePO<sub>4</sub> -Graphite Li-Ion Batteries for Fast Charge Applications. In: J. Electrochem. Soc., Vol. 159, 2012, Iss. 9, A1508-A1519
- Prada, E; Di Domenico, D; Creff, Y; Bernard, J; Sauvant-Moynot, V; Huet, F. (A Simplified Electrochemical and Thermal Aging Model of LiFePO<sub>4</sub> -Graphite Li-Ion Batteries: Power and Capacity Fade Simulations):** A Simplified Electrochemical and Thermal Aging Model of LiFePO<sub>4</sub> -Graphite Li-ion Batteries: Power and Capacity Fade Simulations. In: J. Electrochem. Soc., Vol. 160, 2013, Iss. 4, A616-A628
- Purewal, J; Wang, J; Graetz, J; Soukiazian, S; Tataria, H; Verbrugge, M. W. (Degradation of lithium ion batteries employing graphite negatives and nickel–cobalt–manganese oxide + spinel manganese oxide positives: Part 2, chemical–mechanical degradation model):** Degradation of lithium ion batteries employing graphite negatives and nickel–cobalt–manganese oxide + spinel manganese oxide positives: Part 2, chemical–mechanical degradation model. In: Journal of Power Sources, Vol. 272, 2014, pp.1154–1161
- Rasmussen, C. E. (Gaussian Processes in Machine Learning): Gaussian Processes in Machine Learning.** In: Bousquet, O; Luxburg, U. von; Rätsch, G. (Eds.): Advanced Lectures on Machine Learning (Series: Lecture Notes in Computer Science). Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp.63–71
- Schulz, E; Speekenbrink, M; Krause, A. (A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions):** A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. In: Journal of Mathematical Psychology, Vol. 85, 2018, pp.1–16
- Scikit-Learn (3.3. Metrics and scoring: quantifying the quality of predictions):** 3.3. Metrics and scoring: quantifying the quality of predictions. Accessed: 10.02.2023
- Severson, K. A; Attia, P. M; Jin, N; Perkins, N; Jiang, B; Yang, Z; Chen, M. H; Aykol, M; Herring, P. K; Fragedakis, D; Bazant, M. Z; Harris, S. J; Chueh, W. C; Braatz, R. D. (Data-driven prediction of battery cycle life before capacity degradation):** Data-driven prediction of battery cycle life before capacity degradation. In: Nat Energy, Vol. 4, 2019, Iss. 5, pp.383–391
- Shah, A; Shah, K; Shah, C; Shah, M. (State of charge, remaining useful life and knee point estimation based on artificial intelligence and Machine learning in**

- (lithium-ion EV batteries: A comprehensive review):** State of charge, remaining useful life and knee point estimation based on artificial intelligence and Machine learning in lithium-ion EV batteries: A comprehensive review. In: Renewable Energy Focus, Vol. 42, 2022, pp.146–164
- Sohn, S; Byun, H.-E; Lee, J. H. (Two-stage deep learning for online prediction of knee-point in Li-ion battery capacity degradation):** Two-stage deep learning for online prediction of knee-point in Li-ion battery capacity degradation. In: Applied Energy, Vol. 328, 2022, p. 120204f.
- Statista (Projected global battery demand from 2020 to 2030, by application):** Projected global battery demand from 2020 to 2030, by application. <https://www.statista.com/statistics/1103218/global-battery-demand-forecast/>. Accessed: 11.04.2023
- Statista (E-Scooter-sharing Worldwide):** E-Scooter-sharing - Worldwide. <https://www.statista.com/outlook/mmo/shared-mobility/shared-rides/e-scooter-sharing/worldwide>. Accessed: 11.04.2023
- Sterner, M; Stadler, I. (Handbook of Energy Storage):** Handbook of Energy Storage. Demand, Technologies, Integration / Michael Sterner, Ingo Stadler. Berlin, Heidelberg: Springer, 2019
- Sun, P; Bisschop, R; Niu, H; Huang, X. (A Review of Battery Fires in Electric Vehicles):** A Review of Battery Fires in Electric Vehicles. In: Fire Technol, Vol. 56, 2020, Iss. 4, pp.1361–1410
- U.S. Department of Energy Office of Energy Efficiency and Renewable Energy (How Do All-Electric Cars Work?):** How Do All-Electric Cars Work? <https://afdc.energy.gov/vehicles/how-do-all-electric-cars-work>. Accessed: 11.04.2023
- U.S. Department of Energy Office of Energy Efficiency and Renewable Energy (Solar Photovoltaic Cell Basics):** Solar Photovoltaic Cell Basics. <https://www.energy.gov/eere/solar/solar-photovoltaic-cell-basics>. Accessed: 11.04.2023
- United Nations Framework Convention on Climate Change (The Paris Agreement):** The Paris Agreement. <https://unfccc.int/process-and-meetings/the-paris-agreement>. Accessed: 11.04.2023
- Vetter, J; Novák, P; Wagner, M. R; Veit, C; Möller, K.-C; Besenhard, J. O; Winter, M; Wohlfahrt-Mehrens, M; Vogler, C; Hammouche, A. (Ageing mechanisms in lithium-ion batteries):** Ageing mechanisms in lithium-ion batteries. In: Journal of Power Sources, Vol. 147, 2005, Iss. 1-2, pp.269–281

