Machine Learning Application in Energy Storage System's State Estimation: State of Health (SOH)

Ashkan Nazari

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Computer Science

Lenwood S. Heath, Chair
Naren Ramakrishnan
Michael W. Ellis

May11, 2021
Blacksburg, VA

Keywords: Machine Learning, Artificial Neural Network, Convolutional Neural Network, Long Short-term Memory, Li-ion Battery

Machine Learning Application in Energy Storage System's State Estimation: State of Health (SOH)

Ashkan Nazari

ABSTRACT

There exists an increasing demand for the modern prognostics and health management system for the Li-ion batteries under real-world operation, specifically for the electric vehicle (EV) applications. Since the estimation of battery state of health (SOH) is critical for the safety and the decision making such as warranty analysis, the battery SOH should be estimated accurately. In this work, we have employed measurable data such as current, voltage, and temperature towards developing different deep learning (DL) models to estimate the cell's SOH cycled under a variety of extreme fast charging protocols. The results obtained from the different DL models have been compared with those obtained from the conventional feed forward neural networks (FFNNs). The accuracy of all the developed DL models with long short-term memory (LSTM), convolutional LSTM (ConvLSTM), and deep convolutional neural network (DCNN) architecture are acceptable by industry standards, with mean absolute percentage error (MAPE) less than 3%. The promising results obtained in this study indicate that the presented DL models in this work can be implemented in future battery management systems (BMSs).

Machine Learning Application in Energy Storage System's State Estimation: State of Health (SOH)

Ashkan Nazari

GENERAL AUDIENCE ABSTRACT

There exists an increasing demand for the modern prognostics and health management system for the Li-ion batteries under real-world operation, specifically for the electric vehicle (EV) applications. Since the estimation of battery state of health (SOH) is critical for the safety and the decision making such as warranty analysis, the battery SOH should be estimated accurately. In this work, we have employed measurable data such as current, voltage, and temperature towards developing different deep learning (DL) models to estimate the cell's SOH cycled under a variety of extreme fast charging protocols. The results obtained from the different DL models have been compared with those obtained from the conventional machine learning (ML) methods. The accuracy of all the developed DL models are acceptable by industry standards, with mean absolute percentage error (MAPE) less than 3%. The promising results obtained in this study indicate that the presented DL models in this work can be implemented in future battery management systems (BMSs).

ACKNOWLEDGEMENT

First and foremost, I wish to express my sincere appreciation to my advisor and mentor, Prof. Lenwood S. Heath, for his consistent support and guidance during my M.Sc. at the department of computer science at Virginia Tech. Your intellectual guidance, mentorship and the trust that you placed in me provided an unparalleled creative environment towards finely honing my skills.

I would also like to extend my gratitude to my committee members, Dr. Naren Ramakrishnan and Dr. Michael W. Ellis for their support, suggestions, and encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION AND LITERATURE REVIEW

During the last 20 years, lithium-ion batteries (LIBs) have found their way into a large number of devices. One of the most notable reasons that made battery engineers focus on the developing LIBs is their high energy density. The other reasons that made LIBs the first option as an energy storage system can be attributed to the lack of memory effect, high cycling performance, highly low self-discharging and desirable working voltage [1]. These days, governments are trying to find a solution to mitigate the air pollution in cities by replacing conventional internal combustion engine vehicles (ICEVs) with electric vehicles (EVs). Accordingly, demands for the LIBs used in the EVs are dramatically increasing [2] and prediction of the LIBs life specifically in the EVs can help the manufacturer to build more durable EVs and detect the malfunctioning of the energy storage system in EVs. The capability to estimate state of health (SOH) and remaining useful life (RUL) in LIBs not only decrease the maintenance cost but also facilitate the transition to sustainable energy with electric cars.

1.1 Motivation and contributions

In this work, the application of machine learning toward prediction of battery state of health (SOH) is elaborated. Different machine learning algorithms are evaluated, and the combination of the algorithms have been used toward developing a robust data-driven predictive model. Recently, a research has been done [3] to predict and classify the battery using machine-learning tools. Those authors have introduced several statistical-based features showing strong correlations with the cell cycle life. Although, in their work, it has been mentioned that combining deliberate data generation with data-driven modelling can assist in predicting the behavior of the battery as a complex

dynamical system, implementation of such a predictive model in real world scenarios such as the electric vehicle (EV) industry seems to be challenging. For instance, it has been shown in [3] that using the first 100 cycle data, the cycle life can be predicted in the early stage of the cell degradation. However, in the most of real work applications such as energy storage systems (ESS) and EV industry, remaining cell capacity throughout the cell life should be monitored and only the cell cycle life prediction is not sufficient. Furthermore, in [3] , several statistical-based (e.g. variance of historical data) approaches have been used, requiring cell historical data toward building the features before applying into the data-driven predictive model. Accordingly, the aforementioned issues highlight the need for a state of health (SOH) estimation of Li-ion battery performance to obtain the variations of the cell capacity throughout the battery cell's life, to give an accurate estimation of the cell's remaining capacity at any point. In this work, no feature extraction from the raw charging/discharging data, which requires a lot of domain-specific experience, has been used to build the data-driven predictive model. In fact, the capability of the deep learning models has been employed to conduct the efficient feature extraction from the raw data (current, voltage, and temperature) directly.

Organization of the document

The organization of this document is as follows:

- **Chapter 1**: **Introduction and Literature Review**

In this chapter, the state of health (SOH) concept in Li-ion batteries as the key problem in this research is explained and the role of machine learning to predict the battery state is elaborated. Also, the motivations and the contributions are explained in this chapter.

- **Chapter 2**: **Data Set Explanation**

The data set that has been used in this work has been shown and discussed in this chapter.

- **Chapter 3**: **SOH Model Development Using Deep Learning Methods**

In this chapter, different deep learning models have been employed and explained. Using three different performance metrics, the robustness of data-driven models toward SOH prediction is discussed.

- **Chapter 4**: **Conclusion and Future Work**

  In this chapter the conclusion, limitations of the current work, and the recommendations for the future work are presented.

1.2 Literature Review and Background

There exist several mechanisms for LIB degradation that need to be considered for developing the SOH and RUL models. Over the years, a large number of models are developed that are generally divided into three groups: (i) empirical models, (ii) physics-based models and (iii) data-driven models.

The empirical models are usually obtained from curve-fitting on the experimental battery aging data and usually operate online in a recursive implementation such as a Kalman filter [4-11].

In some cases, the input parameters are also obtained from the physics-based models; they are called semi-empirical models [12-19]. Since, the battery aging process is a complex phenomenon, if the operational condition of a LIB deviates from that of experimental data, the empirical models lack sufficient capability to estimate SOH.

Physics-based (electrochemical) models are obtained through applying partial differential equations (PDEs) to characterize the aging phenomena. In these models, the charge transfer, mass transfer and heat transfer equations are derived and solved using numerical methods such as finite element (FE), finite difference (FD) and finite volume (FV) methods. Although using these models is computationally expensive, the accuracy of the results obtained from the physics-based model are not satisfactory. As a result, in recent years, battery engineers are trying to develop data-driven methods to obtain more accurate and computationally efficient models. Since, the machine-learning based models can represent the nonlinearity involved with the complexity of the battery aging mechanisms, they have become pioneers among the existing models.

1.3 Battery State of Health Estimation: State-of-the-Art

In order to obtain the battery SOH, several machine-learning based models have been developed and introduced by other researchers. One of the most popular machine learning algorithms is the artificial neural network (ANN) that has been used extensively for SOH estimation. In order to define the SOH ground truth for the training process in an ANN, SOH needs to be defined based on the battery quantified specs. Two main battery parameters can be used to define the SOH are battery cell capacity and internal resistance. In order to define the battery cell capacity, several methods have been introduced in the literature. However, the reference performance test (RPT) has been recognized as the most reliable test to measure the cell capacity. In fact, the RPT test can help the measurement to avoid being affected by environmental variables such as temperature that might cause error in the measurement. For the capacity measurement in the RPT, a fully charged cell needs to be discharged with a constant C-rate (1C or 0.2 C). It is recommended that the cell be discharged with C-rate 0.2 to avoid temperature increase within the cell. In order to save time, a 1C discharge rate is also acceptable as long as cell temperature is prevented from highly increasing. The SOH based on the capacity fade can be defined as follows:

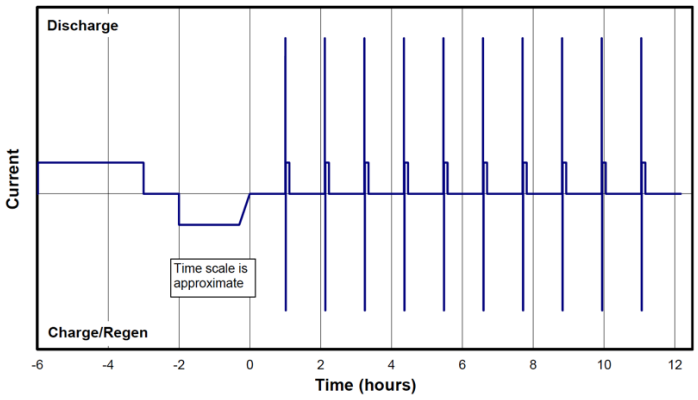
$$SOH_C = (1 - \frac{Capacity_{aged\ cell}}{Capacity_{fresh\ cell}}) \quad 1\text{-}1$$

Also, the SOH can be defined based on the cell internal resistance as follows:

$$SOH_R = (\frac{R - R_{aged}}{R_{fresh} - R_{aged}}) \quad 1\text{-}2$$

In this equation, the R_{aged} is the cell resistance and the end of life (EOL) and the R_{fresh} is the cell internal resistance at the beginning of life (BOL). To measure the cell resistance, a hybrid pulse

power characterization (HPPC) test needs to be conducted. A typical HPPC test profile can be shown in the following figure.



A typical HPPC test profile to measure the battery cell internal resistance [20]

Apart from the aforementioned methodologies to measure the cell capacity, in many studies, an assumed constant current followed by constant voltage (CCCV) profile is used to obtain the SOH based on the energy stored in the cell. In fact, SOH can be calculated through normalizing the stored energy in the cell against the cycle number. A typical battery SOH against the cycle number can be shown in Figure 1-1.

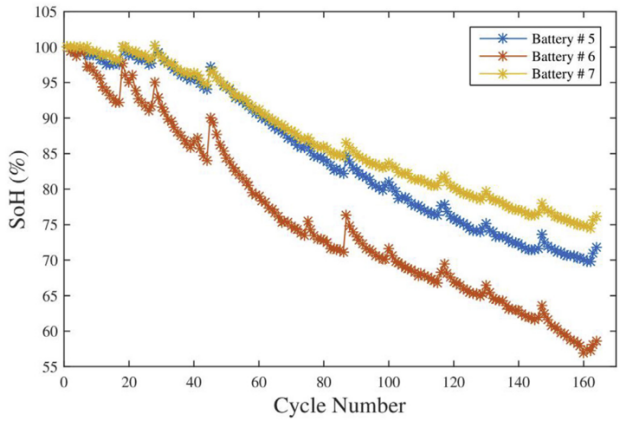


Figure 1-1. The typical SOH against cycle number for different batteries [21]

The literature review shows that different features have been used to estimate the SOH by different researchers. In [21], the voltage curves has been used to build the features toward training an ANN. To obtain the features, the first data collection interval (DCI) is defined for the cell voltage during the charging session. Then the voltage and the voltage slope (velocity) have been used as the input features to train the model. Also, a group method of data handling (GMDH) polynomial neural network (PNN) has been employed in this work to estimate SOH and the relation between input and output is shown in Eq. 1-3, where X is the input vector, x_i , x_j and x_k are the input. variables, c_{ijk} weights and N is the number of the input variables.

$$f(\mathbf{X}) = c_0 + \sum_{i=1}^{N} c_i x_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} x_i x_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} c_{ijk} x_i x_j x_k + \cdots \qquad 1\text{-}3$$

The structure of their algorithm is shown in Figure 1-2. The NASA battery data set has been used in this work.

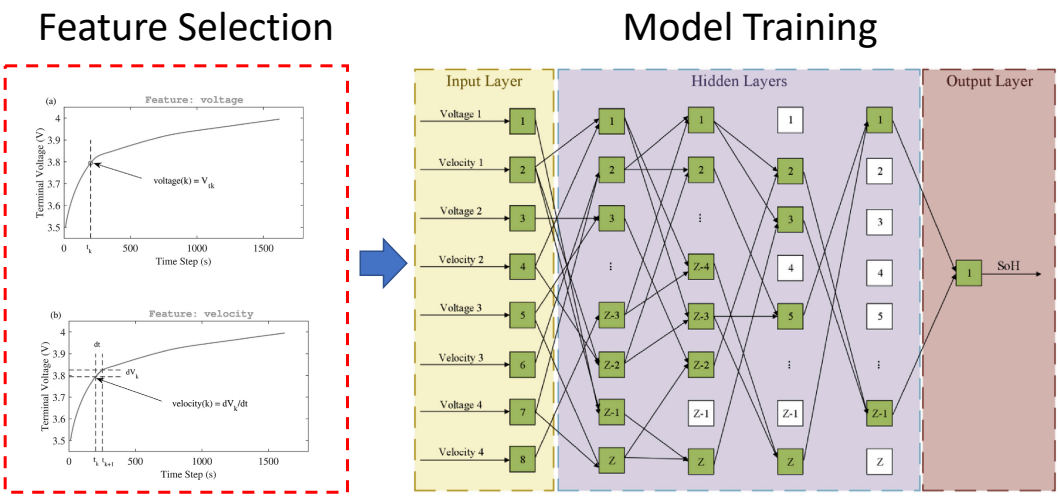


Figure 1-2. The feature selection and the group method data handling (GMDH) polynomial neural network (PNN) used in [21]

In [3], the equivalent circuit model (ECM) representing the battery behaviors of the battery is used to obtain SOH_R. In this work, the internal resistance of the battery, which is functional of state of charge (SOC) and Temperature (T), is replaced with an ANN. Then, the ANN weights are obtained through training the cell resistance within the system. In fact, in this work, instead of using an ANN to obtain the whole battery behavior, only the internal resistance is assumed to be obtained using a trained ANN. Then, the SOH_R can be correlated with the obtained internal resistance.

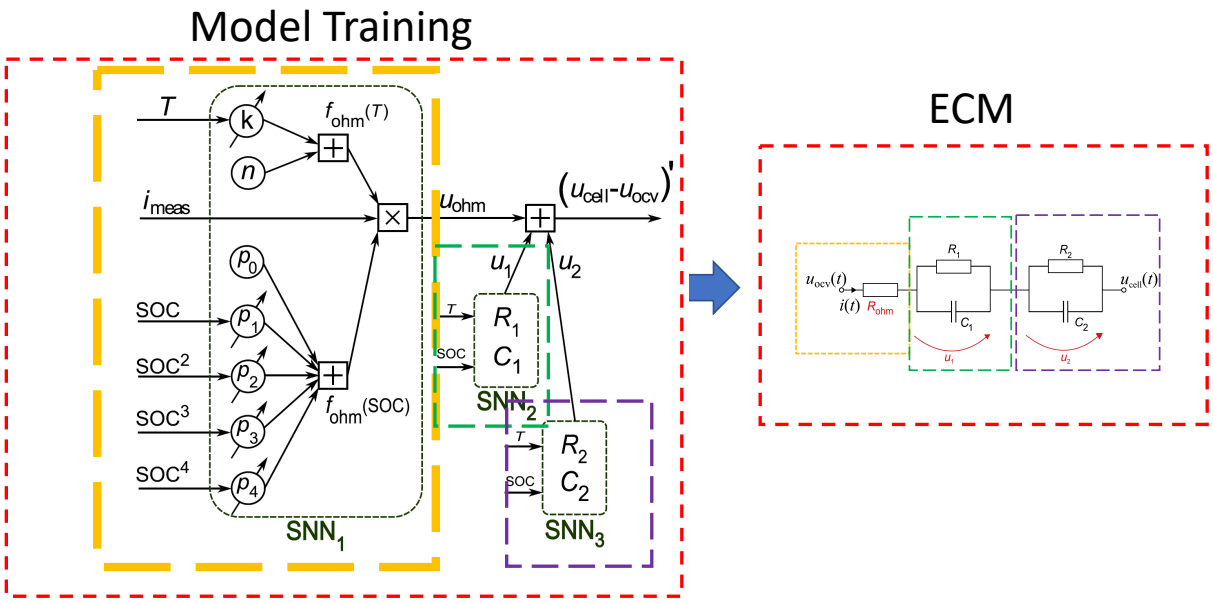


Figure 1-3. Physics-based ANN used to estimate the SOH in [3]

In [22], a probabilistic neural network (PNN) is used to estimate the SOH. In this work, constant current (CC) charging time, voltage drop at the beginning of the discharging session, and the open circuit voltage (OCV) are the features that have been used to train the model.

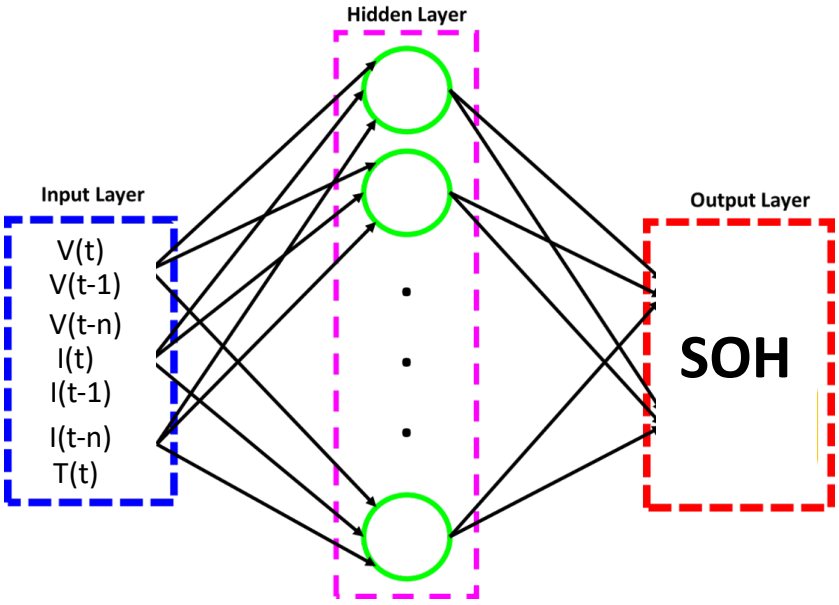


Figure 1-4. Input time-delayed neural networks (ITDNN) used to estimate SOH [23]

In [23] they have used current, voltage, and environmental temperature signals obtained from the sensors. In addition, they have used the time-delayed aforementioned signals to include the effect of the time memory into the ANN.

In another work, a similar procedure has been used to estimate SOH. However, in the time series that have used as inputs into the model, the time-delayed calculated SOH from the preview time step has been used shown in Figure 1-5.

