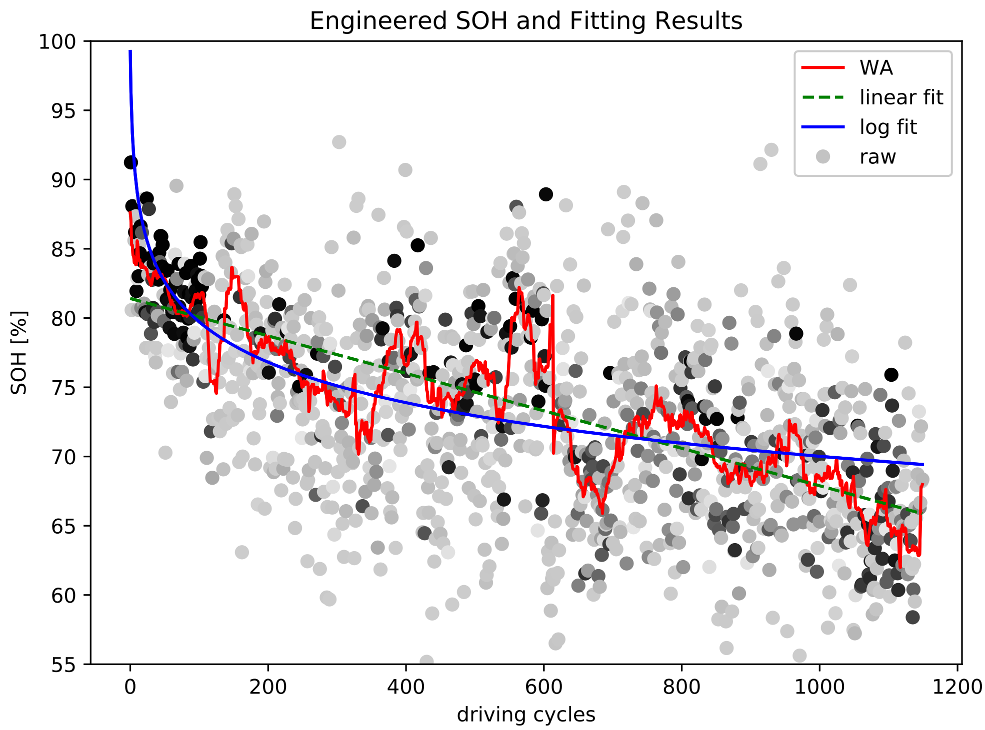
CHALMERS
UNIVERSITY OF TECHNOLOGY



Machine Learning to Predict Future Power and Energy Demands in the Battery Electric Vehicles

Master's thesis in Systems, Control and Mechatronics

SUJITHA DAVIDSON
ZE ZHANG

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Machine Learning to Predict Future Power and Energy Demands in the Battery Electric Vehicles

SUJITHA DAVIDSON
ZE ZHANG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
System, Control and Mechatronics
Chalmers University of Technology
Gothenburg, Sweden 2019

Machine Learning to Predict Future Power and Energy Demands in the Battery Electric Vehicles
SUJITHA DAVIDSON
ZE ZHANG

Supervisor: Christian Fleischer, Volvo Car Corporation
Examiner: Knut Åkesson, Electric Engineering

Master's Thesis 2019
Electric Engineering
System, Control and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Machine Learning to Predict Future Power and Energy Demands in the Battery Electric Vehicles.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Machine Learning to Predict Future Power and Energy Demands in the Battery Electric Vehicles SUJITHA DAVIDSON, ZE ZHANG
Electric Engineering
Chalmers University of Technology

Abstract

With the maturer technique of batteries, the battery electric vehicles are strongly developed in recent years for better ride experience and zero air pollution. In order to promote the battery performance and prolong the battery lifetime, investigating the battery states is necessary. A battery is a complex electrochemical device, whose internal ingredients change over usage and time. With all the factors affecting the battery as weather, temperature, driving behavior, driving load, etc., a battery precise model is difficult to establish.

In this project, a new method based on machine learning technology is introduced and tested. By analyzing and processing the real-world driving data from C30 battery electric vehicles provided by the Volvo Cars, a dataset containing several signals from charging and discharging cycles is obtained. A pipeline mainly about estimating the state of charge and state of health then is proposed to analyze energy demands. The core is a machine learning model which contains two sub-models. The first sub-model is a recurrent neural network for estimating the state of charge (or depth of discharge), and the second one is a nonlinear autoregressive network with exogenous inputs for estimating and predicting the state of health. The whole model is tested on part of the C30 dataset promising results.

Keywords: Battery electric vehicle, battery states, machine learning, neural network, big data.

Acknowledgements

In this project, we would like to state the sincerest appreciation to our thesis work supervisor Christian Fleischer and examiner Professor Knut Åkesson. Mr. Fleischer offered us the valuable opportunity to work on this project in Volvo Cars and provided us a lot of help on data and hardware resources. He also brought us good suggestions on making essential decisions. His support is the premise of our final completion. Professor Åkesson gave us many help on making the research plan and implementing certain methods. He provided us many meaningful feedback on this report as well. In addition, we appreciate the help from Herman Johnsson, who had worked on the C30 dataset before and offered us quite important information. We also need to thank other colleagues in the Traction Battery and other departments in Volvo Cars helping us in different ways.
Finally, we'd like to thank our parents for their care and support.

SUJITHA DAVIDSON, ZE ZHANG, Gothenburg, August 2019

Contents

List of Figures

List of Tables

1

Introduction

In this chapter, for the topic *Machine Learning to Predict Future Power and Energy Demands in the Battery Electric Vehicles*, the history, motivation, expected outcome, limitation are introduced and discussed.

1.1 Background

Automobiles have been the most common way of private transportation for over a hundred years. They bring human beings a lot of convenience as well as some problems. Air pollution, for instance, is one of the side effects of conventional gas cars. Electric vehicles as the new energy vehicles are widely developed these years. They are quiet, easy to maintain, and cleaner. As the most important component, batteries are always the hot spot in electric vehicles' technique and their states are mostly concerned.

After the productization in the automobile industry in the late 1800s, when the Second Industrial Revolution came, the basis of current cars was developed [3]. Amazingly, one of the first tries on electric cars was invented just several years after the productization of the gasoline cars, by British inventor Thomas Parker.

Looking back to history, in the beginning, electric vehicles (EVs) had the same even better popularity compared to gasoline vehicles. Since the massive exploration of oil boosted the development of Internal Combustion Engines (ICE) in the early 1900s, EVs started to lose the competition with ICE cars. However, due to the pollution problem of ICE, EVs were destined to be brought up again, which was the reality that the U.S. Congress decided new legislation "recommending electric vehicles as a means of reducing air pollution" in 1966. Nowadays, as the new energy automobile, EVs are wildly developed across the world by all the major automobile manufacturers. The reasons are not only the pollution reduction demand but also better driving experience such as the quieter driving environment. The recent study [4] shows the vehicle electrification impact on air quality.

Nevertheless, a big problem for EVs is the complexity of batteries. EVs have higher energy utilization rate but the battery volume is limited and its states are hard to evaluate. Here comes the Battery Management Systems (BMSs). The BMS, as its name implies, is a real-time system used to manage the rechargeable battery, in the form of cell or pack. In EVs, the BMSs need to monitor the battery states, such as current, voltage and temperature, optimize the battery performance, predict the battery energy and consumption, record and schedule the charging and discharging events [5]. The more accurate can the estimation on the battery states be, the better can the battery be utilized. To estimate the battery states requires the battery model. The traditional battery model is the Equivalent Circuit Model (ECM). In this report, a data-driven method is proposed. The battery model is represented by a machine learning model.

Figure 1.1: One of Thomas Parker's early cars outside the family home; The Manor House, Upper Green, Tettenhall. Thomas is sat in the middle and on the back seat is possibly his son Alfred. From the Wolverhampton History and Heritage Website, Elwell-Parker, Limited.

1.2 Purpose

This project is meant to build a machine learning model to substitute traditional ECMs. By training the new model with real-world charging and discharging data from Battery Electric Vehicles (BEVs), this machine learning model should be able to present the theoretical battery model and reserve the realistic using influence.
Specifically speaking, we want to build a machine learning model that can estimate the battery states precisely for different BEVs in both short-term and long-term.
In addition, the new model should be of generalization for BEVs with the same battery pack, but it can adjust itself by learning the actual charging and discharging behaviors.
In this project, this model is called the Battery State Model (BSM) and can estimate the current charge load and battery healthy condition as well as the future tendency.

1.3 Limitation

For any practical machine learning problem, one of the most significant factors that affect the performance of the chosen model is the data. People keep saying

that a good dataset for a certain application is the basic guarantee of valid results. However, this is also the limitation for a lot of machine learning methods.
The limitations of this project are from two perspectives. The biggest one is that the raw data from real-world driving test are not well organized. This is because the sensor system to collect the data is old (it is important for this project to have data over years but then the outdated sensor system has to be tolerated). The raw data are not sampled in a fixed period and different sensors work in different frequencies. Some sensors don't have a precise measurement or delicate resolution, which means they are dull to changes and basically can't be used as real-time signals. The even worse occasion is some of them recording wrong states of the battery or vehicle. More details will be seen in the data processing part.
The other fatal issue is that the capacity signal is missing in the sensor system. Since supervised learning is used in this project to estimate and predict capacity, missing labels could mean direct failure. To fix this problem, an engineering way of forming capacity is introduced. This engineering way which generates engineering capacity calculates capacity by definition. The problem then is this method uses charge amount difference and corresponding proportion to full charge to get the actual full charge and the difference of charge amount is calculated by the current signal. Both of them (the proportion value and current signal) are corrupted on different levels and influenced by many factors. This will cause a bad engineering capacity value. To make this better to use, people usually use some filters or fitting methods. More details will be seen in the engineering capacity part.
There is also one man-made issue. Due to the confidential reason, not all information about the data source is accessible for this project. With these limitations, the data processing cost a long time and there wouldn't be sufficient time to explore as many possibilities as possible. This leaves more extension and possibility in the future.

1.4 Application

Electric vehicles have entered the cars market and the number of EVs is increasing very fast. Batteries are usually one of the most expensive components in EVs, so it is very reasonable to squeeze out as much as possible of them. This mainly means that, with the premise of safety, the lifetime of batteries should be prolonged, the power and energy capabilities should be improved, the power density should be large and the usage should be stable. These cannot be achieved if the states of batteries are unknown.
With the proposed methods, the battery management system can estimate the states of the battery accurately under most conditions without knowing the individual usage situation of the battery, just through several signals that are easy to get.
More specifically, BMS can estimate even predict the current battery remaining charge (energy) as well as the healthy condition.

2

Problem Definition

With the fast exponentially falling cost of battery packs for vehicles usage as in battery electric vehicles (BEVs), hybrid electric vehicles (HEVs), plug-in hybrid electric vehicles (PHEVs) [6] [7], there is this big trend that all kinds of EVs are occupying more and more cars market as the new energy automobile. Lithium-ion batteries (LIBs) are the main role in this revolution due to their efficiency, high voltage, high energy density, long cycle lifetime, and environmental benignity [8] [9]. Nowadays, almost all major car companies are developing pure electric vehicles which are referred to as battery electric vehicles (BEVs) in this report. It is necessary to understand the battery behavior in the real-world driving scenario for BEVs. The monitoring and prognosis of cell performance including degradation in LIBs are essential for assuring the reliability and safety of the battery electric vehicles. In this chapter, we define our problem and the research questions that we aim to solve in our thesis.

2.1 Problem Description

A battery is an energy storage device that converts the chemical energy stored in its active materials into electrical energy by means of electrochemical reactions. The basic electrochemical storage element is a cell. A battery consists of one or more cells, connected in series, parallel or both, based on the desired nominal battery voltage and capacity. An electrochemical cell is composed of an anode, a cathode and between them an electrically insulating, but ionically conducting electrolyte enabling ions to move through the cell. There is also a rigid separator, often a polymer, between the anode and cathode to contain the electrolyte and prevent short circuit [10].
The electrodes function as hosts for Li^+. In the charged state, the anode (negative electrode) has a high concentration of lithium and low potential. The cathode (positive electrode) has a low concentration of lithium and high potential. When connecting the electrodes via an external load, electrons move spontaneously from the anode, increasing its potential, through the load to the cathode, decreasing its potential, therefore reducing the driving force, i.e the overall cell voltage. Li^+'s are simultaneously transported by the electrolyte from the anode, through the separator, to the cathode to conserve charge neutrality, figure 2.1. The described processes are that of discharge. The processes are reversed during charging of the battery [10].

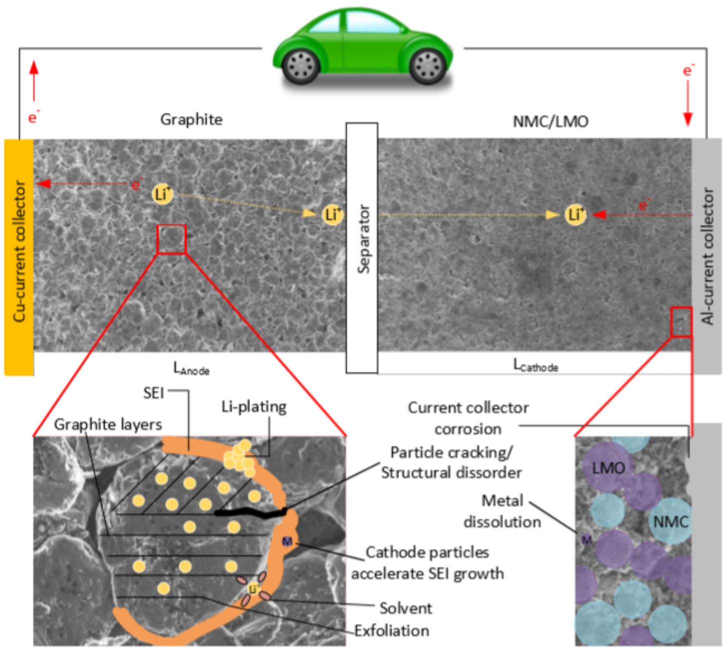


Figure 2.1: Schematic illustration of working of LiB discharge and the aging mechanisms during usage [1].

Performance of the battery degrades with time and use (aging of the battery). Aging of the battery depends on its usage and even the battery ages without use sometimes (calendar aging). Aging is also due to various processes happening inside the battery. The figure 2.2 shows the various aging phenomenon of the battery and explaining them all is out of the scope of this thesis. But it is important to note the complexity of the battery system.

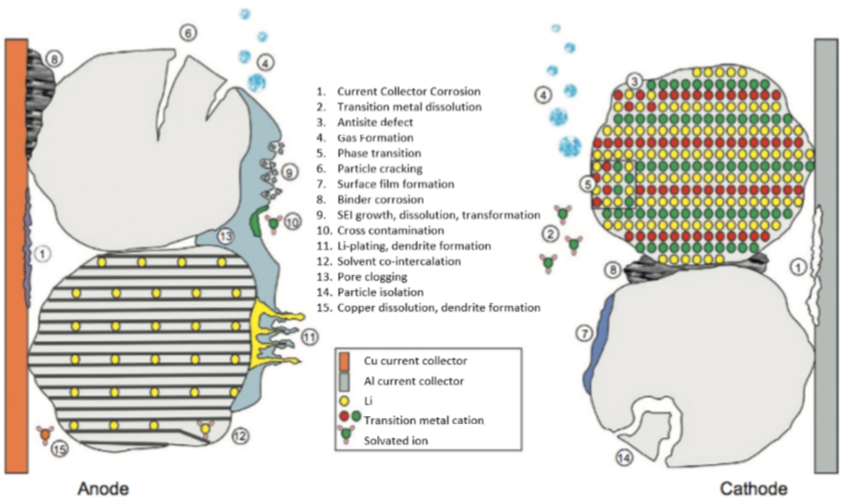


Figure 2.2: Different aging phenomenon of batteries [2].

Based on the literature study, the various major parameters affecting the aging of the battery were found to be,

1. Change in SOC.
2. Voltage and current.
3. Temperature.
4. Current density.
5. Number of charging and discharging cycles.
6. Charging rate, etc.

There is no doubt that the energy and power of the battery is highly related to its charge amount and health state, which can be simply presented by capacity (state of health, SOH) and state of charge (SOC). To know these values, a model of the battery needs to be discovered.

2.2 Modelling of the Battery

Before exploring more on the battery model, specific explanations on relevant terms need to be made. Then, the conditional way of modelling is introduced before the machine learning methods are presented in the next chapter.

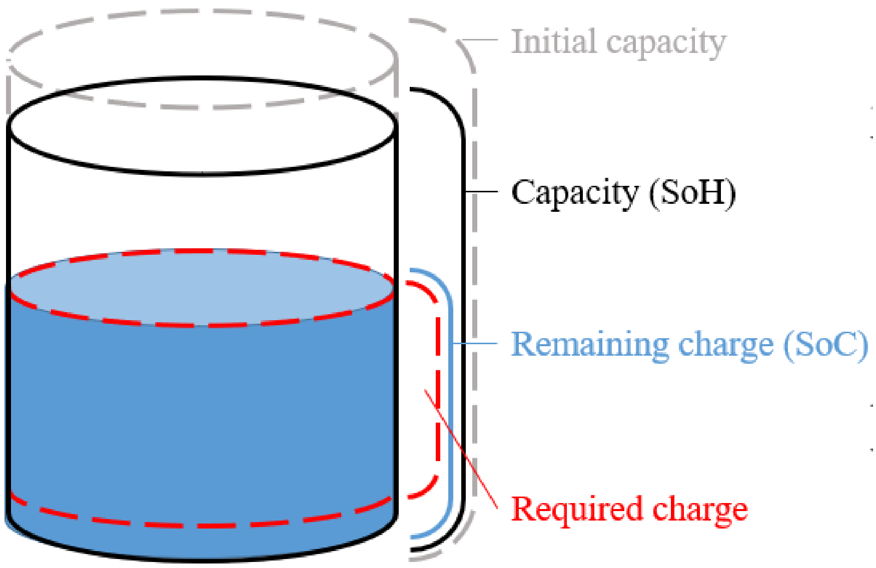


Figure 2.3: This figure shows the straightforward concepts of some battery states. The initial capacity is known from the supplier, which is treated as a constant. The capacity is the long-term state of the battery and state of health (SOH) is its relative representation. The remaining charge shows how much charge left in the current moment which is a short-term variable. State of charge (SOC) is the relative form of the remaining charge.

The performance of the battery can be presented by the battery states. The states of the battery are often continuously estimated in order to make sure that the battery is always operating within the safe limits.

2.2.1 State of Charge and Depth of Discharge

State of Charge (SOC) is the amount of charge stored in the battery, presented in percent of total capacity (0-100%). It is the closest analogy to a fuel gauge in a petrol engine car [11]. The SOC can be estimated in many ways. The most reliable and intuitive way is to completely discharge the battery under known conditions and measure the amount of charge drained. The battery will be depleted and the measured SOC, however known, will no longer be relevant [12].
Depth of Discharge (DOD) is the complement of SOC as shown in equation (2.1). While the SOC is (0% = empty; 100% = full), DOD will be (0% = full; 100% = empty).

$$DOD = 100\% - SOC \tag{2.1}$$

In this project, since the signal directly from the test batteries is DOD instead of SOC, DOD is actually selected to use.

2.2.2 Capacity and State of Health

Capacity is easy to understand as the maximal amount of charge that can be stored in the battery. Customarily, its unit is $A \cdot h$ as ampere per hour. It will decay over time and usage.
The State of Health (SOH) is a general indicator of how well the battery is doing. It is often used to compare the current capability of the battery to the initial 100%. To do so, it is connected to some properties of the battery, often capacity, resistance, self-discharge rate, or power density [12]. SOH is closely connected to aging. In fact, it might be considered a tool to measure aging in the battery. Perhaps most used is the capacity-based SOH, how much charge the battery can store. To simplify this problem, in this project capacity is chosen to represent SOH. This parameter will decrease with age, decreasing the amount of energy that can be stored in the battery. Such degradation is important since it is closely connected to the range of the car. It also means that an older battery will charge quicker since there is less available lithium for transport [11].
To formulate the above battery states, assuming a battery with initial capacity Cap_0 ampere per hour, and real-time capacity Cap ampere per hour,

$$Q_0 = 3600 \times Cap_0 \tag{2.2}$$
$$Q = 3600 \times Cap \tag{2.3}$$
$$SOH = Q/Q_0 \times 100\% \tag{2.4}$$
$$SOC = q/Q \times 100\% \tag{2.5}$$
$$DOD = 100\% - SOC \tag{2.6}$$

where q is the current amount of charge, Q is the current maximal charge capacity and Q_0 is the initial battery capacity for brand new batteries.

2.2.3 Traditional ECMs

Due to the lithium-ion battery having a polarization phenomenon, this phenomenon will lead to the battery dynamic hysteresis characteristics. Because of battery dy-

namic hysteresis characteristics, the open-circuit voltage (OCV) curve of the charging and discharging process is different. The battery OCV of the charging process is higher than discharging [13]. Hence a battery model is needed in order to make estimations of the battery states. A LIB model can be classified as an electrochemical model, physical model, equivalent circuit model (ECM), thermal model, coupled electro-thermal model, and so on. Among them, ECM is the most commonly used in lithium-ion batteries SOC estimation for BEV applications due to their simple model structure [13].
ECMs such as the Rint model, Thevenin model, PNGV model, GNL, n-RC model, and FOM (Fractional Order Model), are shown in figure 2.4. The Rint model is very simple to implement in real-time. However, the model's output equation expressed is only a rough estimate actual terminal voltage of the battery, which may lead to large uncertainties in SOC estimates [14]. The Thevenin model connects a parallel RC network in series based on the Rint model. The PNGV model can be used to describe the changing of open-circuit voltage generated in the time accumulation of load current by adding a capacitor in series based on the Thevenin model [15]. The GNL nonlinear equivalent circuit battery model takes into account the influence of the self-discharge on characteristics of the battery [16]. The n-RC consists of an n parallel RC network in series based on the Thevenin model in order to consider dynamic voltage performances [17].

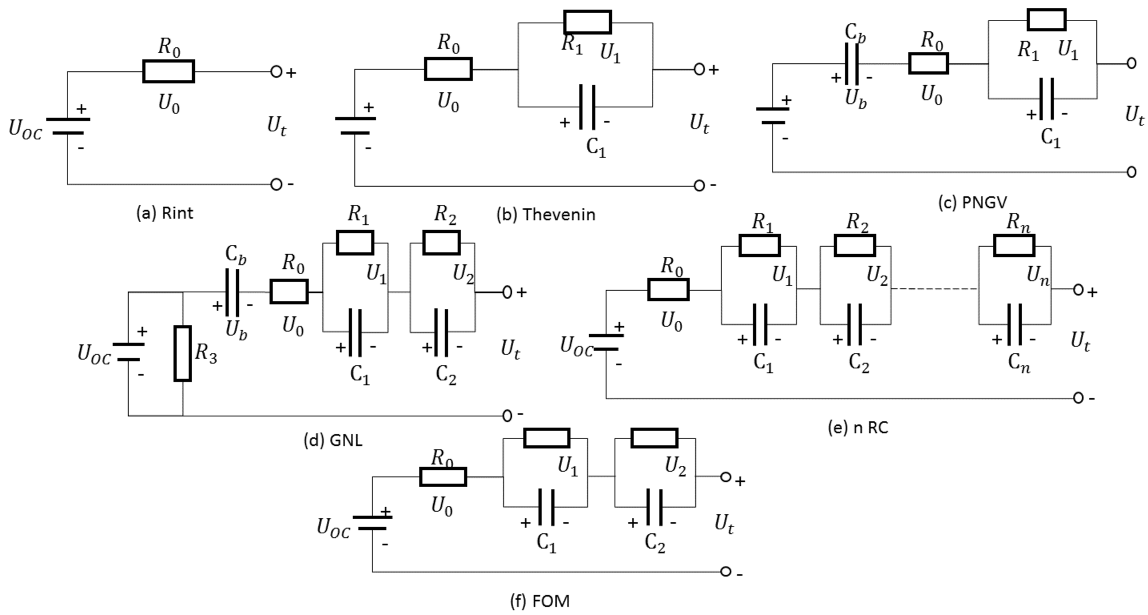


Figure 2.4: Several equivalent circuit models of batteries.

The intuitive phenomenon of polarization effect is that when the charging or discharging process stops, the terminal voltage converges to an equilibrium value slowly rather than immediately. However, since the nonlinear relations among the polarization effect, current, SOC and temperature, the dynamic characteristics of the battery are difficult to model [18].
Battery modeling has an important influence on SOC estimation. Establishing a battery model is a challenge, because of the complex electrochemical and dynamic

environment. Each model may lack accuracy and adaptability to use in different operating conditions. In addition, the battery hysteresis effect is not considered in many ECMs. So, we need to develop a suitable battery model that can work accurately under different load conditions, which is done using machine learning.

2.3 Real-world Situation

For the EV battery systems, an accurate capacity prediction can prevent the battery from over-discharge and charge, thus ensuring battery system safety, making more efficient use of the limited energy, and extending the battery life. Specifically, it can support the precise calculation of the vehicle driving range, provide a better discharging or charging strategy, improve the efficiency of other power sources, and make balance strategies work more effectively. The lithium-ion battery capacity estimation in BEVs has become a major challenge due to its complex electrochemical reactions and performance degradation caused by various factors [19].
In order to improve the accuracy, stability, robustness, economy, and other challenges of the capacity prediction algorithm in electric vehicles, these issues, such as lithium-ion battery hysteresis characteristic, battery model, aging, estimation algorithm, and cell unbalancing, are worth studying in depth and are also solutions to solve SOC estimation problems [13].
The remaining capacity of a lithium-ion battery is affected by many factors, such as external environmental loads, the number of charging and discharging cycles, the value of discharging current and so on. Since the battery capacity is related to several measured features, it is convenient to estimate capacity by using trained connections with multiple features from the current, voltage, state of charge and temperature profiles. These methods avoid understanding complex reaction mechanisms inside batteries to construct mathematical or physical models, thereby having been widely investigated by many researchers. To effectively capture the non-linearity relationship between the features and capacity, various machine learning approaches have been integrated with feature-based methods as well. The prognostic results show that these integrated methods often have a good performance in the battery capacity estimation [20].

2.4 Research Goals

With all the problems and information, our project has several goals to complete.

1. Analyzing and processing the real-world EVs driving data.
2. Using machine learning methods to estimate the battery states including the state of charge (or depth of discharge) and the state of health.
3. The estimation of the state of charge should have a small error. To start with 5% error is acceptable.
4. The estimation of the state of health should follow the ground truth as much as possible. It is an extra goal if battery degradation can be predicted.

3 Theory and Approach

In this chapter, the core idea and main approaches of the new methods are presented.

3.1 OCV-SOC relationship

The methods proposed in this project are based on the fact that the OCV-SOC curve will change with the increasing of life cycles, or in other words the decaying of battery capacity. However, the OCV-SOC curve will change based on a lot of factors, the demand to find a generalized model is brought up. Reference [21] has a review on this model. But most methods are verified in a lab environment while the real situation is more complicated.

In this project, a pipeline to obtain the battery states is proposed based on the batteries' data from real-world driving. The OCV-SOC relationship is represented by a neural network and with the estimated SOC and other signals, the capacity can be estimated and predicted.

3.2 Machine Learning

The traditional way to solve an engineering problem as shown in chapter two about ECMs is to searching for an answer with known rules and inputs. It is pretty straightforward to understand but with an obvious problem, which is how to know the rules. In a lot of cases, accurate rules or models are not just given. They are actually black-box systems. It is then easier to just let the system running in different situations (inputs) and collect the data (outputs) from these tests. So, the target changes into searching for the rules with given inputs and outputs. If a hypothetical model has the same or tolerably similar performance as the real one, the rules are found. Machine learning (ML) methods are designed for this purpose.

As a significant branch of Artificial Intelligence (AI), with sufficient data and reasonable assumption on the model, ML can make the machines learn by themselves to find the model [22]. However, as one of the most typical features of ML, data-driven means the performance of ML is directly linked to the quality and quantity of data, which is normally the biggest issue of ML.

Assuming that there is a potential relationship among the SOC, capacity (SOH) and other signals such as the current, voltage, temperature etc., which can be validated to some extent in the previous chapter, if the relationship can be discovered, the states of the battery can be obtained by simply measuring those signals. However,

the signals from the real world are often corrupted and influenced by a lot of factors. The driver's behavior will also affect the aging problem but it is very hard to consider that into the vehicle or battery model. A good way to explore this is via machine learning methods.
ML is able to find the complex underlying pattern from big data. There are a lot of machine learning methods. The battery state estimation in this project is a typical supervised learning problem, and a regression problem more specifically speaking. Supervised learning means the result of the learning process has a standard answer to compare with and evaluate. This answer is commonly called the label. So, a label can be regarded as a concise representation of a data sample. In this case, SOC and capacity (SOH) are the labels.

3.2.1 Regression Model

The two basic problems of supervised learning are classification and regression. If the desired output is the class of the input, for example, whether an input image contains a dog, cat or other animals, then it is called a classification problem. In a classification problem, the output belongs to a discrete, finite category. Otherwise, if the desired output is a continuous value, it is a regression problem. SOC and SOH are both continuous value from 0% to 100%. Here it is a typical regression problem.
A general format for regression problems can be defined as following, for many samples in the form of $\mathbf{x}^{[i]} = (x_1, x_2, \cdots, x_N)^T$, which means the i-th sample with N features, each of them has a corresponding match target $t^{[i]}$. Note that t is simplified here since it can be a vector as well but this won't influence the calculation. This mapping from $\mathbf{x}$ to t is presented as

$$t = f(\mathbf{x}) \tag{3.1}$$

A simplest assumption is that the relationship between the input $\mathbf{x}$ and output t is linear. Then equation (3.1) can be defined more specific as

$$t = \mathbf{w}^T\mathbf{x} + \mathbf{b} \tag{3.2}$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_N)^T$ and $\mathbf{b} = (b_1, b_2, \cdots, b_N)^T$.
Similarly, the relationship can be polynomial, exponential, logarithmic and so on. For example, the logarithmic fitting is implemented later to smooth the capacity decaying curve for the neural network achieving better performance. The logarithmic fitting is based on the formula (3.2) but the input is rescaled logarithmically as below.

$$t = w \cdot log(x) + b \tag{3.3}$$

After the model is determined, an algorithm to find the optimal parameters is required. The most common one, also the one used in this project, is the least squares approximation. This algorithm defines the loss function as equation (3.4) and finds the minimal loss by setting its gradient to zero.

$$L = \sum_{i=1}^{n}(y_i - t_i)^2 \tag{3.4}$$

where y is the true value and t is the estimation.

3.2.2 Deep Machine Learning

The traditional machine learning methods focus on expressing some relationship in a specific mathematical way, usually as a function. With the growth of data size and complexity in the relationship, one function is not enough anymore. Then more functions of previous functions are introduced to fulfill the complexity. This is artificial neural networks and one repeat of the function is called a layer. With multiple layers, the machine learning models are referred to as deep. So, deep machine learning is actually neural networks with multiple layers.

3.3 Artificial Neural Networks

The artificial neural network (ANN), or neural network (NN) in short, is a branch of machine learning technique which is a major concept in artificial intelligence (AI). Looking at the working mechanism of human brains, the neural signals transmit through nerve cells, as known as neurons. They are regarded as a basic computing unit in our brain. The reason that a human can understand complicated things and distinguish different things is there are billions of neurons analyzing and computing the electric signals converted from the external signals like light, sounds, force, etc. into abstract and summative concepts. The concept of the ANN is a simple mimic of human neurons.

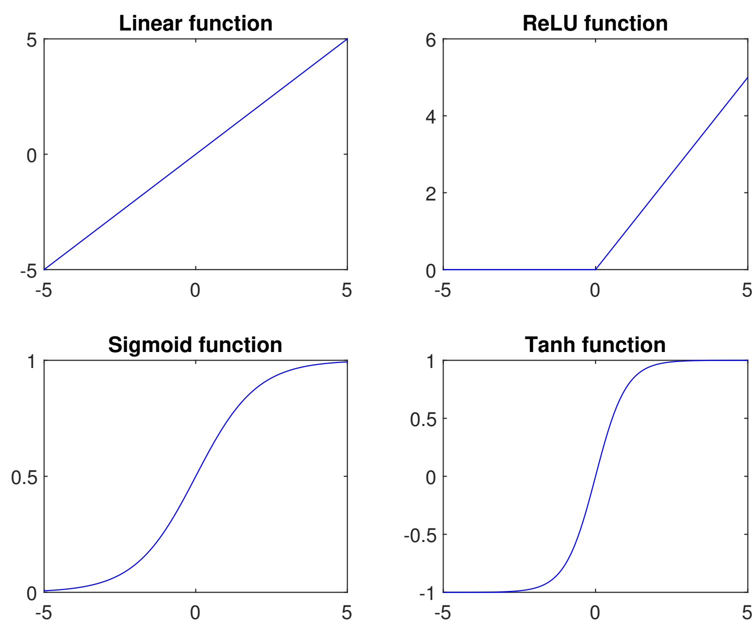


Figure 3.1: Four common activation functions in artificial neural networks.