**Wold, S; Sjöström, M; Eriksson, L. (PLS-regression: a basic tool of chemometrics):**

PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems, Vol. 58, 2001, Iss. 2, pp.109–130

**Xiong, S. (A study of the factors that affect lithium ion battery degradation):** A study of

the factors that affect lithium ion battery degradation. 2019.

**Xu, B; Oudalov, A; Ulbig, A; Andersson, G; Kirschen, D. S. (Modeling of Lithium-Ion**

**Battery Degradation for Cell Life Assessment):** Modeling of Lithium-Ion  
Battery Degradation for Cell Life Assessment. IEEE Trans. Smart Grid, Vol. 9,  
2018, Iss. 2, pp.1131–1140

**Zhang, F; O'Donnell, L. J. (Support vector regression):** Support vector regression: Ma-

chine Learning. Elsevier, 2020, pp.123–140

**Zwicker, M; Moghadam, M; Zhang, W; Nielsen, C. V. (Automotive battery pack manu-**

**facturing – a review of battery to tab joining):** In: Journal of Advanced Joining Processes Volume 1, <https://doi.org/10.1016/j.jaip.2020.100017>, 03.2020  
Automotive battery pack manufacturing – a review of battery to tab joining.  
Journal of Advanced Joining Processes, Vol. 1, 2020, p. 100017f.

## VI Appendix

### A.1. Hyperparameter Tuning Steps

**Table 6.1: Hyperparameter tuning steps and intermediate results for Lasso**

Parameter	Choice	RMSE	R2 score
Alpha	0.5	0.8948	0.3338
Alpha	0.25	0.779	0.495
Alpha	0	0.309	0.9205

**Table 6.2: Hyperparameter tuning steps and intermediate results for ElasticNet**

Parameter	Choice	RMSE	R2 score
Alpha	1	0.9235	0.2903
L1 Ratio	0.5		
Alpha	1	0.8003	0.4671
L1 Ratio	0.25		
Alpha	0.5	0.6826	0.6123
L1 Ratio	0.25		
Alpha	0.25	0.4271	0.8482
L1 Ratio	0.1		
Alpha	0.1	0.4089	0.8608
L1 Ratio	0		
Alpha	0	0.309	0.9205
L1 Ratio	0		

**Table 6.3: Hyperparameter tuning steps and intermediate results for PCR**

Parameter	Choice	RMSE	R2 score
Number of components	2	0.7625	0.5162
Number of components	5	0.6322	0.6675
Number of components	12	0.3864	0.8758
Number of components	19	0.3062	0.9220

## A.2. Performance Metrics by Datasets

**Table 6.4: Performance metrics of the model candidates in dataset Alpha**

Model	RMSE [%SOH]	RMSE Rank	R2	R2 Rank	Training Runtime [s]	Training Runtime Rank
Linear	0.1382	7	0.0540	8	0.0269	3
Ridge	0.1382	7	0.0540	7	0.0018	1
Lasso	0.1383	9	0.0527	9	0.0957	7
ElasticNet	0.1383	9	0.0527	9	0.0660	5
Polynomial	0.1337	4	0.1142	4	0.0660	5
GPR	0.1107	2	0.3924	2	263.8524	10
SVR	0.1189	3	0.2989	3	4.0557	8
PCR	0.1382	6	0.0541	6	0.0355	4
PLS	0.1381	5	0.0550	5	0.0104	2
MLPR	0.1058	1	0.4452	1	23.5409	9