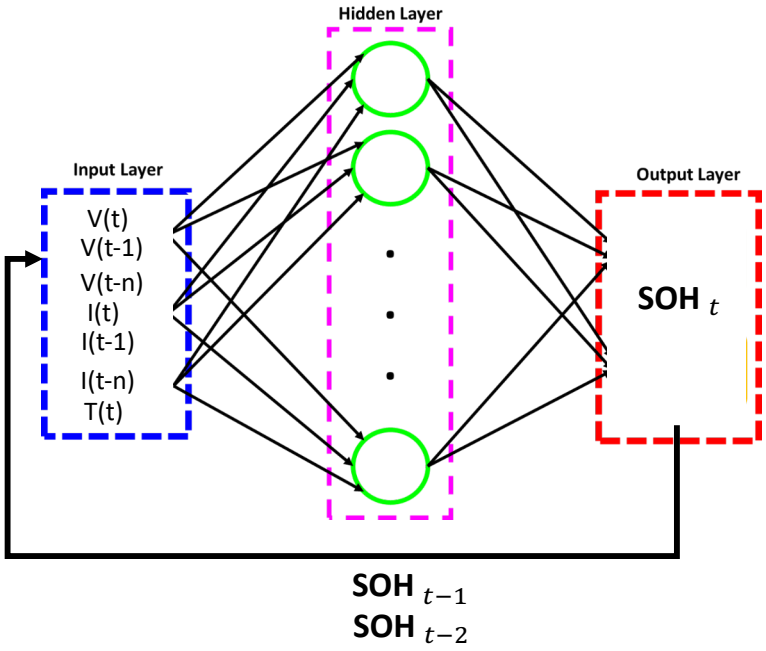
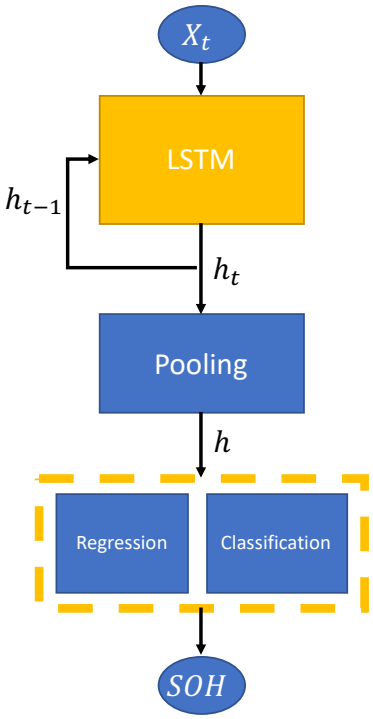


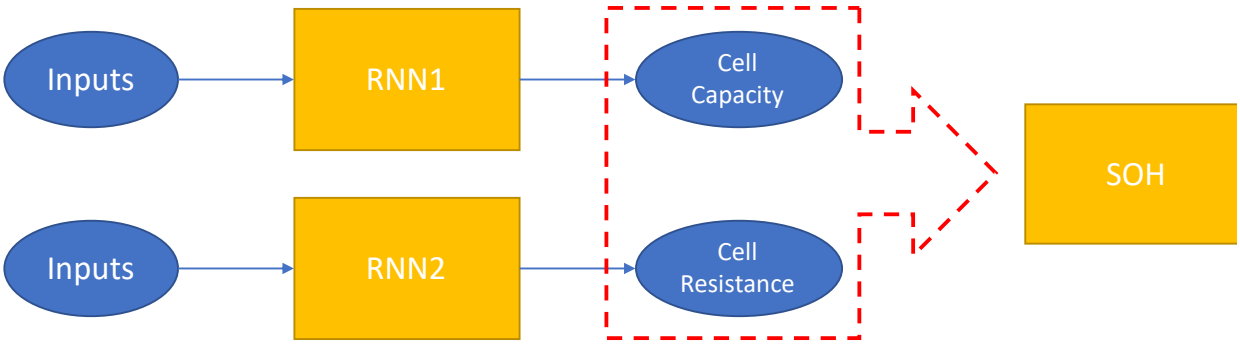
Figure 1-5. Input time-delayed neural networks (ITDNN) used to estimate SOH [24]

Another type of algorithm that has been extensively used for different applications, as well as battery state estimation, is the recurrent neural network (RNN). Since RNN enjoys a memory for the input dataflow, it can be considered an advantage to predict the time dependent phenomenon such as aging. As a result, the power of RNN to estimate the battery SOH has been reflected in some literature.



LSTM architecture to calculate SOH using regression or classification [8, 25-27]

In [26], the authors have used both internal resistance (obtained from EIS) and cell capacity to train an RNN toward SOH estimation.



LSTM architecture to calculate SOH using both internal resistance and cell capacity [26]

One of the models that has been also used to find the non-linear relationship between input features and output SOH, is sparse Bayesian models [28]. For instance, in [29], Monte Carlo and Bayesian

models have been used to obtain parameters of a predictive model representing the degradation trajectory.

In [28], the sample entropy of the voltage sequence is also used as a feature to estimate SOH. The sparse Bayesian predictive methodology (SBPM) has been used to find the correlation between the SOH and the sample entropy. The procedure to obtain the sample entropy is shown in Table 1-1.

In [30], the RNN and FNN have been combined to estimate the SOH.

Table 1-1. The procedure used in [28] to sample entropy algorithm

Step	Description
1	$\boldsymbol{U}_m(k_s) = \{s(k_s + l): 0 \leq l \leq m-1\}, k_s = 1, \dots, N_c - m + 1$
2	$d[\boldsymbol{U}_m(k_s), \boldsymbol{U}_m(n)] = max\{\lvert s(k_s + l) - s(n + l)\rvert: 0 \leq l \leq m-1\}$
3	$B_{k_s}^m(r) = \frac{1}{N_c - m - 1} W^m(k_s), k_s = 1, \dots, N_c - m$ $A_{k_s}^m(r) = \frac{1}{N_c - m - 1} W^{m+1}(k_s), k_s = 1, \dots, N_c - m$
4	$(m, r, N_c) = -\ln\left[\frac{A^m(r)}{B^m(r)}\right]$

One of the methods that has been extensively used in SOH estimation is Support Vector Machine (SVM) [31].

An SVM training data set consists of n points with input X of dimensionality S and output Y.

$$\{x_i, y_i\} \text{ where } i = 1, \dots, n.\ y_i \in \Re, X \in \Re^S \qquad 1\text{-}4$$

In order to train the SVM model, Eq. 1-5 is needed to minimize the difference between the prediction and the ground truth values.

$$f(x) = \langle w, x \rangle + b \tag{1-5}$$

The input parameters will shape the output using a kernel function, such as radial basis function (RBF) kernel k, where γ is the kernel option.

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \tag{1-6}$$

Another study [32] has combined the equivalent circuit model (ECM) with a relevance vector machine (RVM) to obtain the SOH using SVM. In fact, the RVM has been used to obtain the internal resistance for regression and the corresponding SOH for the studied battery is obtained. In another work [68], the algorithm to obtain the SOC and SOH has been combined using an adaptive sigma point Kalman filter and an SVM. It is shown in this work that the accuracy of the SOC estimation affects the accuracy of SOH and vice versa [33].

In [34], first voltage boundaries are made. Then, at each charging session, the recorded relative capacity values with voltage intervals in the specific voltage region is used as the input vector shown in Figure 1-6. In another work [35], the RF has been used to assess the importance of different factors involving the pack degradation. The summary of the features they have used to train their model is presented in Table 1-2.

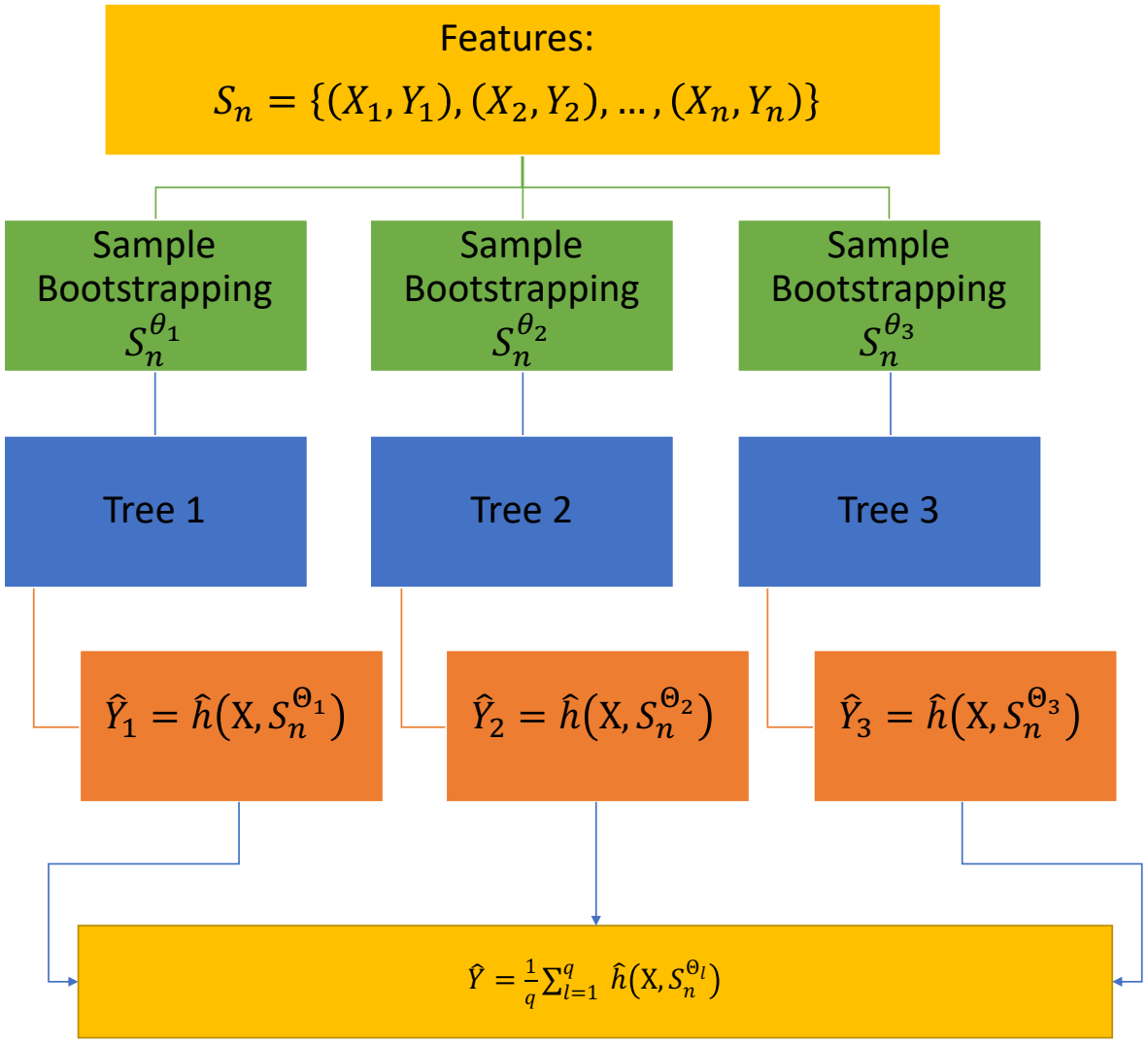


Figure 1-6. The architecture of the random forest regression (RFR) used in [34]

Table 1-2. List of the features has been used at different event to train the RF model [35]

Driving	**Charging**	**Idle (Parking)**
Total mileage since the first use	Total number of charging events since the first use	parking matrix corresponding to the time spent on parking lots split with respect to temperature and SOC
Total discharged energy since the first use	Charging power	

Histogram of the daily covered distances	State of charge variation
Histogram of average driving speed per driving event	Charging temperature
Histogram of initial SOC per driving event	
Driving matrix corresponding to the discharged energy with respect to the temperature and SOC	

1.4 Degradation Modes

The main degradation modes for a Li-ion battery can be summarized to 3 categories as follows:

I. Loss of active material (LAM)

II. Loss of lithium inventory (LLI)

III. Increase of internal resistance (IIR)

The stress factors and the degradation mechanisms corresponding to each degradation mode is depicted in Figure 1-7.

In this work, since the batteries have cycled under extreme fast charging protocols, detecting the exact degradation mechanism is highly difficult. However, two degradation mode of LLI and LAM are imagined as the main degradation modes. Some methods for estimating degradation mechanism are suggested by other researchers such as incremental capacity analysis (ICA) or differential voltage analysis (DVA), however, the investigation of the degradation mechanism is out of the scope of this work.

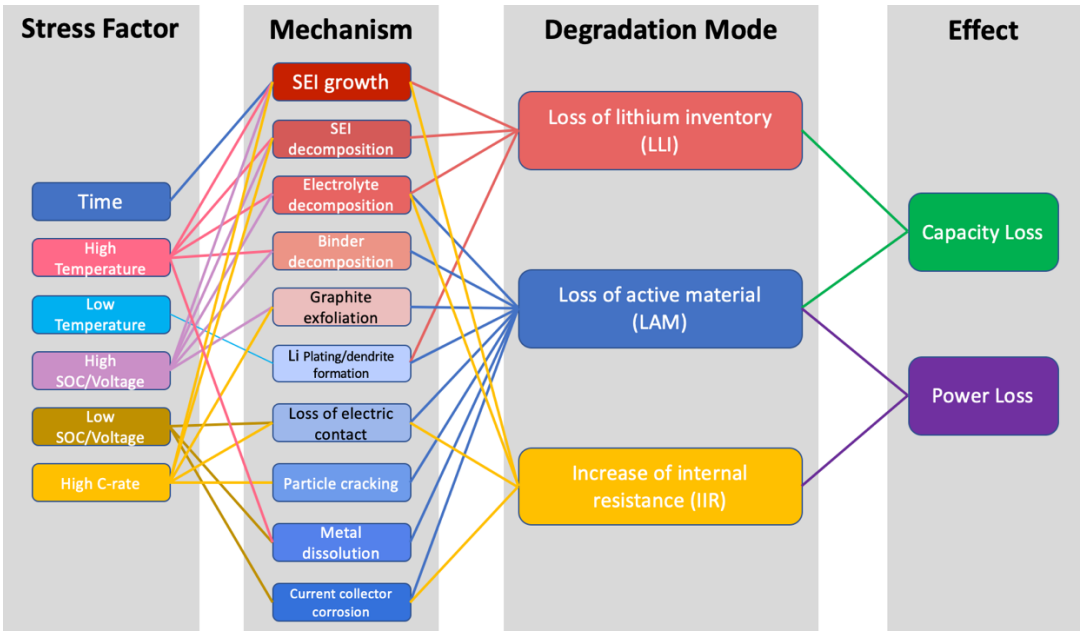


Figure 1-7. Li-ion battery degradation mechanisms and corresponding degradation modes

2 DATA SET EXPLANATION

With regard to the target of this work which is utilizing machine learning toward estimation of Li-ion battery state of health, there should be a database to provide enough information for the battery degradation. Among the data sets available, in this work, the largest publicly available datasets including more than 90000 cycles has been used. In order to obtain the cycling data at different conditions, a combination of different charging conditions has been used to build the charging protocols. A commercial battery cell has been cycled at the controlled temperature at different extreme fast-charging protocols. All the cells have been discharged at the same discharge rate (4.4 A at the 2.0 V cut-off voltage).

Table 2-1. The specifications of Li-ion battery used to generate the datasets

Brand	Model	Nominal Capacity	Nominal Voltage	Standard Charge/Discharge rate	Higher/Lower Cut-off Voltage
A123Systems	APR18650M1A	1100mAh	3.30V	1500mA/5500 mA	3.60V/2.00V
					

The charging protocols that have been used to charge the cell at different C-rates have been depicted in the following figure. The fast charging protocols are made of two constant current (CC) steps followed by a constant voltage (CV) step.

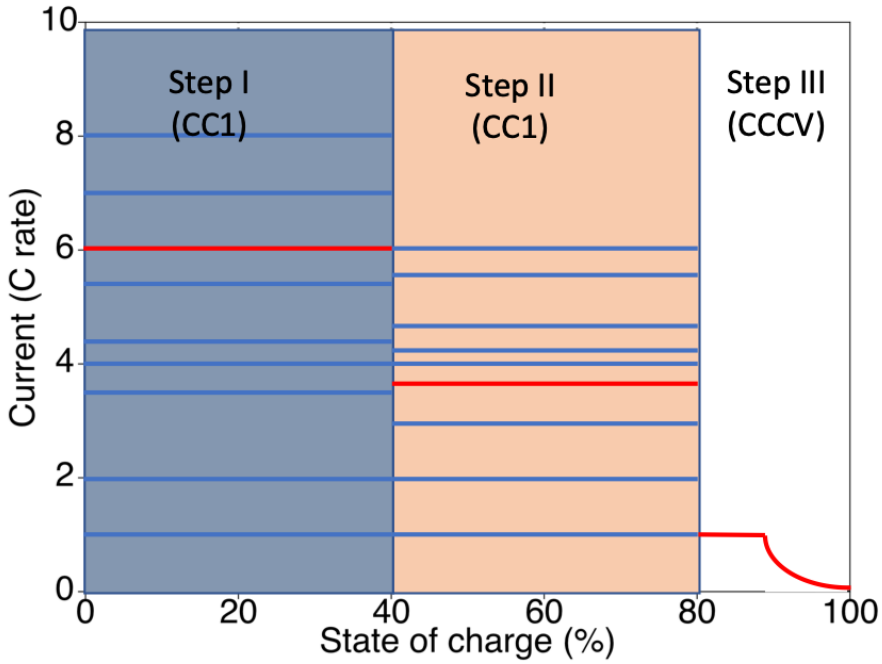


Figure 2-1.The fast charging protocols used to charge the cell at different C-rates

The combination of the protocols at the 1st and 2nd steps shape an extreme fast charging policy, which let the cell be charged up to 80% state of charge (SOC) within approximately 10 minutes. As shown in Figure 2-1, the horizontal lines at step I and step II show the first and second applied currents to the cell. In order to represent the fast charging protocols, each protocol has been shown in the format of CC1(SOC1)-CC2(SOC2)-CCCV(SOC3). The SOC1, SOC2 SOC3 represent the SOC that the cell has been charged up to during the charging step. In fact, SOC1 and SOC 2 are the SOCs at which the current switches. In this work, the SOC2 and SOC3 are always 80% and 100%, respectively. Also, the CCCV step. After the 80% SOC, all cells have been charged with constant current-constant voltage (CCCV) in which they have been charged galvanostatically at 1C to 3.6 V and then they have been charged potentiostatically at 3.6 V.