In ANNs, the basic computing unit is known as the Perceptron, which is multiple inputs, signal output, nonlinear function. For N inputs x_1, x_2, $\cdots$, x_N, the perceptron calculated the weighted summation with a bias b and perform a nonlinear function f to get the final output y. Its formula is

$$y = f(w_1x_1 + w_2x_2 + \cdots + w_Nx_N + b) \tag{3.5}$$

where the nonlinear function f is called the *activation function* and is often selected as the linear function, sigmoid function, tanh function or ReLU function as shown in figure 3.1.

A perceptron is a single layer neural network and obviously not a deep one. However, by regarding one perceptron as one node and setting several nodes per layer and several layers in total, a basic deep feedforward neural network (FNN) is formed as shown in figure 3.2. There are one input layer, m hidden layers, and one output layer. Since the input layer doesn't perform any calculation, this neural network is called a $m+1$ layers neural network. All the parameters related to the architecture of the neural network are hyperparameters, including the number of hidden layers, the number of nodes in each layer and so on. Normally, the activation functions in all hidden layers are the same noted as f_h.

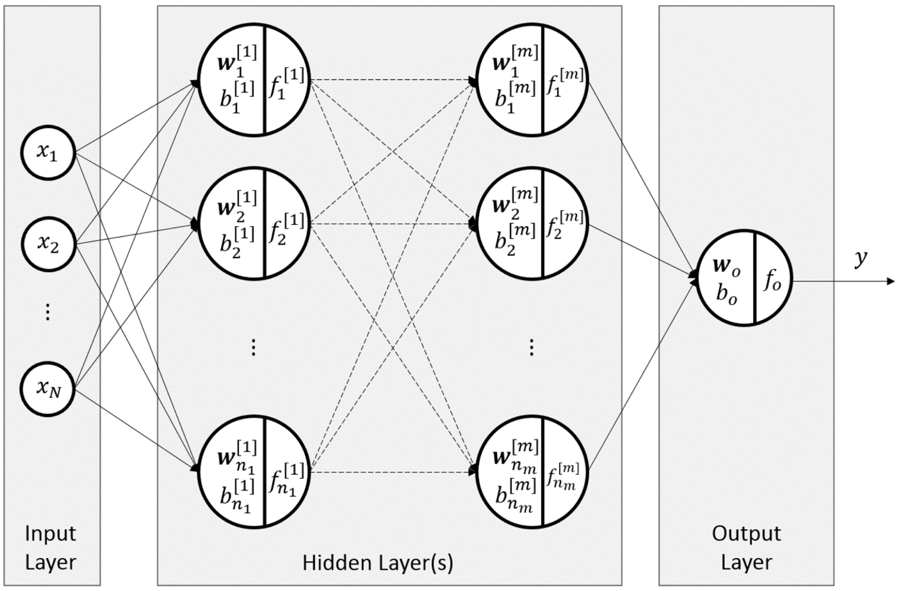


Figure 3.2: A classic deep neural network with complete structure hyperparameters noted.

Besides the hyperparameters that shows the structure of a neural network, weights and biases are the parameters in a neural network. Now a vector, $\mathbf{S} = (n_1, n_2, ..., n_m)^T$, can be used to represent the architecture of the deep neural network, which means the neural network is of m layers and the i-th layer has n_i nodes. So, in the i-th node of j-th layer, the calculation is

$$a_i^{[j+1]} = f_i^{[j]}\left((\mathbf{w}_i^{[j]})^T\mathbf{a}^{[j-1]} + b_i^{[j]}\right) \tag{3.6}$$

where the weight $\mathbf{w}_i^{[j]} = (w_{i,1}^{[j]}, w_{i,2}^{[j]}, \cdots, w_{i,n_{j-1}}^{[j]})^T$, and $\mathbf{a}^{[j]} = (a_1^{[j]}, a_2^{[j]}, \cdots, a_{n_j}^{[j]})^T$ is the output of j-th layer (there are two special cases: the input $\mathbf{x} = \mathbf{a}^{[0]}$ and output $y = a^{[m+1]}$).

3.3.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a kind of Neural Networks that have great performance in image recognition and classification. The basic principle is regional feature detection. By collecting subtle features like regional edges and shadows and aggregating them to form larger features like shapes and colors, CNNs can construct a feature pyramid while the top is the detected object's label or class. The "convolution" operation shown as equation (3.7) is executed by the convolutional layers, which are the key to extract features.

$$o(i, j) = \sum_m \sum_n c(m, n)x(i - m, j - n) \tag{3.7}$$

In equation (3.7), $o(i, j)$ is the output point at i-th row and j-th column, and c is the filter, x is the input. The values m and n mean every element in the filter.
CNNs can be used to detect spatial features inside an image and similar situations. The main idea is to capture the changes in successive values. Thus, CNNs actually fit any application fulfilling this feature, which means CNNs can deal with more than just pictures or videos.
In this project, CNNs couldn't give a satisfactory result on the C30 dataset, thus not included in this report.

3.3.2 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are another kind of Neural Networks that are designed to deal with sequential information such as texts and time series.
The essential difference between RNNs and other feedforward neural networks is that RNNs have the memory function. In equation (3.8), the bold part reserve the previous output as the memory and combine it with current inputs. To compare it with the equation (3.5), it is obvious the dissimilarity is the term my_{k-1} which stands for the history.

$$y_k = f(w_1x_1 + w_2x_2 + \cdots + w_Nx_N + \boldsymbol{my_{k-1}} + b) \tag{3.8}$$

A simple RNN node designed according to the equation (3.8) cannot hold a lastingly good performance with the time prolonging. Therefore, an advanced recurrent neural network, the long short-term memory neural network (LSTM) [23], is invented as the remedy to the problem that the simple RNNs cannot preserve a valid long memory, which manifests exploding or vanishing gradient.

$$i_t = \sigma(x_tU^i + h_{t-1}W^i + b_i) \tag{3.9}$$
$$f_t = \sigma(x_tU^f + h_{t-1}W^f + b_f) \tag{3.10}$$
$$o_t = \sigma(x_tU^o + h_{t-1}W^o + b_o) \tag{3.11}$$
$$\tilde{C}_t = \tanh(x_tU^g + h_{t-1}W^g + b_C) \tag{3.12}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.13}$$
$$h_t = \tanh(C_t) * o_t \tag{3.14}$$

From the equation (3.9) to (3.14), the mechanism of the LSTM node is presented.

3.4 NARX

The Nonlinear Auto-Regressive model with Exogenous inputs (NARX) is an important class of discrete-time nonlinear systems that can be mathematically represented as,

$$y(n+1) = f[y(n), \cdots, y(n-d_y+1); u(n), u(n-1), \cdots, u(n-d_u+1)] \tag{3.15}$$

where $u(n) \in \mathbb{R}$ and $y(n)\epsilon\mathbb{R}$ denote, respectively, the input and output of the model at discrete time step n, while $d_u \geq 1$ and $d_y \geq 1$, $d_u \leq d_y$, are the input-memory and output-memory orders. In a compact vector form, eq. (3.15) may be written as,

$$y(n+1) = f[y(n); u(n)] \tag{3.16}$$

where the vectors $y(n)$ and $u(n)$ denote the output and input regressors respectively. The nonlinear mapping $f(.)$ is generally unknown and can be approximated, for example, by a standard multi-layer perceptron (MLP) network [24]. The resulting connectionist architecture is then called a NARX network, a powerful class of dynamical models which has been shown to be computationally equivalent to Turing machines [25].
In this thesis, the nonlinear mapping of $f(.)$ is done using Feedforward Neural Network (FFN), Random Forest Regressor (RFR) and XGB Regressor (XGB). Hence, in the following sections, RFR and XGB are discussed.
The estimations are done using FFN, RFR and XGB and linear regressor and the predictions are made using NARX.

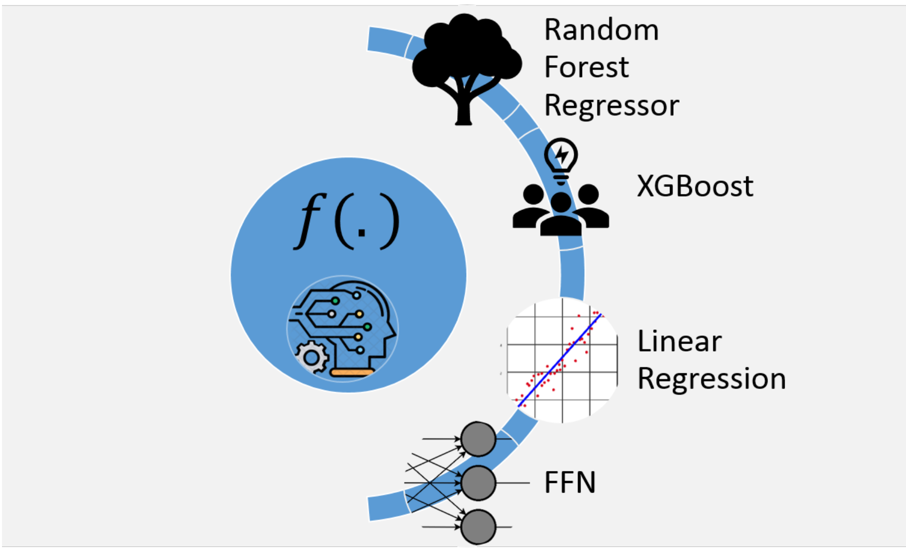


Figure 3.3: The concept of $f(.)$ approximation.

3.4.1 Random Forest Regressor

Random Forest is a flexible machine learning algorithm that produces great results most of the time with minimum time spent on hyper-parameter tuning. It is powerful due to its simplicity and also because it can be used for both classification and

regression problems.
Random Forest is an ensemble algorithm. It uses the bagging technique that reduces the overfitting. Bagging is an interesting idea of combining the predictions of n different models each of which was only somewhat predictive while their predictions were not correlated with each other. That would mean that the n models would have profound different insights into the relationships of the data. If an average of those n models is taken, then it brings in effectively the insights from each of them [26]. So this way averaging models is the technique behind the ensemble algorithm.
A random forest algorithm has two concepts that give it the name random. They are the random sampling of training observations when building trees and random subsets of features for splitting nodes.
Each tree in a random forest learns from a random sample of the training observations. The samples are drawn with replacement, known as bootstrapping, which means that some samples will be used multiple times in a single tree. The idea is that by training each tree on different samples, although each tree might have high variance with respect to a particular set of the training data, overall, the entire forest will have lower variance but not at the cost of increasing the bias [26].
The other main concept in the random forest is that each tree sees only a subset of all the features when deciding to split a node. In Skearn this can be set by specifying $max\_features = sqrt(n\_features)$ meaning that if there are 16 features, at each node in each tree, only 4 random features are considered for splitting the node [26].
Sklearn of scikit-learn is used for implementing the random forest regressor. The figure 3.4 shows how a random forest algorithm works.

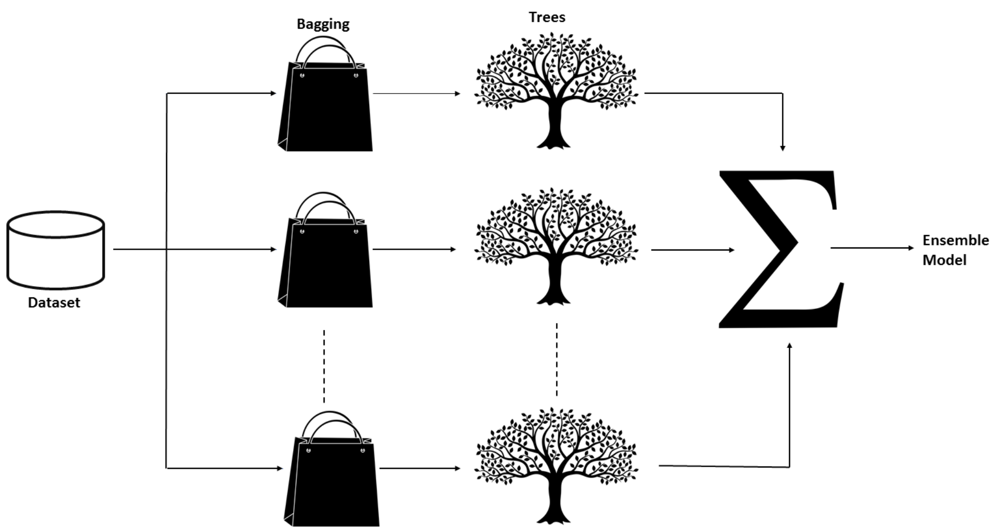


Figure 3.4: Random forest - a way of bagging decision trees.

The number of estimators represents the number of trees in the forest. Usually the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training process considerably, therefore we do a parameter search to find the effective number of estimators.
Depth of each tree in the forest is an important parameter. The deeper the tree, the more splits it has and it captures more information about the data.

However, according to sklearn documentation for the random forest, the search for a split does not stop until at least one valid partition of the node samples will be found, even if it requires to effectively inspect more than $max\_features$ features.
The key insight of the random forest regressors is to construct multiple models which are better than nothing with a significantly lower risk of overfitting.

3.4.2 XGBoost Regressor

XGBoost (Extreme Gradient Boosting) is a scalable end-to-end tree boosting system. It uses the technique of Boosting along with bagging of the random forest.
Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model.
Here is an instance to understand boosting. A subset is created from the original dataset. Initially, all data points are given equal weights. A base model is created on this subset. This model is used to make predictions on the whole dataset. Errors are calculated using the actual values and predicted values. The observations which are incorrectly predicted, are given higher weights. Another model is created and predictions are made on the dataset. This model tries to correct the errors from the previous model. Similarly, multiple models are created, each correcting the errors of the previous model.
The final model (strong learner) is the weighted mean of all the models (weak learners). Thus, the boosting algorithm combines a number of weak learners to form a strong learner. The individual models would not perform well on the entire dataset, but they work well for some part of the dataset. Thus, each model actually boosts the performance of the ensemble [27].
The figure 3.5 shows the idea behind the XGBoost algorithm.

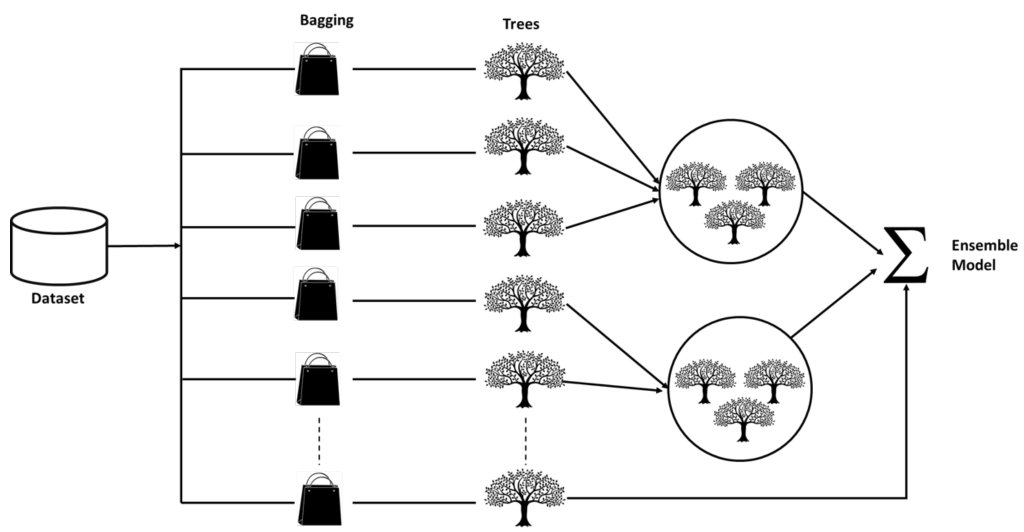


Figure 3.5: XGBoost Principle.

XGBoost is one of the fastest implementations of gradient boosted trees. It uses parallel computing. That is when XGBoost is executed, by default, it would use

all the cores of the computer. The whole point of gradient boosting is to find the function which best approximates the data. When building a gradient boosted decision tree, a challenge is to decide how to split a current leaf.
It is done by considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.
Due to the complexity, XGBoost algorithm is not fully explained here.

3.4.3 Linear Regressor

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. A linear regression line has an equation of the form $Y = a + bX$, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$) [28]. The coefficients and intercept are estimated by least squares, i.e., setting them equal to the unique values that minimize the sum of squared errors within the sample of data to which the model is fitted. And the model's prediction errors are typically assumed to be independently and identically normally distributed.

3.5 Engineered State of Health

In the first chapter, the limitations of this project are mentioned, where the most fatal one is the absence of the capacity signal. However, according to the definition of SOH, there is a way to form SOH using other signals. In this way, the formed SOH is called engineered SOH.
If the charge quantity as shown in equation (2.2) to (2.6) in the battery can be determined, with the DOD signal the capacity can be calculated.

$$SOH = \frac{\int_{t_1}^{t_2} I(t)dt}{DOD(t_2) - DOD(t_1)} \tag{3.17}$$

In equation (3.17), SOH is regarded as a constant in short-term, like during one driving. Thus, it can be calculated in this way. However, the current signal might be not stable and the SOC signal is often inaccurate, which makes the engineered SOH very noisy.
To weaken the noise and obtain a feasible dataset for training and testing, some approaches are implemented. Common sense is that longer the charging or discharging process is, better accuracy the sensor can provide. Meanwhile, longer process means a larger difference between the start and end values of DOD. So, the log data is weighted by the value difference of DOD. Another fact is the SOH does not change too much within a short duration. So, to smooth the SOH, we can calculate the average over a short duration. Combining these ideas, the weighted

average is calculated. For instance, in a period of 5 days, the engineered SOHs and its corresponding DOD differences in i-th day are SOH_i and ΔDOD_i.

$$SOH_{WA} = \frac{\sum_{i=1}^{5} SOH_i \cdot \Delta DOD_i}{5} \tag{3.18}$$

In this case, the duration is selected as 30 days. The result is shown in figure 3.6.

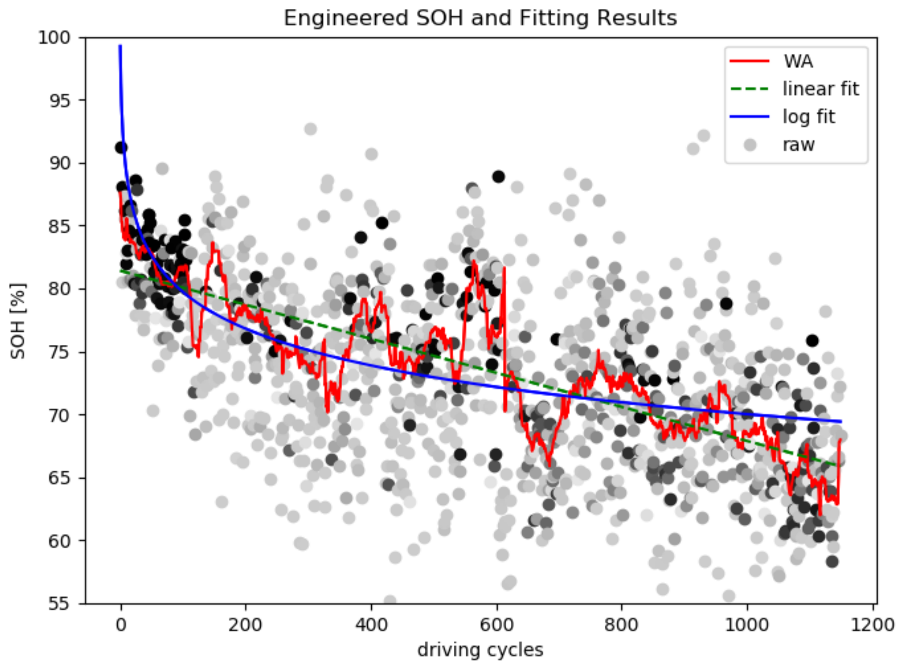


Figure 3.6: Several ways to smooth the engineered SOH. For the raw data points, deeper color means a larger weight. The linear fit curve shows a rough tendency while the logarithmic fit gives better following to the original data.

To figure out the trend of the SOH change, fitting methods are used. The raw data are fitted into two models, linear one and logarithmic one. The formulas are

$$y = ax + b \tag{3.19}$$
$$y = a \log x + b \tag{3.20}$$

Later, the logarithmic fitting results are actually used for the machine learning process.

4

Data Description and Processing

4.1 C30 Real Fleet Dataset - Volvo Cars Corporation

Figure 4.1: This is one Volvo C30 Battery Electric Vehicle. C30 is used mainly for testing. They are assigned to different users who mostly driving around the Gothenburg city. The sensor system will log all the defined signals and all the records are sent to Volvo Cars.

4.1.1 Volvo C30

The electric C30 has a 24 kWh lithium-ion battery, supplied by US manufacturer EnerDel [29]. The electric motor, the motor controller, the charger and other drive-train components are supplied by Swiss manufacturer BRUSA Elektronik AG. [30]. Top speed is estimated by Volvo at 130 km/h (81 mph), with acceleration from 0–50 kph (0–31 mph) in 4 seconds. The C30 Electric can be recharged from a regular household power SOCket and a full charge takes about 7 hours. The all-electric range is up to 150 kilometers (93 mi). The batteries are installed where the fuel

tank normally sits and also in a special compartment in the C30's central tunnel, and as a result, the luggage compartment is unchanged [31]. In August 2012, Volvo ordered a second supply of battery packs from EnerDel to build an internal test fleet together with Siemens [32] [33].
The Volvo C30 Electric is equipped with three climate systems: one supplies the passengers with heating or cooling; one cools or warms the battery pack as necessary; the electric motor and power electronics are water-cooled. The climate control in the passenger cabin features a bioethanol-powered heater, a solution chosen by Volvo to get heating without compromising the battery driving range, but the driver has the option to run the climate unit on electricity from the batteries. Ethanol is the default mode and the ethanol tank can carry 14.5 liters (3.8 US gal) of bioethanol. Volvo has Tested the C30 in winter conditions in temperatures as low as $20°C$ $(4°F)$ [34].
The C30 DRIVe was one of the five finalists to the 2011 Green Car Vision Award [35]. In July 2012, the Volvo C30 electric was named the "Green Car of the Year" in China at the 4th China New Energy Mobility Summit [36].

4.1.2 Dataset

In 2012, Volvo Cars Corporation launched around 150 C30 Vehicles in the first generation and an additional 150 in the second generation, which was launched a few years later. These cars were equipped with modules that could send measurements to the Volvo Cars Corporation through GSM network. The data was collected with an event-based logging system that is, the data was only when there were changes in the values of the signals. From this, it can be concluded that no data was logged when the car was idle. On the other hand, intensive use can give a sampling frequency higher than once per second. There are also built-in systems that make the car wake up and send a signal with a set interval if it is not already in use. The cars were monitored for a period of six years and hence, a large collection of data points were collected that are separated from seconds to even weeks.
There were about 60 signals recorded in the cars. After careful inspection of the data, only the signals that are important for this thesis were used to build the model. Finally, the selected signals are battery output current, battery output voltage, battery working temperature, driving distance, the depth of discharge, power and estimated time to empty the battery.

4.2 Preprocessing

Data preprocessing is a necessary step in any machine learning procedure. It is a technique in which raw data is transferred into an understandable and feasible format. Real-world data are often incomplete, inconsistent, and/or lacking in certain behaviors. Moreover, real-world data are likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Figure 4.2 shows how data preprocessing is done usually.

4.2.1 Error detection

In the raw data, at some moments, the sensor system doesn't work properly, which leads to some wrong values like zero, negative values or sudden jump or drop. For some signals like the voltage, total driving distance, and DOD, they shouldn't have negative values or zeros for some of them. The whole cycle is traversed and all the wrong values are replaced by the previous valid value. As for the sudden changes, they are detected through the difference of the original signal. If there is abnormally large value in the differential sequence, the location of the targeted sudden value is found.

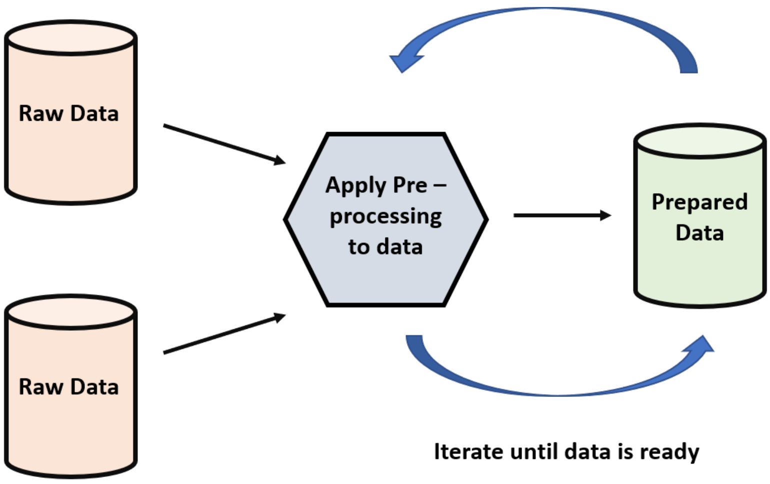


Figure 4.2: Preprocessing in machine learning.

It is also very common to see the missing values, which requires imputation. Since the resolution of the original signal is in second, the signal is so much precise than needed. Imputation is just simply holding the previous valid value.
The strategy here is quite easy because the data are of a large size and complex algorithm will lead to extremely slow processing.

4.2.2 Integration and resampling

The raw data are sparse. There are many repeating seconds containing information of different signals. In order to implement resampling, there can't be any repeating time moments. All the information in the same second needs to be integrated. This step costs a lot of computing power. The final method is to find all the extra seconds in one cycle and remove them all at once.
After selecting relevant features out of all the signals, the charging and discharging cycles need to be recognized. Fortunately, there is one signal "ChargeMode" recording the current charging condition. It is not noise-free and perfectly accurate. By combining other signals as speed, current, etc., the error part can be mostly removed and the charging and driving cycles can be divided.
In the end, all the cycles need to be resampled so that they can be fed into the machine learning model. Linear interpolation is implemented to fulfill this task,

which is forming a linear function between two points and calculate the value of a certain point in between.

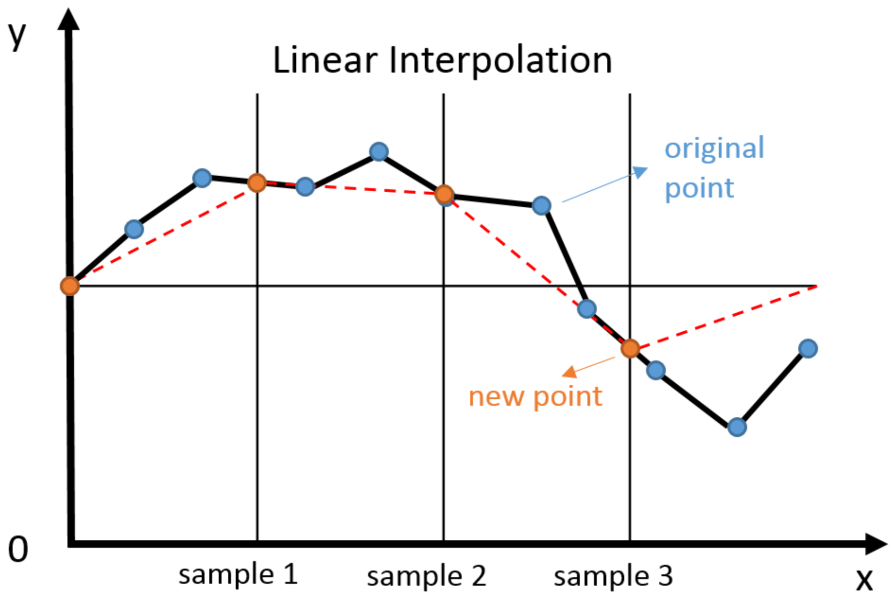


Figure 4.3: The concept of linear interpolation is to regard the unknown points as points on the linear function formed by the previous and next known points.

4.2.3 Prepared data

The whole procedure of preprocessing is shown in figure 4.4.

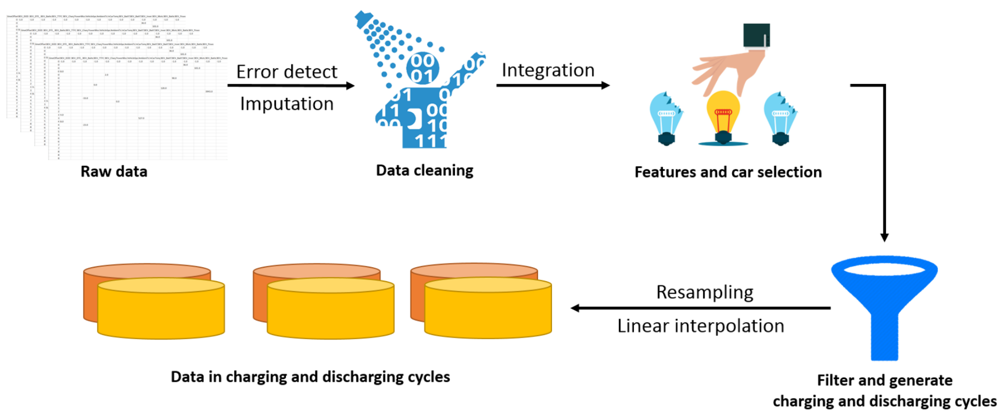


Figure 4.4: The complete procedure of data preprocessing contains error correction, data interpolation, feature detection and resampling. The result is a dataset of separately driving and charging cycles.

As a company confidential policy, the real data are not allowed to expose. A similar sample is shown in figure 4.5.

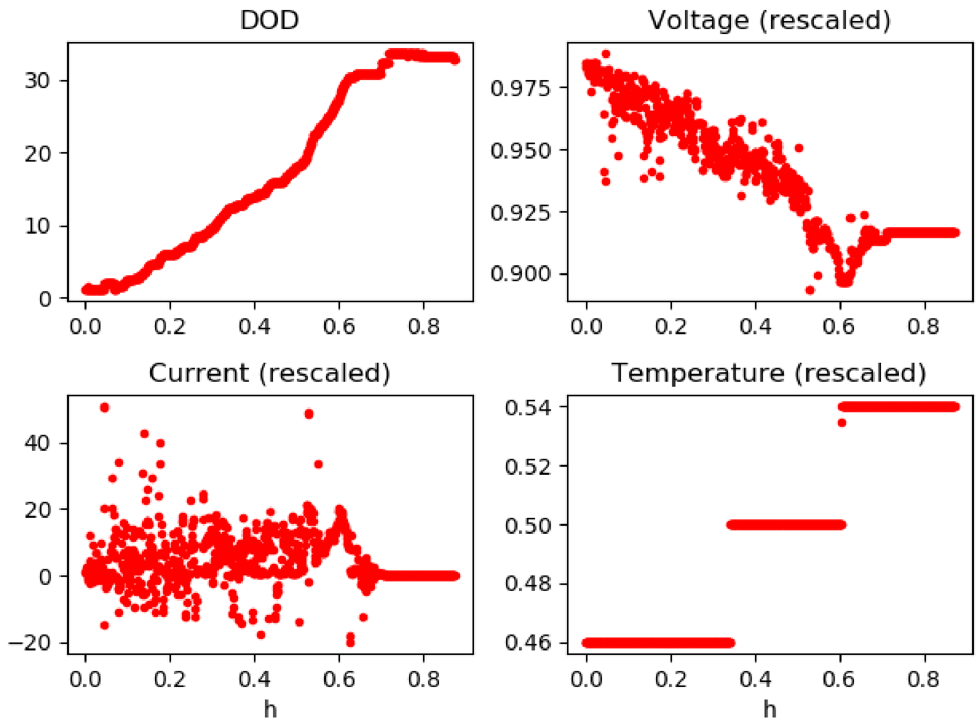


Figure 4.5: A sample of processed data.

5

Battery States Estimation and Energy Synthesis

In this chapter, the technical details and results are presented. The whole project obeys a pipeline shown in figure 5.1.

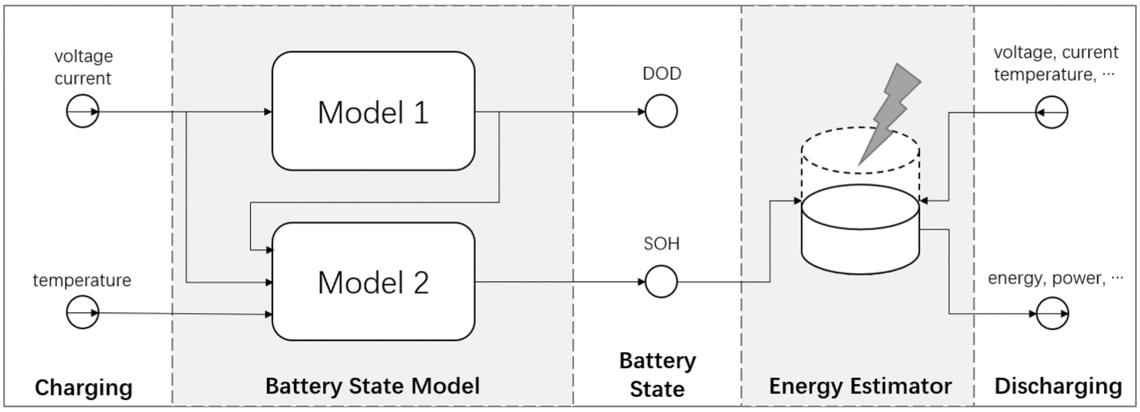


Figure 5.1: The pipeline consists of two parts. The front one is Battery State Model (BSM) which takes the electric and temperature signals from the charging process and outputs the states, DOD and SOH of the battery. The other part is an energy estimator that takes several signals from the discharging process and evaluates the energy-consuming situation.