**Table 6.5: Performance metrics of the model candidates in dataset Beta**

Model	RMSE [%SOH]	RMSE Rank	R2	R2 Rank	Training Runtime [s]	Training Runtime Rank
Linear	0.1189	5	0.0705	5	0.0643	6
Ridge	0.1212	7	0.0343	7	0.0062	2
Lasso	0.1233	9	0.0000	9	0.0069	3
ElasticNet	0.1233	9	0.0000	9	0.0016	1
Polynomial	0.1119	4	0.1765	4	0.6408	8
GPR	0.0968	1	0.3840	1	350.1981	10
SVR	0.1038	3	0.2905	3	0.0539	5
PCR	0.1231	8	0.0027	8	0.0976	7
PLS	0.1193	6	0.0637	6	0.0192	4
MLPR	0.1005	2	0.3350	2	30.6397	9

**Table 6.6: Performance metrics of the model candidates in dataset Delta**

Model	RMSE [%SOH]	RMSE Rank	R2	R2 Rank	Training Runtime [s]	Training Runtime Rank
Linear	0.3062	5	0.9220	4	0.0040	4
Ridge	0.3166	9	0.9166	9	0.0008	1
Lasso	0.3090	7	0.9205	7	0.0054	7
ElasticNet	0.3090	7	0.9205	7	0.0049	5
Polynomial	0.5401	10	0.7573	10	0.0049	5
GPR	0.2316	3	0.9554	3	0.2306	9
SVR	0.2167	2	0.9609	2	0.0105	8
PCR	0.3062	5	0.9220	5	0.0023	2
PLS	0.3062	5	0.9220	5	0.0025	3
MLPR	0.1897	1	0.9701	1	3.1652	10

### A.3. Cross Validation Metrics by Datasets

**Table 6.7: Cross validation performance of model candidates in dataset Alpha**

Model	CV Runtime [s]	CV Runtime Rank	CV Score	CV Score Rank	CV STD	CV STD Rank
Linear	0.0318	4	0.0644	4	0.0135	4
Ridge	0.0037	1	0.0644	3	0.0135	4
Lasso	0.0638	7	0.0639	6	0.0144	7
ElasticNet	0.0619	6	0.0639	6	0.0144	7
Polynomial	0.4507	8	0.1290	1	0.0394	9
GPR	0.0224	2	0.0644	4	0.0135	4
SVR	1.9503	9	-0.0061	9	0.0066	1
PCR	0.0435	5	0.0637	8	0.0137	6
PLS	0.0272	3	0.0644	2	0.0133	2
MLPR	5.1946	10	-79319.3	10	82608.8	10

**Table 6.8: Cross validation performance of model candidates in dataset Beta**

Model	CV Runtime [s]	CV Runtime Rank	CV Score	CV Score Rank	CV STD	CV STD Rank
Linear	0.0443	7	0.0730	4	0.0043	1
Ridge	0.0130	4	0.0728	6	0.0044	3
Lasso	0.0035	1	0.0730	2	0.0045	4
ElasticNet	0.0036	2	0.0730	2	0.0045	4
Polynomial	0.5327	8	0.0873	1	0.1689	9
GPR	418.7037	10	0.0730	4	0.0043	1
SVR	0.0380	5	0.0444	9	0.0063	8
PCR	0.0391	6	0.0724	7	0.0046	6
PLS	0.0111	3	0.0716	8	0.0056	7
MLPR	6.6663	9	-59488.8	10	59472.0	10

**Table 6.9: Cross validation performance of model candidates in dataset Delta**

Model	CV Runtime [s]	CV Runtime Rank	CV Score	CV Score Rank	CV STD	CV STD Rank
Linear	0.0039	4	0.6726	3	0.4450	4
Ridge	0.0021	1	0.8922	1	0.0473	3
Lasso	0.0066	6	0.6464	6	0.5086	7
ElasticNet	0.0061	5	0.6464	6	0.5086	7
Polynomial	0.0588	8	0.1646	8	0.8184	9
GPR	0.1732	9	0.7882	2	0.0425	2
SVR	0.0092	7	-0.1285	9	0.0321	1
PCR	0.0033	2	0.6726	5	0.4450	6
PLS	0.0038	3	0.6726	4	0.4450	4
MLPR	0.9137	10	-33225013.7	10	22824498.6	10