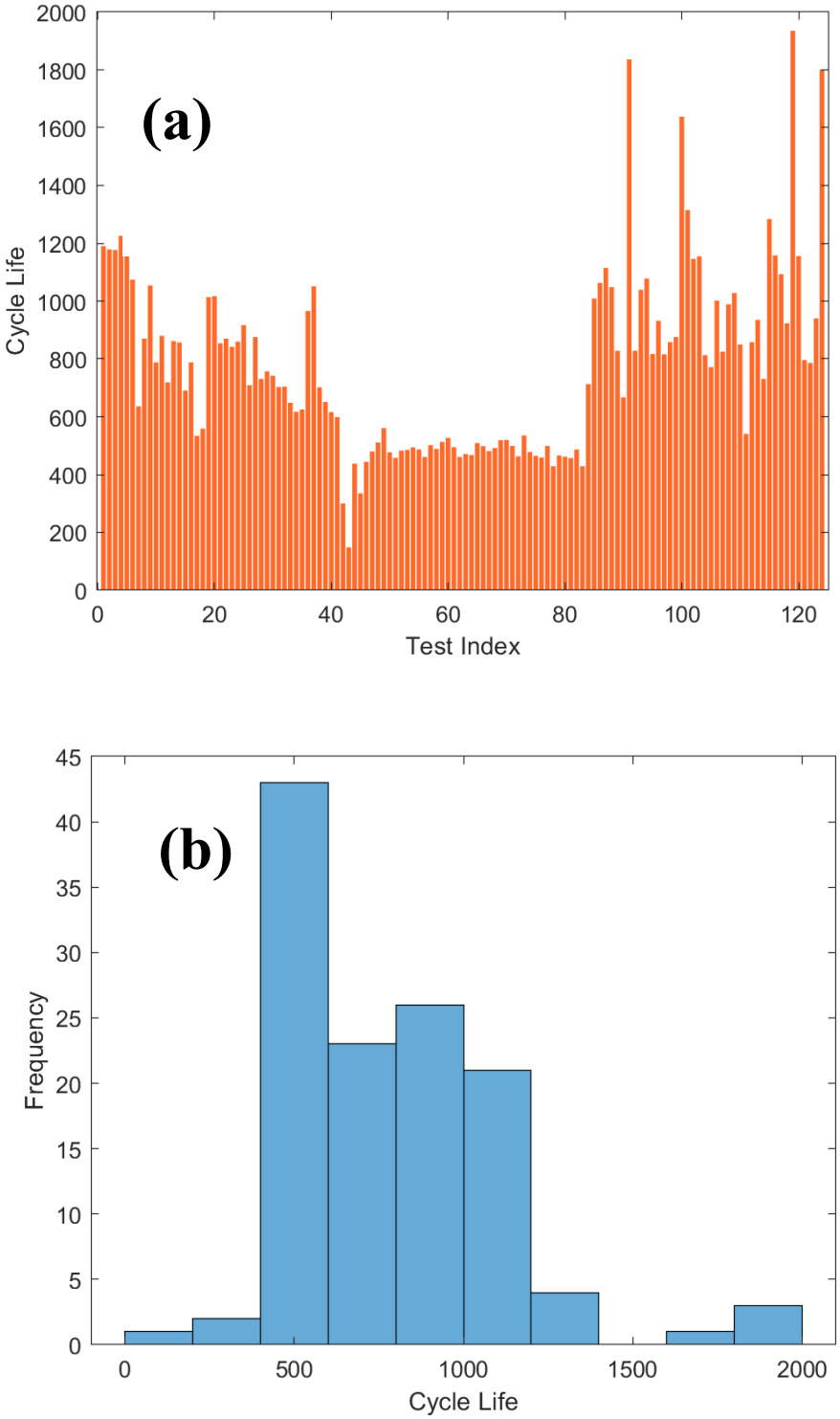
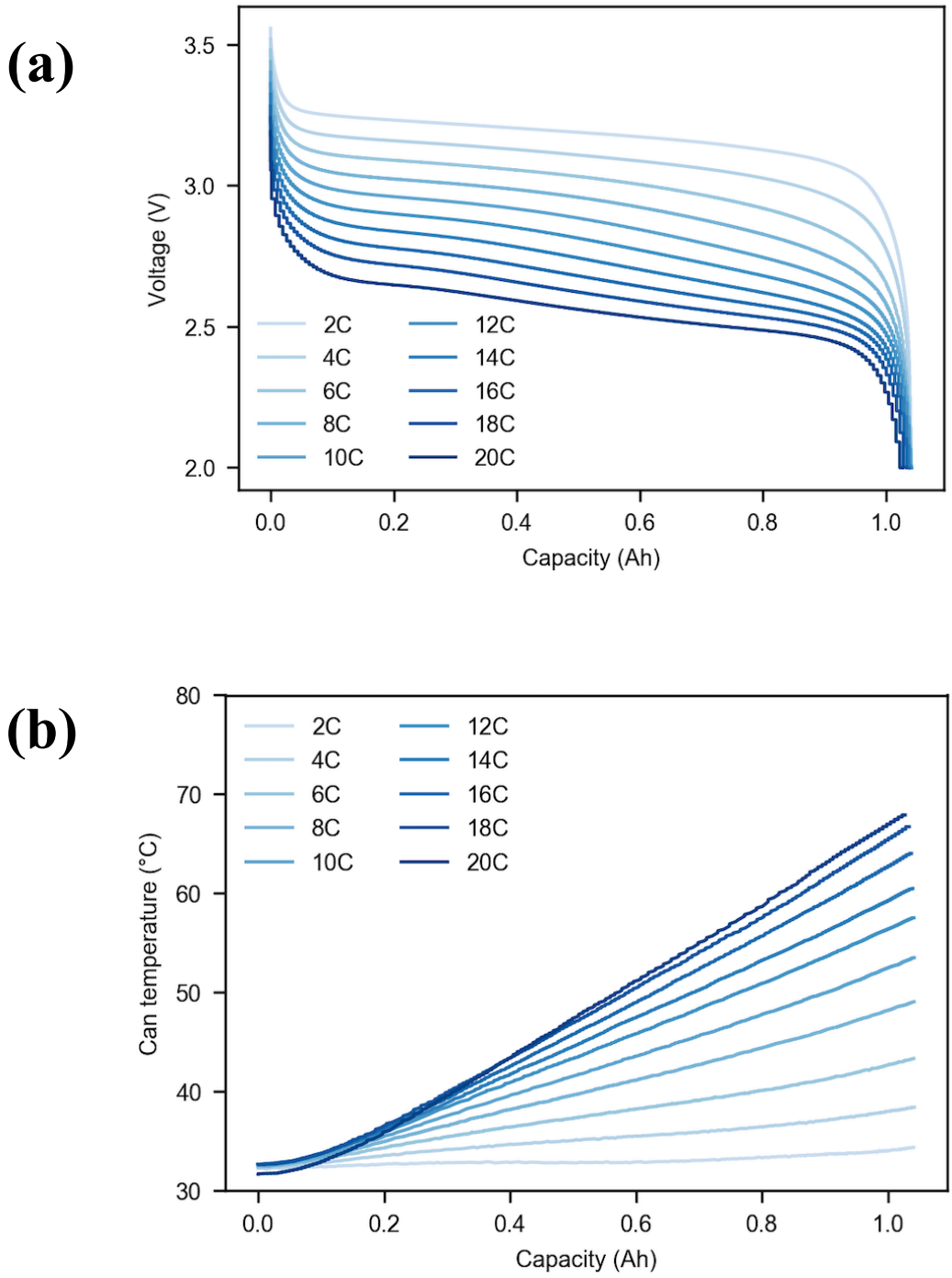


Figure 2-2. Illustration of the cells' cycle lives within the dataset used in this work

Figure 2-2 shows the distribution of the cycle life shaped within the datasets. As is obvious, the diverse cycle life from ~ 200 cycles to ~ 2000 cycles within 124 cells are shaped showing the effect of the different extreme fast charging protocols on the cells' cycle lives.



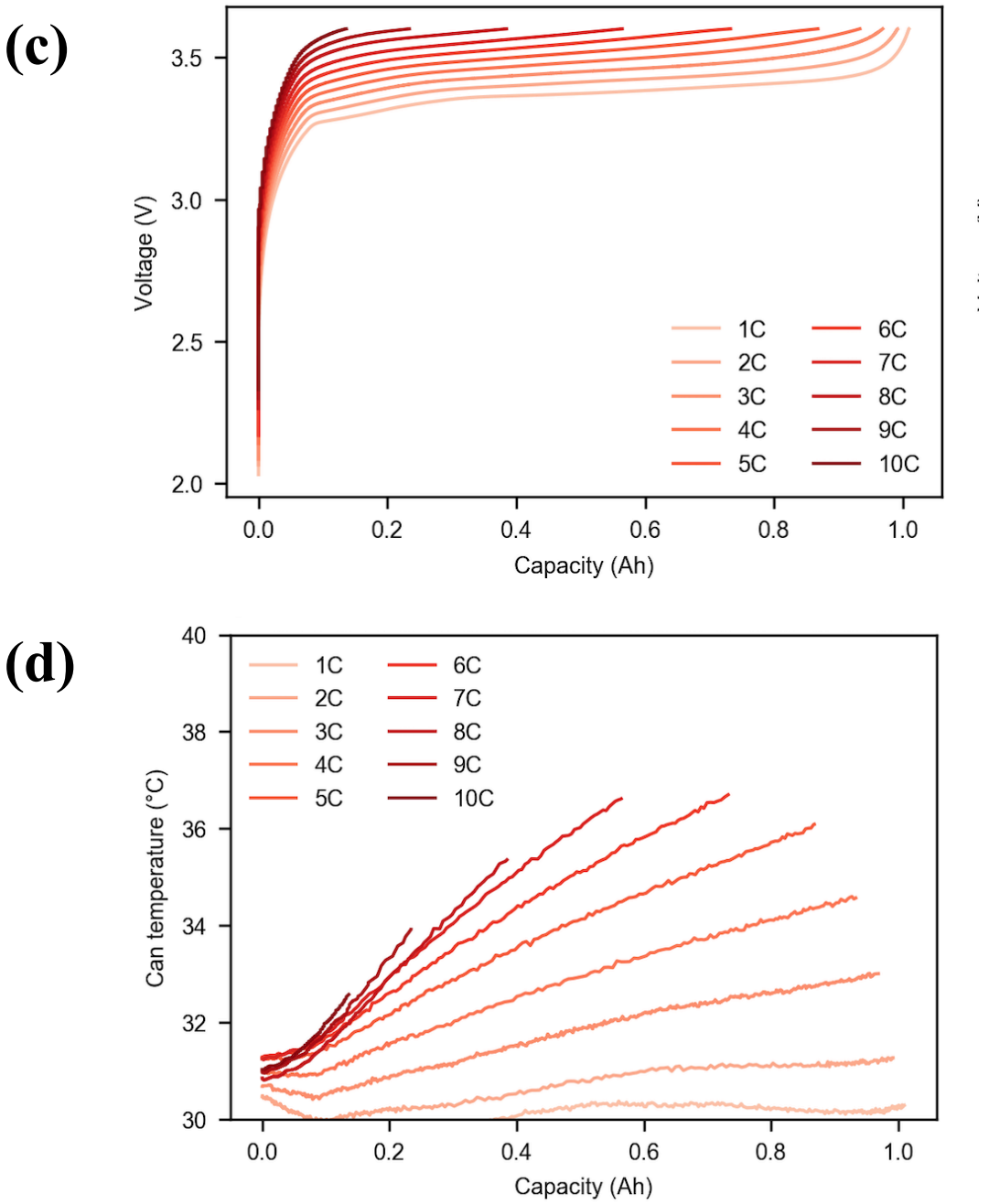


Figure 2-3. The effect of fast charging/discharging on the cell's can temperature

Since, the charging protocols that have been used to generate the data sets are made of extreme fast charging policies, considerable heat generation has been made as the irreversible reaction

within the cell. As a result of the heat generation in the cell, a variety of the temperature distribution has been observed and shown in Figure 2-3.

3 SOH Model Development Using Machine Learning Algorithms

3.1 Introduction

Recently, research has been done [36] to predict and classify the battery using machine-learning tools. Those authors have introduced several statistical-based features showing strong correlation with the cell cycle life. Table 3-1 shows most of the features that have been used in their work.

Table 3-1. The features introduced in [36] for battery cell cycle life prediction

Feature #	Description
1	Cell remaining capacity (discharge) at cycle # 2
2	Maximum change in cell remaining capacity (discharge) at cycles 1 to 100
3	Cell remaining capacity (discharge) at cycle # 100
4	Slope of the linear fit to the capacity fade curve, cycles 2 to 100
5	Intercept of the linear fit to capacity fade curve, cycles 2 to 100
6	Slope of the linear fit to the capacity fade curve, cycles 91 to 100
7	Intercept of the linear fit to capacity fade curve, cycles 91 to 100
8	Standard logarithm of minimum of ΔQ100-10(V)
9	Standard logarithm of mean of ΔQ100-10(V)
10	Standard logarithm of variance of ΔQ100-10(V)
11	Standard logarithm of skewness of ΔQ100-10(V)
12	Standard logarithm of value at 2V of ΔQ100-10(V)
13	Standard logarithm of summation of ΔQ100-10(V)
14	Standard logarithm of summation of ΔQ100-10(V) squared
15	Standard logarithm of energy difference cycles 10 to 100
16	Standard logarithm of energy difference cycles 10 to 100 squared
17	**Output: Cycle life**

In [36], it has been shown that difference of the discharge capacity curves as a function of voltage between the 100th and 10th cycles ΔQ100-10(V) can be used to obtain the cycle life. In fact, using the first 100 cycles data is enough to predict the cell cycle life.

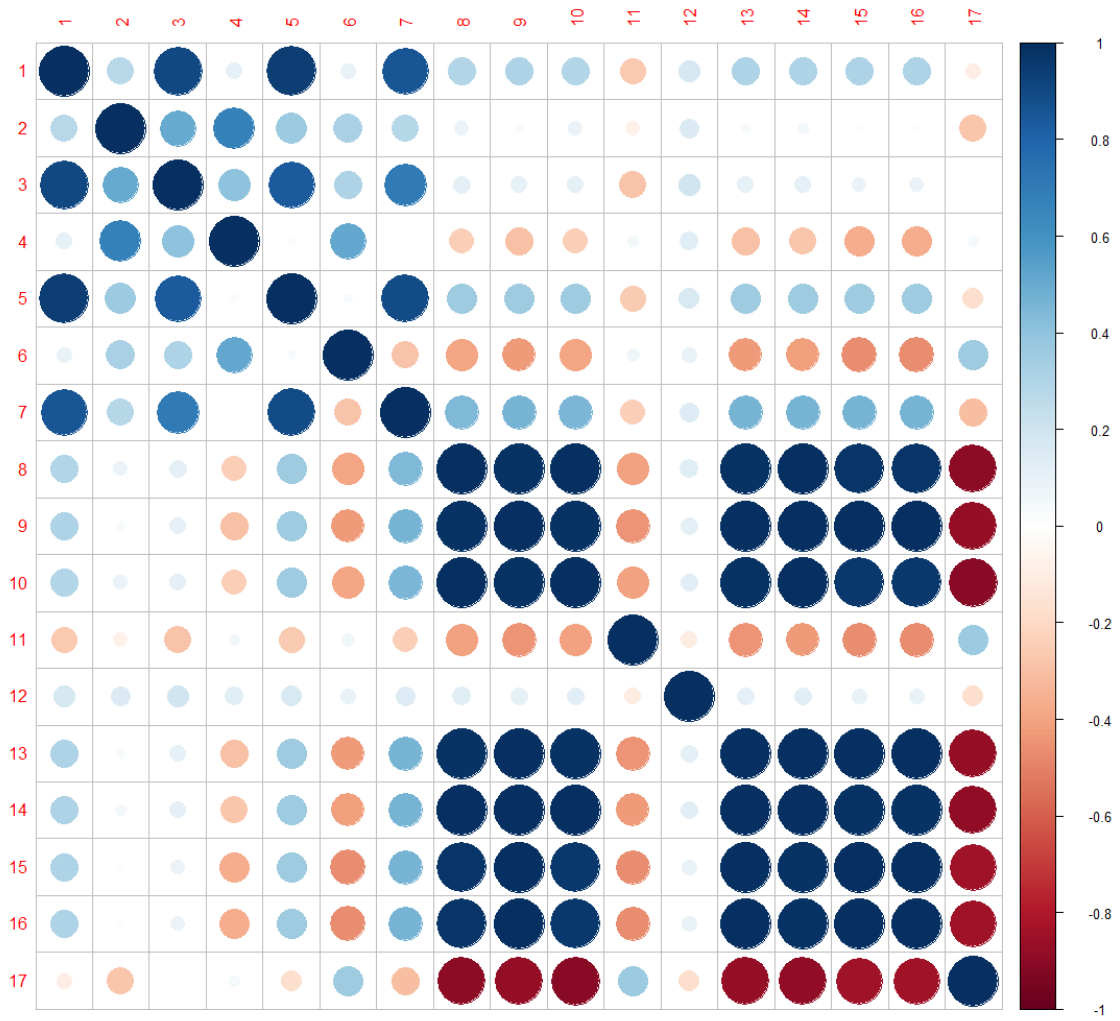


Figure 3-1. Pearson correlation matrix of the features introduced in [36]

As shown in Figure 3-1, the features obtained from the minimum, mean, and variance of the difference of the discharge capacity curves as a function of voltage between the 100th and 10th

cycles have highly strong correlations with the cycle life (feature # 17). As a result, a simple linear regression is capable of predicting the logarithm of cycle life. Although, the presented methodology in [36] shows a promising solution to predict some of the battery's complex behaviors, still it relies on much data-preprocessing to extract the statistics-based features before applying into the data-driven predictive model, which makes it challenging toward implementation for real world applications such as the EV industry. In addition, to obtain the features toward applying into the data-driven predictive model, the historical data of the first 100 cycles needs to be available.

Accordingly, the aforementioned issues highlight the need for a state of health (SOH) estimation of a Li-ion battery to obtain the variations of the cell capacity throughout the battery cell's life without feature extraction from the raw charging/discharging data, which requires a lot of domain-specific experiences to build the data-driven predictive model. In fact, the capability of the deep learning models has been employed to conduct the efficient feature extraction from the raw data (current, voltage, and temperature) directly.

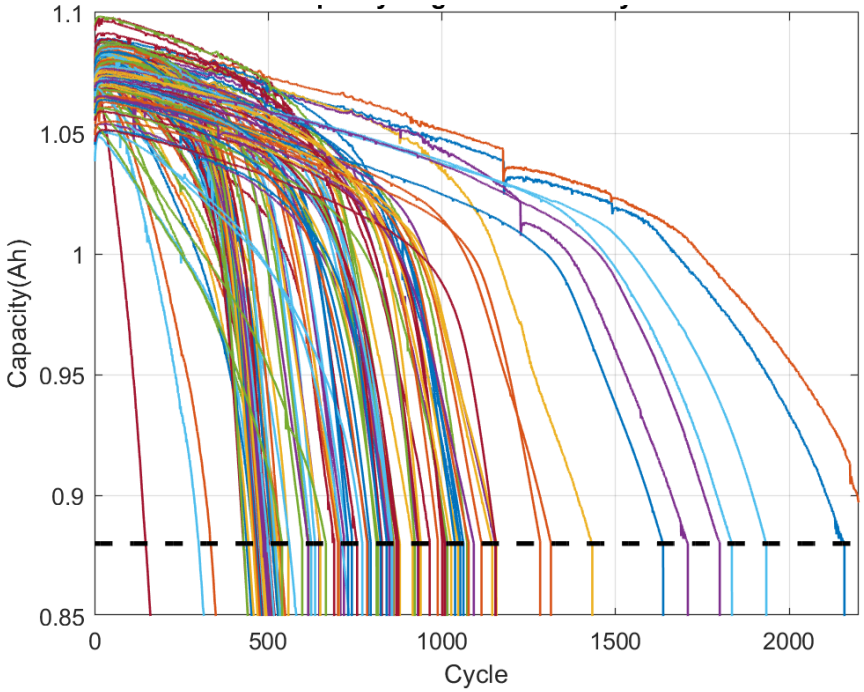


Figure 3-2. Cell capacity against the cycle number for different charging policies. The dash line shows the 80% nominal capacity

Figure 3-2 shows the cell capacities against the cycle number under different cycling conditions. As is obvious, most of the fresh cells possess discharge capacity between 1.05 to 1.10 Ah; however, they will end up with different degradation paths. In the EV industry, 80% of the nominal fresh cell capacity is considered as the end of life (EOL). In fact, at the end of life, the cell degradation rate highly increases, and it reaches to the "elbow point" shown with a dash line in Figure 3-2. Another point that needs to be mentioned regarding the Figure 3-2 is the different initial capacity for different cells. Mainly, the cell manufacturing and formation process cause some deviation of the fresh cell capacity from the nominal capacity defined by the cell manufacturer, in this study, a maximum deviation of 4.5% from the nominal capacity is observed for all 124 cells.