5.1 Battery State Model

As discussed in previous chapters, the battery states are the basis to analyze energy and power. The battery states are mainly about the battery's health condition and the remaining quantity of charge, which are SOH (capacity) and SOC/DOD.

5.1.1 Estimation of DOD

As shown in figure 5.1, Model 1 is the one estimating the depth of discharge, which is a recurrent neural network with long short-term memory units. The structure of the RNN is shown in table 5.1.

Layer	Output Shape	Number of Parameters
Input	(s,t,2)	0
LSTM	(s,t,8)	384
Dense	(s,t,1)	9
		Total:393

Table 5.1: The structure of Model 1, RNN. s is the number of samples, and t is the length of a sample, which is about one thousand in this case. The input layer has two channels as the current and voltage, and the LSTM layer has eight channels.

The training and test are on 10 cars. Each car has about 1000 samples. The training set is formed by 9 cars out of 10 and the left one is the test set. Therefore, there are about 9000 samples in the training set which contains about 8000 for training and about 1000 for validation, and about 1000 samples for test. The training is conducted for 30 epochs on a GPU, Quadro M2000M. Each epoch costs about 8 seconds. The training process and test result are shown in figure 5.2.

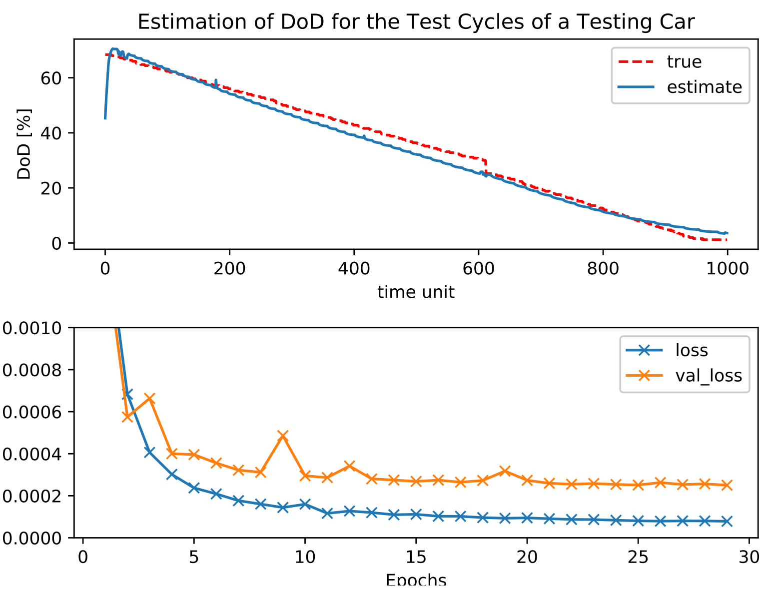


Figure 5.2: The training loss graph shows a bit overfitting but the test result is pretty good. The estimation can follow the true curve within an acceptable error.

To cross-test the RNN model, four more tests are implemented with a random test car. Then the mean squared errors between the estimation and true of each test are counted as seen in figure 5.3.

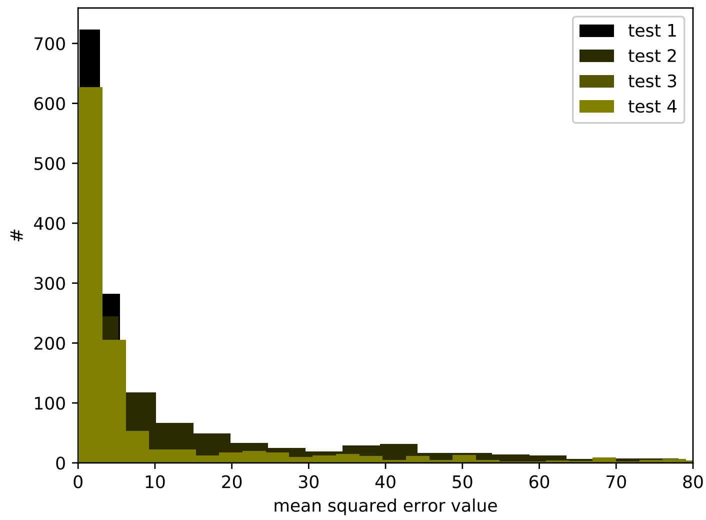


Figure 5.3: The statistics show that this RNN model is quite stable and mostly with small errors. Actually, most big errors are due to the wrong data but not the model.

5.1.2 Estimation of SOH

In figure 5.1, Model 2 is used to estimate the battery state of health, which is a nonlinear autoregressive network with exogenous inputs (NARX).
In this method, first a model is trained to predict the target value one step ahead. Then the predicted one-step ahead value is reused to produce the prediction of the next step. By iterating this procedure, one is able to obtain a multi-step prediction. The Data obtained after splitting the log files to charging and discharging cycles are converted to a single record by taking the mean of the selected features. The selected features are DOD, temperature, current, voltage and temperature with the label as capacity obtained from the logarithmic fitting of engineering SoH.
The training input data is the mean DOD, max DOD, min DOD, mean Temperature, mean voltage and mean current with the label as the Capacity. We use only the charging cycles since the current is almost constant during charging.
The training is performed on 22 cars and 1 car is used for testing. The cross-validation is done and the results for all the cars are shown here.
On the whole, there are about 430,000 cycles altogether. Each test has different cycles based on the usage of cars. The training is done on a GPU, Quadro M2000M. The $f(.)$ fitting is done using RFR, XGB and FFN. The FireTS module that supports sklearn interface is used to perform the training. The exogenous delay is set to 0 and the exogenous order is set to 5 based on experiments. The multi-step prediction is done directly and No future inputs are used in the prediction.
Results of one car is shown here. But four cars that are driven for longer cycles are chosen to illustrate the performance of the model as the prediction of the battery's health needs to be done for the long term. Note the final assessment is based on all the cars. The rest of the results are presented in the appendix.

Training Parameters	Values
n_estimators	100
criterion	MSE
bootstrap	True

Table 5.2: Parameters of RFR.

5.1.2.1 NARX + Random Forest Regressor

The default parameters of the Random forest regressor showed best when tested with different values. The number of estimators used is 100 for both random forest regressor and XGBoost for comparison. The table 5.2 shows some of the tuned parameters and their values that were used.

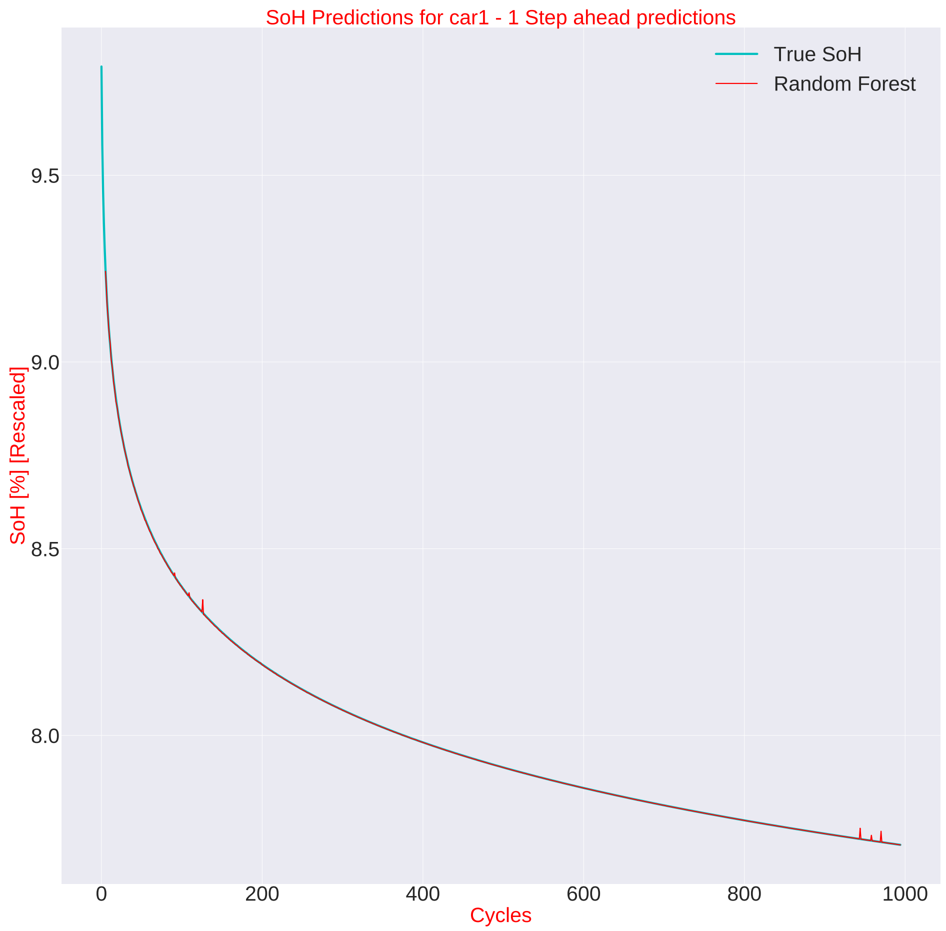


Figure 5.4: 1 Step predictions for Car 1 using Random Forest Regressor.

5.1.2.2 NARX +XGBoost

Due to time restriction, only few parameters were modified. The parameters chosen for the XGBoost are learning rate, $eta = 0.3$ and $n\_estimators = 100$.

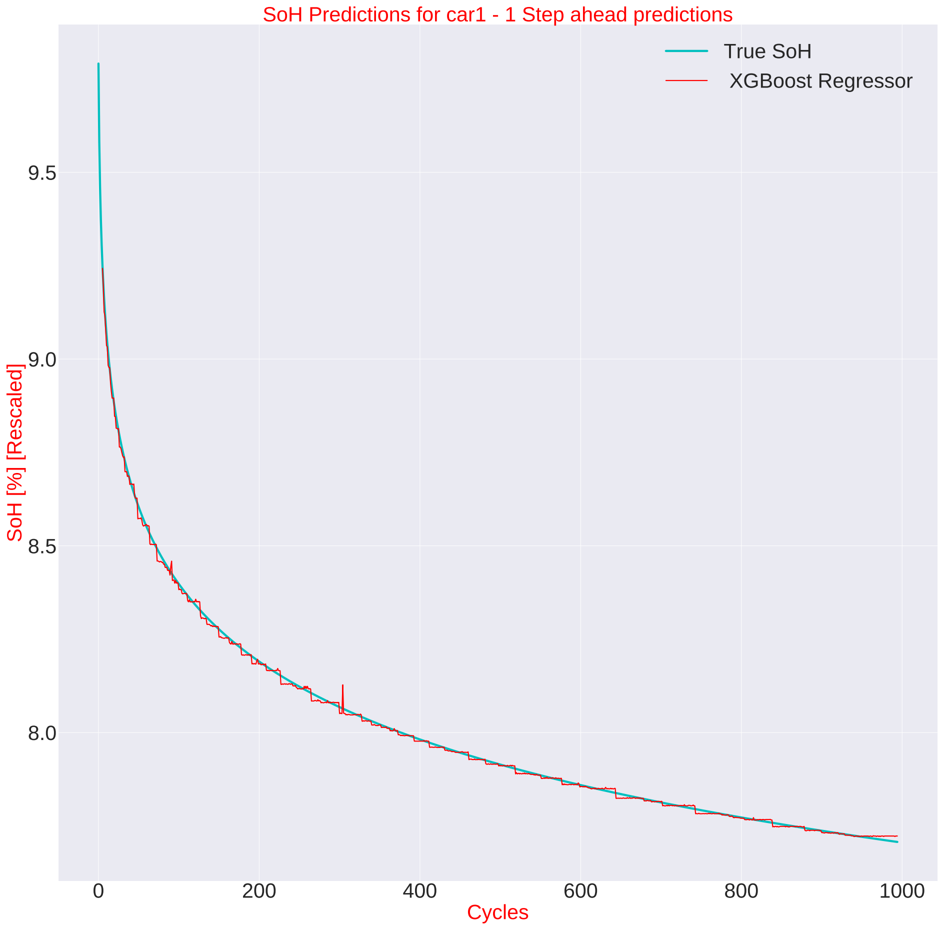


Figure 5.5: 1 Step predictions for Car 1 using XGBoost Regressor.

5.1.2.3 NARX + Feed Forward Network

A traditional feed forward network with 16 hidden layers was used and its architecture is shown in the table 5.3. The input layer has 6 channels as the mean DOD, min DOD, max DOD, mean current, mean temperature and mean voltage. The hidden layers have 16 and 32 neurons separately. Training parameters are shown in table 5.4. Other parameters are set to the default.

Layer	Output Shape	Number of Parameters
Input	(None,6)	0
Hidden Layer 1	(None,16)	96
Hidden Layer 2	(None,32)	544
		Total:673

Table 5.3: The structure of the model FNN.

Training Parameters	Values
Solver	Adam (0.001)
Activation Function	relu
#iterations	107
Training Loss	0.28
Loss Function	MSE
Tolerance	0.0001
Batch Size	200

Table 5.4: Training parameters of FFN.

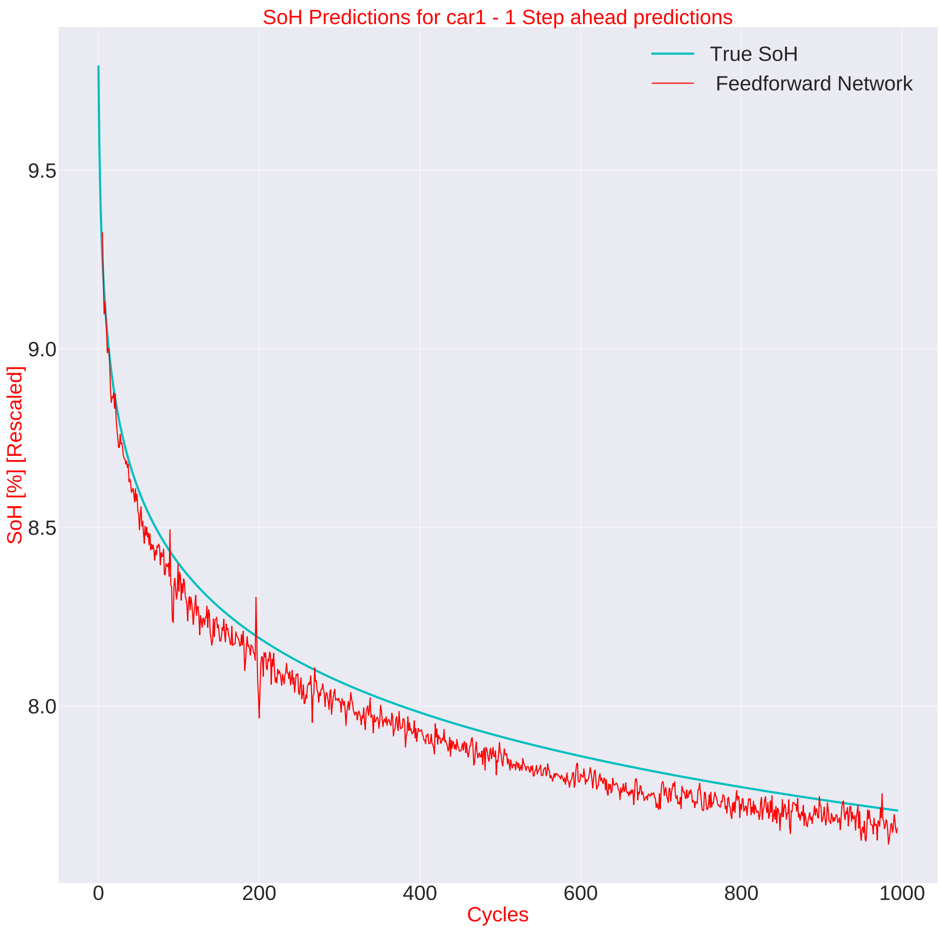


Figure 5.6: 1 step ahead predictions for Car 1 using Feed Forward Network.

5.1.2.4 NARX + Linear Regressor

The linear regressor is implemented with the only parameter $fit\_intercept = True$.

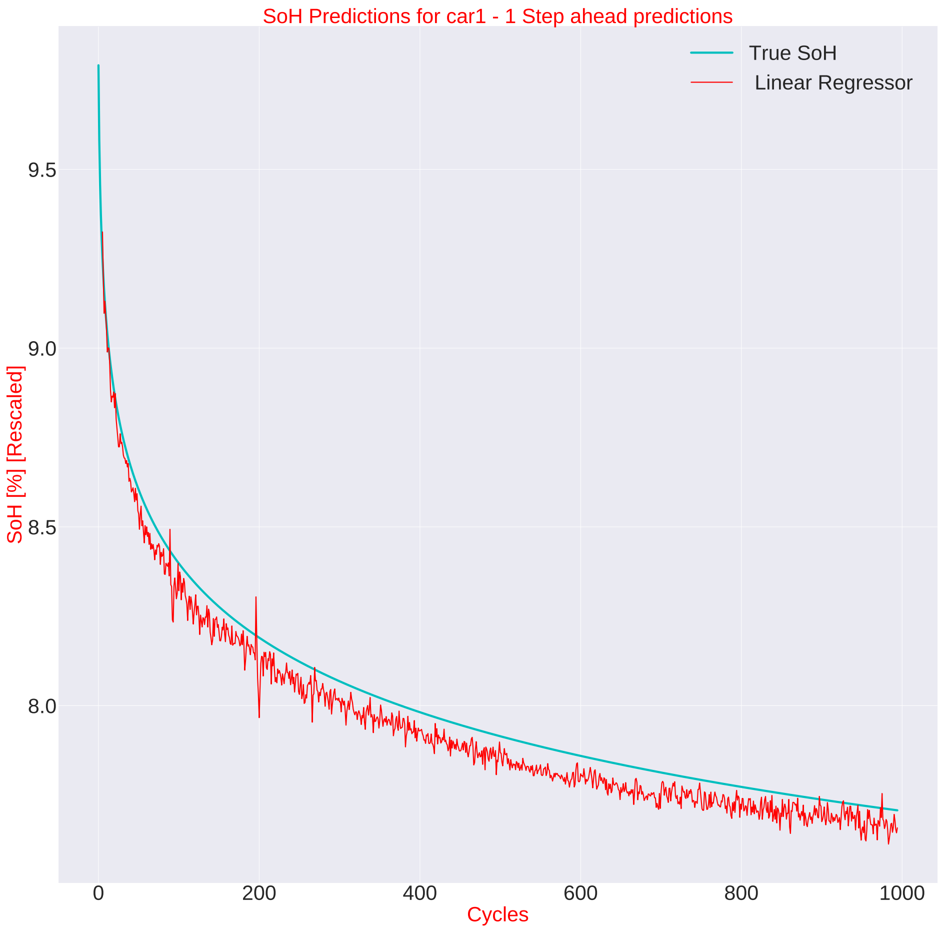


Figure 5.7: 1 step ahead predictions for Car 1 using Linear Regressor.

5.2 Energy Synthesis

To conclude the complete energy synthesis, an example would be helpful.
Assuming an electric car with an initial capacity 60 $A \cdot h$, which is 216000 Coulomb charge in total, after charging the battery state model has an estimation of 80% SOH and 50% SOC. The remaining charge is

$$216000 \times 80\% \times 50\% = 86400\ C \tag{5.1}$$

If the average or predictive discharging current is about 10 A, then the time to empty the battery is 2.4 h.

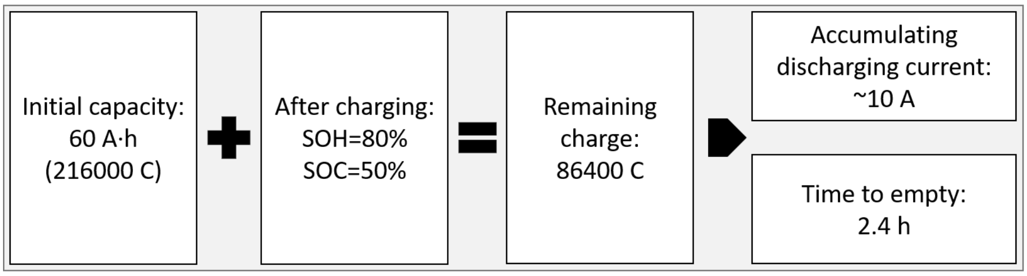


Figure 5.8: A easy example about how to use the pipeline. Note that every time the vehicle is charged, there would be an updated SOH.

6
Battery Degradation

NARX model is highly powerful in the prediction of time series. This approach has not been applied to any practical battery degradation problem. In this thesis, we can observe how decent predictions can be made when machine learning models are nested together with a non-linear auto-regressive exogenous model for real-time data.
The results from the prediction of the battery aging are presented here. The step-ahead predictions were made for 10, 50 and 100 steps ahead with each step denotes a charging cycle. Since the battery degradation is affected by driving behavior, it is assumed to be indirectly model through the exogenous input given to the NARX.
The four cars that were driven for longer cycles are chosen to illustrate the performance of the model as the prediction of the battery's health needs to be done for as farther as possible. Note the final assessment is based on all the cars.

6.1 Battery Degradation Prediction

6.1.1 NARX + Random Forest Regressor

The default parameters of the Random forest regressor showed best when tested with different values.
The number is estimators used is 100 for both random forest regressor and XGBoost for comparison. The table 5.2 shows some of the tuned parameters and their values that were used.

- **10 Steps Ahead Prediction**

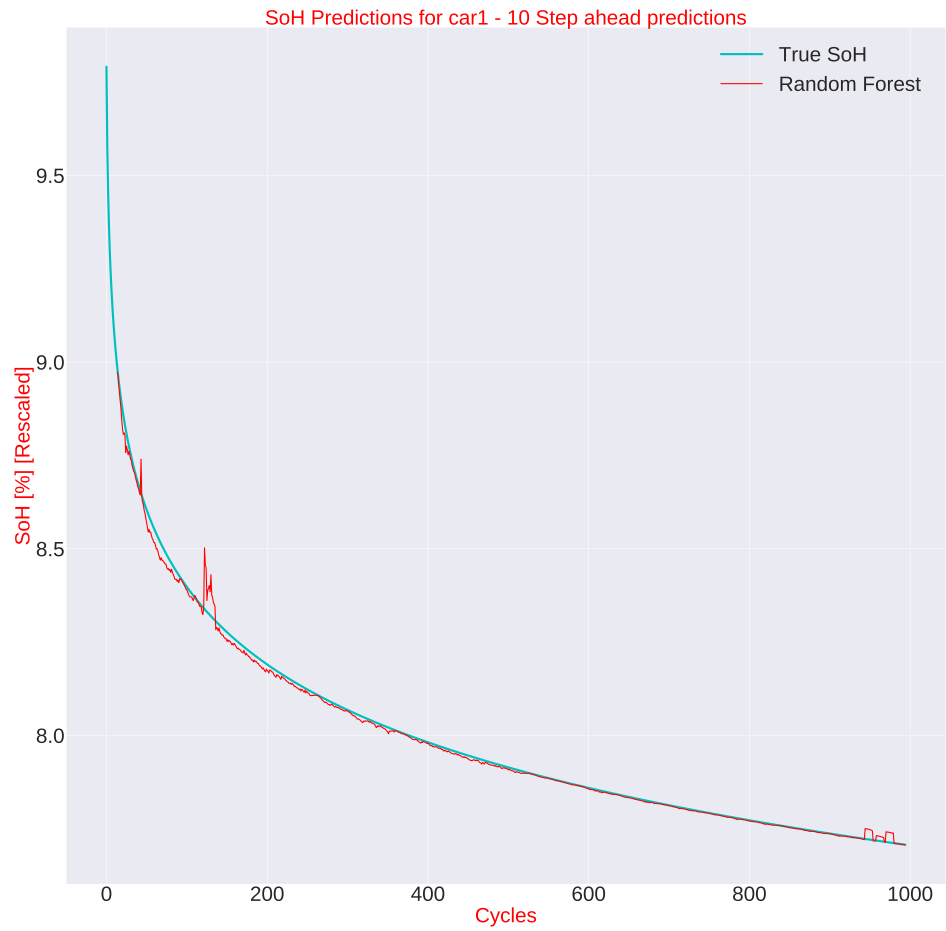


Figure 6.1: 10 steps ahead predictions for Car 1 using Random Forest Regressor.

- **50 Steps Ahead Prediction**

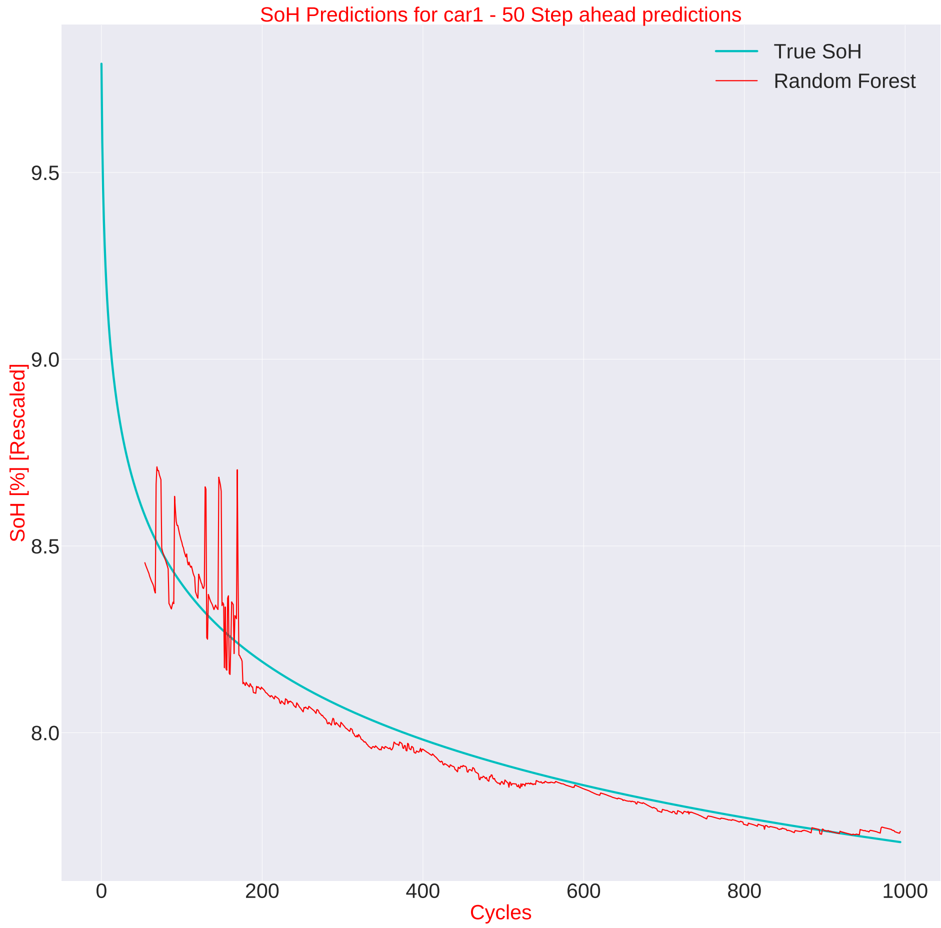


Figure 6.2: 50 steps ahead predictions using Random Forest Regressor.

- **100 Steps Ahead Prediction**

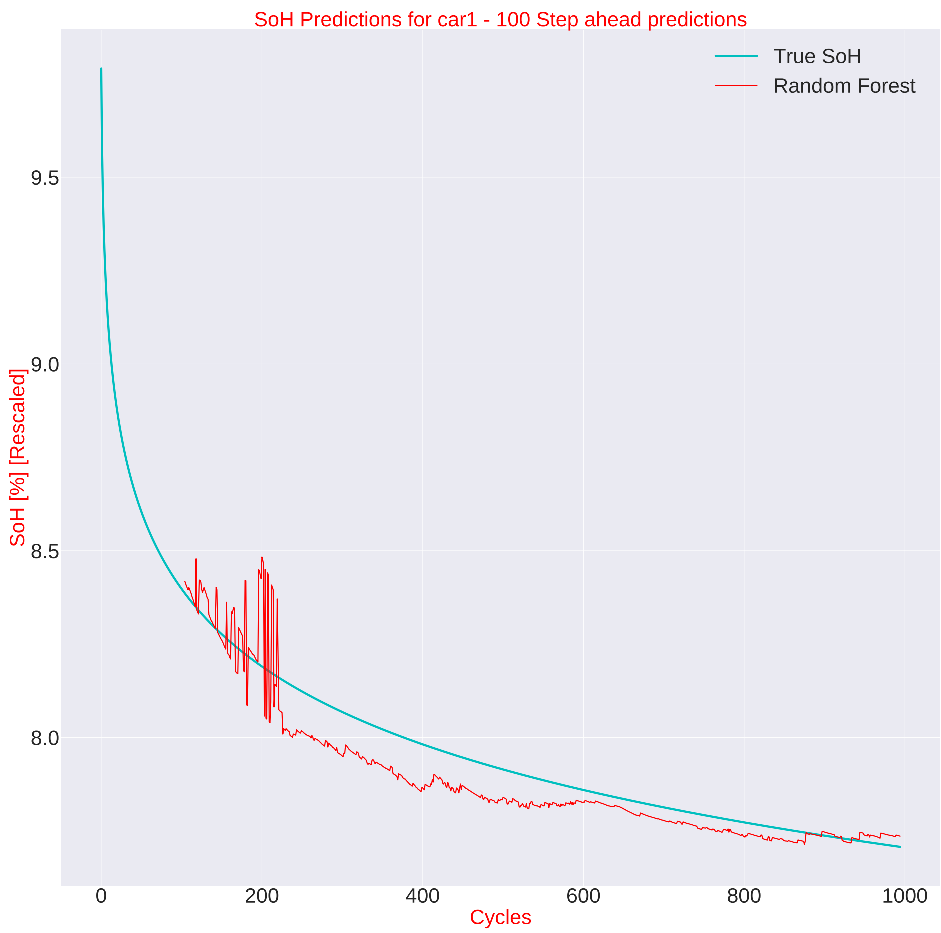


Figure 6.3: 100 steps ahead predictions for Car 1 using Random Forest Regressor.

6.1.2 NARX +XGBoost

After several testing, the parameters were chosen for the XGBoost regressor. There are a huge amount of parameters that can be altered and experimented with. Due to time restrictions, only few parameters were modified as the learning rate, $eta = 0.3$ and $n\_estimators = 100$.

- **10 Steps Ahead Prediction**

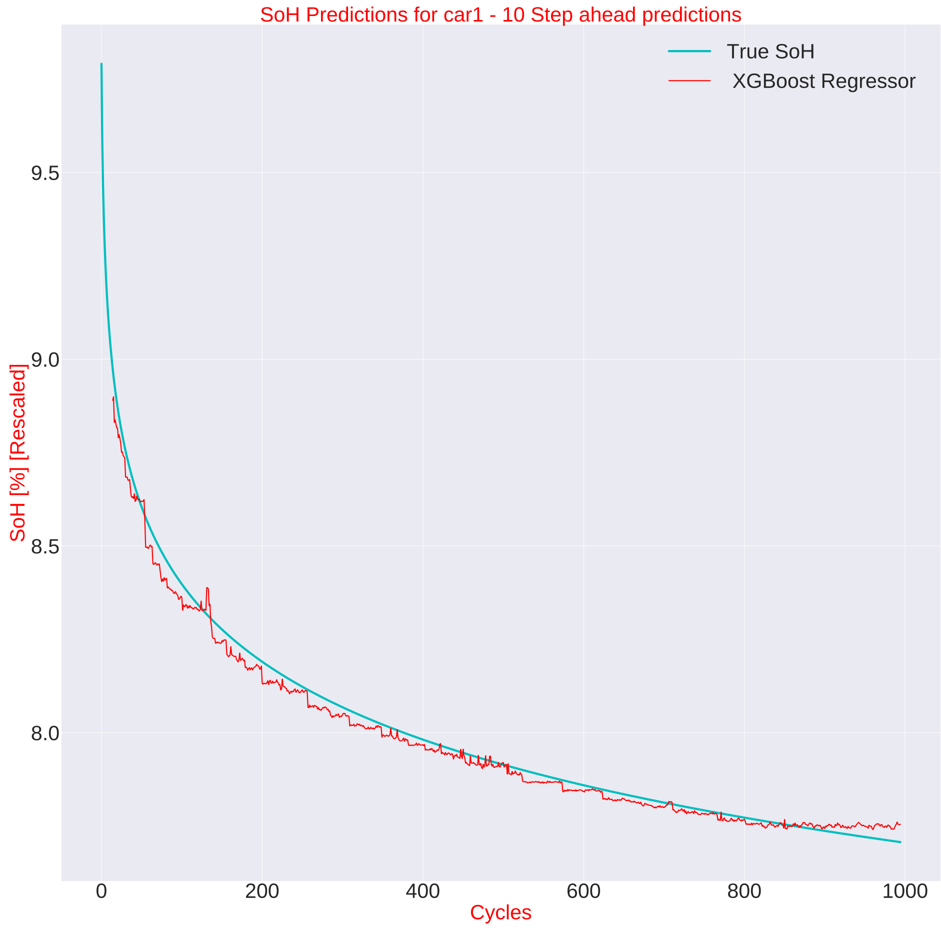


Figure 6.4: 10 steps ahead predictions for Car 1 using XGBoost Regressor.