3.2 General Deep Learning Models Architecture

In this part, all the methods that have been used to build the deep learning models are explained. Before explaining the deep learning architectures, a multiple regression (MR) and a multiple auto regression (MAR) model have been employed and their accuracy is discussed as follows:

$$y_n = \beta_0 + \beta_1 y_{n-1} + \beta_2 y_{n-2} + \cdots + \beta_{n-1} y_1 + \epsilon_n \quad (3\text{-}1)$$

The multiple auto regression model is shown in Eq. 3-1, the output from the previous cycle (SOH_{n-1}) works as the predictor for the next cycle output (SOH_n). The order of an autoregression refers to the number of immediately preceding values in the series that are used to predict the value at the present time (cycle) which is considered 10 in this work. To compare the MR and MAR models, the prediction error is calculated from Eq. 3-2.

$$Prediction\ Error = \frac{\sum_1^N \left| \frac{\widetilde{SOH_n} - SOH_n}{SOH_n} \right| \times 100\%}{N} \quad (3\text{-}2)$$

The available data from 2 different battery cells has been utilized to train the MR and MAR models and the corresponding prediction error is shown in Table 3-2.

Table 3-2. The estimation error based on the multiple regression models

Regression Model	Battery Cell #1	Battery Cell #2
MR	22.6 %	18.3 %
MAR	20.1 %	14.5 %

As it is obvious in Table 3-2, the MAR model can improve the prediction accuracy compared to a simple MR model. However, still the accuracy of the evaluated regression model is not acceptable by the industry standards; meaning that more advance models should be employed to predict SOH.

3.2.1 FFNN

Artificial neural networks are inspired computationally by human brain's performance, and they can be used to predict the complex behavior of dynamic systems. The first architecture of artificial neural networks has been made using a simple perceptron. In fact, the input has been connected to the output using linear or nonlinear functions. Researchers tried to improve the performance of artificial neural networks through increasing the number of hidden layers that connect the input to output layer. As a result, the concept of a deep neural network has been introduced that is representative of several hidden layers between the input and output within the architecture artificial neuron network. The simplest version of artificial neural networks is a feed forward neural network (FFNN) shown in Figure 3-3.

$$h^{i+1}\left(b^i, \mathbf{W}^i, h^i\right) = f\left(b^i + \mathbf{W}^i h^i\right) \qquad 3\text{-}3$$

To obtain the activations $h^i$of every layer, the activations $h^{i-1}$of the layer below is needed which can be calculated from Eq. 3-3 where $h^i$and $h^{i+1}$are the activations of hidden layers i and $i+1$. Also $\mathbf{W}^i$ is the matrix of the weight and b^i is the bias vector in layer i, which is shown with the blue attribute in Figure 3-3.

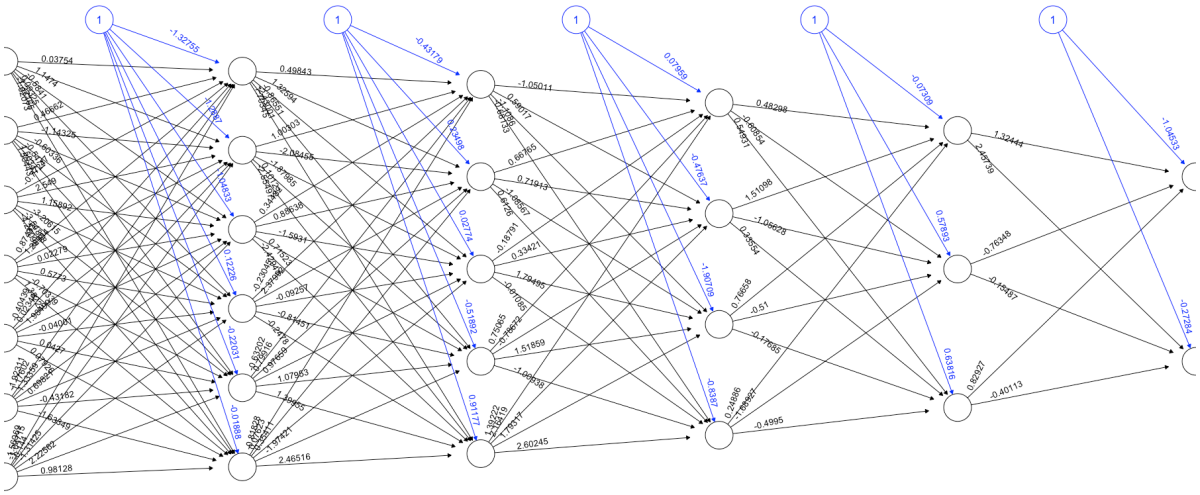


Figure 3-3. A typical artificial neural network (ANN) with multiple hidden layer

Different functions f can be used as the activation function. In this work, we have mostly used the Hyperbolic Tangent, Sigmoid, and Rectified Linear functions shown in Eq. 3-4, 3-5, and 3-6.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3-4}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3-5}$$

$$\text{ReLu}(x_i) = max(x_i, 0) \tag{3-6}$$

In order to train the model, we need to define a cost function (error) and minimize the cost using optimization procedures. Gradient descent is the method that is highly popular to train the model in which backpropagation is utilized to obtain the gradients. If we assume that the cost function will be as in Eq. 3-7, then the gradient of the cost function can be calculated to obtain the weight matrix of the output layer through Eq. 3-8 to 3-13.

$$E(x, \hat{y}, \theta) = \frac{1}{2}\|f_\theta(x) - \hat{y}\|^2 \quad 3\text{-}7$$

$$h_j^{i+1}\left(b_j^i, \mathbf{W}_j^i, h^i\right) = \tanh\left(b_j^i + \mathbf{W}_j^i h^i\right) \quad 3\text{-}8$$

$$\frac{\partial E}{\partial \mathbf{W}_{ij}^{N-1}} = \frac{\partial \frac{1}{2}\|f_\theta(x) - \hat{y}\|^2}{\partial \mathbf{W}_{ij}^{N-1}} = \quad 3\text{-}9$$

$$(y_i - \hat{y}_i)\frac{\partial f_\theta(x)}{\partial \mathbf{W}_{ij}^{N-1}} = (y_i - \hat{y}_i)\frac{\partial h_i^N}{\partial \mathbf{W}_{ij}^{N-1}} = \quad 3\text{-}10$$

$$(y_i - \hat{y}_i)[1 - \tanh^2(b_i^{N-1} + \mathbf{W}_i^{N-1} h^{N-1})]h_j^{N-1} \quad 3\text{-}11$$

$$\frac{\partial E}{\partial \mathbf{W}_{ij}^{n-1}} = \delta_i^n \frac{\partial h_i^n}{\partial \mathbf{W}_{ij}^{n-1}} \quad 3\text{-}12$$

$$\nabla_\theta E_\theta(x) = \left\{\left(\frac{\partial E}{\partial \mathbf{W}^i}, \frac{\partial E}{\partial b^i}\right) \mid i = 0 \cdots N - 1\right\} \quad 3\text{-}13$$

Using this method, we need to calculate the derivative of the cost function towards the parameters $\nabla_\theta E_\theta(x, \hat{y})$. As a result, we can obtain the vector of the parameters in the space of θ showing the direction of increasing cost. As a result, the cost can reduce through moving in the opposite direction. This method is sometimes challenging, as $\nabla_\theta E_\theta(x, \hat{y})$ is a local measure and only provides the gradient for the particular input X and parameters θ.

The parameters change as the input changes, and, as result, the gradient can change. A solution for this issue is to select part of the input dataset through randomly sampling and conduct a small update to the parameters in the opposite direction of the gradient $\theta = \theta - \lambda \nabla_\theta E_\theta(x, \hat{y})$. In this approach, λ_o is the initial learning rate and we can adjust it to decrease over time.

$$\lambda_{\text{learning}}(n) = \lambda_o \cdot \gamma^n \quad 3\text{-}14$$

For instance, the learning rate can exponentially decay at the n'th iteration shown in Eq 3-14.

3.2.2 Convolutional Neural Network (CNN)

Although the artificial neural network is highly flexible and powerful to find the complexity of the relation between the input and output throughout its hidden layers, the computational load grows dramatically as the size of the input or the number of the hidden units increase.

As a result, it has been tried to find an efficient way to connect the input to the output layer. For the first time, this innovative solution has been tried in the image recognition problem. In this approach, despite the neural networks that all the units from one layer get connected to the all units in the next layer, the neurons connect only locally to neurons that correspond to the same neighborhood and representative of the parts from the same object.

Accordingly, the convolutional neural networks (CNN) has been introduced.

In CNN, instead of using matrix multiplication, convolutions will be used. The operation of 2-dimensional convolution is shown in Eq. 3-15.

$$S(i,j) = (X * W)(i,j) = \sum_{m} \sum_{n} X(i-m, j-n) W(m,n) \qquad 3\text{-}15$$

where X is an input matrix and W is a kernel matrix used to build the CNN architecture. A typical structure for a CNN architecture, shown in Figure 3-4, consists of a convolution layer, a pooling layer and a fully-connected layer.

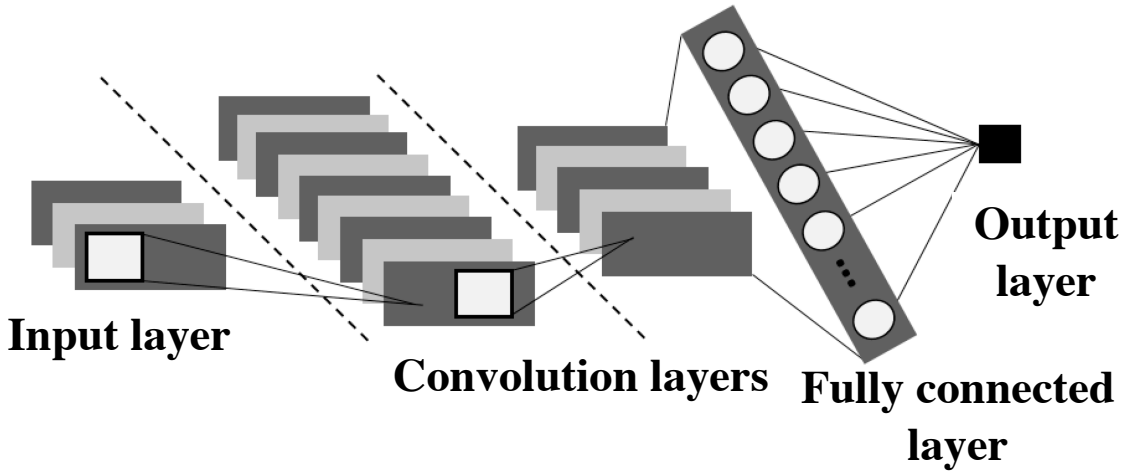


Figure 3-4. The structure of convolution of neural network (CNN) used in this work

In a CNN, the convolution layer extracts the features from the input data and send them to the activation function. In addition, to avoid a huge number of channels sent to the activation function, we can use a pooling layer such as “Maxout Pooling” to keep the maximum value for local groups of channels and provide robust learning. The architecture of the CNN can be made of several convolution and pooling layers toward shaping global features. The max pooling step in this work has been done through the following steps:

I. Pick a window size (usually 2 or 3).

II. Pick a stride moving range of pixels (usually 2).

III. Slide the window across the filtered input

IV. For each window, we take the maximum value.

The last two layers of the CNN in this work are made of a fully connected layer and a regression layer. In the fully connected layer, there is voting by the set of values to determine the output of the regression layer.

3.2.3 Long Short-Term Memory (LSTM)

When the nature of the problem is involved with the time series data, inputs are constantly changing, the time is a main factor in the analysis. Long Short-Term Memory (LSTM) is a kind of RNN that can consider the time memory for the future prediction.

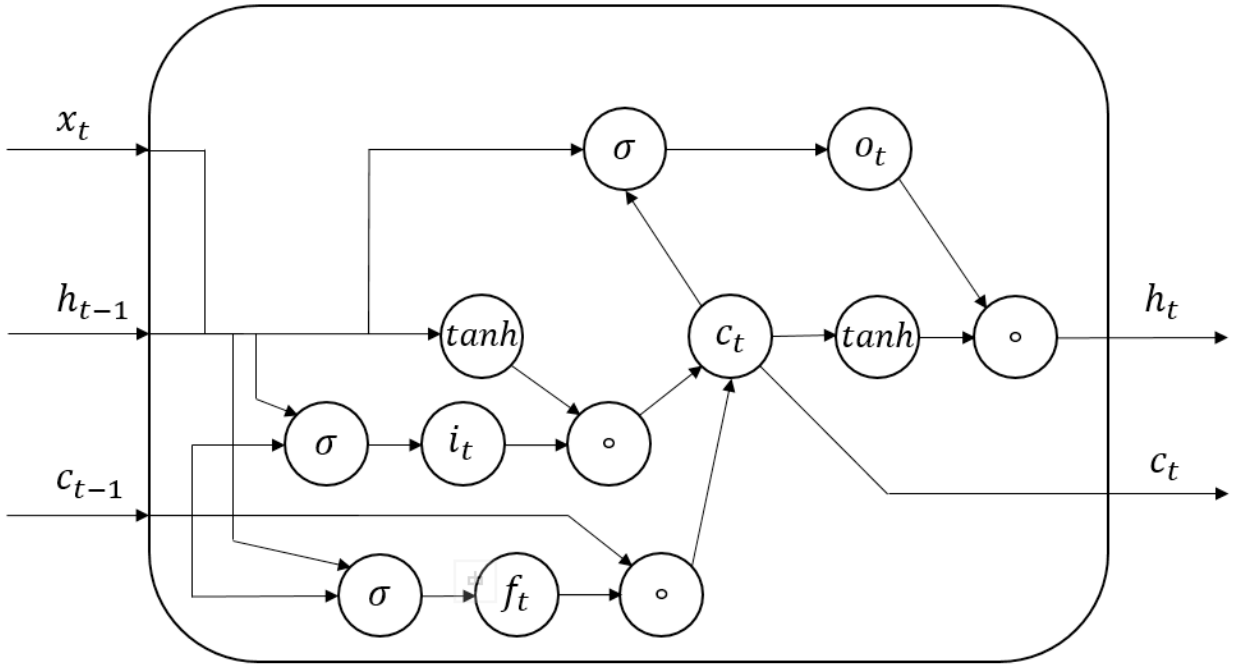


Figure 3-5. The LSTM architecture used in this study

Figure 3-5 shows the overview of the LSTM used in this study. It is obvious that LSTM can get the inputs and unroll the input data in time toward learning from each time step. More specifically, f_t, i_t, c_t, o_t, and h_t are calculated from Eq. 3-16 to 3-20 as follows:

$$f_t = \sigma\left(W_{fc} \circ c_{t-1} + W_{fh} h_{t-1} + W_{fx} x_t + b_f\right) \quad 3\text{-}16$$

$$i_t = \sigma(W_{ic} \circ c_{t-1} + W_{ih} h_{t-1} + W_{ix} x_t + b_i) \quad 3\text{-}17$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh\left(W_{ch} h_{t-1} + W_{cx} x_t + b_c\right) \quad 3\text{-}18$$

$$o_t = \sigma(W_{oc} \circ c_t + W_{oh} h_{t-1} + W_{ox} x_t + b_o) \quad 3\text{-}19$$

$$h_t = o_t \circ \tanh\left(c_t\right) \quad 3\text{-}20$$

where σ and tanh are the sigmoid and hyperbolic tangent activation functions, w are the different weight matrices, and x_t is the input at the present timestep, respectively. In addition, h_t and $h_{t-1}$are the cell at current time step and the previous time step, respectively. C_{t-1} is the cell's previous memory state, and C_t is the cell's current memory state. The ($\circ$) operator denotes the Hadamard product. Also, there are three gates, namely the input gate i_t, the forget gate g_t and the output gate o_t.

3.2.4 Convolutional Long Short-Term Memory (ConvLSTM)

ConvLSTM has the capability to enhance the ability of LSTM to store the extracted information and it contains hidden state and cell state units. In contrast with LSTM where the input and hidden states are multiplied by the weights directly, in ConvLSTM, the convolution with the weights needs to be applied which is shown in Figure 3-6.

$$f_t = \sigma\left(W_{fc} \circ c_{t-1} + W_{fh} * h_{t-1} + W_{fx} * x_t + b_f\right) \quad 3\text{-}21$$

$$i_t = \sigma(W_{ic} \circ c_{t-1} + W_{ih} * h_{t-1} + W_{ix} * x_t + b_i) \quad 3\text{-}22$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh\left(W_{ch} * h_{t-1} + W_{cx} * x_t + b_c\right) \quad 3\text{-}23$$

$$o_t = \sigma(W_{oc} \circ c_t + W_{oh} * h_{t-1} + W_{ox} * x_t + b_o) \quad 3\text{-}24$$

$$h_t = o_t \circ \tanh\left(c_t\right) \quad 3\text{-}25$$

The (∘) operator and (∗)denote to the Hadamard product the convolution operator, respectively. It is noteworthy to mention that if the size of the kernel is changed to 1×1, the ConvLSTM will be changed to the LSTM architecture.

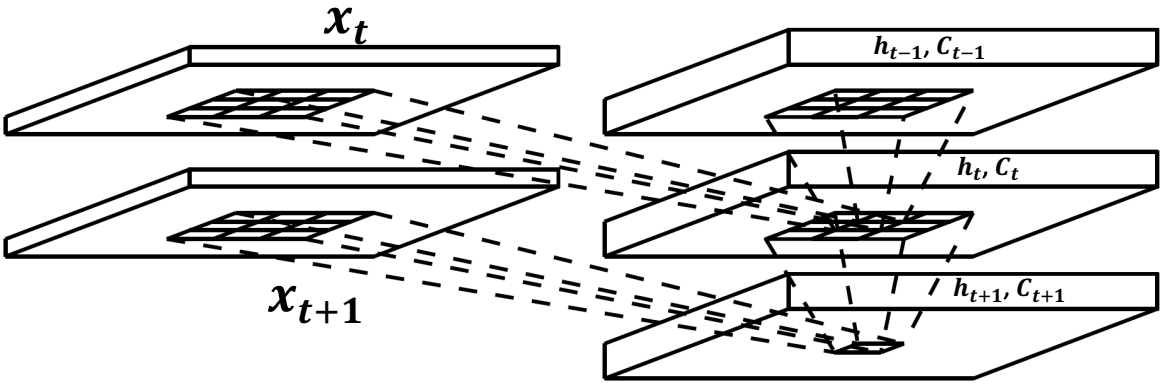


Figure 3-6. ConvLSTM architecture used in this work

3.3 Traditional Methods for SOH Estimation

In the traditional ML model development to predict the SOH, some algorithms are required to extract the input features representative of cell capacity such as the cell charging time, the sample entropy, initial cell temperature, charge voltage, and final charge current [37-41]. Figure 3-7 summarizes the difference between the conventional ML methods and the novel deep learning methods.

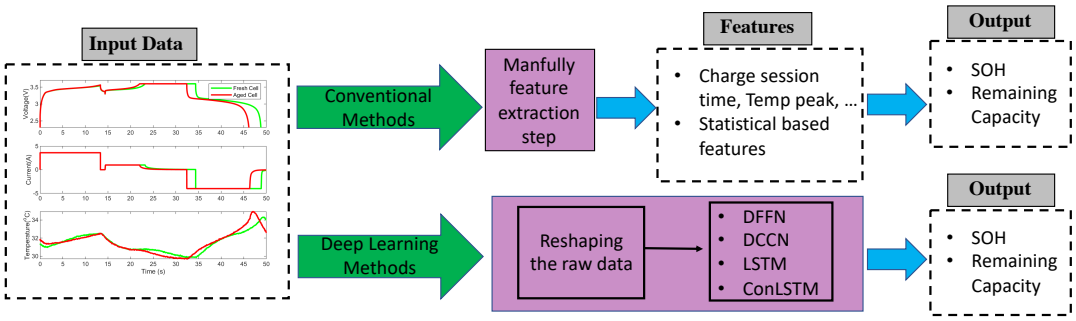


Figure 3-7. The conventional machine learning and deep learning methods for battery cell SOH estimation

There are some drawbacks associated with the conventional SOH estimation method. For illustration, one of the features used in conventional methods has been evaluated in the existing data sets employed in this work.

For instance, in the conventional machine learning methods for the online estimation of the battery SOH, charging time has been extensively utilized as the main input feature to build the predictive model. However, in this work, it has been observed that there is inconsistency between the charging time as an input feature and the corresponding remaining capacity or SOH.

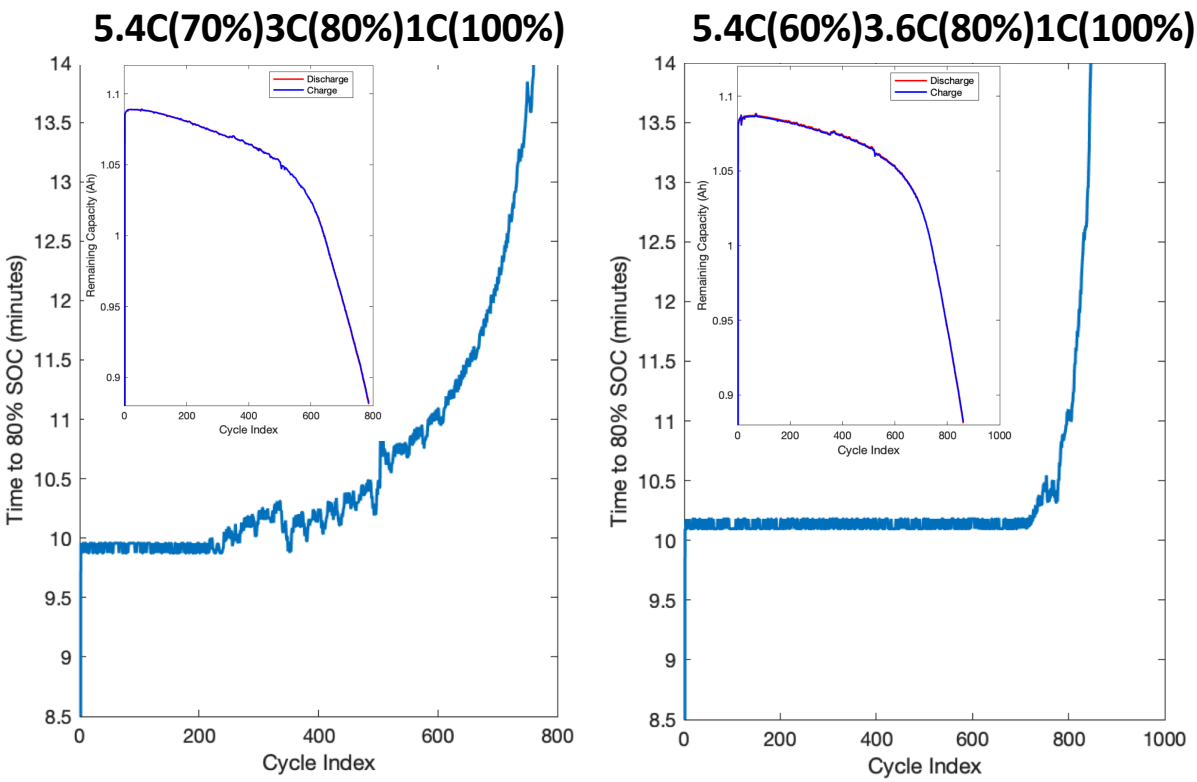


Figure 3-8. The charging time to 80% SOC against the cycle number for two different charging protocols

Figure 3-8 shows the charging time from 0% to 80% SOC versus the cycle number for two different charging protocols. It is obvious that, for the cell with the charging protocol 5.4C(70%)3C(80%)1C(100%) Figure 3-8 (left), the charging time is highly noisy and training the model with such a feature brings considerable uncertainty into the prediction from the ML model. In addition, the experimental data for the cell with charging protocol 5.4C(60%)3.6C(80%)1C(100%) shown in Figure 3-8 (right) shows that the charging time does not change with the cycle number up to cycle 700. Accordingly, it cannot provide sufficient information for the ML model to predict the SOH. In contrast with the tradition ML method to extract features toward training the model, in this work, we have employed the complete set of raw

data during the cell partially charge session as the input without the extraction or selection of characteristic features toward training different deep learning model.

3.4 Methodology

In this part, the methodology that the models have been trained and tested are explained. The original dataset has been made of 124 cell data. See Figure 3-9.

In this work, first the cells have been categorized in 4 groups based on the cycle life. The minimum and the maximum cycle life are148 and 1935, respectively. In order to obtain the number of the cell at each category, a constant bin width of 490 cycles has been used and the distribution of the cycle life within the 4 groups has been obtained.

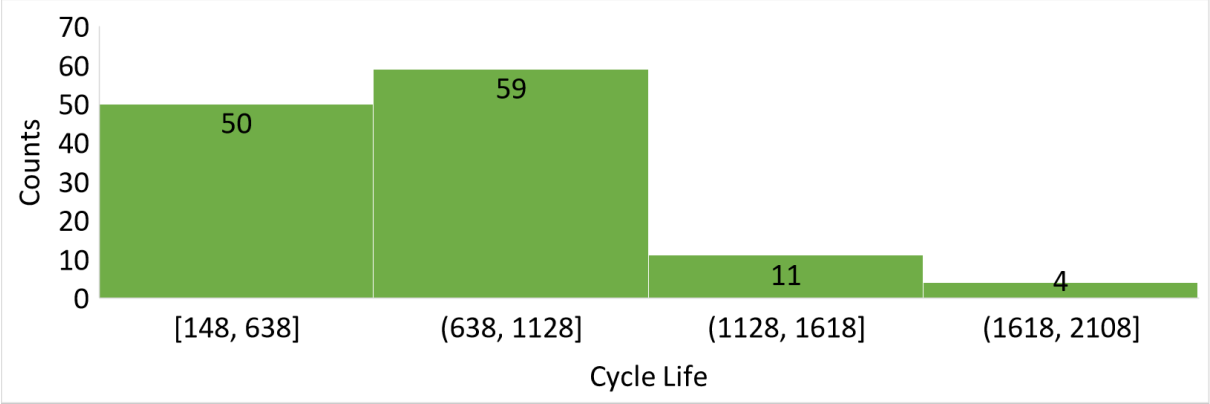


Figure 3-9. The number of the cell distributed at each bucket with different cycle life

Most of the cells have cycle life between 148 and 1128 cycles. Also, the average cycle life of all the cells is 774 cycles. From each group, randomly one cell has been sampled and totally 4 cell data have been used to train the model. As it is explained in the previous section, the data sets have been made of current, voltage, and temperature time series for each cell at each cycle. Since, in the real world, specifically in the EV application, there exist more possibilities to control the current during the cell charging than during discharging, in this work, the data during the charge session

has been used to build the model. In addition, only part of the charge session data has been used to develop the model for the following reasons:

1. In the real scenario, accessing to the full charge session from SOC 0% to SOC 100% is difficult, since the battery might never experience highly low SOC (e.g. less than 5%).
2. The optimized charging protocols can be modified with regard to the cell SOH; however, the process of slow charging at the end of the charge session is a constraint that needs to be maintained given the Li-ion battery's specification.
3. All the charging protocols need to be followed by a constant voltage (CV) mode at the end of the charging session due to technical issues for which the explanation is out of the scope of this work.

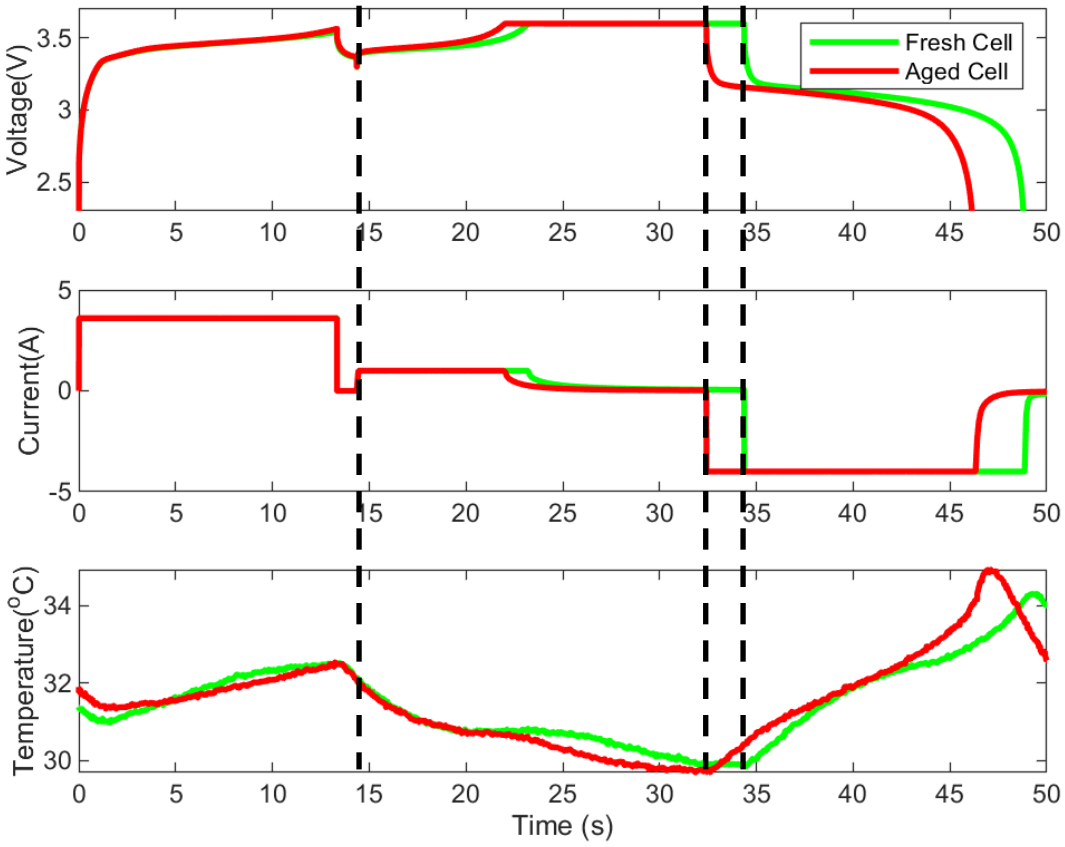


Figure 3-10. The voltage, current and temperature time series of the fresh and aged cell

Given the aforementioned reasons, the charging session data has been used partially to build the model. As it is shown in Figure 3-10, the dashed lines show the start and the end of the partially charging session interval. As is obvious, the end of the partially charged session interval has changed as the cell ages. In order to find the input data from the raw data, an algorithm has been used to find the partially charging session.

Table 3-3. The algorithm used to find the partially charging session from the raw data

Pseudocode	**Description**

1:	**function** PartialChargeData (Dataset, $I_{cut-off}$, $V_{cut-off}$)	
2:	**for** $i = 1, 2, 3, \ldots 124$ **do**	Cover all the cell
3:	$N \leftarrow last\ cycle\ for\ cell\ \#\ i$	Update Cycle number
4:	**for** $n = 1, 2, 3, \ldots, N$ **do**	Cover all the cycles
5:	**if** *I(t) > 0 &…* $I(t) > I_{cut-off}$ *&…* *V (t) >* $V_{cut-off} \cdots$	Find the sampling time for each cycle
6:	$t_{sample} \leftarrow t$	Update the sampling time
7:	**end if**	
8:	$I_p(i, n) \Leftarrow I(t_{sample})$	Store the partial current
9:	$V_p(i, n) \Leftarrow V(t_{sample})$	Store the partial voltage
10:	$T_p(i, n) \Leftarrow T(t_{sample})$	Store the partial temperature
11:	**end for**	
12:	**end for**	
13:	**end function**	

Table 3-3 shows the detils of the algorithm used to obtain the input data toward training the model in this work. The $I_{cut-off}$ and $V_{cut-off}$ have been considered 1.001 A and 3V, respectively. The values for the $I_{cut-off}$ and $V_{cut-off}$ are selected to make sure the charge session data points from 80% state of charge (SOC) to 100% SOC are extracted correctly from the raw data.

After the partial charge session has been obtained using the algorithm shown in Table 3-3, the current, temperature and voltage curves have been discretized to 100 sections. Then average value of each discretized section has been obtained to make the matrix shown in Eq. 3-26, where $\bar{V}_i$, $\bar{I}_i$, and $\bar{T}_i$ respectively represent the average of discretized values of the voltage, current, and temperature for the jth segment measured at the jth time interval of a partial charge session for the cycle # n.

$$\text{Cycle Input} = \begin{bmatrix} \bar{V}_1 & \bar{I}_1 & \bar{T}_1 \\ \vdots & \vdots & \vdots \\ \bar{V}_j & \bar{I}_j & \bar{T}_j \\ \vdots & \vdots & \vdots \\ \bar{V}_{100} & \bar{I}_{100} & \bar{T}_{100} \end{bmatrix}_{100\times 3} \tag{3-26}$$

In the next step, each cycle input matrix is reshaped and made a matrix with dimension 1×300. Then, all the cycle inputs are stacked to shape the input data for the partial charging session of each battery cell studied in this work.

$$\begin{matrix} \left[\bar{V}_1 \quad \cdots \quad \bar{V}_j \quad \cdots \quad \bar{V}_{100} \quad \bar{I}_1 \quad \cdots \quad \bar{I}_j \quad \cdots \quad \bar{I}_{100} \quad \bar{T}_1 \quad \cdots \quad \bar{T}_j \quad \cdots \quad \bar{T}_{100}\right]_{1\times 300} & \text{from cycle 1} \\ \vdots \\ \left[\bar{V}_1 \quad \cdots \quad \bar{V}_j \quad \cdots \quad \bar{V}_{100} \quad \bar{I}_1 \quad \cdots \quad \bar{I}_j \quad \cdots \quad \bar{I}_{100} \quad \bar{T}_1 \quad \cdots \quad \bar{T}_j \quad \cdots \quad \bar{T}_{100}\right]_{1\times 300} & \text{from cycle n} \\ \vdots \\ \left[\bar{V}_1 \quad \cdots \quad \bar{V}_j \quad \cdots \quad \bar{V}_{100} \quad \bar{I}_1 \quad \cdots \quad \bar{I}_j \quad \cdots \quad \bar{I}_{100} \quad \bar{T}_1 \quad \cdots \quad \bar{T}_j \quad \cdots \quad \bar{T}_{100}\right]_{1\times 300} & \text{from cycle N} \end{matrix} \tag{3-27}$$

The matrix shown in Eq. 3-28 is the input data from cycle 1 to N used to train the model.

$$\begin{bmatrix} \bar{V}_1^1 & \cdots & \bar{V}_j^1 & \cdots & \bar{V}_{100}^1 & \bar{I}_1^1 & \cdots & \bar{I}_j^1 & \cdots & \bar{I}_{100}^1 & \bar{T}_1^1 & \cdots & \bar{T}_j^1 & \cdots & \bar{T}_{100}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{V}_1^N & \cdots & \bar{V}_j^N & \cdots & \bar{V}_{100}^N & \bar{I}_1^N & \cdots & \bar{I}_j^N & \cdots & \bar{I}_{100}^N & \bar{T}_1^N & \cdots & \bar{T}_j^N & \cdots & \bar{I}_{100}^N \end{bmatrix}_{\text{N}\times 300} \tag{3-28}$$

Each cycling is made of the charge session with a variety of the charging protocols followed by a discharge session with constant discharge. All the cells studied in this work have been discharged with 4C discharge rate. As a result, the discharge capacity obtained from each cycle n has been used to obtain the ground truth output to train the model.

$$Y_n = SOH_n = \left(\frac{Discharge\ Capacity_{cyle\#1}}{Discharge\ Capacity_{cyle\#n}}\right) \quad 3\text{-}29$$

It is noteworthy to mention that the discharge capacity in the data sets is directly the output from the battery cycler. However, the discharge capacity can also be calculated using the coulomb counting method, which is the time integration of the discharge current over the discharge time for the entire full discharge cycle shown in Eq. 3-30.

$$Discharge\ Capacity_{cyle\#n} = \int_{T=begining\ of\ discharge}^{T=end\ of\ discharge} I(t)dt \quad 3\text{-}30$$

The discharge capacity obtained from the Eq. 3-30 and the one directly obtained from the battery cycler raw data are in agreement.

3.5 Network Training

In this section, more details about the training algorithm of the deep learning models as well as the final architecture of the trained model are discussed. In this work, the data preprocessing has been done in MATLAB, and the deep learning model development has been done in a Python 3.6 environment, with TensorFlow 2. Keras deep learning library has been employed to build the model architecture. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow [42, 43]. In this work, for all the deep learning models, the popular optimizer “Adam” has been employed for the network training. Adam is an algorithm for first-

order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The implementation is highly straightforward and computationally efficient to use. Also, Adam needs small memory, is invariant to diagonal rescaling of the gradients, and is efficient in training the model that is involved with a large data and/or parameter [44].

3.5.1 Deep feed forward neural network (FNN) Architecture

As has been already discussed, a feed forward neural network has been shaped of multiple layers, and each layer has several neurons consisting of the nonlinear activation according to the weighted connections to the previous layer. We have trained two different FNN, one with 1 hidden layer and 10 neurons. The other is made of 4 hidden layers and 30 neurons at each layer.

3.5.2 LSTM Architecture

In this study, we have tried to train several bidirectionally wrapped LSTM with different layer and node dimension sizes toward obtaining the most efficient architecture. The best performing LSTM architecture after the training and evaluation process (using the performance matrix) was obtained to be a LSTM network with one zero masking layer, 6 LSTM layers with 80 nodes per layer and three dense layers.