- **50 Steps Ahead Prediction**

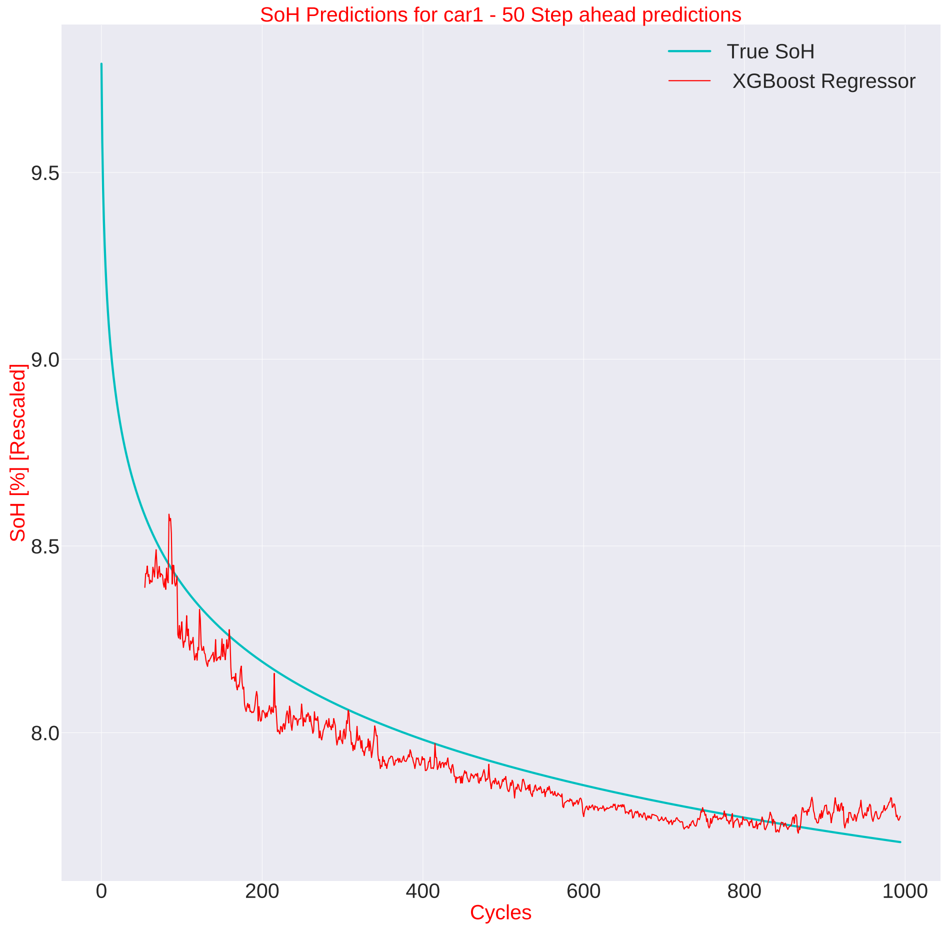


Figure 6.5: 50 steps ahead predictions for Car 1 using XGBoost Regressor.

- **100 Steps Ahead Prediction**

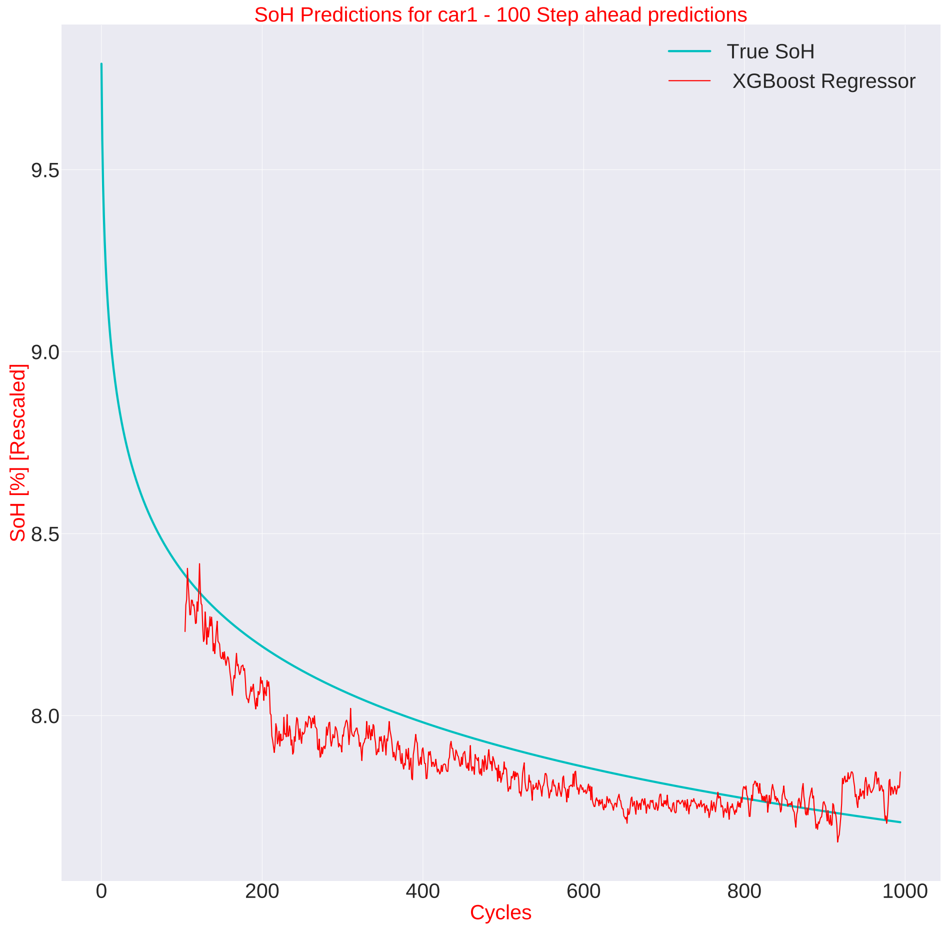


Figure 6.6: 100 steps ahead predictions for Car 1 using XGBoost Regressor.

6.1.3 NARX + Feed Forward Network

A traditional feed forward network with 16 hidden layers were used. The architecture of the FFN is shown in table 5.3. The training parameters are shown in table 5.4. The other parameters are set to the default.

- **10 Steps Ahead Prediction**

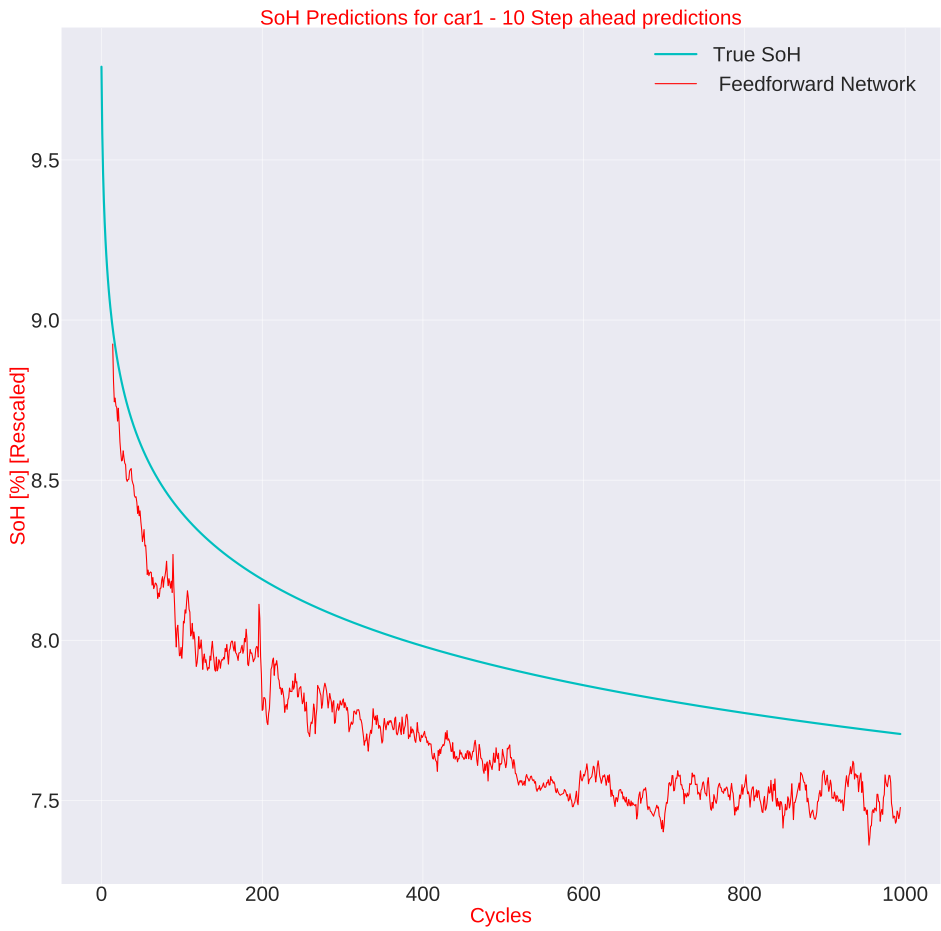


Figure 6.7: 10 steps ahead predictions for Car 1 using Feed Forward Network.

- **50 Steps Ahead Prediction**

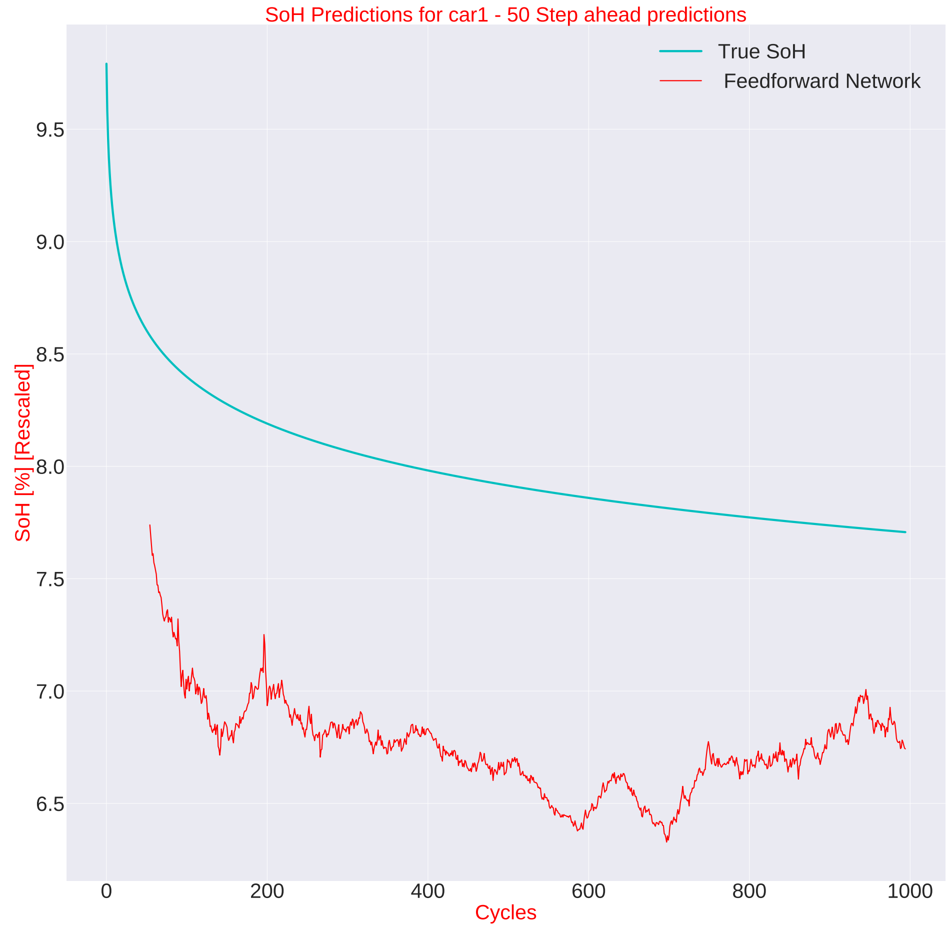


Figure 6.8: 50 steps ahead predictions for Car 1 using Feed Forward Network.

- **100 Steps Ahead Prediction**

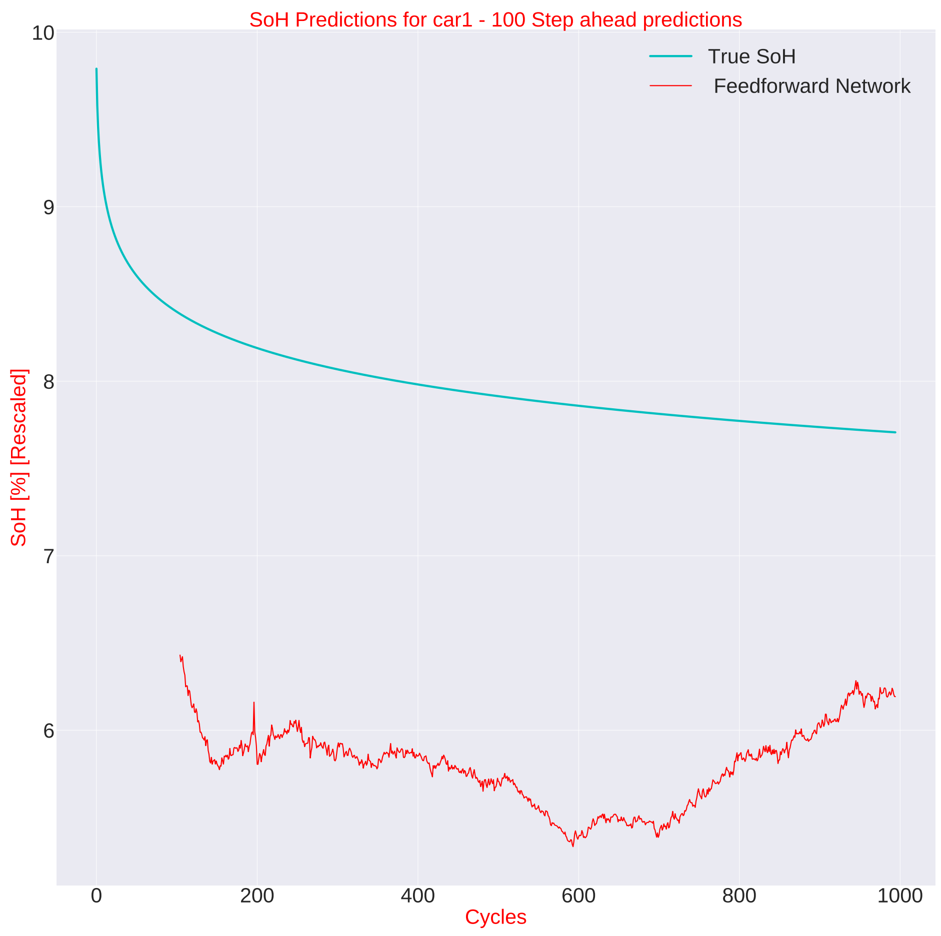


Figure 6.9: 100 steps ahead predictions for Car 1 using Feed Forward Network.

6.1.4 NARX + Linear Regressor

Linear Regressor is implemented with $fit\_intercept = True$.

- **10 Steps Ahead Prediction**

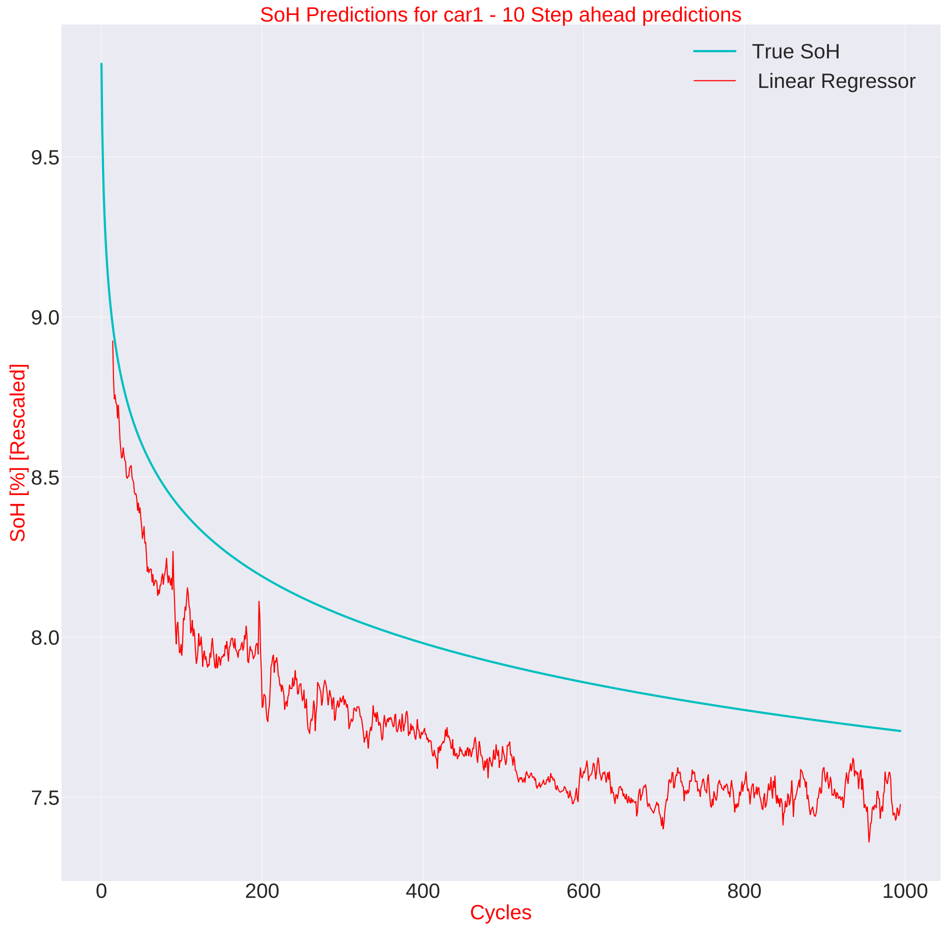


Figure 6.10: 10 steps ahead predictions using Linear Regressor for four different cars.

- **50 Steps Ahead Prediction**

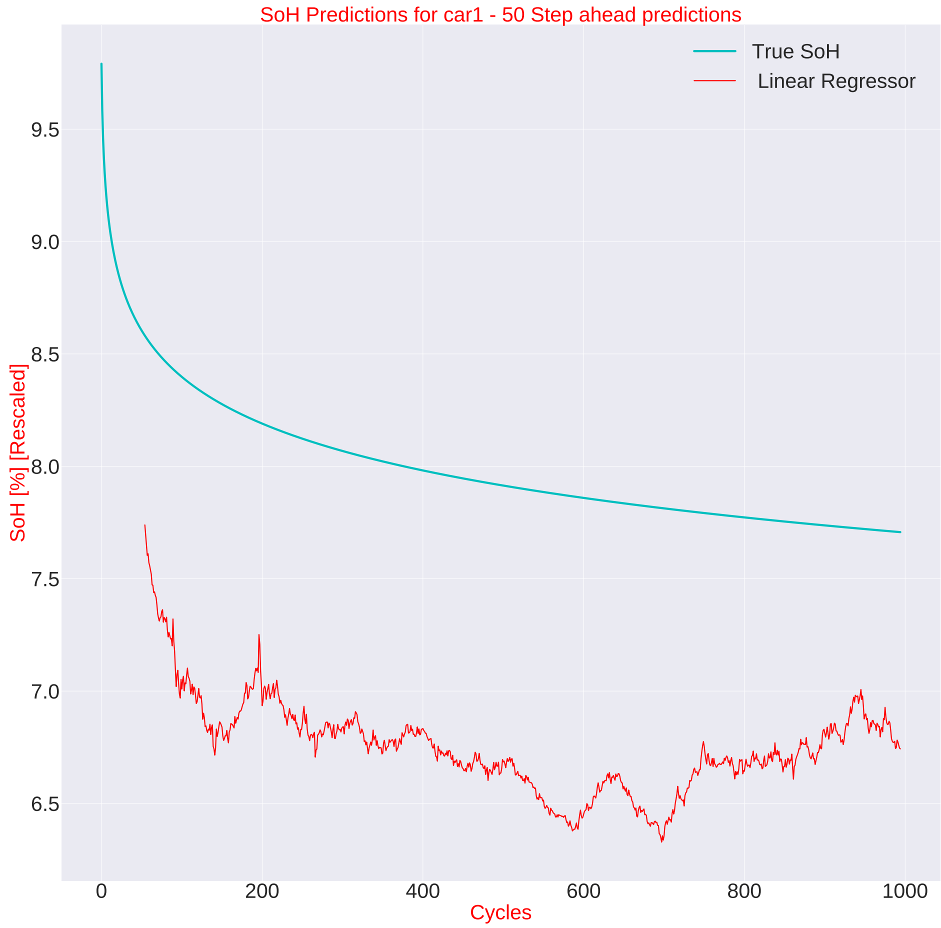


Figure 6.11: 50 steps ahead predictions using Linear Regressor for four different cars.

- **100 Steps Ahead Prediction**

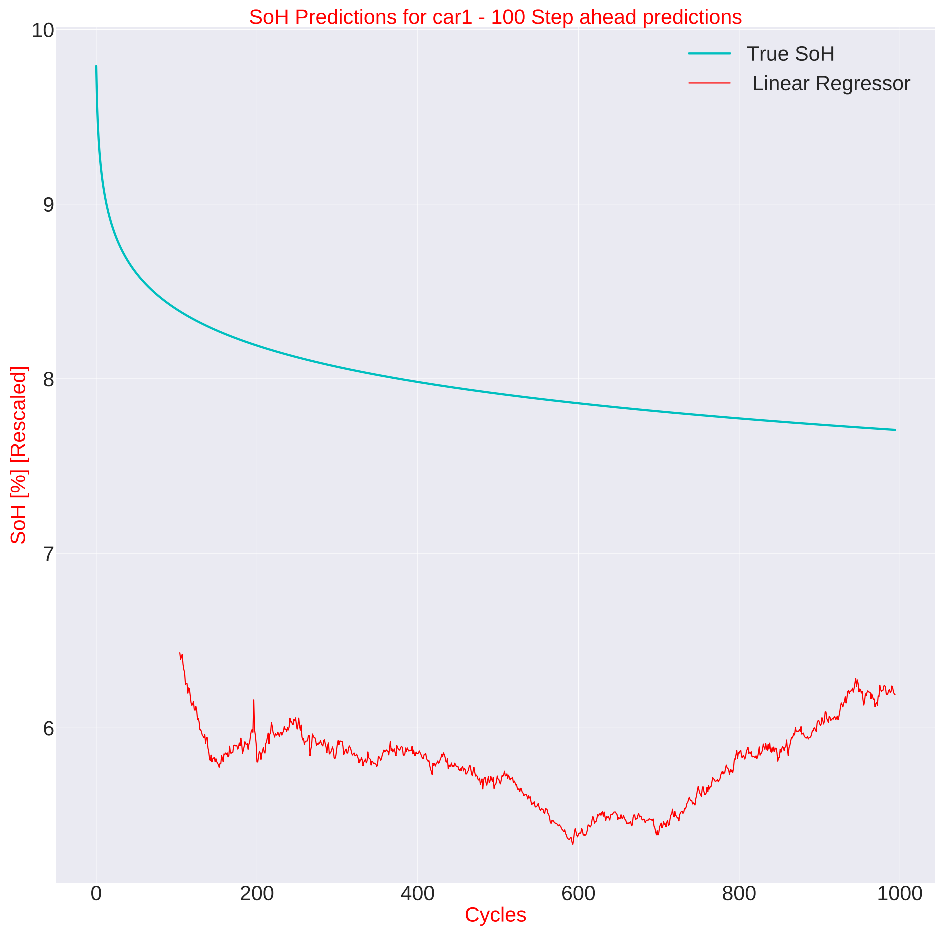


Figure 6.12: 100 steps ahead predictions using Linear Regressor for four different cars.

6.2 Evaluation of the Models

Validation and Evaluation of a Data Science Model provides more color to our hypothesis and helps evaluate different models that would provide better results against our data.
In this thesis, the models are evaluated based on Mean Square Error (MSE), R^2 score and variance and they are implemented using the scikit-learn package.

6.2.1 Mean Square Error

Mean square Error (MSE) is the average of the square of the errors. The larger the number the larger the error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6.1}$$

6.2.2 R^2 Score

R^2 Score varies between 0 and 100%. It is closely related to the MSE, but not the same.

$$R^2 = 1 - \frac{MSE(Model)}{MSE(Base)} \tag{6.2}$$

$$\frac{MSE(Model)}{MSE(Base)} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - \hat{y}_i)^2} \tag{6.3}$$

where MSE (model) is the MSE of the predictions against the actual values, and MSE (baseline) is the MSE of mean predictions against the actual values.
In other words, how good our regression model as compared to a very simple model that just predicts the mean value of target from the train set as predictions.

6.2.3 Variance

Variance is a measure of how far observed values differ from the average of predicted values, and the goal is to have a low value.

6.2.4 Cross-validation

Cross-validation is a model validation technique for assessing how the results of statistical analysis (model) will generalize to an independent data set. It is mainly used in prediction, and one wants to estimate how accurately a predictive model will perform in practice. The goal of cross-validation is to define a data set to test the model in the training phase (i.e. validation data set) in order to limit problems like overfitting, underfitting and get an insight on how the model will generalize to an independent dataset [37].
In this thesis, for evaluation, 1 fold cross-validation is used that is the prediction function was learned using 21 cars (folds), and the fold (1 car) left out was used for tests.
This metric helps us validate which the model which will perform best on unseen data.
The tabulation shows the 1 fold cross-validated evaluation of the models for their corresponding steps of prediction.
It can be observed that NARX along with XGBoost is good for long term predictions. The performance of the RFR and linear regressor along with NARX is almost the same for 5 and 10 steps ahead predictions. But in the 50 and 100 steps ahead predictions, Linear regressor is better than the random forest regressor and the

Model	MSE	R^2 Score	Variance
NARX + RFR	0.0001	0.9999	$4.06 * 10^{-16}$
NARX + XGBoost	0.004713	0.9981	0.0048
NARX + FNN	0.0382	0.9885	0.0086
NARX + LR	0.0025	0.9991	0.00087

Table 6.1: Model Evaluation - 1 step ahead prediction

Model	MSE	R^2 Score	Variance
NARX + RFR	0.1640	0.9574	17.0625
NARX + XGBoost	0.08607	0.9705	0.0930
NARX + FNN	2.1090	0.4492	0.4712
NARX + LR	0.1193	0.9590	0.0386

Table 6.2: Model Evaluation - 10 steps ahead prediction.

Model	MSE	R^2 Score	Variance
NARX + RFR	1.4176	0.6686	17.0625
NARX + XGBoost	0.7018	0.8277	3.3319
NARX + FNN	32.5093	-7.2936	29.8446
NARX + LR	1.6392	0.5600	1.2088

Table 6.3: Model Evaluation - 50 steps ahead prediction.

Model	MSE	R^2 Score	Variance
NARX + RFR	1.5265	0.6272	98.3473
NARX + XGBoost	0.6109	0.8783	3.0238
NARX + FNN	35.8976	-7.6927	55.4745
NARX + LR	1.8124	0.6713	1.3791

Table 6.4: Model Evaluation - 100 steps ahead prediction.

feed forward neural network. The XGBoost along with NARX performed the best when compared to the rest of the models. NARX + XGBoost has a variance of 3 approximately for 50 steps and 100 steps ahead predictions which could be due to the presence of outliers in the data.
In this thesis, we have shown that the NARX along with machine learning techniques can successfully use its output feedback loop to improve its predictive performance in complex time series prediction tasks.

7
Discussion and Conclusion

In this chapter, the whole project will be discussed and some conclusions are drawn. Some possible future work is mentioned as well.

7.1 Discussion

This project is aimed at the estimation and prediction of battery states of battery electric vehicles. The whole project is based on a quite reasonable hypothesis, which is the battery states such as SOC and SOH can be reflected in several signals like current, voltage and temperature.

The study is done with real-world big data of the dataset C30 from Volvo Cars. Analyzing the real-world data is the necessary step for applying machine learning method on battery management systems. At the same time, it is a huge challenge. The C30 dataset is collected from Volvo C30 battery electric cars in about 8 years so the battery decaying problem is clear to observe. To have data over a long time is the key to study battery degradation. The challenge, on the other hand, is that dealing with such big data requires a long time and great computing power. In the end, only a small part of the whole dataset is utilized for this reason.

The other issue of the data is caused by real-world impacts. The data is really confusing with a lot of wrong logging and corrupting noise. The preprocessing took a long time as well.

After cleaning the data, the next step is to find the right machine learning model. According to the features of our problem, neural networks and regression models are suitable. Recurrent neural networks are used to estimate the depth of discharge after every charging process. Nonlinear autoregressive networks with exogenous input have a good performance in estimating and predicting the state of health decaying.

The above procedure can be concluded as a pipeline shown in figure 5.1. A simplified energy synthesis is done according to the pipeline. Although the battery state model is not complicated yet, final results show a good state.

7.2 Conclusion

We can see in both DOD estimation and SOH estimation the evaluations of the models are good, which means machine learning methods have the potential to be utilized in battery states estimation of battery electric vehicles.

In spite of some small errors, the total performance is satisfying. The error in DOD

estimation is mainly within 5%, which meets the acceptable error. There are some exceptions that show big errors. To solve the issue, more data processing should be implemented to exclude the wrong data. Meanwhile, SOH estimation and prediction based on engineered SOH labels are really accurate in a long period. This is very important for EVs to evaluate the battery health condition.
All in all, machine learning methods are capable to handle battery states estimation and battery degradation prediction problems in BEVs. This project is meaningful and pioneering for further study.

7.3 Future development

To face complex situations and high demands on the battery of BEVs, an intelligent battery adaptive management system must be developed. Machine learning methods are really strong tools to achieve the goal.
This project is just the start of applying machine learning methods on battery management systems. Due to limited time and resource, only some machine learning methods with simplified structures are tried on a subset of the whole dataset. Therefore, more work can be done in the future from several aspects.
Firstly, more data should be analyzed. To get a universal model and a general evaluation on the model, the model should be trained and tested on a dataset of sufficient big size. With stronger computing power and enough time in the future, the actual involved data should be so much more than the one used in this project.
Secondly, more models should be introduced and with more complicated architectures. In this way, complex battery behaviors are able to be understood.
This project is a pre-validation of applying machine learning methods on battery management problem. With more work being done and more resources involving in the future, we believe the realization of an intelligent battery management system can be achieved soon.

Bibliography

[1] P. N. J. Vetter, K.-C. M. J. B. M. W. M. W.-M. C. V. M.R. Wagner, C. Veit, and A. Hammouche, “Ageing mechanisms in lithium-ion batteries,” in *Journal of Power Sources*, Orlando, USA, 2005, pp. 269–281.

[2] V. L. M.A. Danzer and F. M. 14, “Aging of lithium-ion batteries for electric vehicles,” in *Advances in Battery Technologies for Electric Vehicles*, 2005, p. 359–387.

[3] C. Sulzberger, “An early road warrior: Electric vehicles in the early years of the automobile,” *Power and Energy Magazine, IEEE*, vol. 2, pp. 66 – 71, 06 2004.

[4] J. L. Schnell, V. Naik, L. W. Horowitz, F. Paulot, P. Ginoux, M. Zhao, and D. E. Horton, “Air quality impacts from the electrification of light-duty passenger vehicles in the united states,” *Atmospheric Environment*, vol. 208, pp. 95 – 102, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1352231019302183

[5] A. Vezzini, “15 - lithium-ion battery management,” in *Lithium-Ion Batteries*, G. Pistoia, Ed. Amsterdam: Elsevier, 2014, pp. 345 – 360. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780444595133000157

[6] A. Ulvestad, “A Brief Review of Current Lithium Ion Battery Technology and Potential Solid State Battery Technologies,” *arXiv e-prints*, p. arXiv:1803.04317, Mar 2018.

[7] B. Nykvist and M. Nilsson, “Rapidly falling costs of battery packs for electric vehicles,” *Nature Climate Change*, vol. 5, pp. 329–332, 03 2015.

[8] D. Deng, “Li-ion batteries: basics, progress, and challenges,” *Energy Science Engineering*, vol. 3, 09 2015.

[9] J. Tarascon and M. Armand, “Issues and challenges facing rechargeable lithium batteries,” *Nature*, vol. 414, pp. 359–67, 12 2001.

[10] Y. K. H. D. Abruña and J. C. Henderson, “Batteries and electrochemical capacitors,” in *Physics Today*, 2008, p. 43–47.

[11] C. F. Wladislaw Waag and D. U. Sauer, “Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicl,” in *Journal of Power Sources*, 2014, p. 321 – 339.

[12] J. L. J. H. a. M. O. Languang Lu, Xuebing Han, “A review on the key issues for lithium-ion battery management in electric vehicles,” in *Journal of Power Sources*, 2013, p. 226:272 – 288.

[13] L. C. Y. L. W. Z. H. W. Ruifeng Zhang, Baohua Li and W. Wang, “State of the art of lithium-ion battery soc estimation for electrical vehicles.” in *Energy*, 2018, pp. 11, 1820.

[14] D. S. D. Nejad, S.; Gladwin, "A systematic review of lumped-parameter equivalent circuit models for real-time estimation of lithium-ion battery states." in *J. Power Sources*, 2016, pp. 316, 183–196.

[15] X. R. F. J. He, H., "Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach." in *Energies*, 2011, pp. 4, 582–598.

[16] W. Z. X. J. Xing S., Chen S., "Unifying electrical engineering and electronics engineering: Proceedings of the 2012 international conference on electrical and electronics engineering. in lecture notes in electrical engineering." in *Springer*, 2014, p. Volume 238.

[17] R. H. H. Sun, F.; Xiong, "A systematic state-of-charge estimation framework for multi-cell battery pack in electric vehicles using bias correction technique." in *Energy*, 2016, pp. 162, 1399–1409.

[18] C. Z. Yuxiang Zhang and S. Zhu, "State-of-charge estimation of li-ion batteries based on a hybrid model using nonlinear autoregressive exogenous neural networks," in *IEEE PES Asia-Pacific Power and Energy Engineering Conference*, 2018, p. 772–777.

[19] L. M. H. A. M. Hannan, M.A., "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations." in *Energy*, 2017, pp. 78, 834–854.

[20] X. Hu, "Remaining capacity estimation of lithium-ion batteries based on the constant voltage charging profile," 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6034863/

[21] C. Zhang, J. Jiang, L. Zhang, S. Liu, L. Wang, and P. C. Loh, "A generalized soc-ocv model for lithium-ion batteries and the soc estimation for lnmco battery," *Energies*, vol. 9, no. 11, 2016. [Online]. Available: https://www.mdpi.com/1996-1073/9/11/900

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*, Aug 2006.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9(8), pp. 1735–1780, 1997.

[24] J. M. P. Junior and G. A. Barreto, "Slong-term time series prediction with the narx network: An empirical evaluation." in *Elsevier Science*, 2007.

[25] C. L. G. H. T. Siegelmann, B. G. Horne, "Computational capabilities of recurrent narx neural networks," in *IEEE Transactions On Systems, Man, and Cybernetics*, 1997.

[26] G. Drakos, "Random forest regression model explained in depth," 2019. [Online]. Available: https://gdcoder.com/random-forest-regressor-explained-in-depth/

[27] A. SINGH, "A comprehensive guide to ensemble learning," 201. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

[28] Y. courses, "Linear regression," 1997. [Online]. Available: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

[29] VolvoCars, "Welcome to volvocars-pr.com," 2011. [Online]. Available: Media.volvocars.com

[30] G. C. Congress, "Brusa supplying complete electric drivetrain for at least 250 volvo c30 electrics," 2013. [Online]. Available: https://www.greencarcongress.com/2011/09/brusa-20110921.html

[31] ——, "Volvo car corporation begins production of the volvo c30 electric for leasing customers in europe," 2011. [Online]. Available: https://www.greencarcongress.com/2011/06/c30-20110608.html

[32] ——, "Enerdel secures second order to supply li-ion battery pack for volvo c30 electric car," 2012. [Online]. Available: https://www.greencarcongress.com/2012/08/enerdel-20120827.html

[33] VolvoCars, "International – volvo cars," 2015. [Online]. Available: Media.volvocars.com

[34] G. C. Congress, "Volvo c30 electric wins 'green car of the year' in china as 100-day test drives start," 2012. [Online]. Available: https://www.greencarcongress.com/2011/03/c30-20110328.html#tp

[35] G. C. Journal, "Green car's top 5 vehicles that envision the road ahead," 2011. [Online]. Available: http://www.greencar.com/articles/green-cars-top-5-vehicles-envision-road-ahead.php

[36] A. Green, "Volvo c30 electric equipped with three climate systems; bioethanol heater," 2011. [Online]. Available: https://www.autobloggreen.com/

[37] G. Drakos, "Cross-validation," 2018. [Online]. Available: https://towardsdatascience.com/cross-validation-70289113a072