3.5.3 ConvLSTM Architecture

In the ConvLSTM, 3×3 kernel size has been used to extract the feature into the model. The rest of the configurations are similar to the LSTM architecture. Regularization was employed in the network in the shape of dropout layers, utilizing a certain probability for each node to be randomly deactivated from being trained in the training step to reduce the chance of overfitting.

3.5.4 DCNN Architecture

The DCNN model has been trained using 50 epochs and a minibatch of 256 observations. The initial learning rate of 0.01 for all the convolutional layers and fully connected (FC) layers have been considered. The learning rate will be adjusted for better efficient training at each 10 epochs by factor of 8. The summary of the DCNN architecture is shown in Table 3-4.

Table 3-4. The summary of the DCNN model architecture.

Layer	Filter Size	Number of Kernels	Stride Size	Number of Weights	Number of Biases
Input	$100 \times 3 \times 1$	_	_	_	_
Convolution 1	$1 \times 2 \times 1$	32	(1,1)	64	32
Max Pooling	$3 \times 1 \times 1$	32	(3,1)	_	_
Convolution 2	$3 \times 1 \times 1$	64	(1,1)	192	64
Convolution 3	$3 \times 1 \times 1$	100	(1,1)	300	100
Convolution 4	$3 \times 1 \times 1$	100	(1,1)	300	100
FC 1	100×1	_	_	30000	100
FC 2	100×1	_	_	10000	100
FC 3	1×1	_		100	1

3.6 Results and Discussion

In this part, the final results are shown and discussed. As is discussed in the "Methodology" section, 4 battery cells have been randomly selected from 4 different buckets as the nominated training data set. All the deep learning models have been trained, using based nominated training data. In the model training process, 20% of the cells' data have been used for model validation.

After the models have been trained, 100 cells have been selected to test the model. It is noteworthy to mention that the cells that have been used to test the model have never been used for the training process and the model is totally blind to testing the data set.

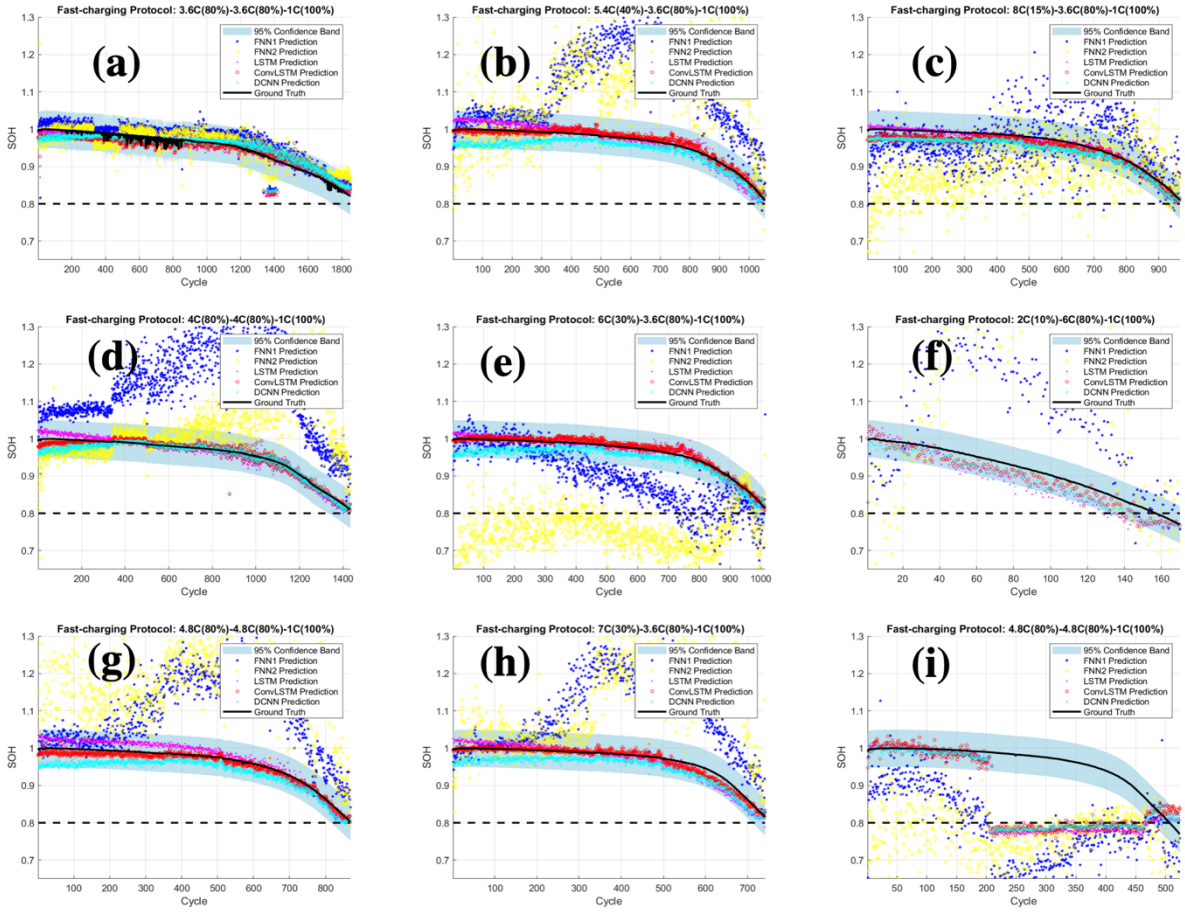


Figure 3-11. The SOH estimation obtained from different deep learning algorithm for selected test data with different charging protocols

The prediction results are shown in Figure 3-11 for 9 battery cells. All of the prediction results for all the 104 cells (4 cells from the training data and 100 cells from testing data) are shown in the APPENDIX section. The specifications of the 9 cells that their results have been shown above, are listed in Table 3-5. In order to provide a guideline for the non-battery experts, the cells' specs are

highlighted with different colors in which the green and the yellow colors represent the best and the worst specs, respectively.

Table 3-5. Cells' specification showing the cycle life and the cell condition at beginning of life (BOL)

Index	Charging Protocol	Cycle Life	Initial Capacity (mAh)	Initial IR (mOhm)
1	3.6C(80%)-3.6C(80%)-1C(100%)	1900	1070.7	16.4
2	4C(80%)-4C(80%)-1C(100%)	1226	1079.7	16.4
3	4.8C(80%)-4.8C(80%)-1C(100%)	870	1093.9	16.3
4	5.4C(40%)-3.6C(80%)-1C(100%)	1054	1083.0	17.0
5	6C(30%)-3.6C(80%)-1C(100%)	1014	1079.3	16.9
6	7C(30%)-3.6C(80%)-1C(100%)	742	1076.6	17.0
7	8C(15%)-3.6C(80%)-1C(100%)	966	1083.3	16.8
8	2C(10%)-6C(80%)-1C(100%)	148	1072.1	17.6
9	4.8C(80%)-4.8C(80%)-1C(100%)	495	1068.0	17.5

Best Worst

As is obvious in Figure 3-11, the LSTM, ConvLSTM, and DCNN outperform compared to FNN models. It is worth mentioning that the tested cells have a variety of the beginning of life (BOL) conditions such a different initial capacity and internal resistance (IR). Still, the deep learning models, specifically ConvLSTM and DCNN, show promising results. In the next part, the performance metrics to quantify the accuracy of the predictive models are presented.

3.6.1 Performance Metric

The performance metric that has been used in this work is based on the results obtained from the prediction ($\widetilde{SOH}$) and the results obtained from the real experimental data (SOH) as the ground truth. Accordingly, three different evaluation metrics, are defined. In order to evaluate the model performance for each single testing cell, the absolute percentage error (APE) is defined.

$$APE = \left|\frac{\widetilde{SOH} - SOH}{SOH}\right| \times 100\% \tag{3-31}$$

In fact, APE defines the relative error at each predicted point by the model. For instance, for the cell with 700 cycles data, the model output is a vector with size 700×1. On the other hand, we need a performance metric for different cells with varying charging protocols to have evaluation over the robustness of the model on the cell performance under different charging conditions. As a result, the mean absolute percentage error (MAPE) score of each model on each battery cell is introduced as the second evaluation metric. Eq. 3-32 shows how the MAPE score is calculated where n is the number of data points. In addition, MAPE highlights the average accuracy of each model over the whole life of the battery cell from cycle 1 to cycle n.

$$MAPE = \frac{\sum_{i=1}^{n} APE_i}{n} \tag{3-32}$$

The third evaluation metric is the mean deviation from the MAPE which in this work is called MSigma of the model for each cell shown in 3-33.

$$MSigma = \frac{\sum_{i=1}^{n} |APE_i - MAPE|}{n} \tag{3-33}$$

MSigma highlights the consistency of the accuracy of the model to predict the $\widetilde{SOH}$ by indicating the average deviation of the predicted $\widetilde{SOH}$ value. Both performance metrics show superior performance with lower values

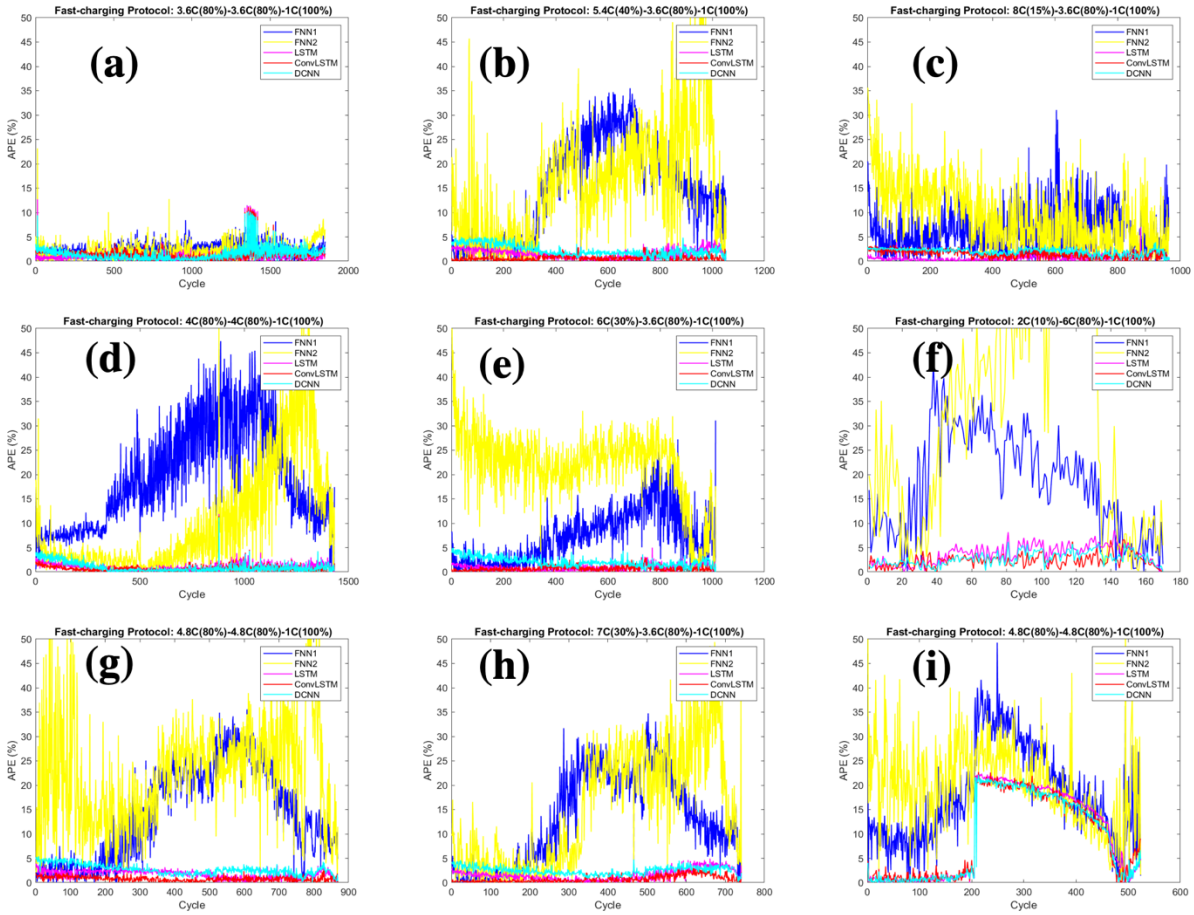


Figure 3-12. The SOH estimation absolute percentage error obtained from different deep learning algorithm for selected test data

The introduced APE metric has been shown in Figure 3-12 for the 9 cells that their prediction results have been shown in Figure 3-11. Also, the corresponding box plots for APE for each cell has been shown in Figure 3-13.

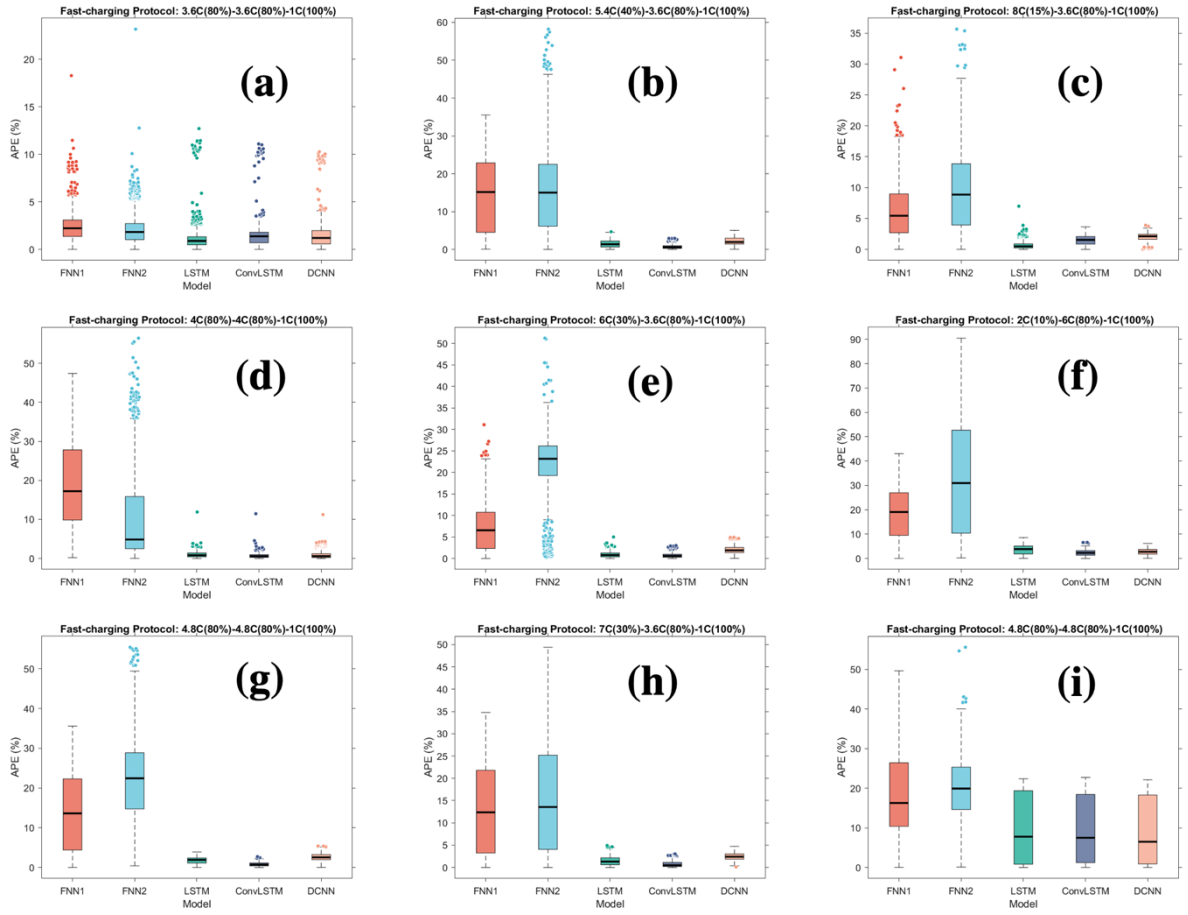


Figure 3-13. The SOH estimation absolute percentage error variability obtained from different deep learning algorithm for selected test data

In the results shown in Figure 3-12 and Figure 3-13, it is obvious that LSTM, ConvLSTM and DCNN possess the lowest average APE compared to the FNN models. Between LSTM, ConvLSTM and DCNN models, the ConvLSTM shows slightly higher accuracy. In order to compare the overall performance of different models, the average values of MAPE and MSigma from different tested cells for each model is calculated and shown in Figure 3-14.

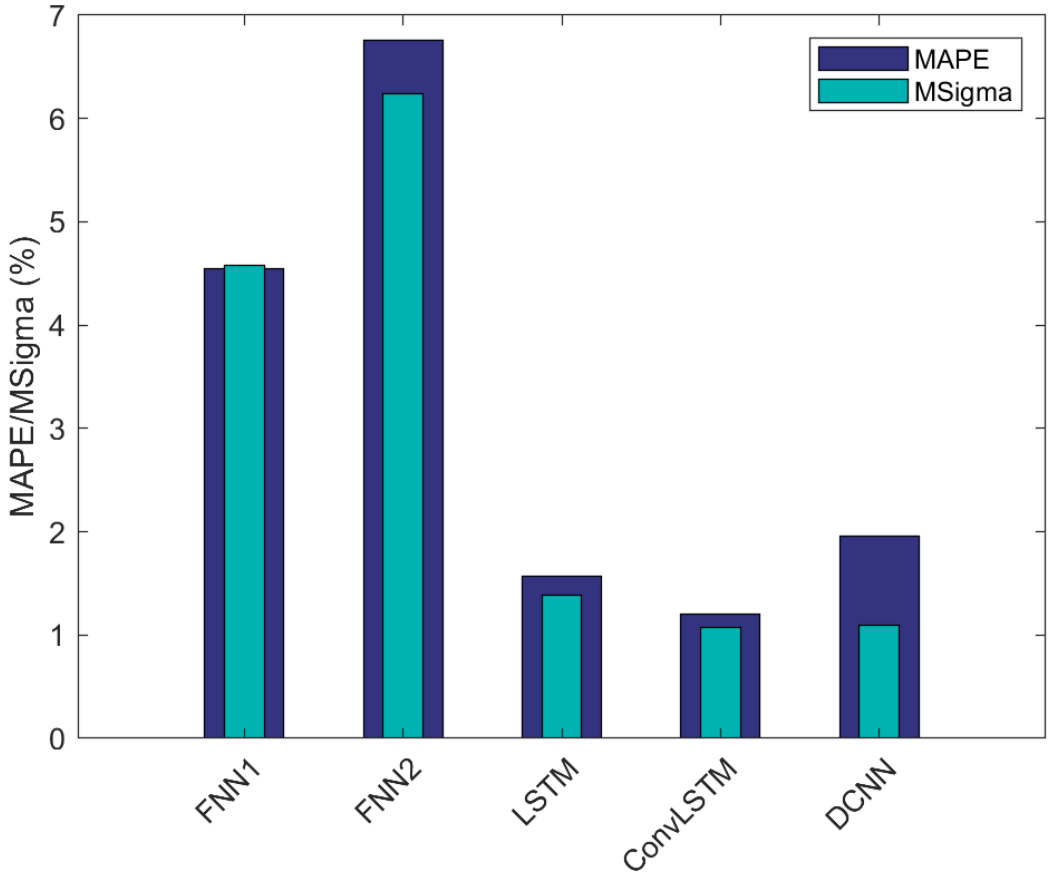


Figure 3-14. Comparison of the overall MAPE and MSigma between FNN, LSTM, ConvLSTM, and DCNN models

The overall performance of the studied deep learning models shows that LSTM, DCNN, and LSTM generally outperform compared to both FNN models.