A

Appendix 1

A.1 Estimation of SOH

A.1.1 NARX + Random Forest Regressor

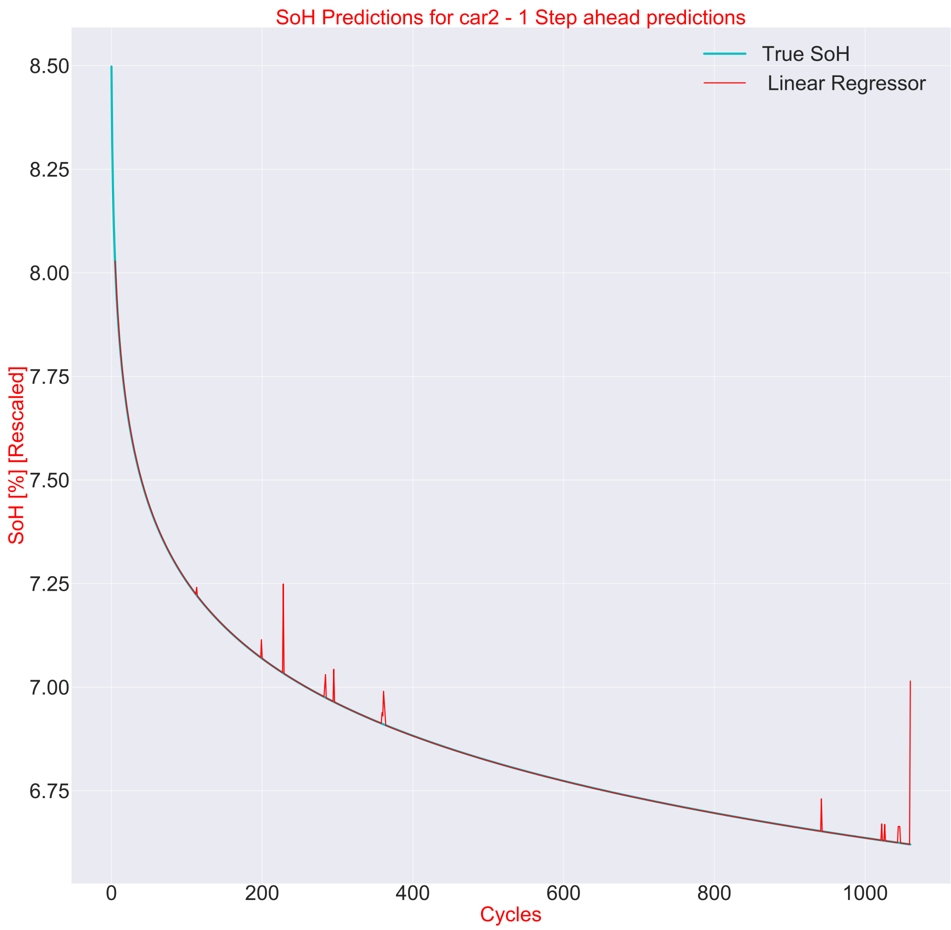


Figure A.1: 1 Step predictions for Car 2 using Random Forest Regressor

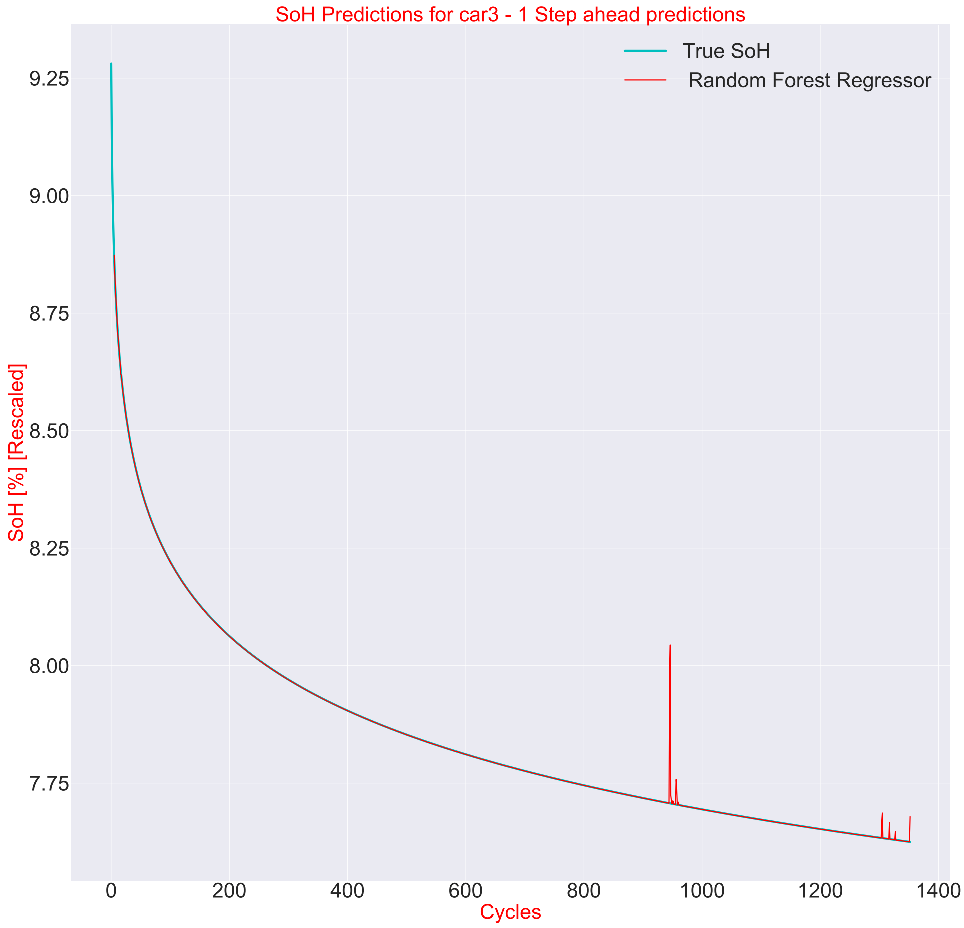


Figure A.2: 1 Step predictions for Car 3 using Random Forest Regressor

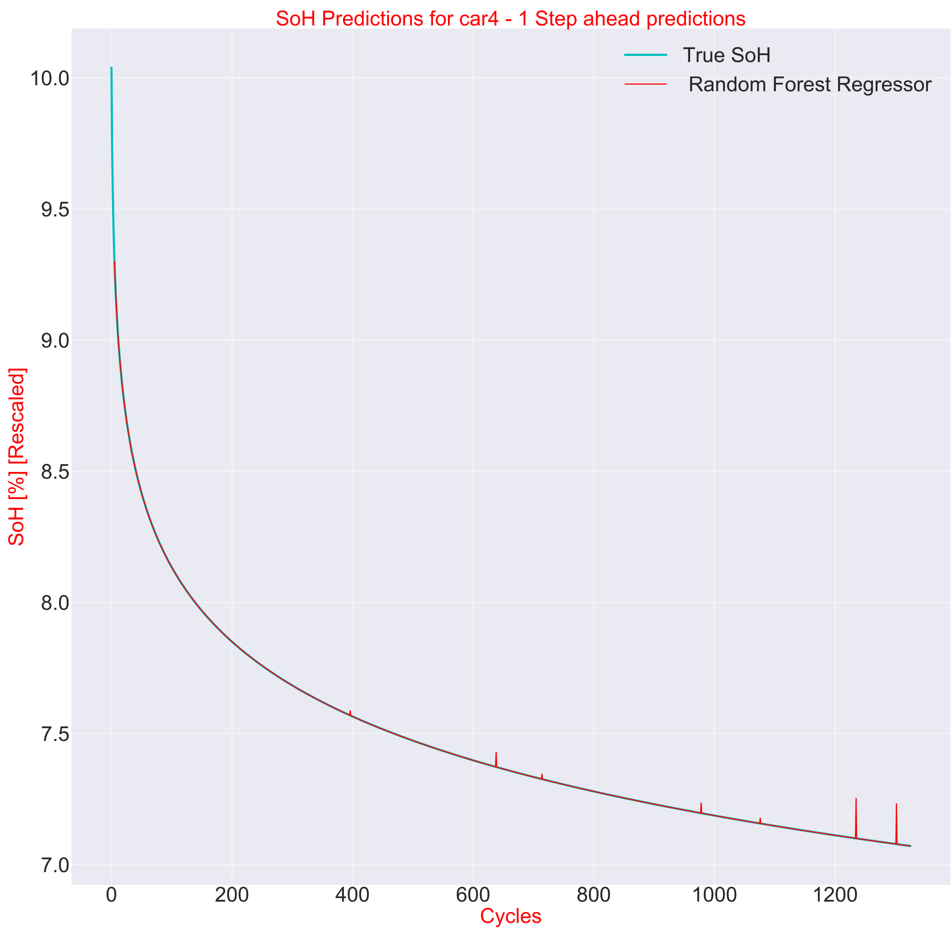


Figure A.3: 1 Step predictions for Car 4 using Random Forest Regressor

A.1.2 NARX + XGBoost

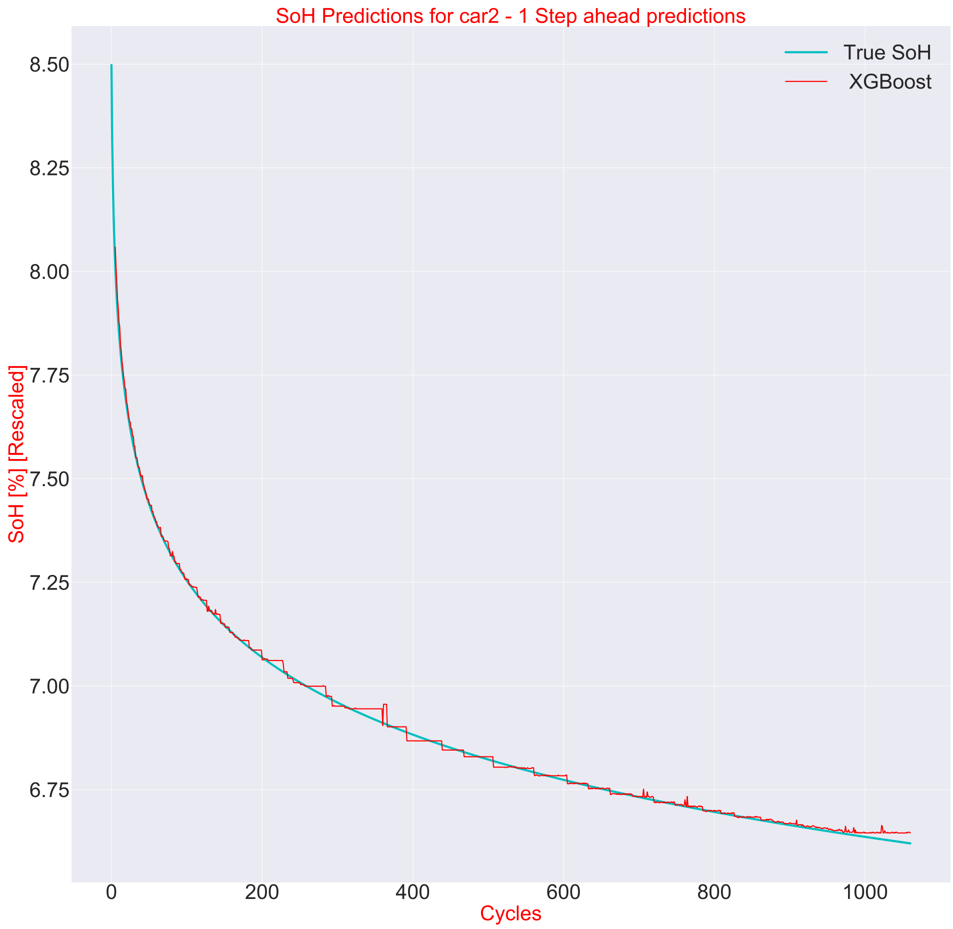


Figure A.4: 1 Step predictions for Car 2 using XGBoost Regressor

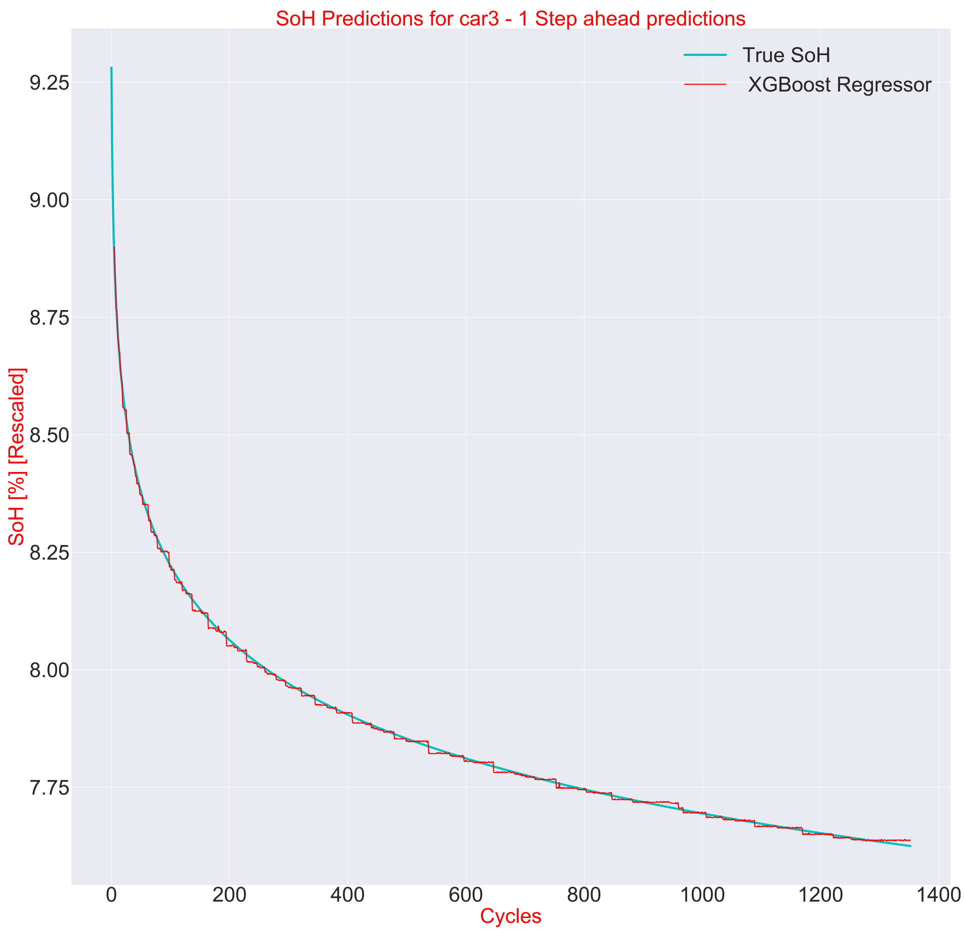


Figure A.5: 1 Step predictions for Car 3 using XGBoost Regressor

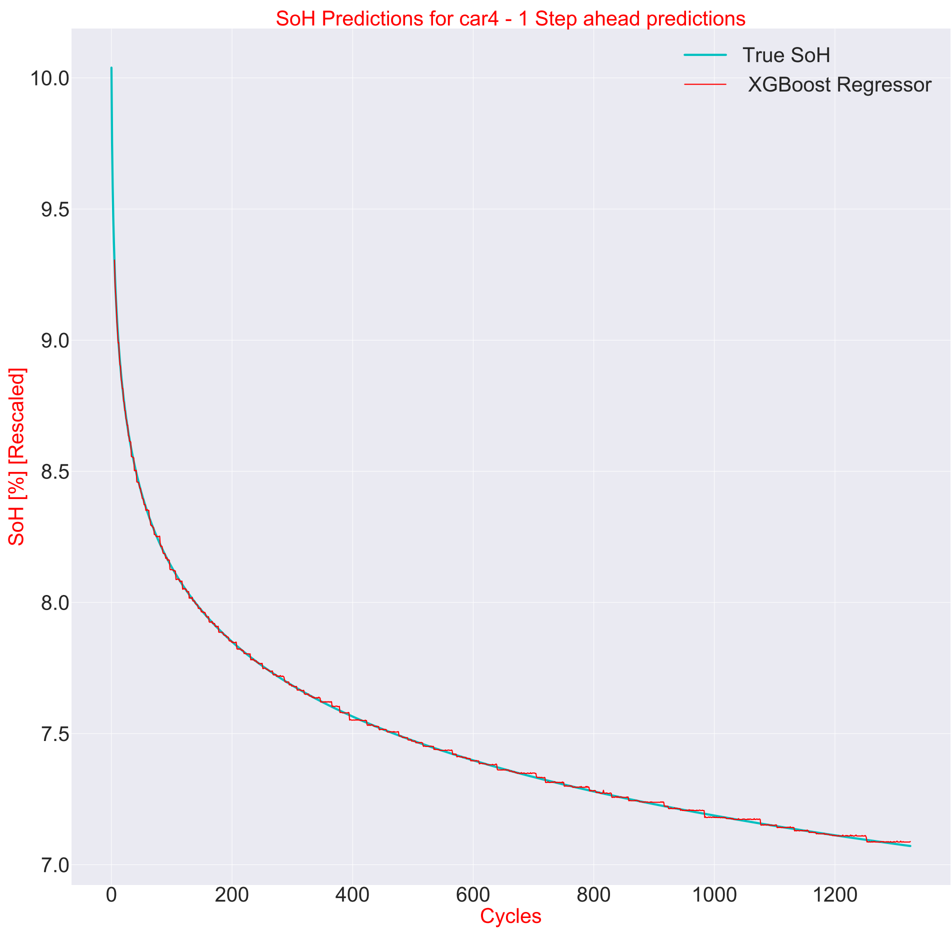


Figure A.6: 1 Step predictions for Car 4 using XGBoost Regressor

A.1.3 NARX + Feed Forward Network

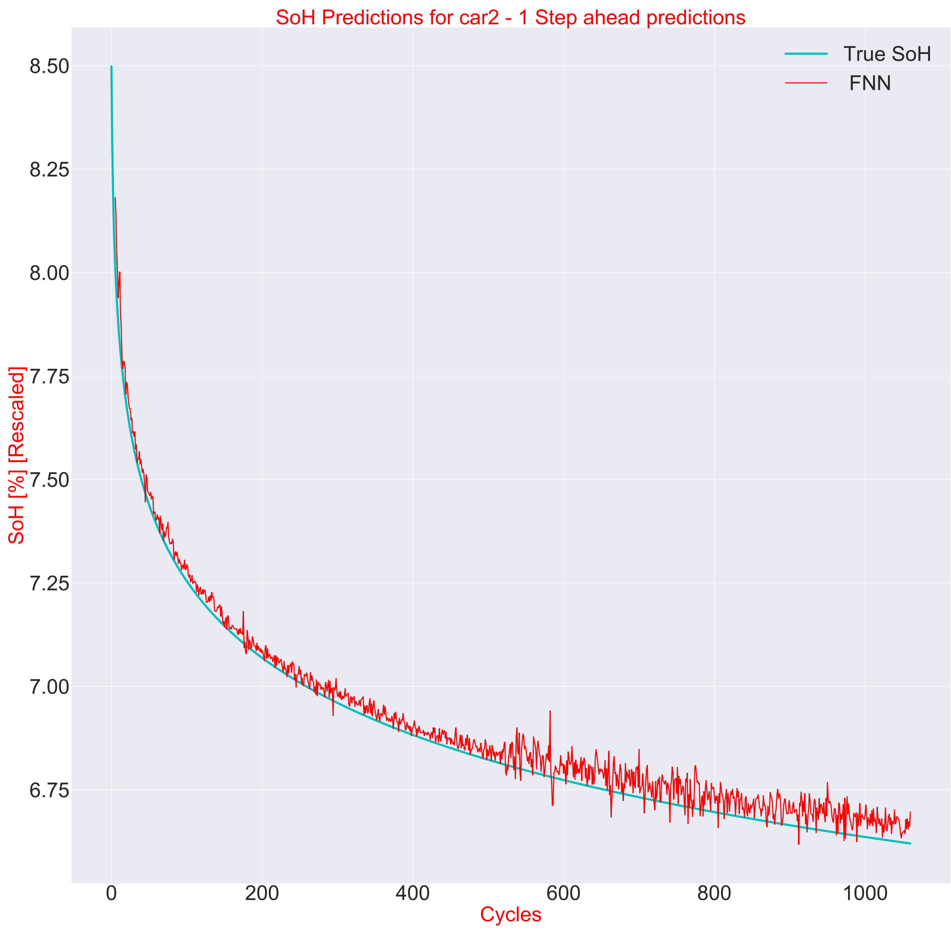


Figure A.7: 1 Step predictions for Car 2 using Feed Forward Network

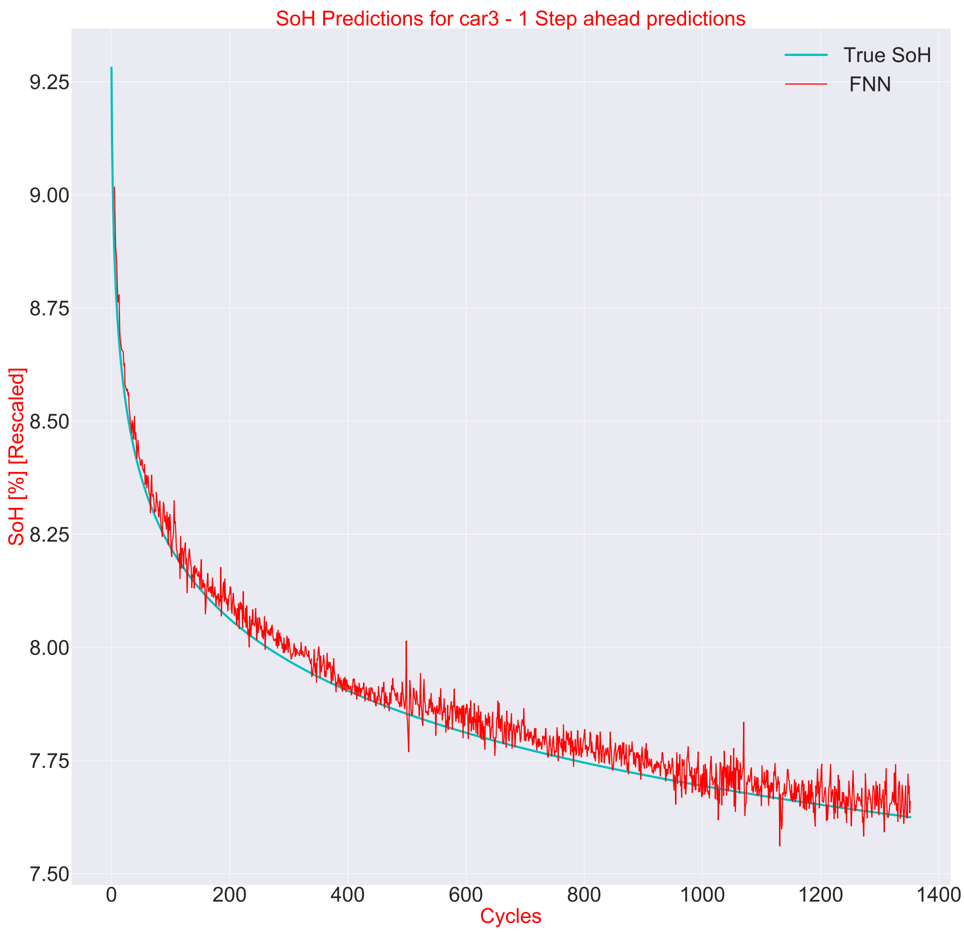


Figure A.8: 1 Step predictions for Car 3 using Feed Forward Network

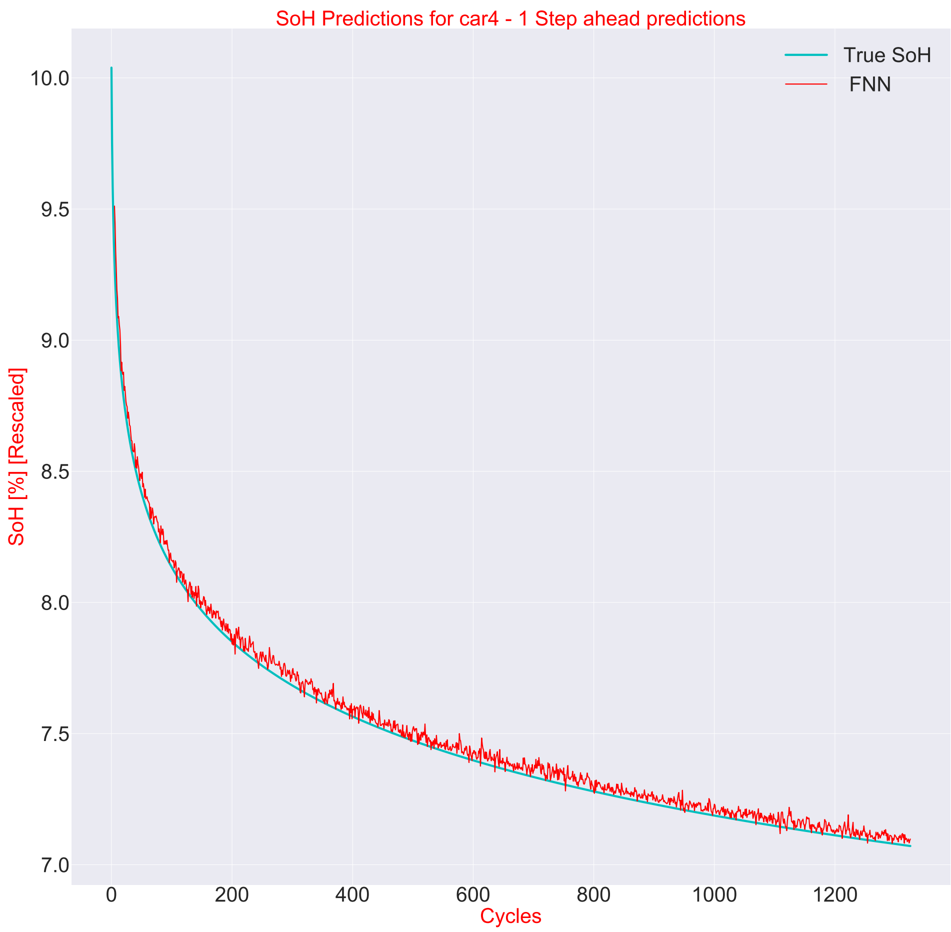


Figure A.9: 1 Step predictions for Car 4 using Feed Forward Network

A.1.4 NARX + Linear Regressor

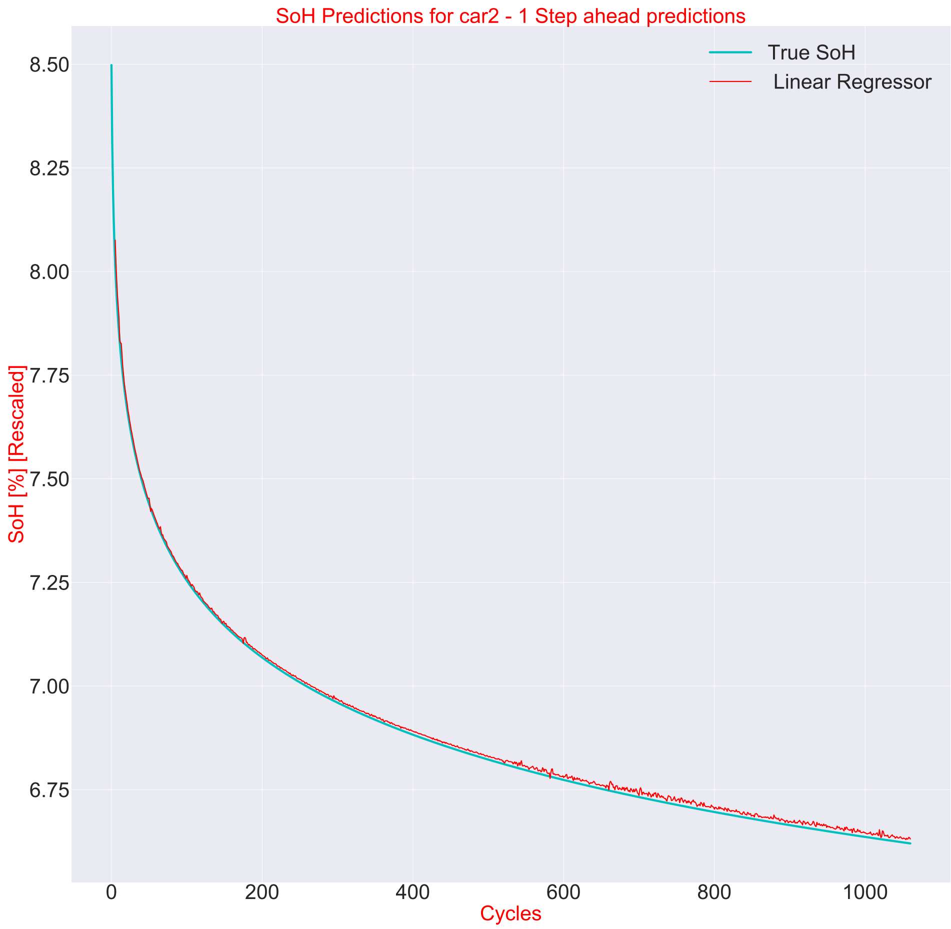


Figure A.10: 1 Step predictions for Car 2 using Linear Regressor

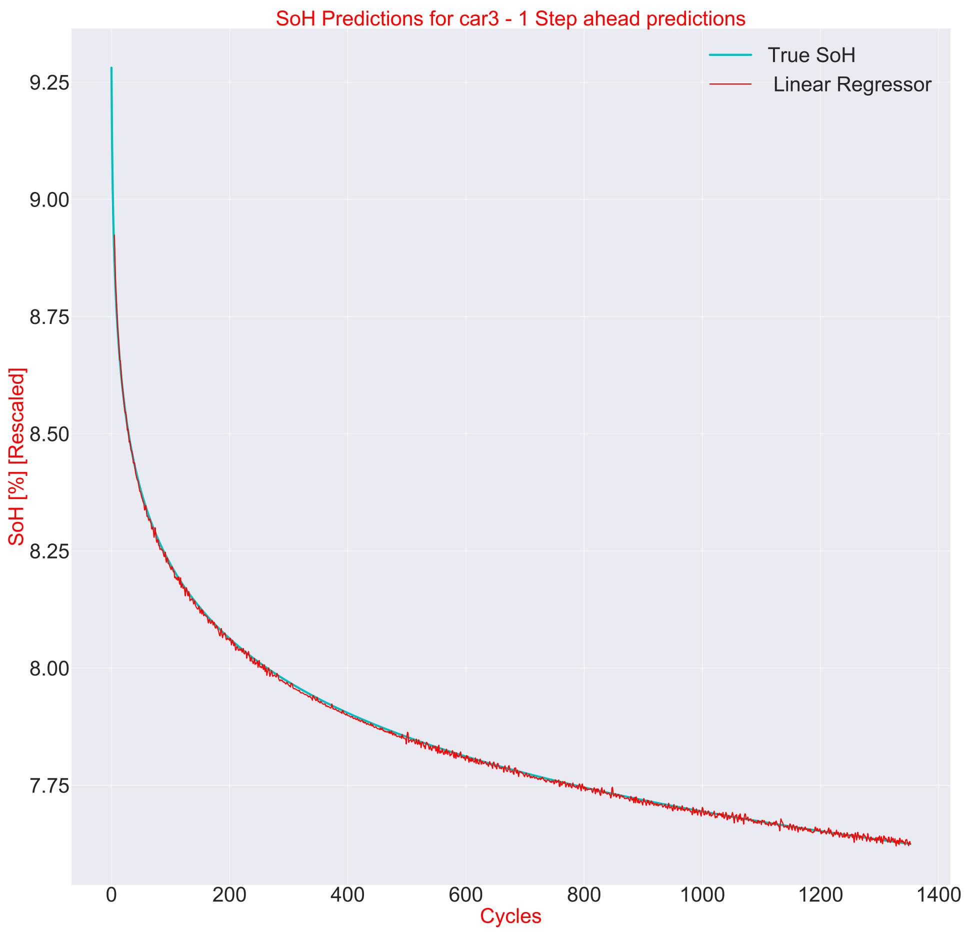


Figure A.11: 1 Step predictions for Car 3 using Linear Regressor

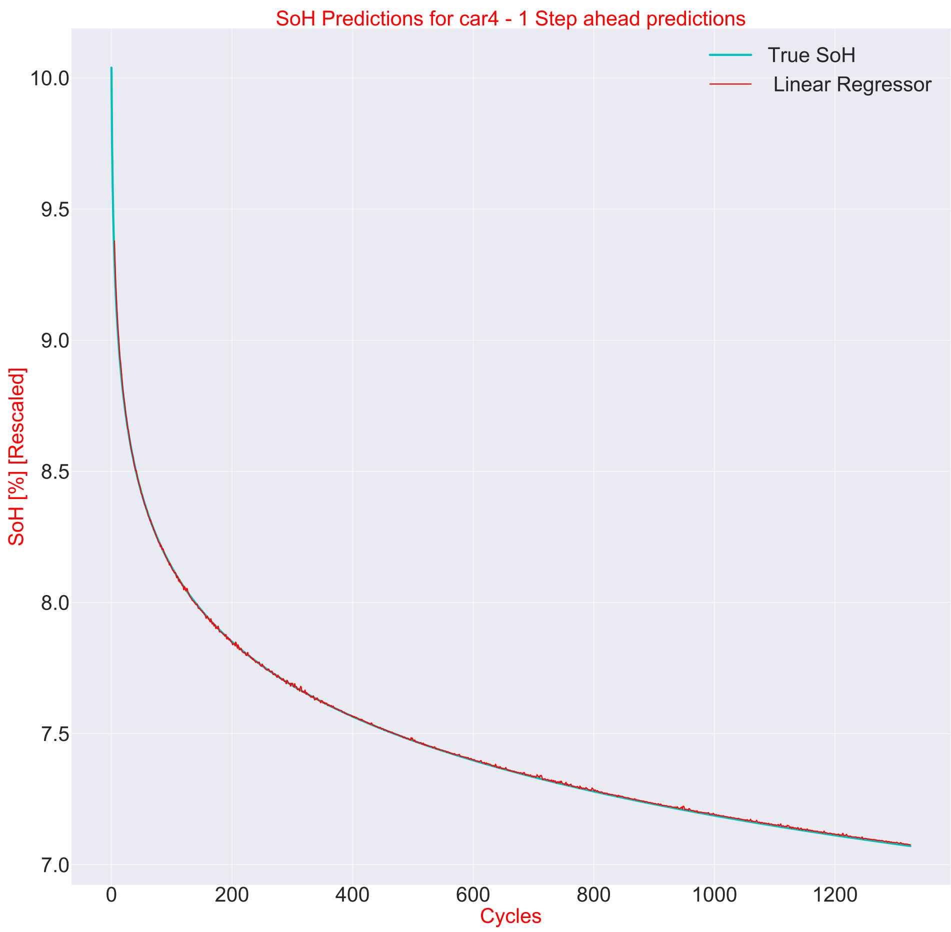


Figure A.12: 1 Step predictions for Car 4 using Linear Regressor

B

Appendix 2

B.1 Prediction of Battery Degradation using NARX + Random Forest Regressor

B.1.1 10 Step Ahead Predictions

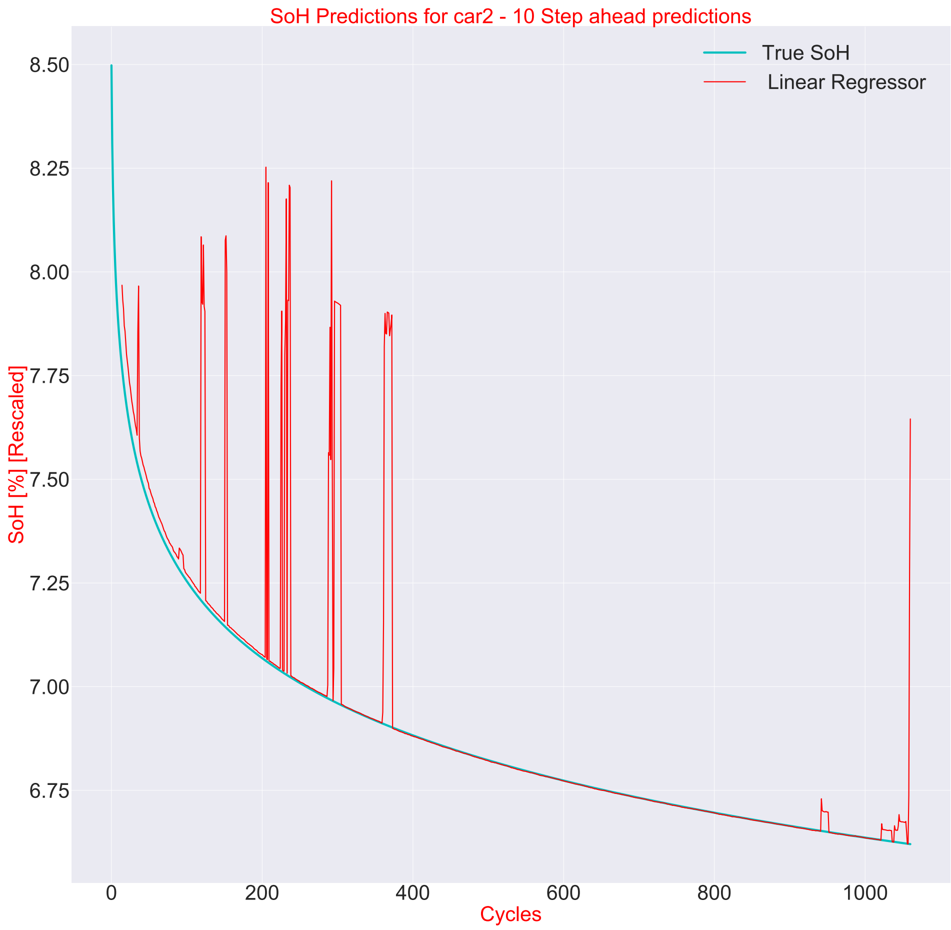


Figure B.1: 10 Step predictions for Car 2 using Random Forest Regressor

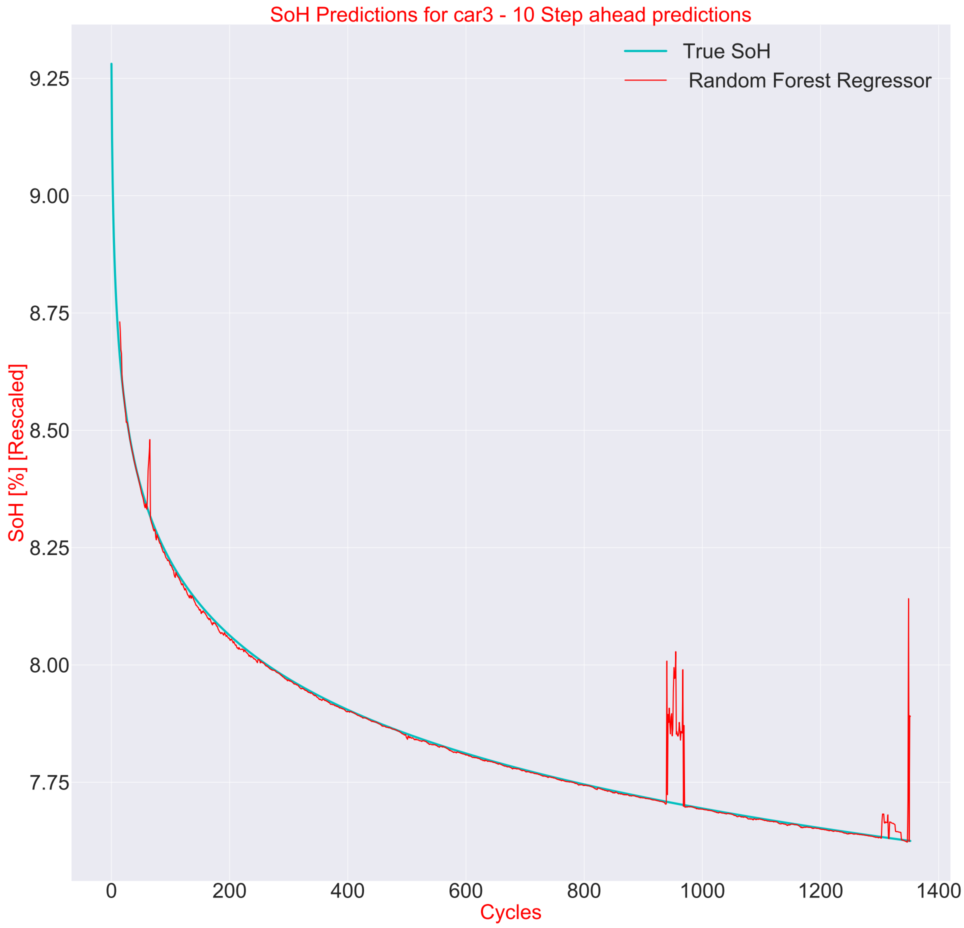


Figure B.2: 10 Step predictions for Car 3 using Random Forest Regressor

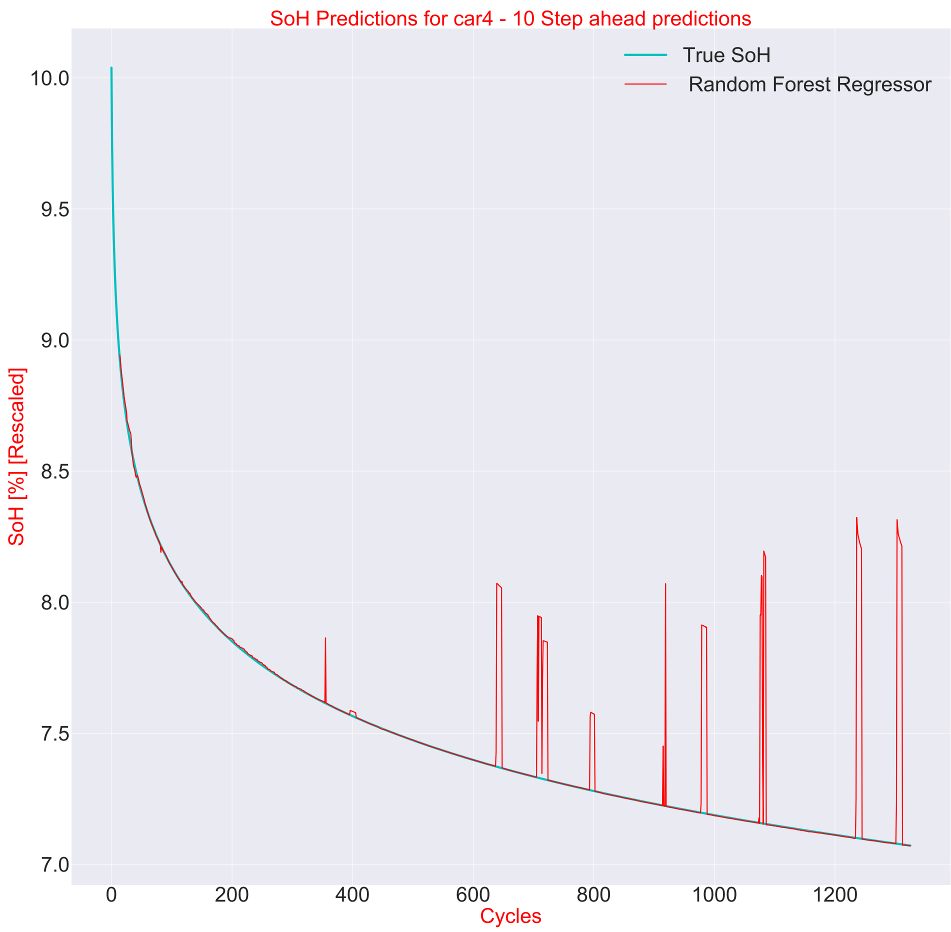


Figure B.3: 10 Step predictions for Car 4 using Random Forest Regressor

B.1.2 50 Step Ahead Predictions

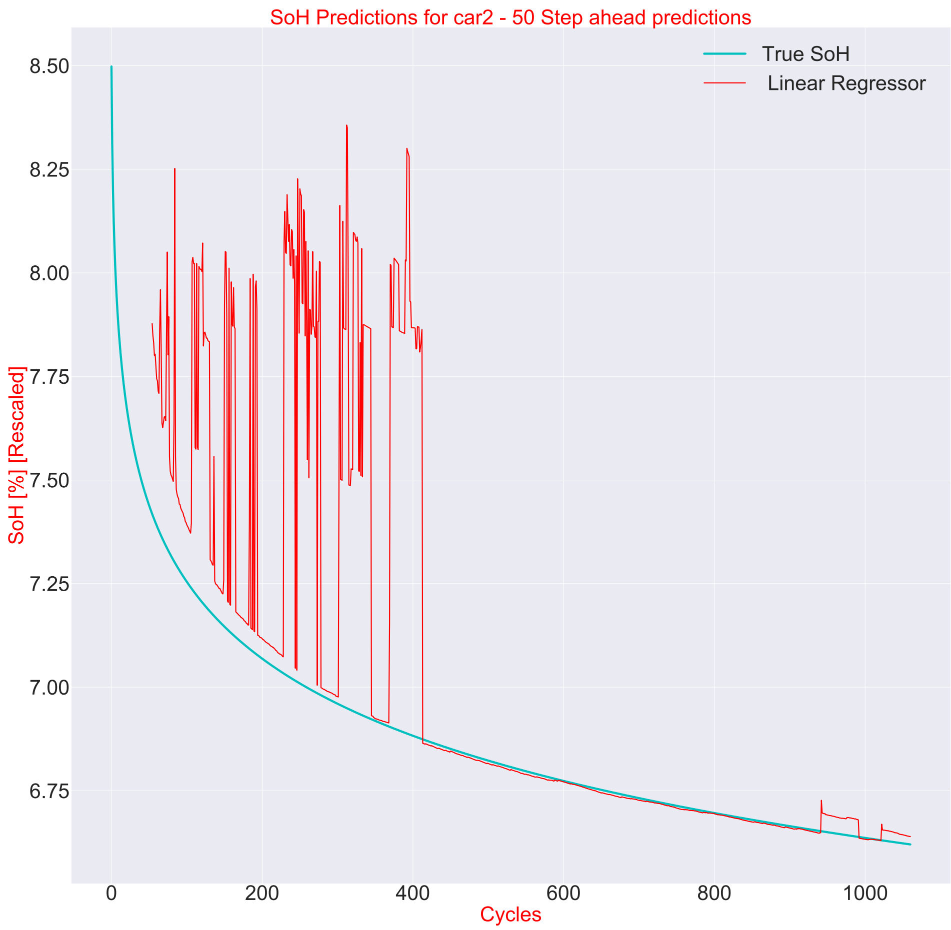


Figure B.4: 50 Step predictions for Car 2 using Random Forest Regressor

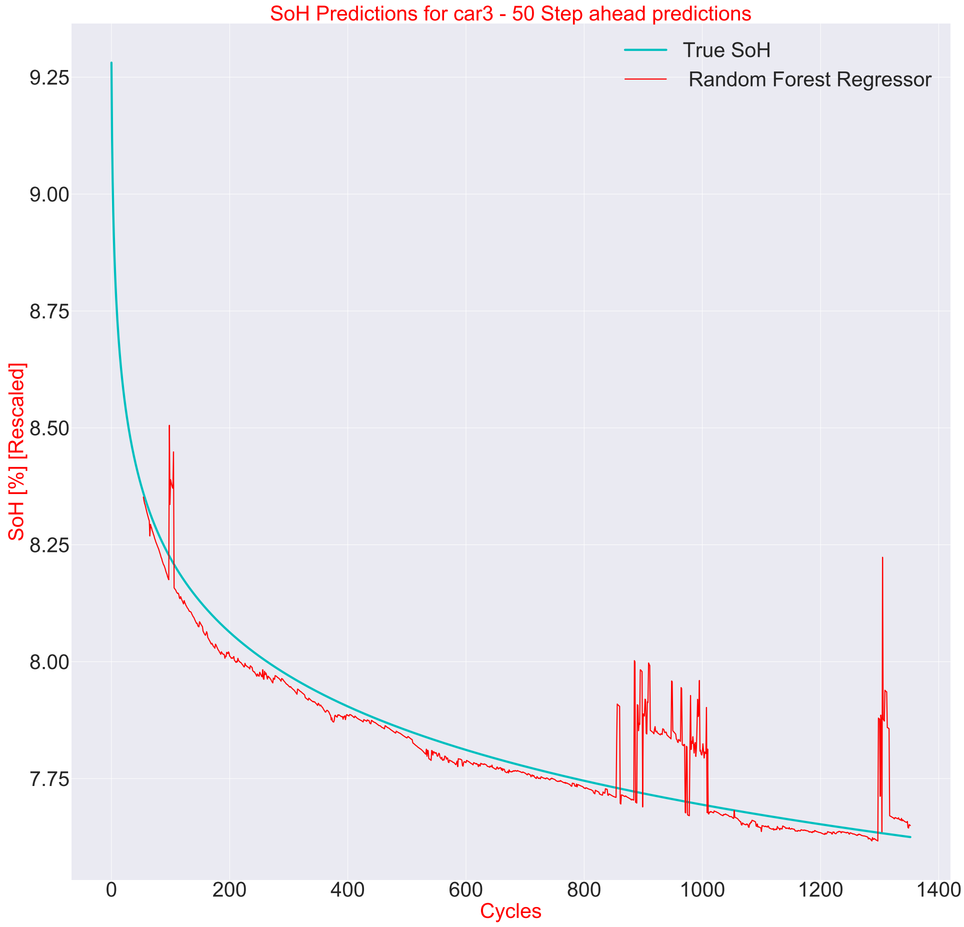


Figure B.5: 50 Step predictions for Car 3 using Random Forest Regressor

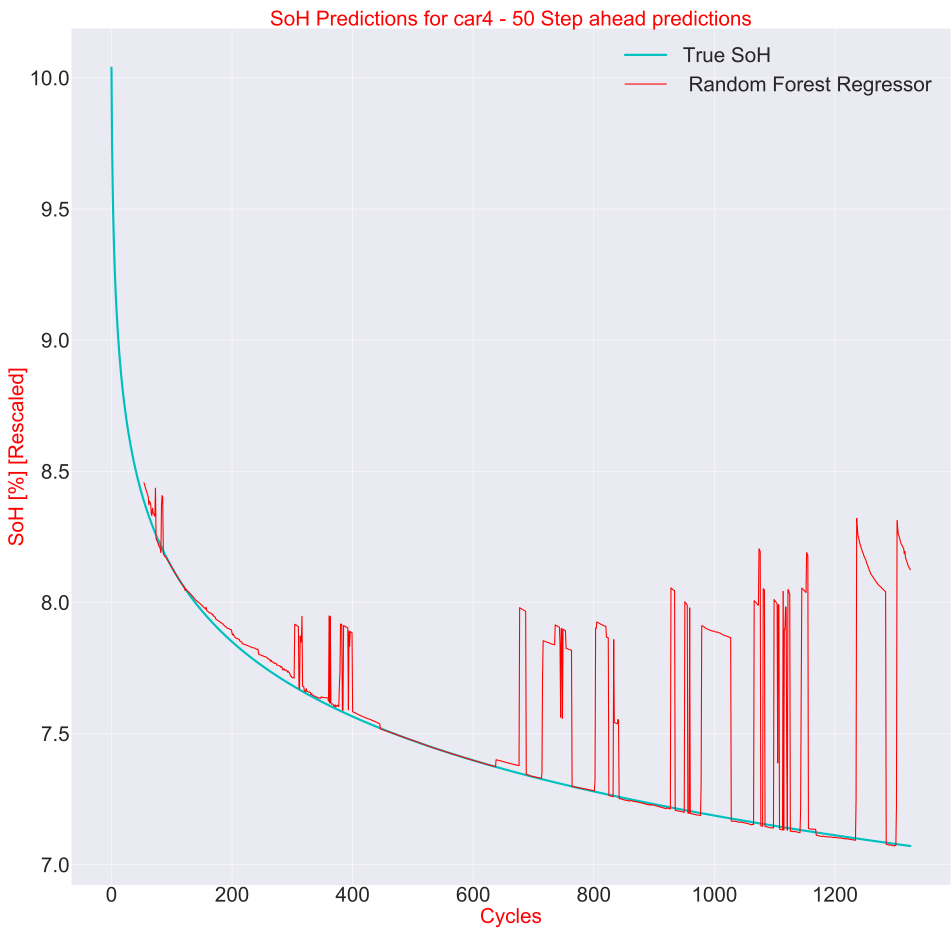


Figure B.6: 50 Step predictions for Car 4 using Random Forest Regressor

B.1.3 100 Step Ahead Predictions

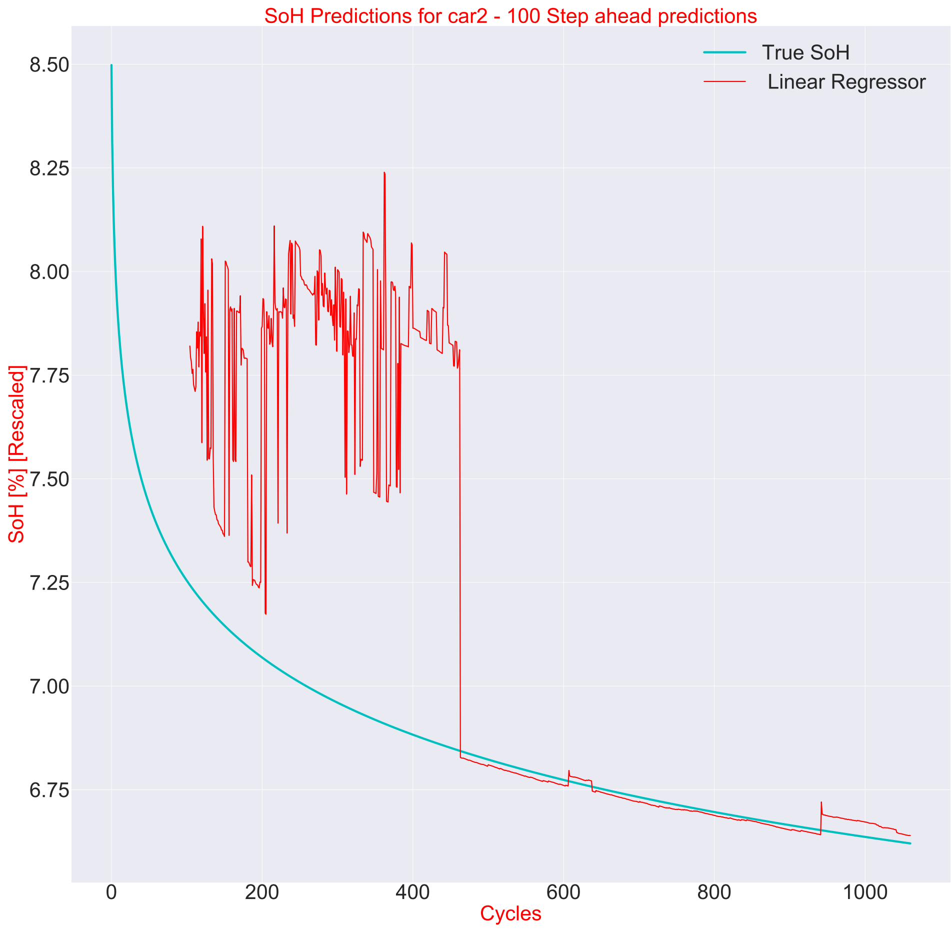


Figure B.7: 100 Step predictions for Car 2 using Random Forest Regressor

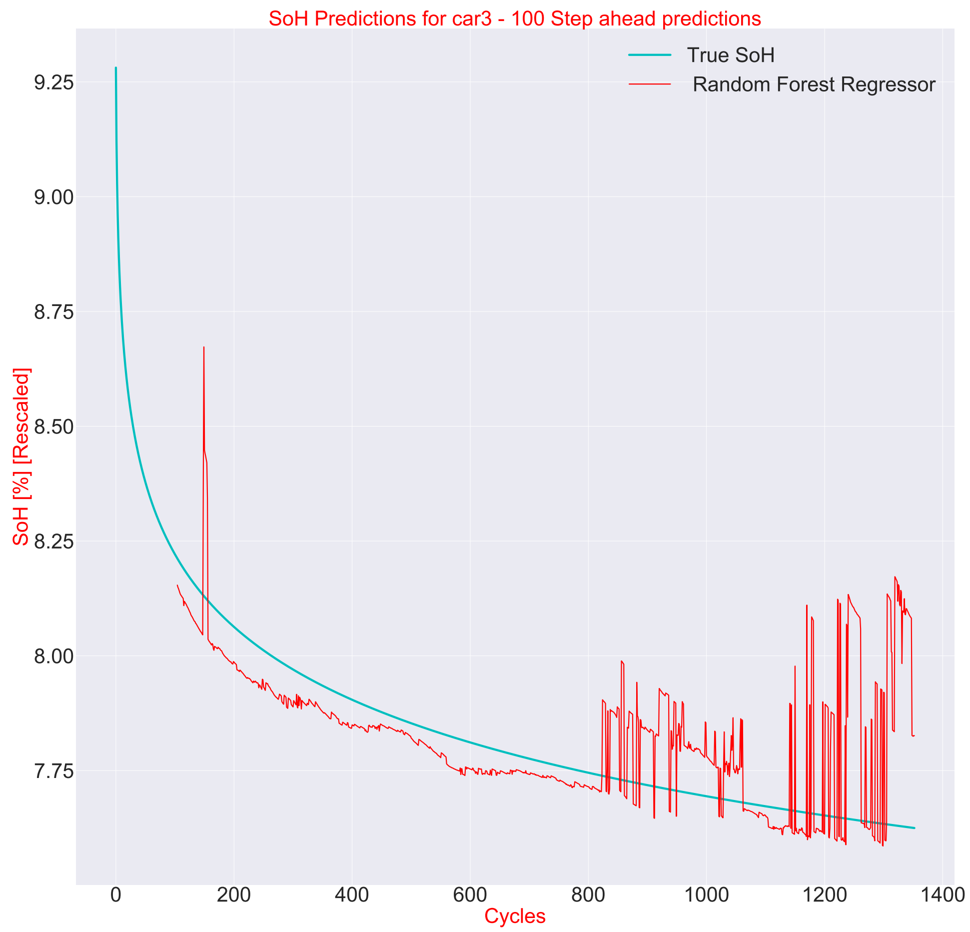


Figure B.8: 100 Step predictions for Car 3 using Random Forest Regressor

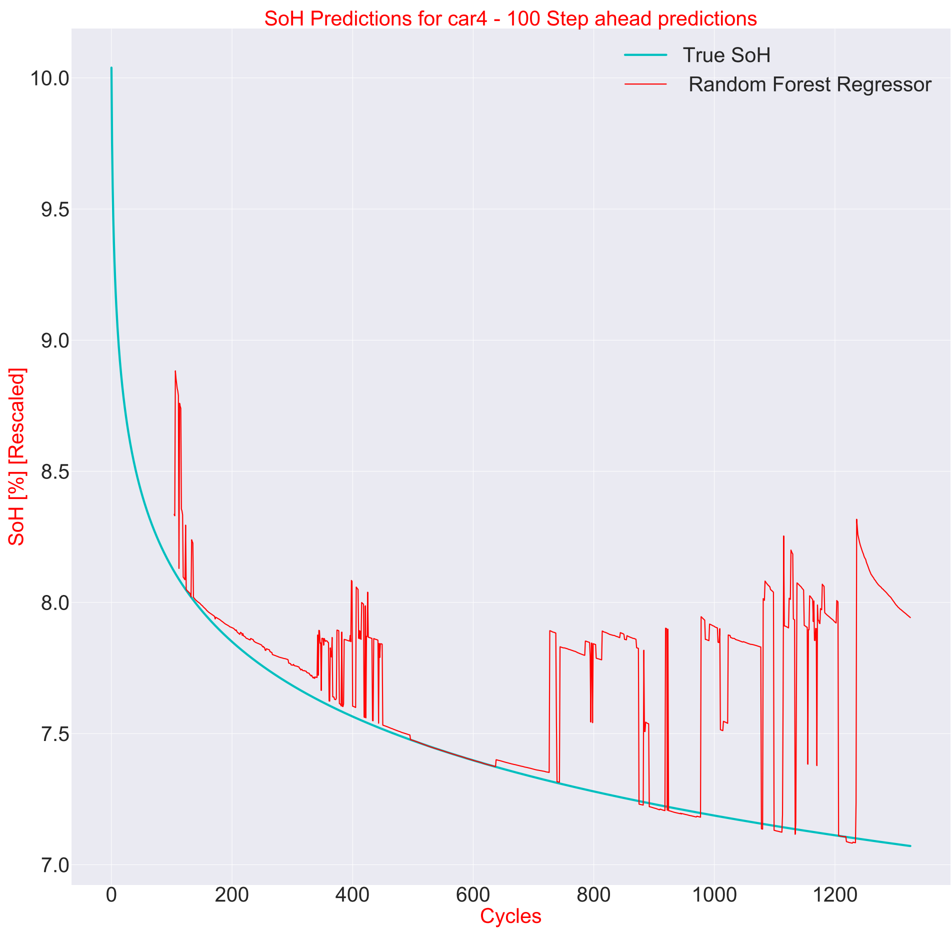


Figure B.9: 100 Step predictions for Car 4 using Random Forest Regressor

C
Appendix 3

C.1 Prediction of Battery Degradation using NARX + XGBoost Regressor

C.1.1 10 Step Ahead Predictions

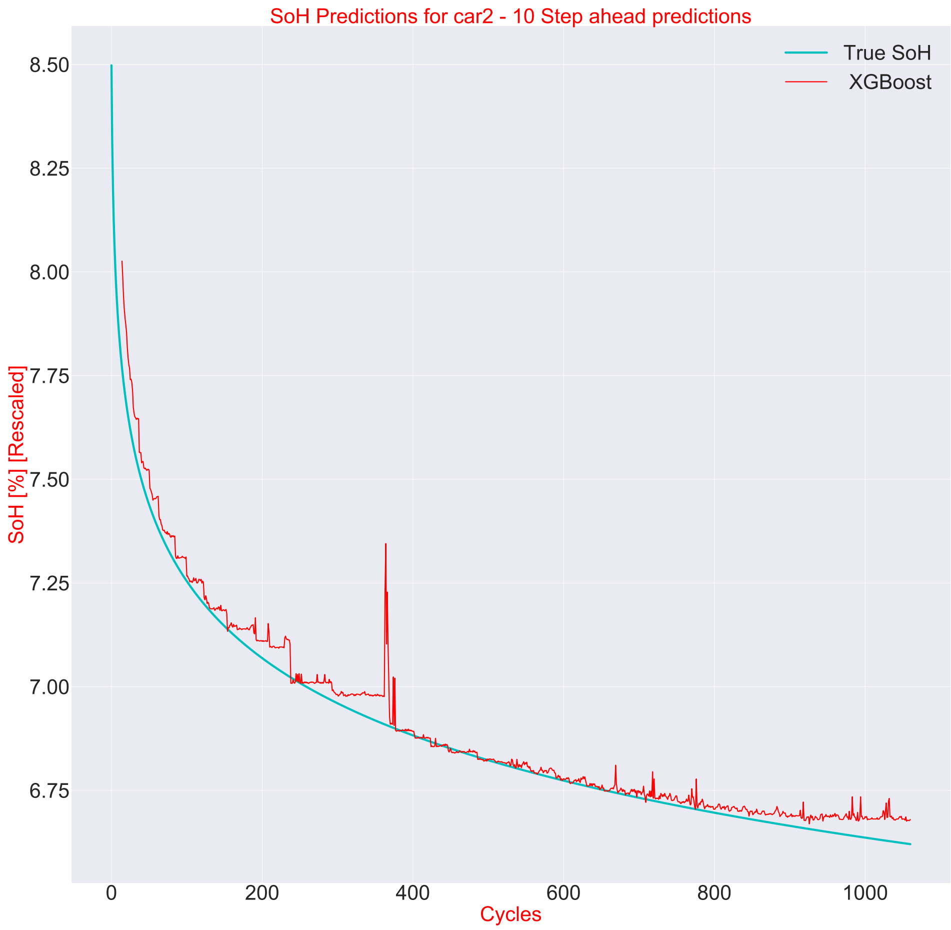


Figure C.1: 10 Step predictions for Car 2 using XGBoost Regressor

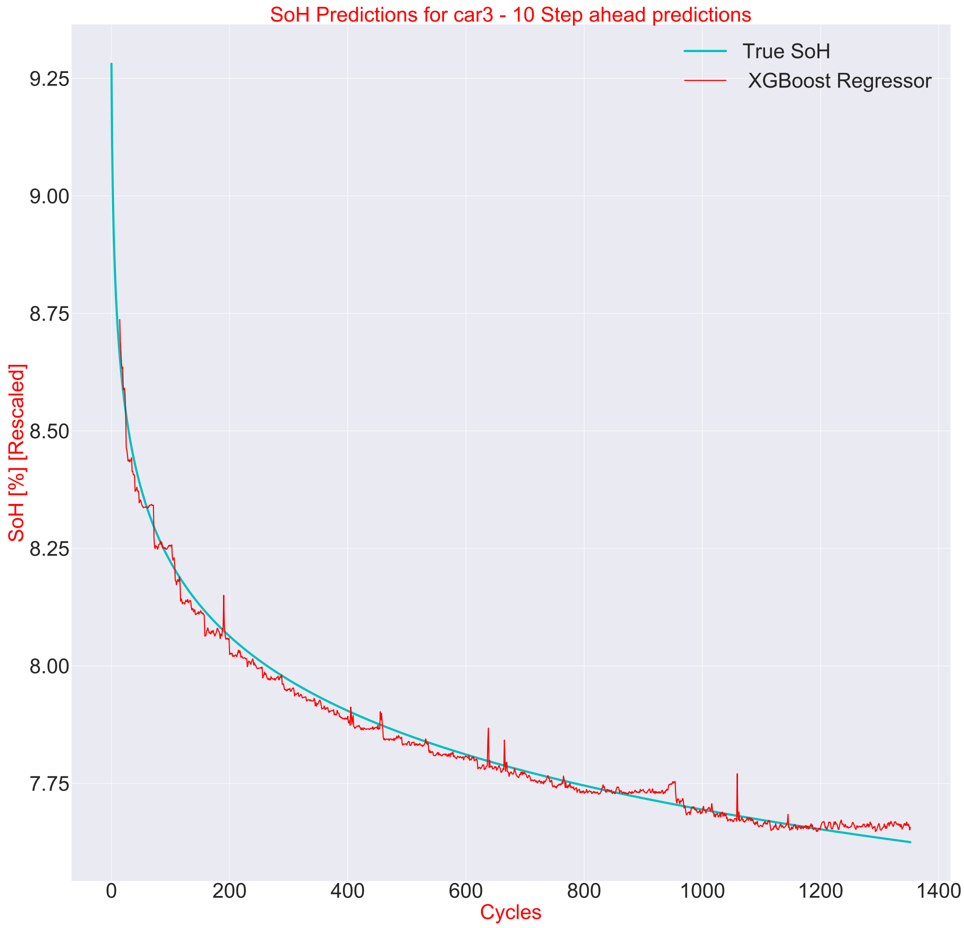


Figure C.2: 10 Step predictions for Car 3 using XGBoost Regressor

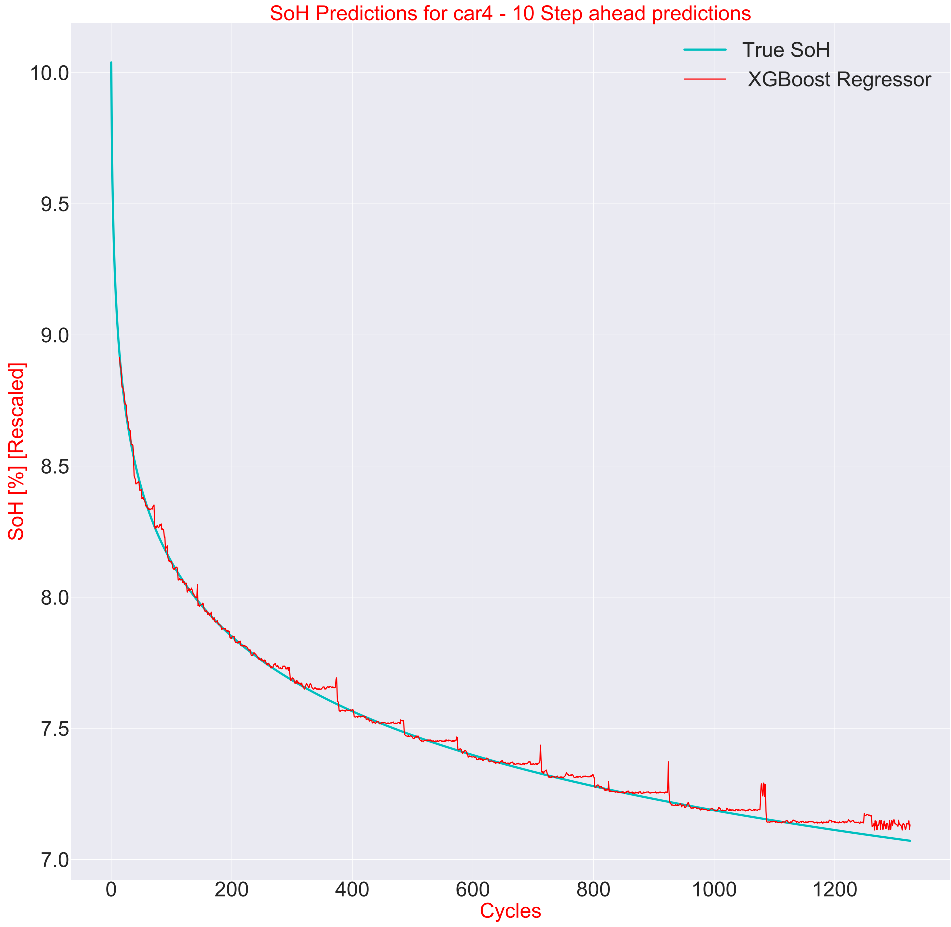


Figure C.3: 10 Step predictions for Car 4 using XGBoost Regressor

C.1.2 50 Step Ahead Predictions

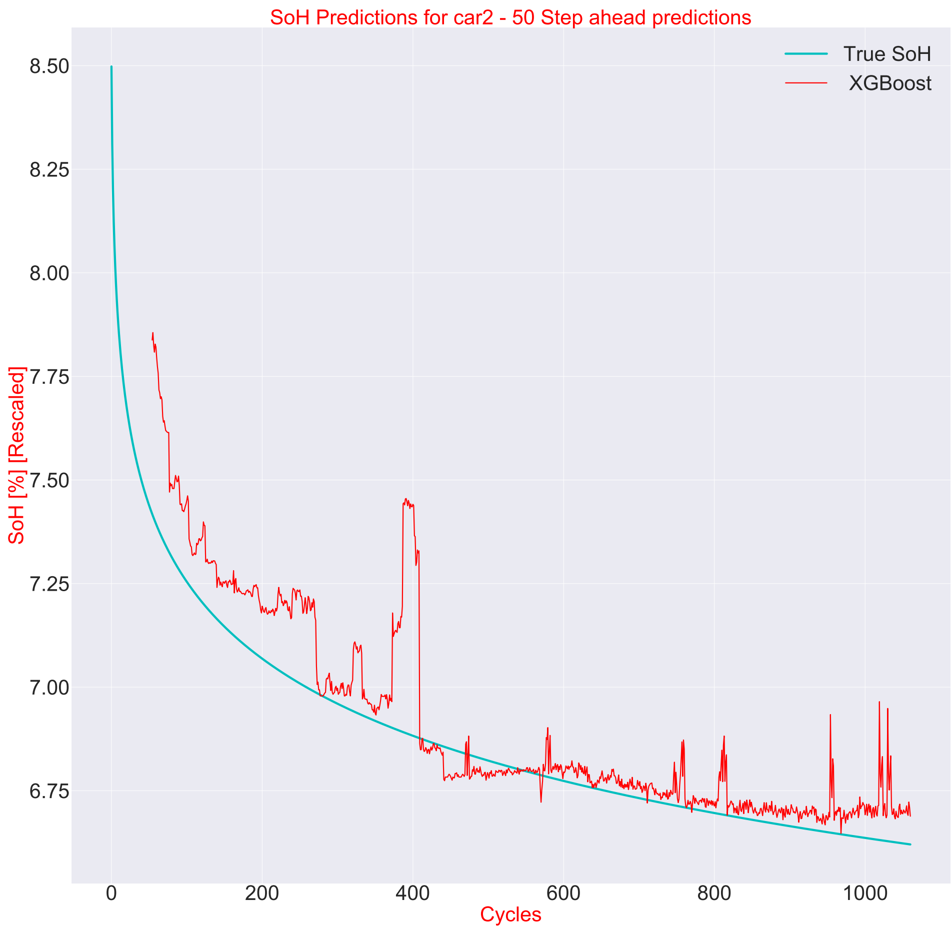


Figure C.4: 50 Step predictions for Car 2 using XGBoost Regressor

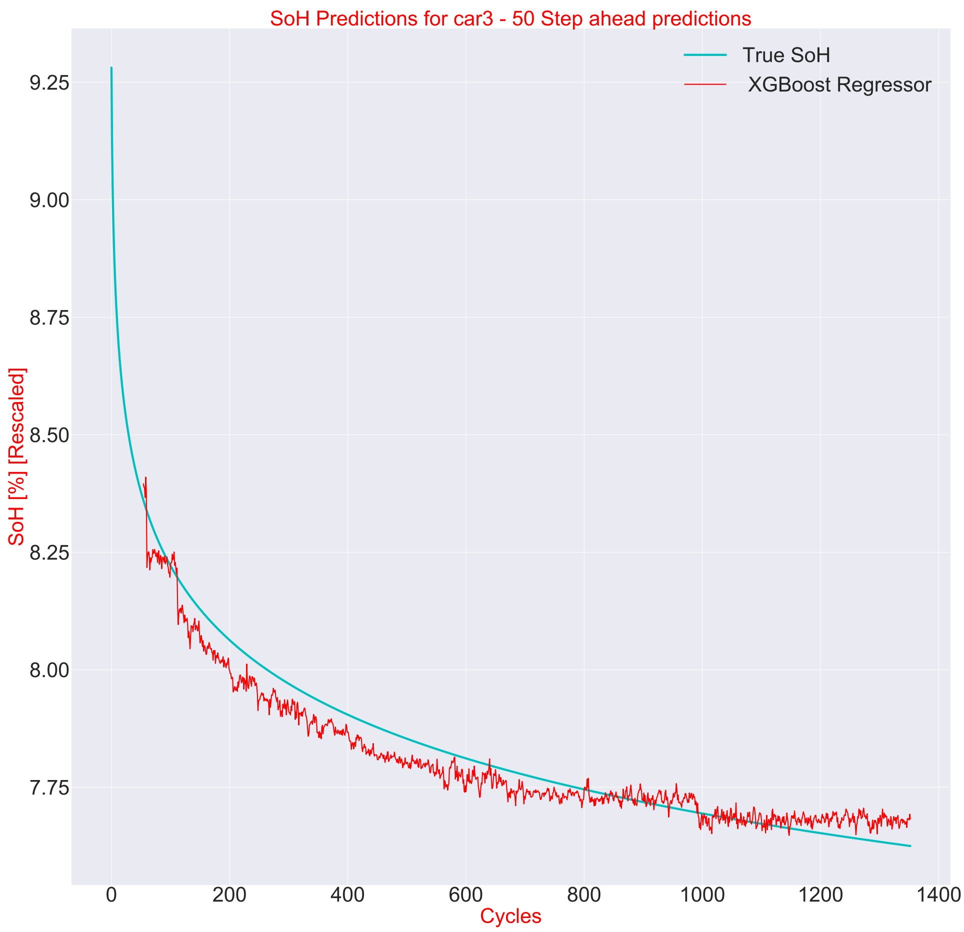


Figure C.5: 50 Step predictions for Car 3 using XGBoost Regressor

Figure C.6: 50 Step predictions for Car 4 using XGBoost Regressor

C.1.3 100 Step Ahead Predictions

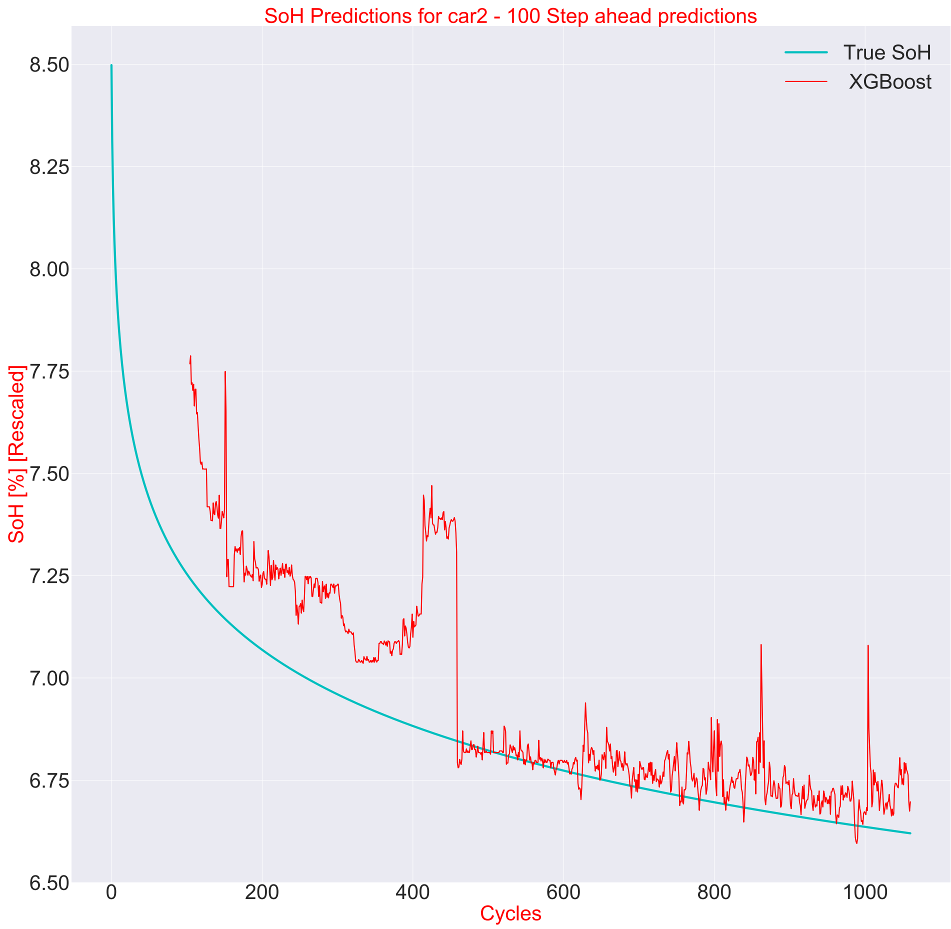


Figure C.7: 100 Step predictions for Car 2 using XGBoost Regressor

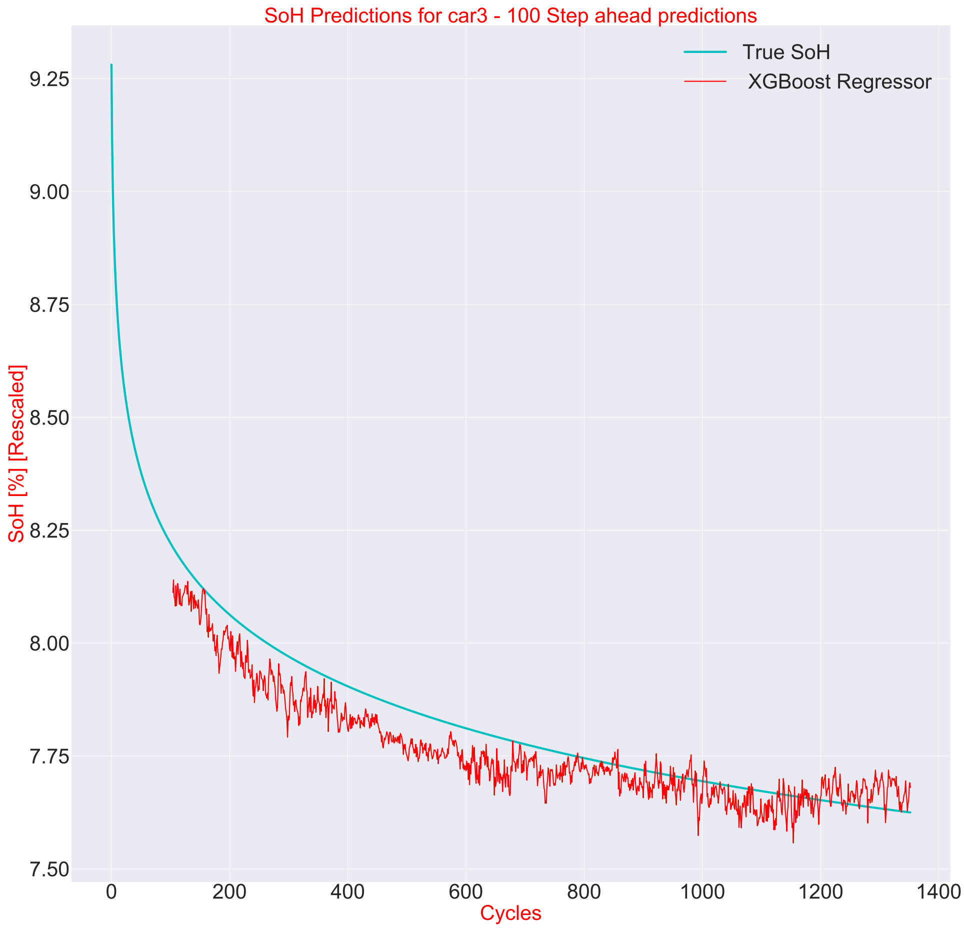


Figure C.8: 100 Step predictions for Car 3 using XGBoost Regressor

Figure C.9: 100 Step predictions for Car 4 using XGBoost Regressor

D

Appendix 4

D.1 Prediction of Battery Degradation using NARX + Feed Forward Network

D.1.1 10 Step Ahead Predictions

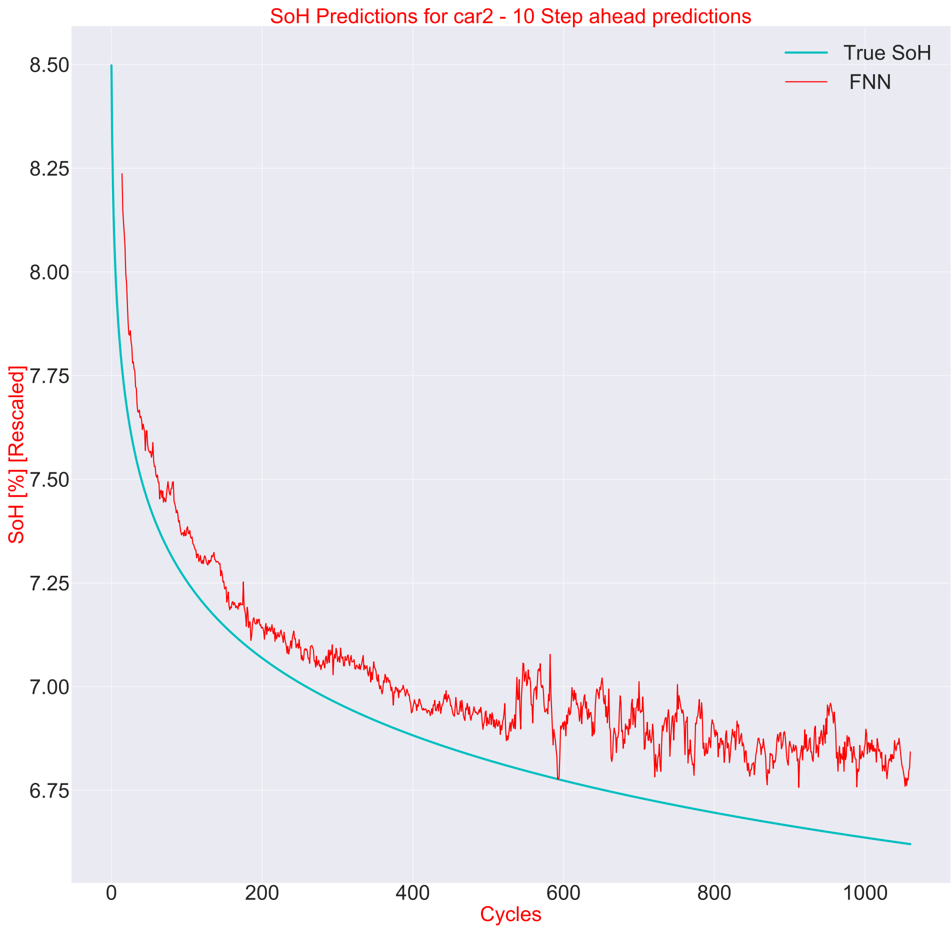


Figure D.1: 10 Step predictions for Car 2 using Feed Forward Network