The LSTM model shows lower overall MAPE compared to DCNN. As a result, LSTM has provided more accurate results compared to DCNN. However, the overall MSigma value for the DCNN model is lower than that of LSTM, showing that the DCNN outperforms in terms of the consistency of the accuracy for different testing protocols.

Table 3-6. Comparison of the overall MAPE and MSigma

Model	FNN1	FNN2	LSTM	ConvLSTM	DCNN
MSigma (%)	4.58	6.23	1.38	1.07	1.10
MAPE (%)	4.54	6.75	1.57	1.20	1.96

Among all the deep learning models studied in this work, ConvLSTM has the best performance both from the MAPE and MSigma point of views, indicating the fact that ConvLSTM architecture possessed both the accuracy (MAPE) of the LSTM and constancy (MSigma) of DCNN. In other words, the feature extraction capability of the DCNN architecture and time memory capability of the LSTM architecture can combine toward building a highly robust architecture. Such a hybrid architecture is highly applicable in the Li-ion battery SOH estimation in EV industry.

3.7 Unexpected Results

The prediction results obtained from the developed deep learning model for SOH estimation shows an unexpected SOH estimation for the battery cell charged with the charging protocol 4.8C(80%)-4.8C(80%)-1C(100%) shown in Figure 3-11 (i). Also, Figure 3-12 indicates that the APE dramatically increases after cycle 200. In order to investigate the reasons for such an unexpected prediction, the raw data from the current, Voltage, and temperature sensors are tracked. Figure 3-15 shows the raw sensor for 9 different cells with different charging protocols. Although the voltage signals for all the 9 cells shows similar trends, the temperature sensor data for the cell with the charging protocol 4.8C(80%)-4.8C(80%)-1C(100%) shown in Figure 3-15 (c) with blue color deviates from the normal trends. As a result of such an anomaly, APE for the SOH prediction has jumped from less than 5% to more than 20%.

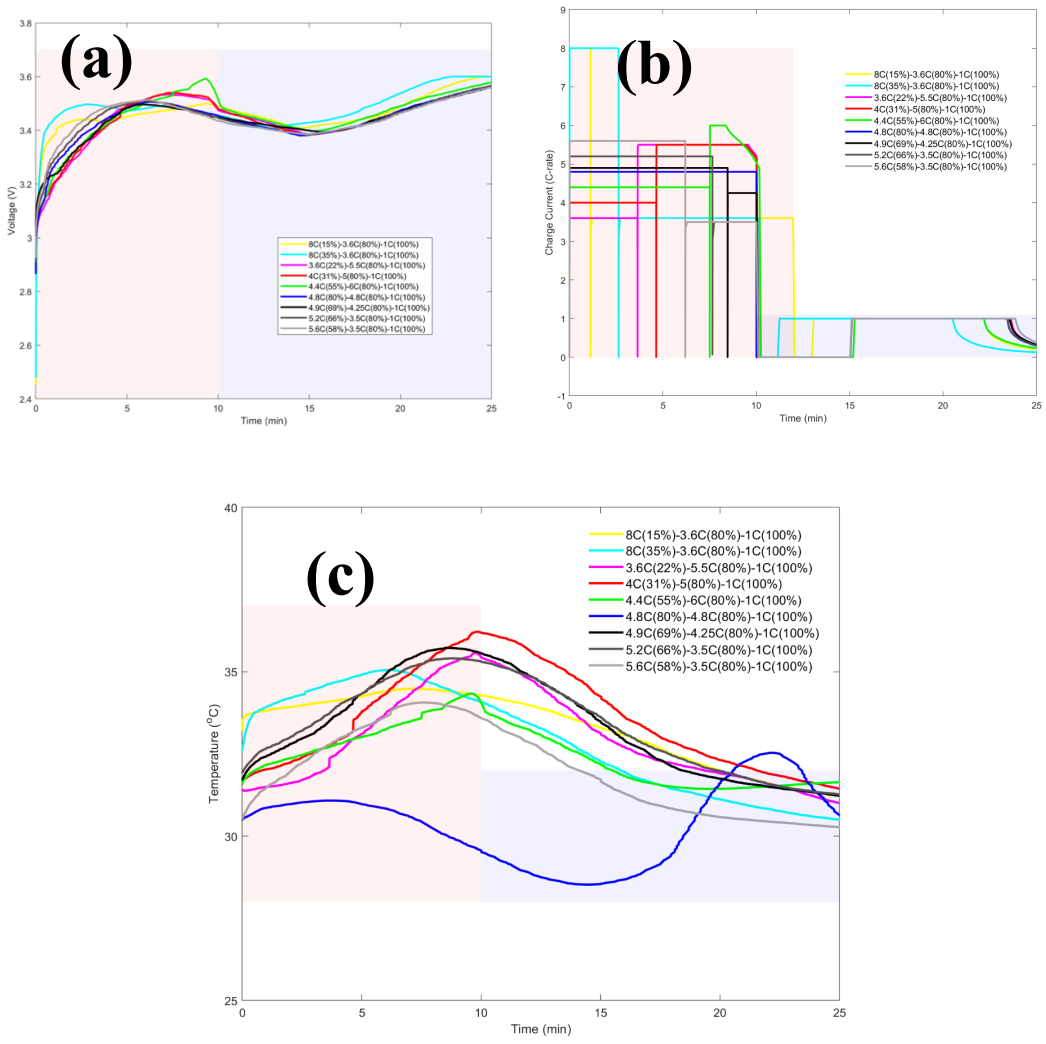


Figure 3-15. The current, voltage and temperature observed for 9 different cells under different charging policy

Such a dramatic change in the SOH that is caused by the malfunctioning sensor can be a potential capability for anomaly detection in the EV battery pack. In fact, if, for any reason, the predicted SOH shows a dramatic change, it can be the sign of a sensor error originated from malfunctions in the battery pack.

4 CONCLUSION AND FUTURE WORK

4.1 Conclusion Summary

This work aimed to apply different machine learning algorithm specifically the deep learning algorithm to obtain the battery state of health (SOH) that is highly applicable for the EV application under real-world operation. The presented deep learning models can be employed in the future battery management systems (BMS) with the outstanding capability to utilize the intelligent health prognostics algorithms. Some of the highlights of this study are as follows:

- Four different deep learning models have been developed and employed for the SOH estimation.
- The battery SOH is estimated from the partial charging data without the extracting any feature manually.
- LSTM, ConvLSTM, and DCNN architectures have been employed to capture the effect of the time memory on the battery degradation.
- The accuracy all deep learning models with LSTM, ConvLSTM, and DCNN architecture are acceptable by industry standards, with overall MAPE of less than 3%
- The LSTM model shows lower overall MAPE compared to DCNN; however, the overall MSigma value for the DCNN model is lower than that of LSTM, showing the fact that the DCNN model outperforms in terms of the consistency of the accuracy for different testing protocols.

- The ConvLSTM has the best performance among all the studied deep learning models with MAPE and MSigma of 1.20% and 1.07%, respectively.
- The employed LSTM, ConvLSTM, and DCNN architectures in this work show promising capability for the anomaly detection that is highly applicable for the state of safety (SOS) prediction.

To the best of the author's knowledge, at the time of the literature review for this work, no comparable work has been found in the same domain, which comprehensively implements and evaluates the performance of LSTM, ConvLSTM and DCNN architectures. In addition, the algorithm presented in this work to obtain the partial charge session data from the raw data of the current, voltage and temperature is highly applicable and paves the way toward employing a deep learning model for the online SOH estimation in the battery management systems (BMS).

4.2 Limitations

In this work, the deep learning models have been developed based on the available data sets for the Li-ion battery with LFP/Graphite chemistry. However, to investigate the robustness of the existing models on other commercial LIBs with different chemistry, such as LMO/Graphite, LCO/Graphite, and NMC/Graphite, more data from the other commercial cells is needed.

4.3 Future work

In this work, it has been shown that the existing deep learning models have potential toward the anomaly detection of the cell performance. Such a capability can be applied toward developing algorithms to enhance the battery back safety in the EV industry.

In addition, the algorithms presented in this work to obtain the partial charging session toward training the deep learning model can be implemented on the cloud in the shape of a data pipeline to enhance the efficiency of the data preprocessing before model training. Furthermore, using cloud-based implementation provides the opportunity to obtain new data sets from different resources through online measurement and uploading to the cloud.

Last but not least, the training process of the existing models can be extended to different types of real-world datasets such as common drive cycles.

REFERENCES

[1] J. Lu, T. Wu, and K. Amine, "State-of-the-art characterization techniques for advanced lithium-ion batteries," *Nature Energy,* vol. 2, no. 3, p. 17011, 2017.

[2] J. Zheng, X. Sun, L. Jia, and Y. Zhou, "Electric passenger vehicles sales and carbon dioxide emission reduction potential in China's leading markets," *Journal of Cleaner Production,* vol. 243, p. 118607, 2020.

[3] D. Andre, A. Nuhic, T. Soczka-Guth, and D. U. Sauer, "Comparative study of a structured neural network and an extended Kalman filter for state of health determination of lithium-ion batteries in hybrid electricvehicles," *Engineering Applications of Artificial Intelligence,* vol. 26, no. 3, pp. 951-961, 2013.

[4] X. Han *et al.*, "A review on the key issues of the lithium ion battery degradation among the whole life cycle," *ETransportation,* vol. 1, p. 100005, 2019.

[5] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, and D. U. Sauer, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *Journal of Energy Storage,* vol. 30, p. 101557, 2020.

[6] H. He, R. Xiong, and H. Guo, "Online estimation of model parameters and state-of-charge of LiFePO4 batteries in electric vehicles," *Applied Energy,* vol. 89, no. 1, pp. 413-420, 2012.

[7] G. L. Plett, "Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2: Simultaneous state and parameter estimation," *Journal of power sources,* vol. 161, no. 2, pp. 1369-1384, 2006.

[8] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, and D. U. Sauer, "Online capacity estimation of lithium-ion batteries with deep long short-term memory networks," *Journal of Power Sources,* vol. 482, p. 228863, 2021.

[9] Y. Zou, X. Hu, H. Ma, and S. E. Li, "Combined state of charge and state of health estimation over lithium-ion battery cell cycle lifespan for electric vehicles," *Journal of Power Sources,* vol. 273, pp. 793-803, 2015.

[10] J. Wu, Z. Wei, W. Li, Y. Wang, Y. Li, and D. Sauer, "Battery thermal-and health-constrained energy management for hybrid electric bus based on soft actor-critic DRL algorithm," *IEEE Transactions on Industrial Informatics,* 2020.

[11] S. Wang *et al.*, "Impact of battery degradation models on energy management of a grid-connected DC microgrid," *Energy,* vol. 207, p. 118228, 2020.

[12] W. Li *et al.*, "Parameter sensitivity analysis of electrochemical model-based battery management systems for lithium-ion batteries," *Applied Energy,* vol. 269, p. 115104, 2020.

[13] W. Li *et al.*, "Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented Kalman filter," *Journal of Power Sources,* vol. 476, p. 228534, 2020.

[14] D. W. Limoge and A. M. Annaswamy, "An adaptive observer design for real-time parameter estimation in lithium-ion batteries," *IEEE Transactions on Control Systems Technology,* vol. 28, no. 2, pp. 505-520, 2018.

[15] B. Jenkins, A. Krupadanam, and A. M. Annaswamy, "Fast adaptive observers for battery management systems," *IEEE Transactions on Control Systems Technology,* vol. 28, no. 3, pp. 776-789, 2019.

[16] A. Bartlett, J. Marcicki, S. Onori, G. Rizzoni, X. G. Yang, and T. Miller, "Model-based state of charge estimation and observability analysis of a composite electrode lithium-ion battery," in *52nd IEEE Conference on Decision and Control*, 2013: IEEE, pp. 7791-7796.

[17] S. J. Moura, M. Krstic, and N. A. Chaturvedi, "Adaptive PDE observer for battery SOC/SOH estimation," in *Dynamic Systems and Control Conference*, 2012, vol. 45295: American Society of Mechanical Engineers, pp. 101-110.

[18] L. Zheng, L. Zhang, J. Zhu, G. Wang, and J. Jiang, "Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model," *Applied Energy,* vol. 180, pp. 424-434, 2016.

[19] C. Zou, C. Manzie, D. Nešić, and A. G. Kallapur, "Multi-time-scale observer design for state-of-charge and state-of-health of a lithium-ion battery," *Journal of Power Sources,* vol. 335, pp. 121-130, 2016.

[20] J. P. Christophersen, "Battery Test Manual For Electric Vehicles, Revision 3," Idaho National Lab.(INL), Idaho Falls, ID (United States), 2015.

[21] J. Wu, Y. Wang, X. Zhang, and Z. Chen, "A novel state of health estimation method of Li-ion battery using group method of data handling," *Journal of Power Sources,* vol. 327, pp. 457-464, 2016.

[22] H.-T. Lin, T.-J. Liang, and S.-M. Chen, "Estimation of battery state of health using probabilistic neural network," *IEEE transactions on industrial informatics,* vol. 9, no. 2, pp. 679-685, 2012.

[23] H. Chaoui, C. C. Ibe-Ekeocha, and H. Gualous, "Aging prediction and state of charge estimation of a LiFePO4 battery using input time-delayed neural networks," *Electric Power Systems Research,* vol. 146, pp. 189-197, 2017.

[24] A. A. Hussein, "Capacity fade estimation in electric vehicle li-ion batteries using artificial neural networks," *IEEE Transactions on Industry Applications,* vol. 51, no. 3, pp. 2321-2330, 2014.

[25] G.-W. You, S. Park, and D. Oh, "Diagnosis of electric vehicle batteries using recurrent neural networks," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 6, pp. 4885-4893, 2017.

[26] A. Eddahech, O. Briat, N. Bertrand, J.-Y. Deletage, and J.-M. Vinassa, "Behavior and state-of-health monitoring of Li-ion batteries using impedance spectroscopy and recurrent neural networks," *International Journal of Electrical Power & Energy Systems,* vol. 42, no. 1, pp. 487-494, 2012.

[27] C. Zhang, Y. Zhu, G. Dong, and J. Wei, "Data-driven lithium-ion battery states estimation using neural networks and particle filtering," *International Journal of Energy Research,* vol. 43, no. 14, pp. 8230-8241, 2019.

[28] X. Hu, J. Jiang, D. Cao, and B. Egardt, "Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling," *IEEE Transactions on Industrial Electronics,* vol. 63, no. 4, pp. 2645-2656, 2015.

[29] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method," *Journal of Power Sources,* vol. 196, no. 23, pp. 10314-10321, 2011.

[30] J. Kim, S. Lee, and B. Cho, "Complementary cooperation algorithm based on DEKF combined with pattern recognition for SOC/capacity estimation and SOH prediction," *IEEE Transactions on Power Electronics,* vol. 27, no. 1, pp. 436-451, 2011.

[31] V. Klass, M. Behm, and G. Lindbergh, "A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation," *Journal of Power Sources,* vol. 270, pp. 262-272, 2014.

[32] B. Saha, S. Poll, K. Goebel, and J. Christophersen, "An integrated approach to battery health monitoring using Bayesian regression and state estimation," in *2007 IEEE Autotestcon*, 2007: Ieee, pp. 646-653.

[33] P.-H. Michel and V. Heiries, "An adaptive sigma point kalman filter hybridized by support vector machine algorithm for battery SoC and SoH estimation," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015: IEEE, pp. 1-7.

[34] Y. Li *et al.*, "Random forest regression for online capacity estimation of lithium-ion batteries," *Applied energy,* vol. 232, pp. 197-210, 2018.

[35] K. S. Mawonou, A. Eddahech, D. Dumur, D. Beauvois, and E. Godoy, "State-of-health estimators coupled to a random forest approach for lithium-ion battery aging factor ranking," *Journal of Power Sources,* vol. 484, p. 229154, 2021.

[36] K. A. Severson *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy,* vol. 4, no. 5, pp. 383-391, 2019.

[37] X. Hu, S. E. Li, Z. Jia, and B. Egardt, "Enhanced sample entropy-based health management of Li-ion battery for electrified vehicles," *Energy,* vol. 64, pp. 953-960, 2014.

[38] D. Yang, X. Zhang, R. Pan, Y. Wang, and Z. Chen, "A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve," *Journal of Power Sources,* vol. 384, pp. 387-395, 2018.

[39] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Gaussian process regression for forecasting battery state of health," *Journal of Power Sources,* vol. 357, pp. 209-219, 2017.

[40] G. E. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," 2010.

[41] C. Hu, G. Jain, C. Schmidt, C. Strief, and M. Sullivan, "Online estimation of lithium-ion battery capacity using sparse Bayesian learning," *Journal of Power Sources,* vol. 289, pp. 105-113, 2015.

[42] "Tensorflow: An end-to-end open source machine learning platform." https://www.tensorflow.org/. (accessed April 15, 2021).

[43] "Keras: deep learning API written in Python." (accessed April 15, 2021).

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

APPENDIX A

In this section, the SOH estimation results obtained from different deep learning algorithms for the selected test data with different charging protocols are presented.