Figure D.2: 10 Step predictions for Car 3 using Feed Forward Network

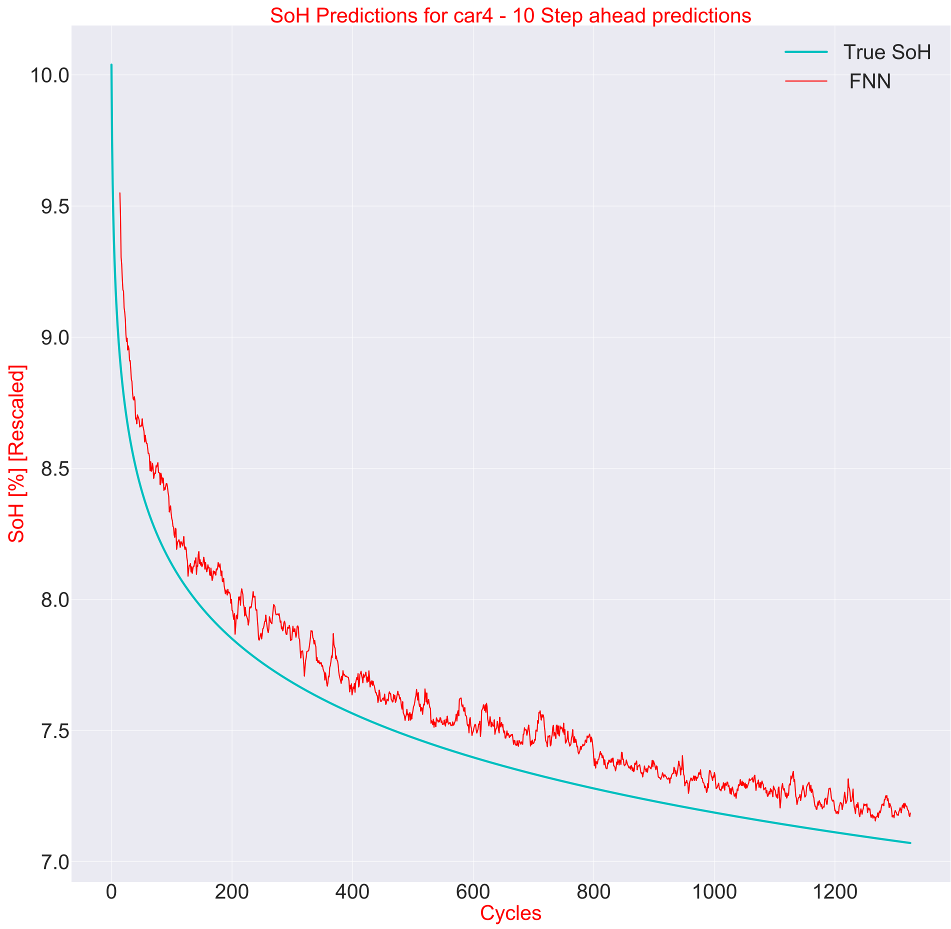


Figure D.3: 10 Step predictions for Car 4 using Feed Forward Network

D.1.2 50 Step Ahead Predictions

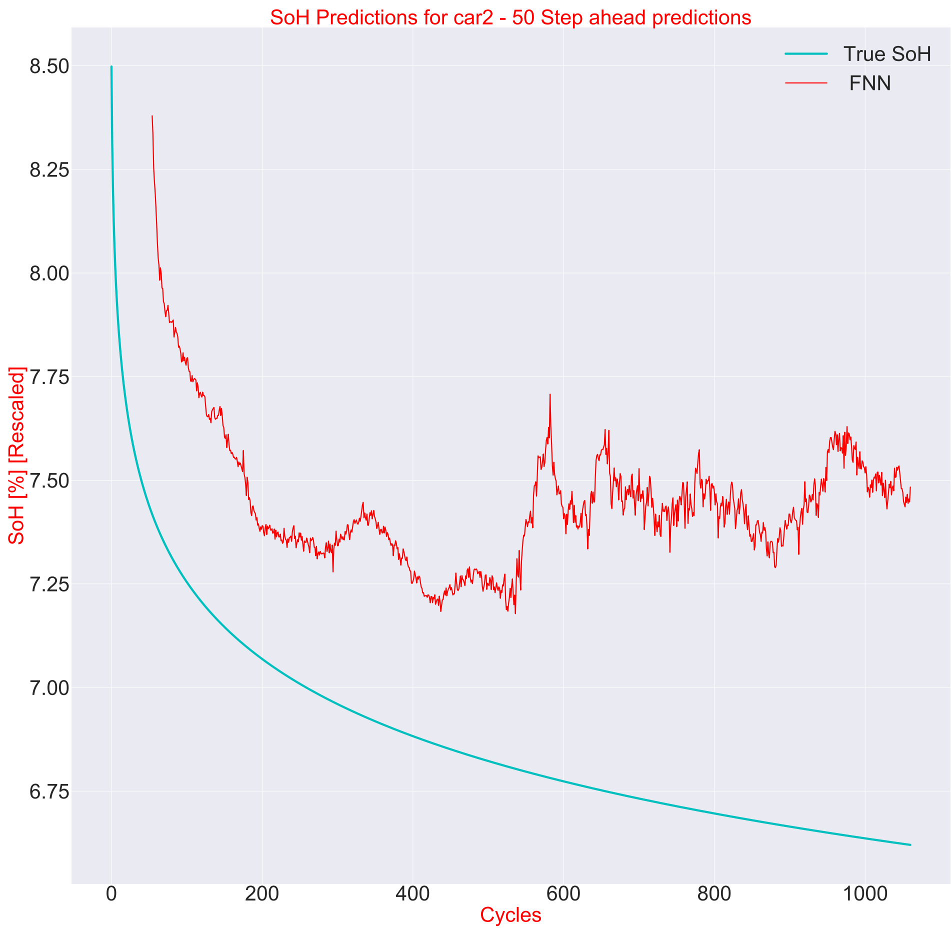


Figure D.4: 50 Step predictions for Car 2 using Feed Forward Network

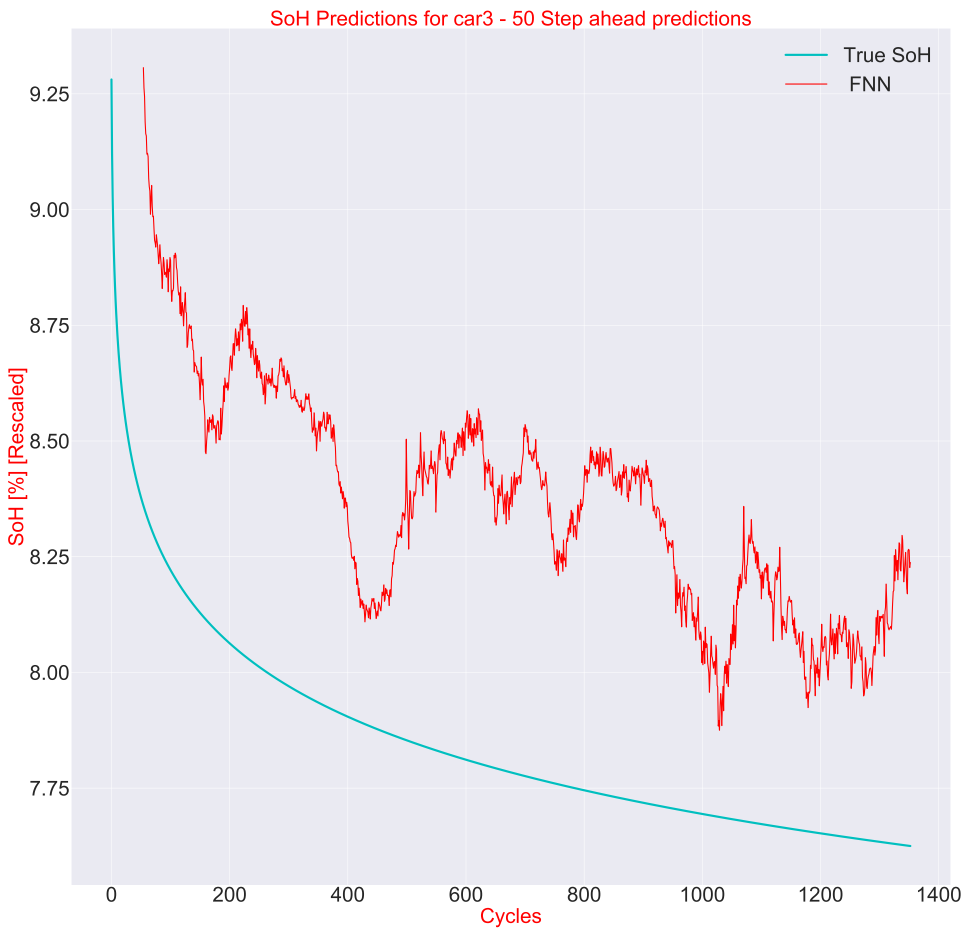


Figure D.5: 50 Step predictions for Car 3 using Feed Forward Network

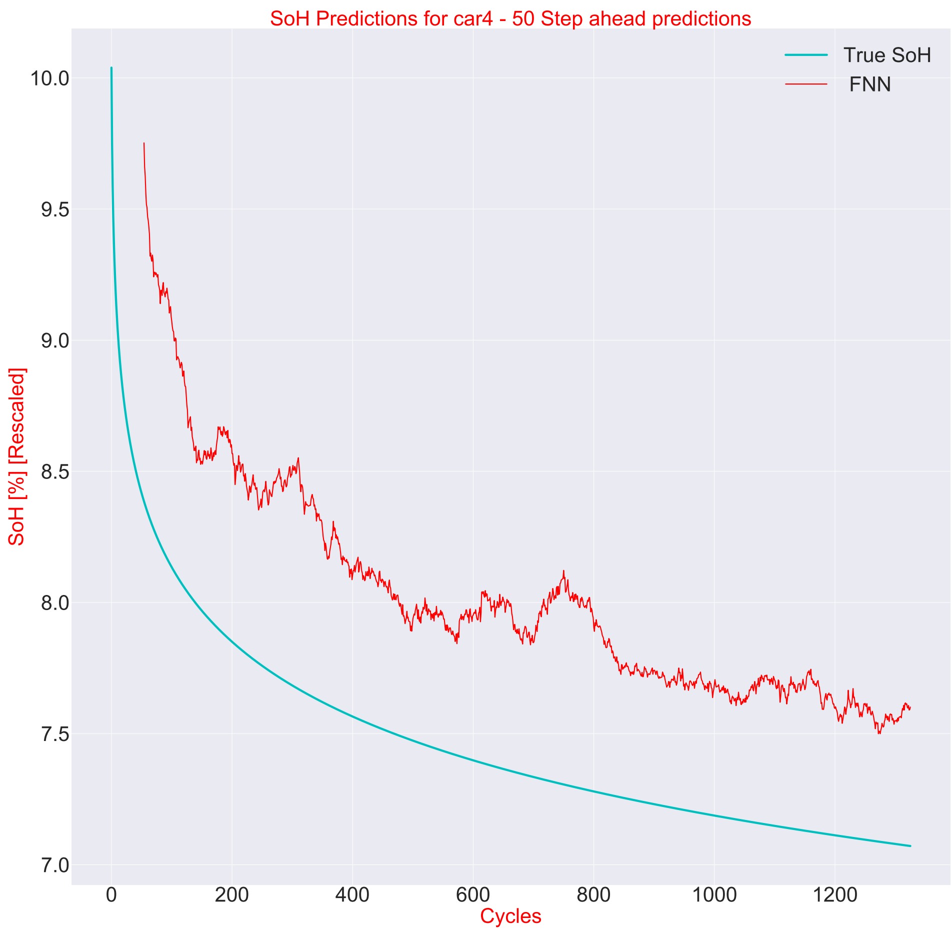


Figure D.6: 50 Step predictions for Car 4 using Feed Forward Network

D.1.3 100 Step Ahead Predictions

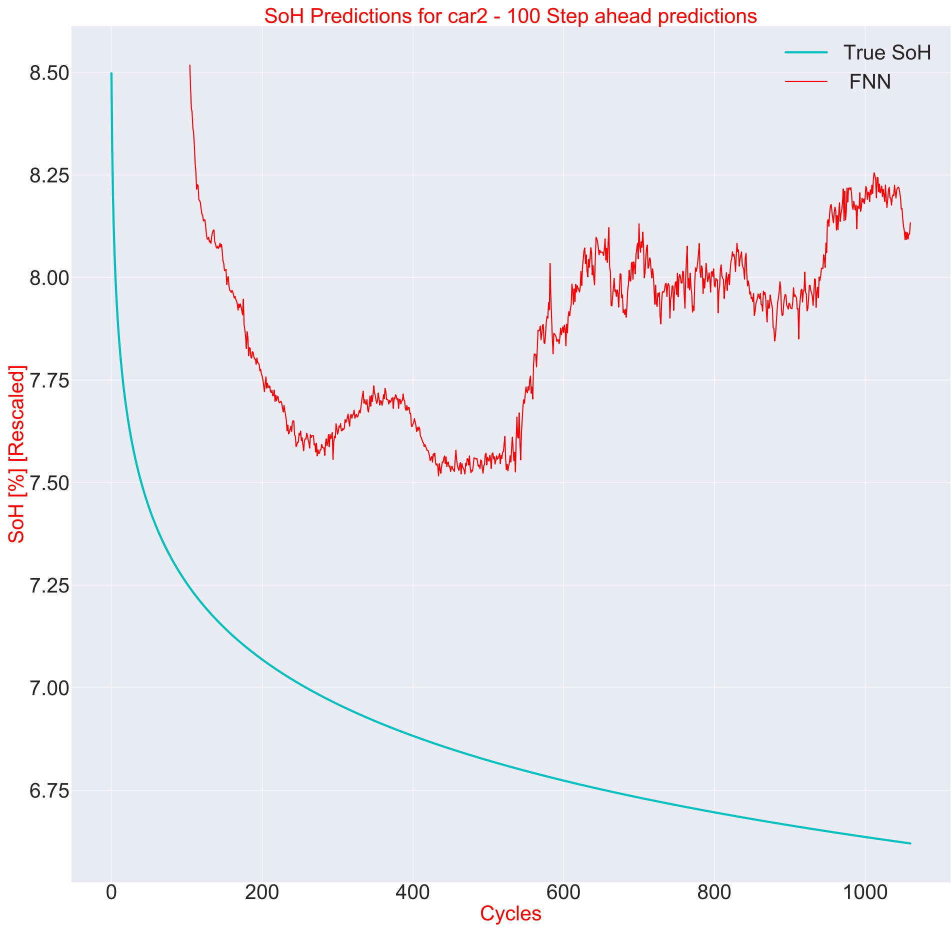


Figure D.7: 100 Step predictions for Car 2 using Feed Forward Network

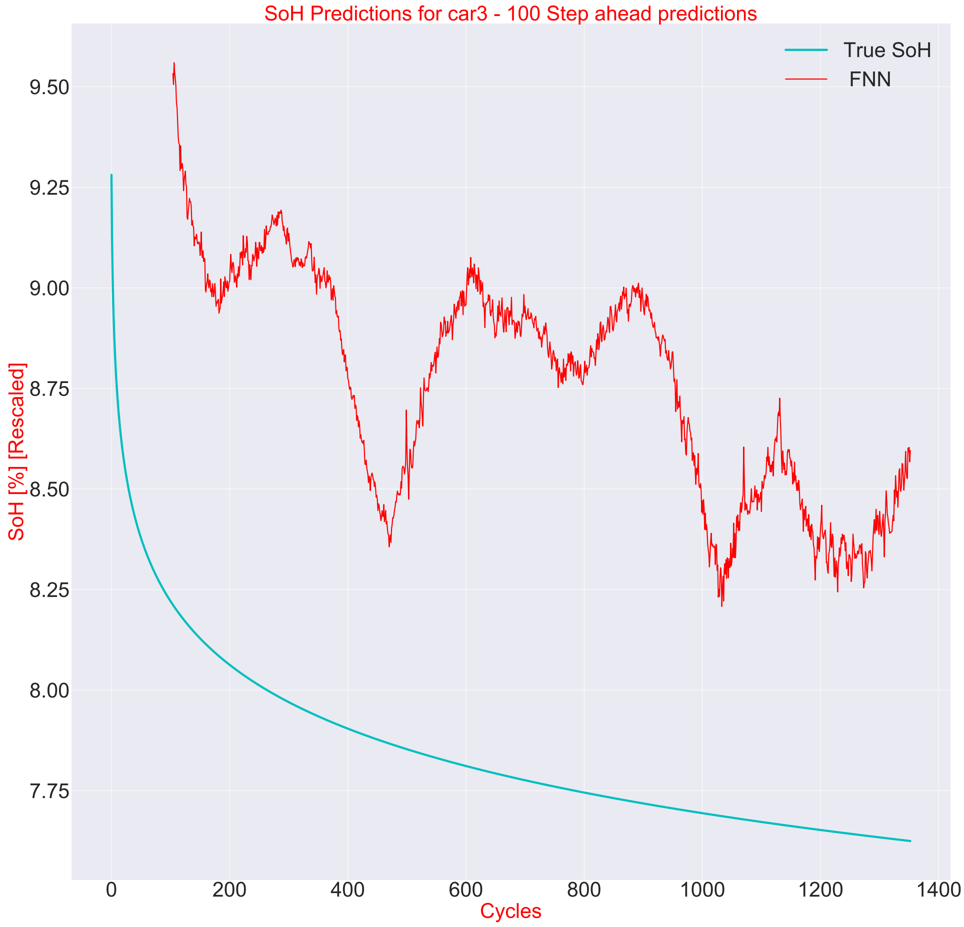


Figure D.8: 100 Step predictions for Car 3 using Feed Forward Network

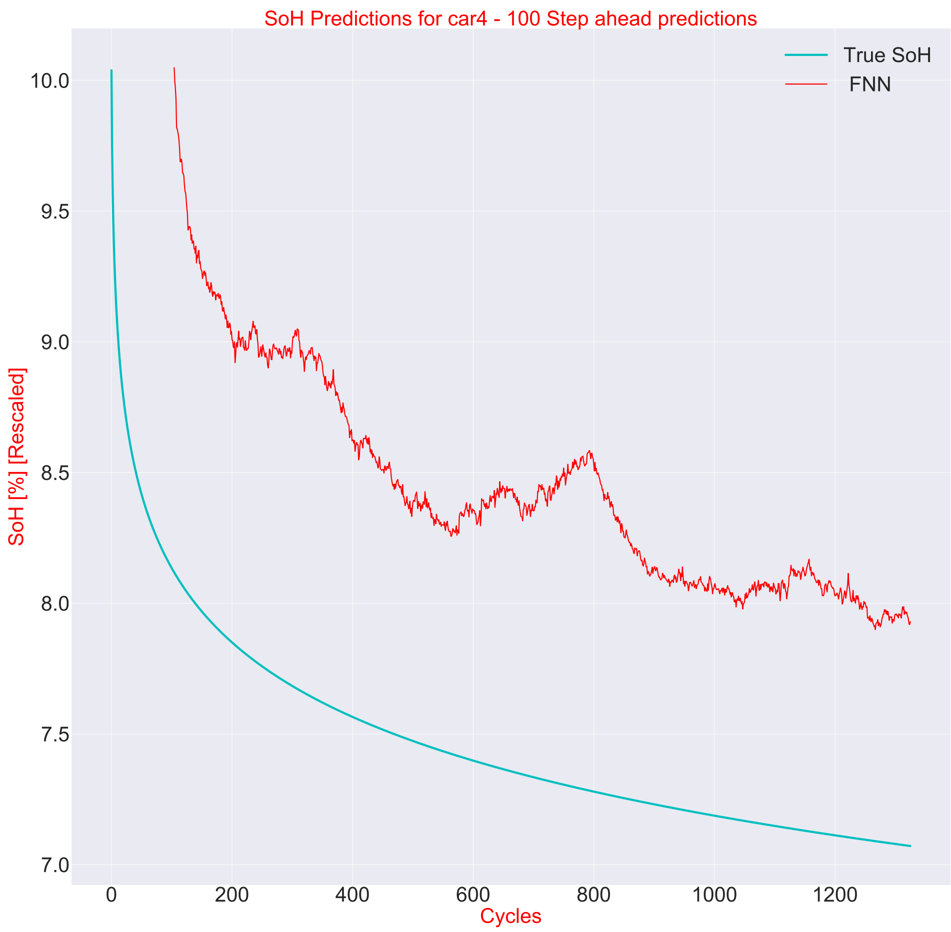


Figure D.9: 100 Step predictions for Car 4 using Feed Forward Network

E

Appendix 5

E.1 Prediction of Battery Degradation using NARX + Linear Regressor

E.1.1 10 Step Ahead Predictions

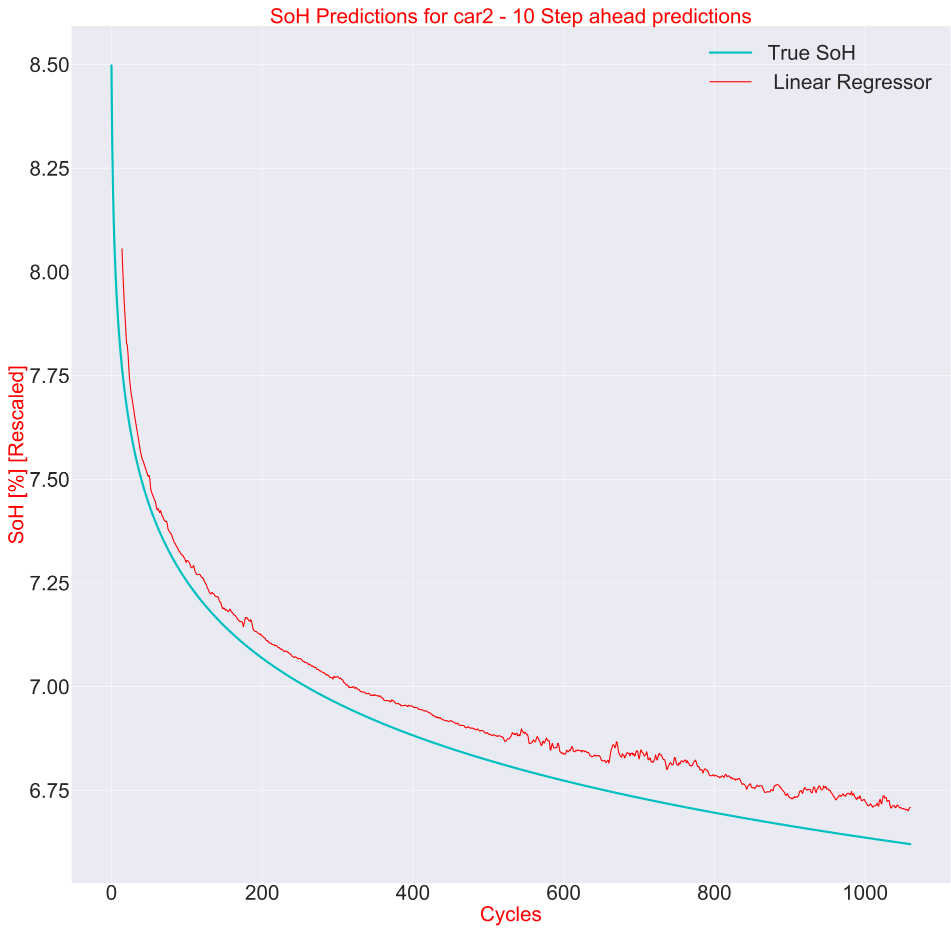


Figure E.1: 10 Step predictions for Car 2 using Linear Regressor

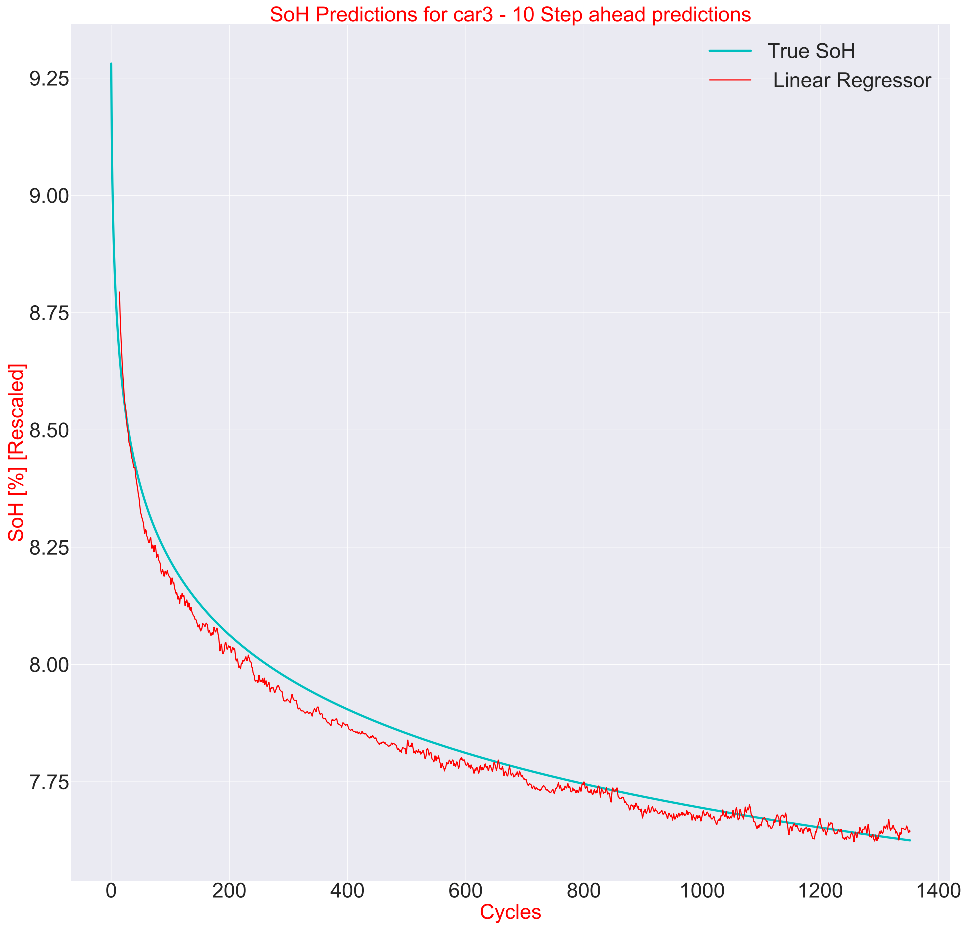


Figure E.2: 10 Step predictions for Car 3 using Linear Regressor

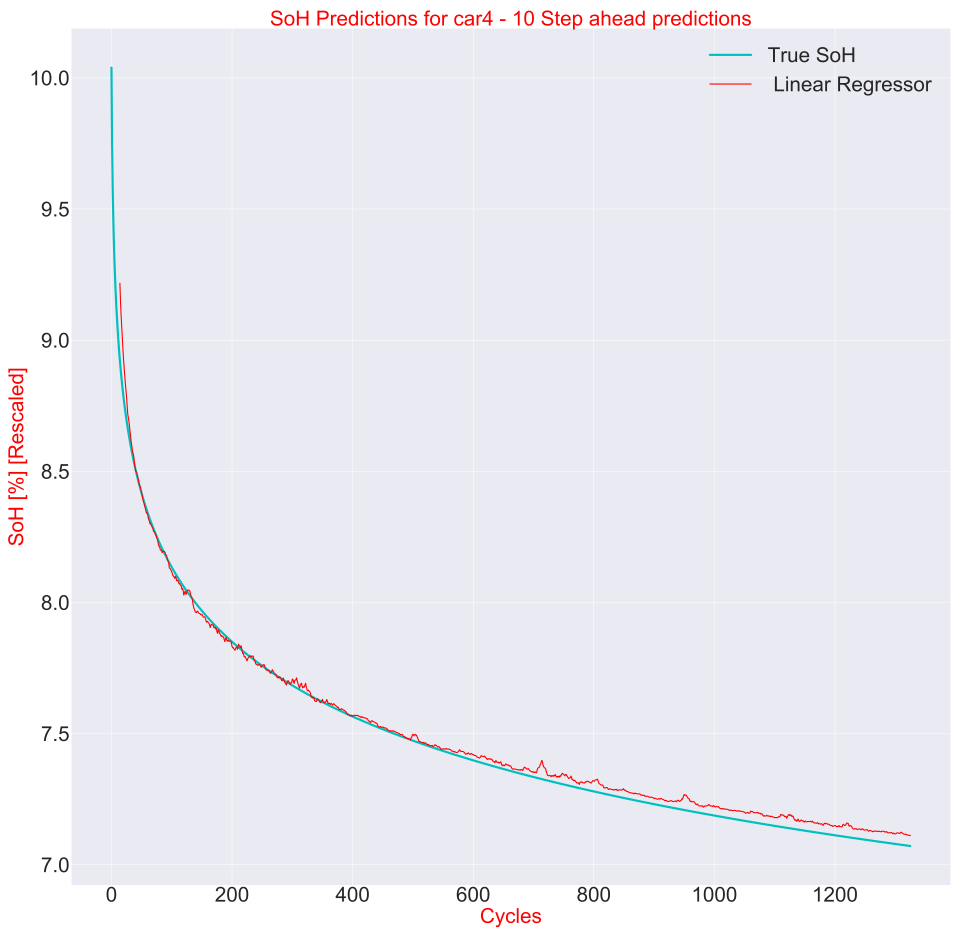


Figure E.3: 10 Step predictions for Car 4 using Linear Regressor

E.1.2 50 Step Ahead Predictions

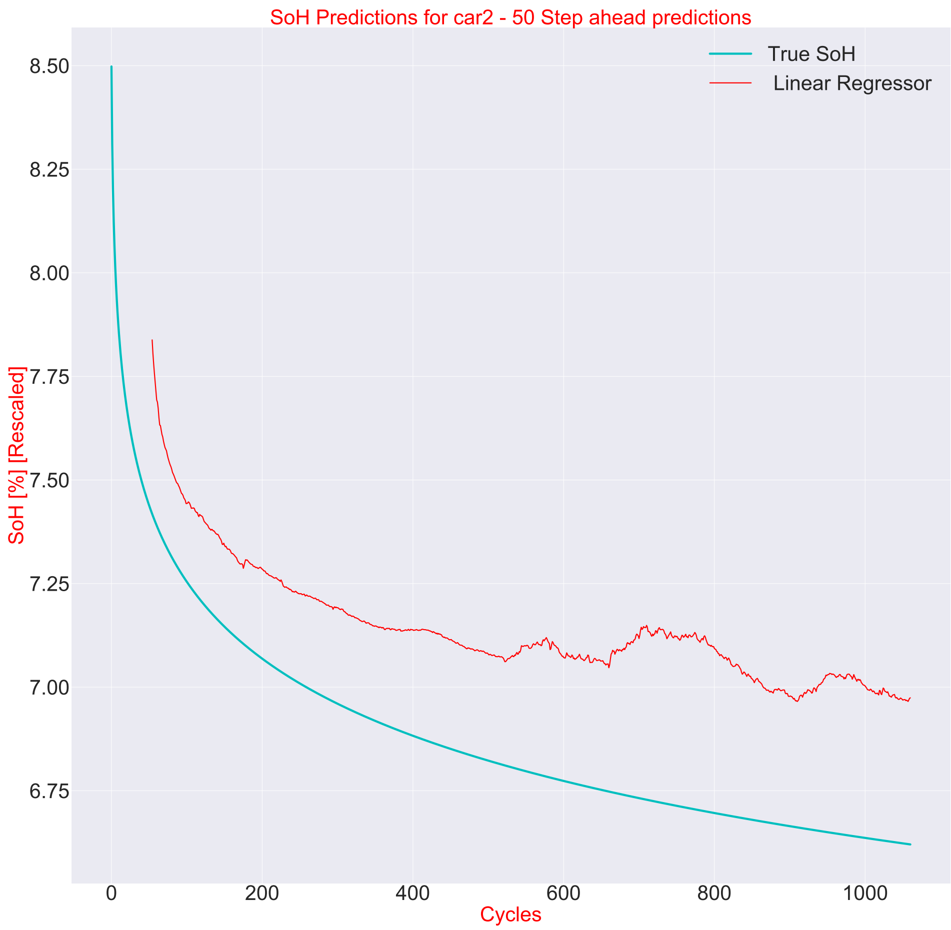


Figure E.4: 50 Step predictions for Car 2 using Linear Regressor

Figure E.5: 50 Step predictions for Car 3 using Linear Regressor

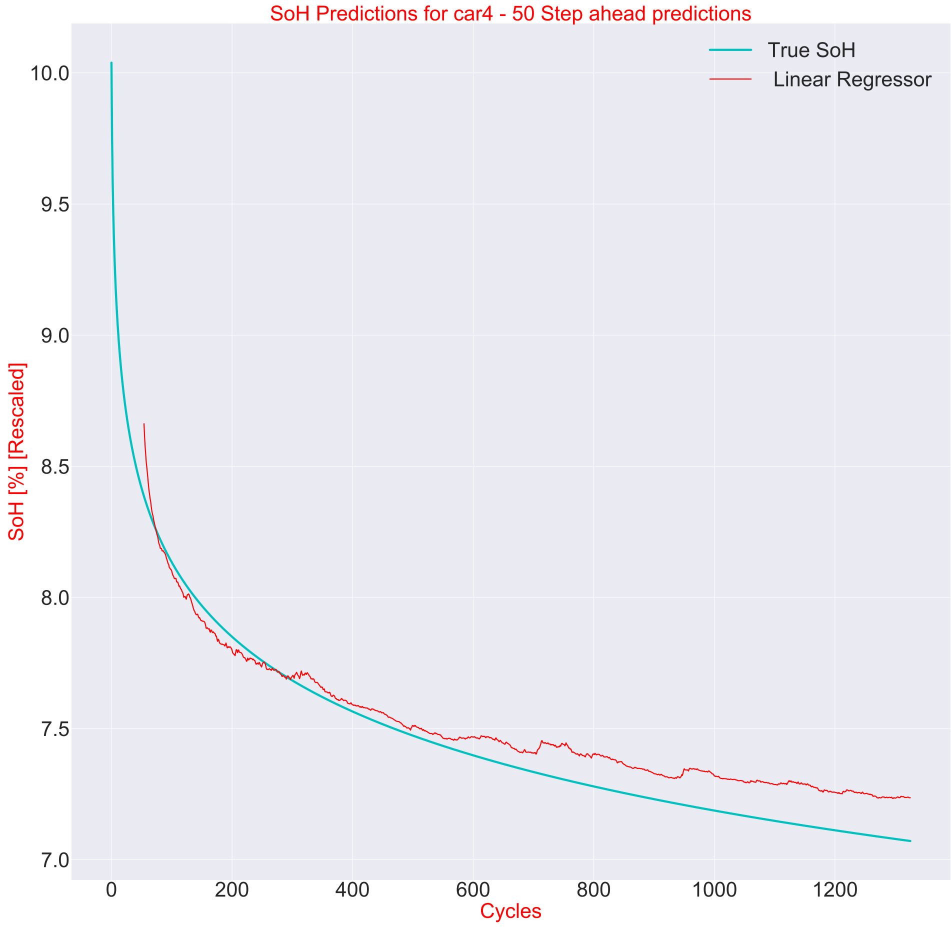


Figure E.6: 50 Step predictions for Car 4 using Linear Regressor

E.1.3 100 Step Ahead Predictions

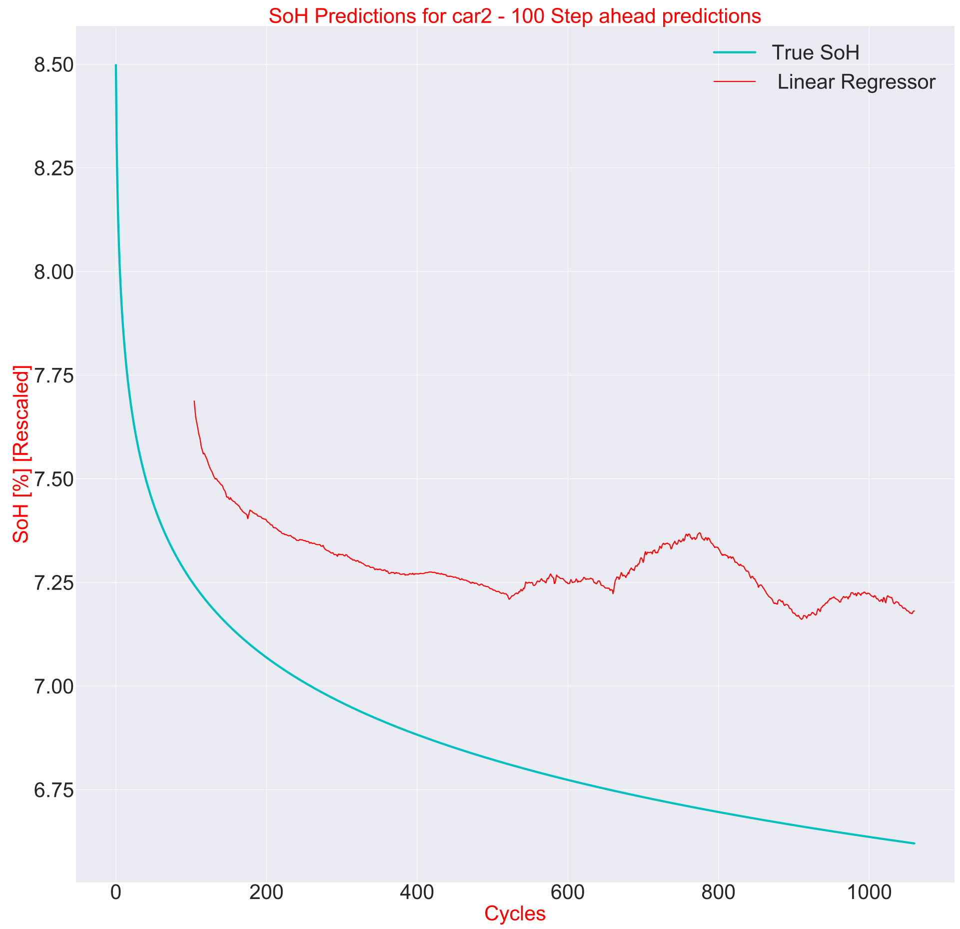


Figure E.7: 100 Step predictions for Car 2 using Linear Regressor

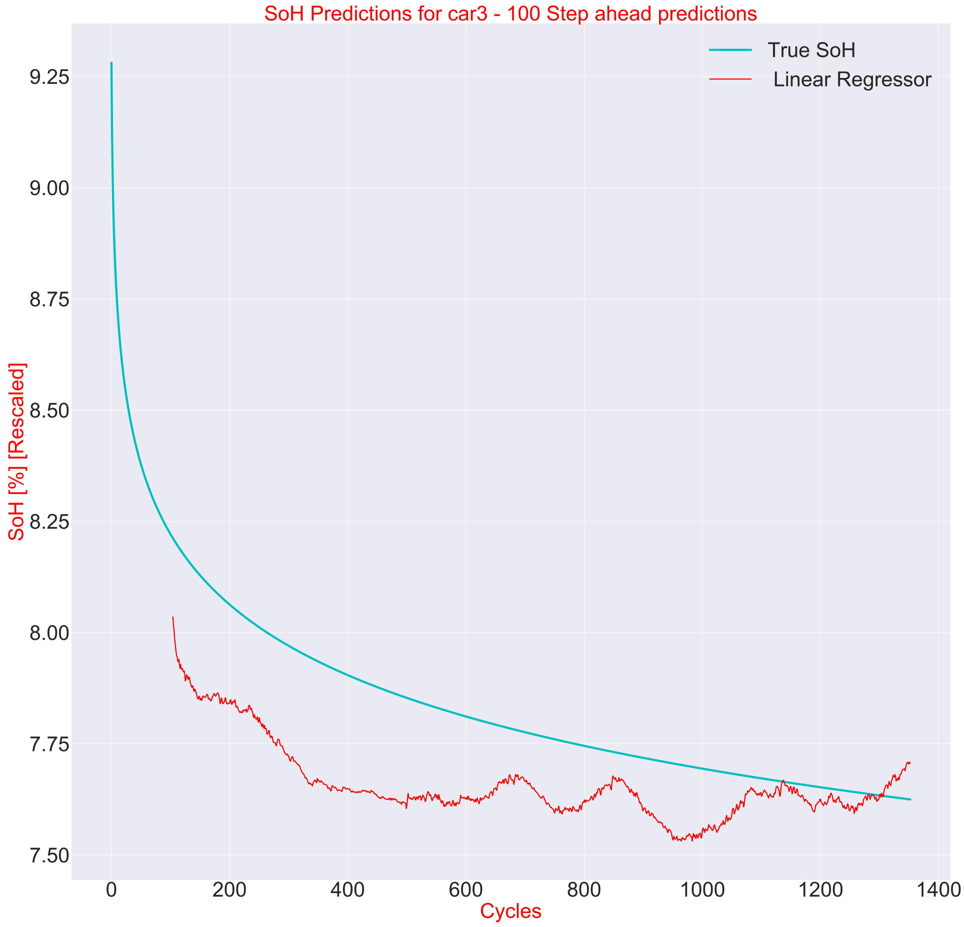


Figure E.8: 100 Step predictions for Car 3 using Linear Regressor

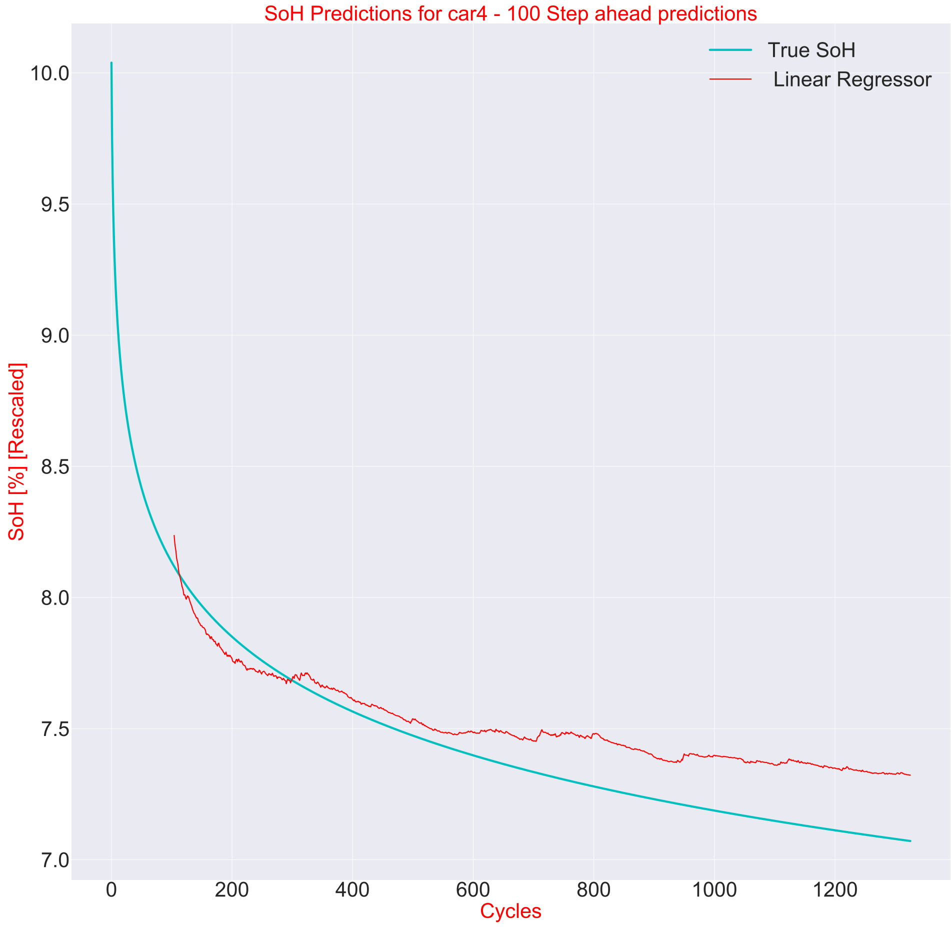


Figure E.9: 100 Step predictions for Car 4 using Linear Regressor