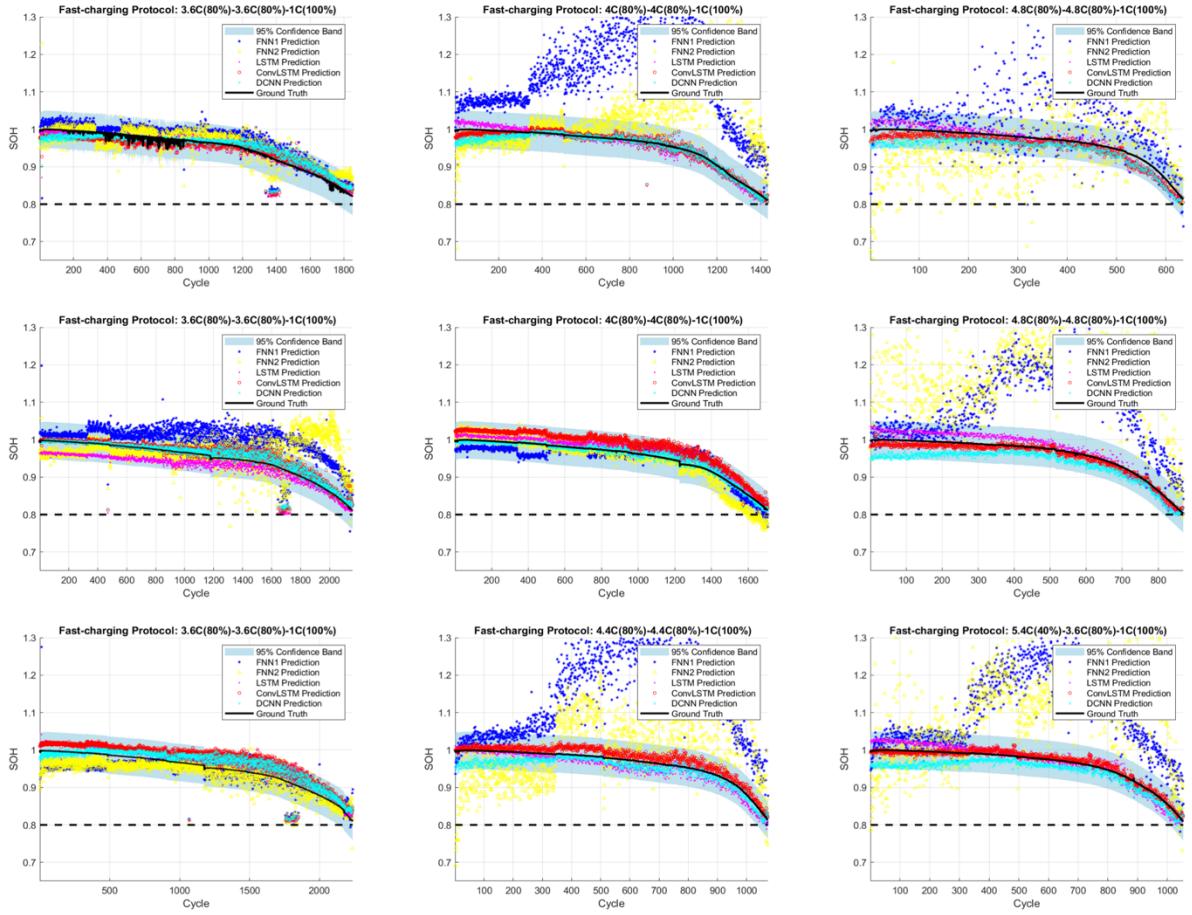


Figure A-1. The SOH estimation obtained from different deep learning algorithms for the selected test data with different charging protocols

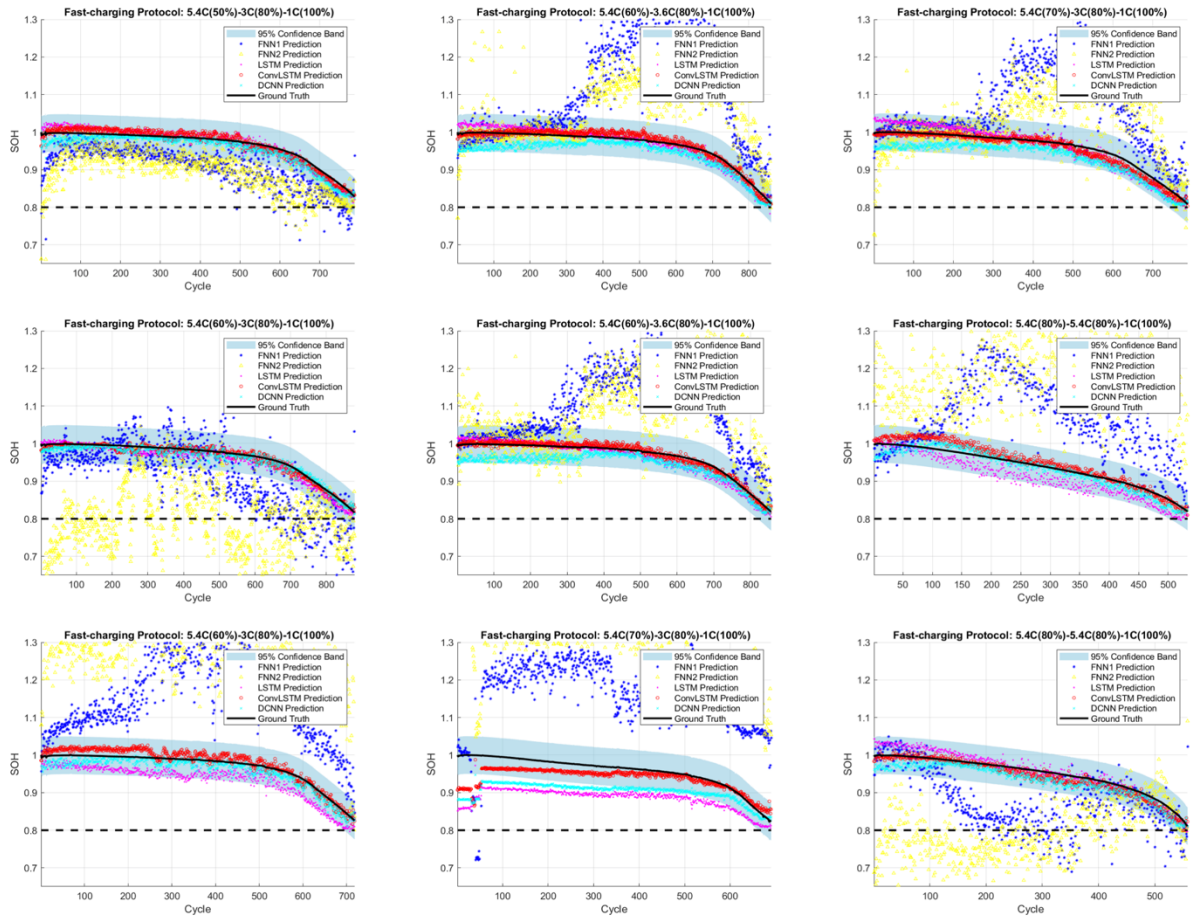


Figure A-2. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

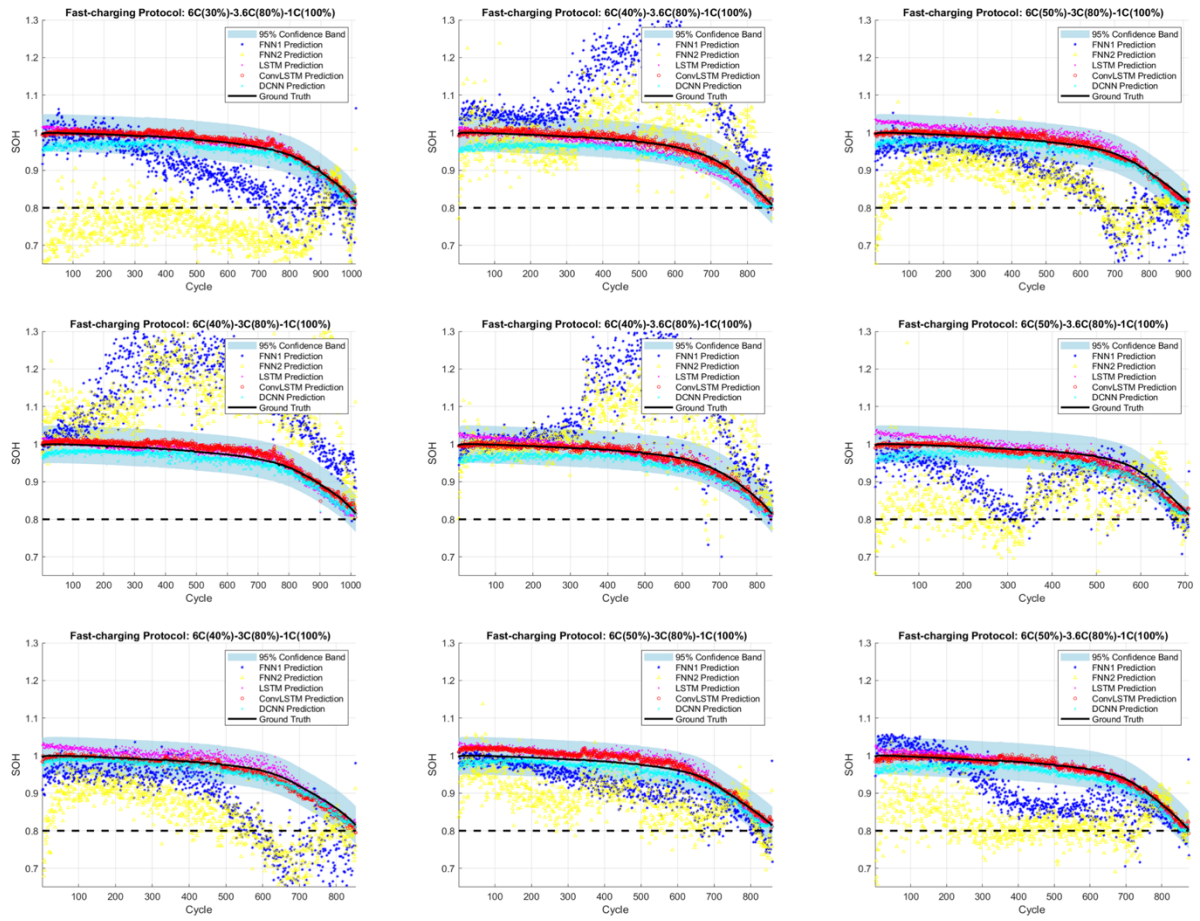


Figure A-3. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

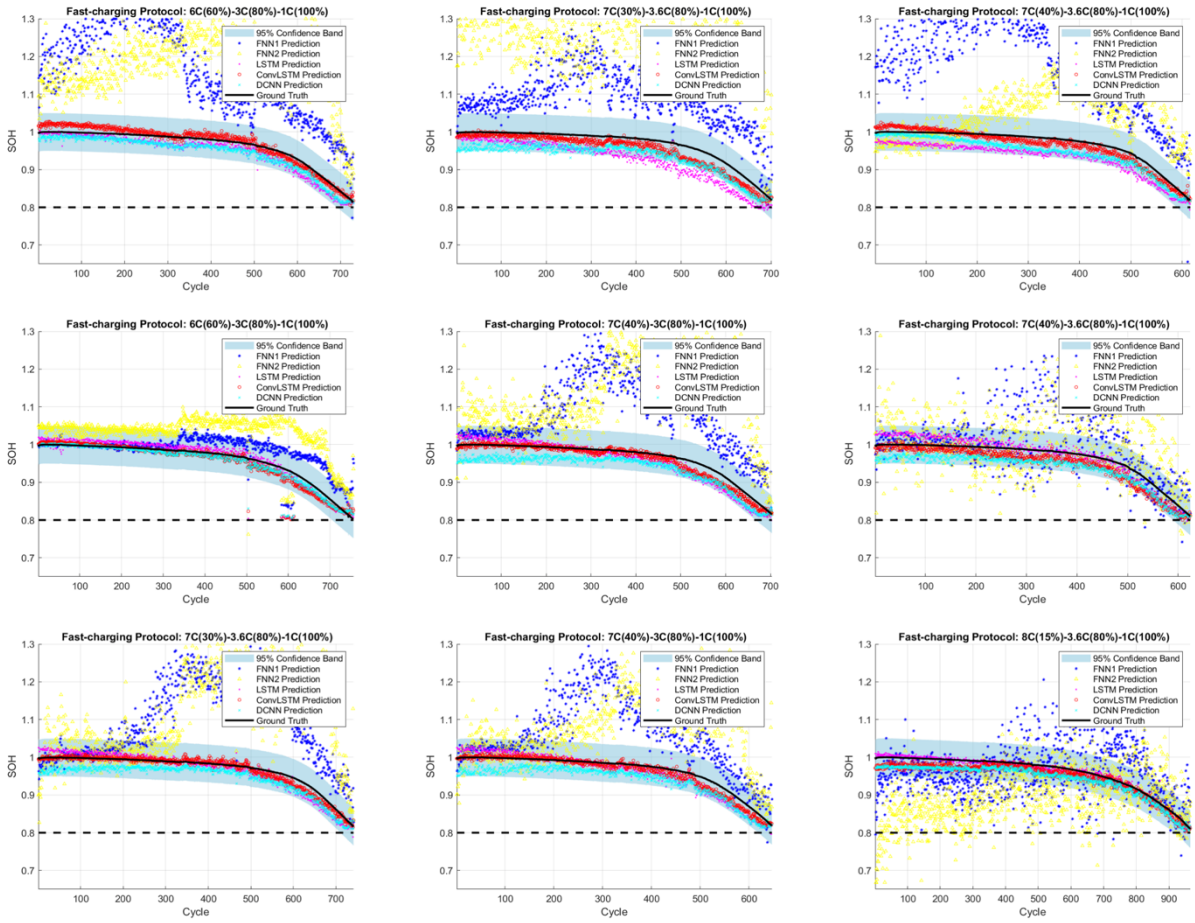


Figure A-4. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

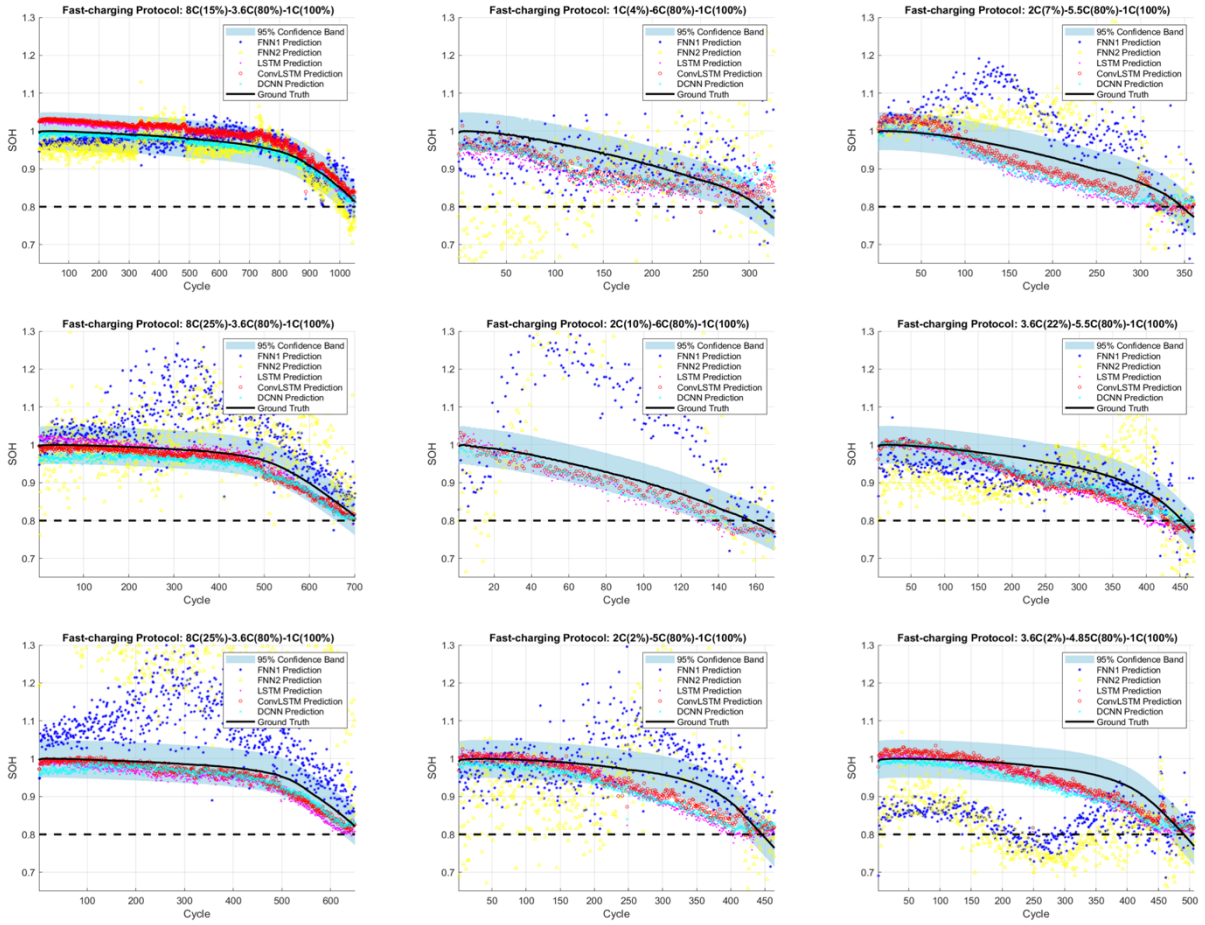


Figure A-5. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

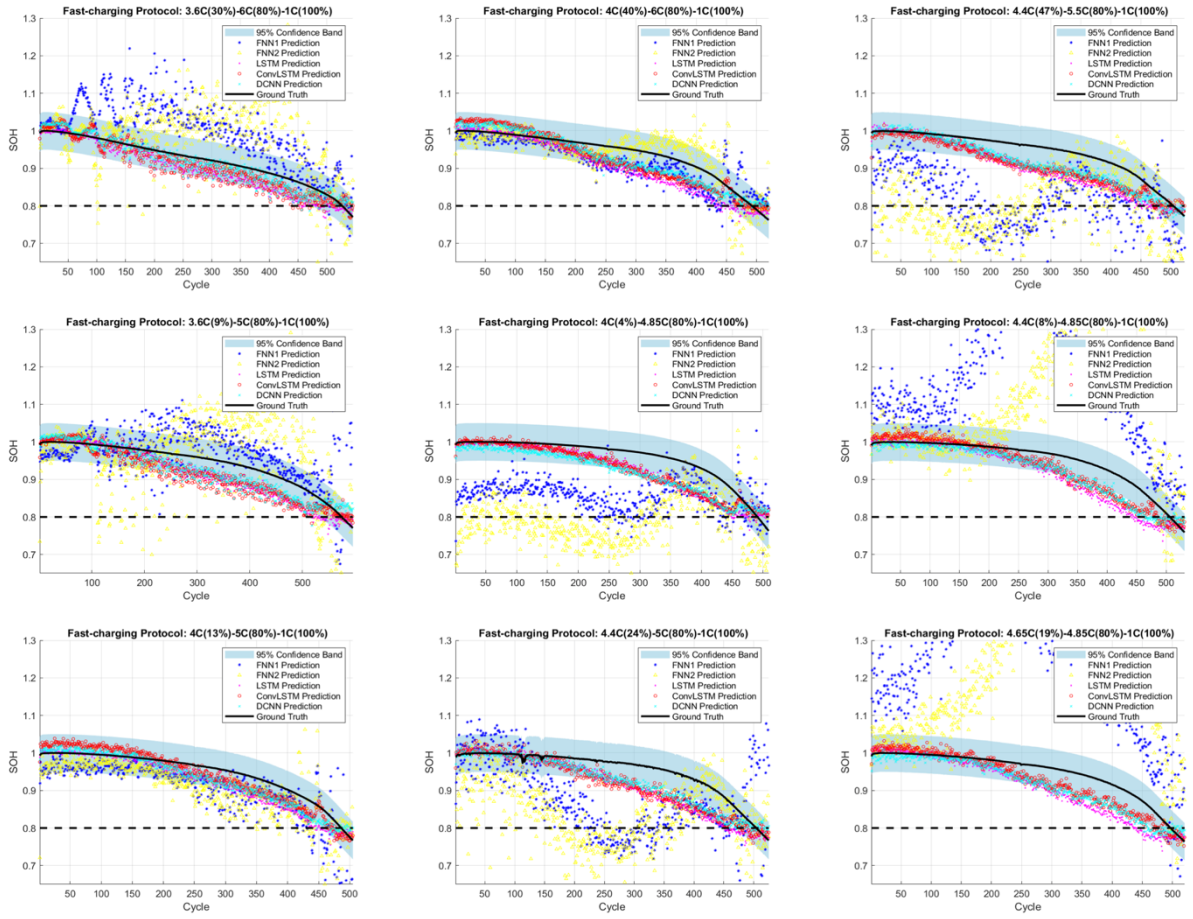


Figure A-6. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

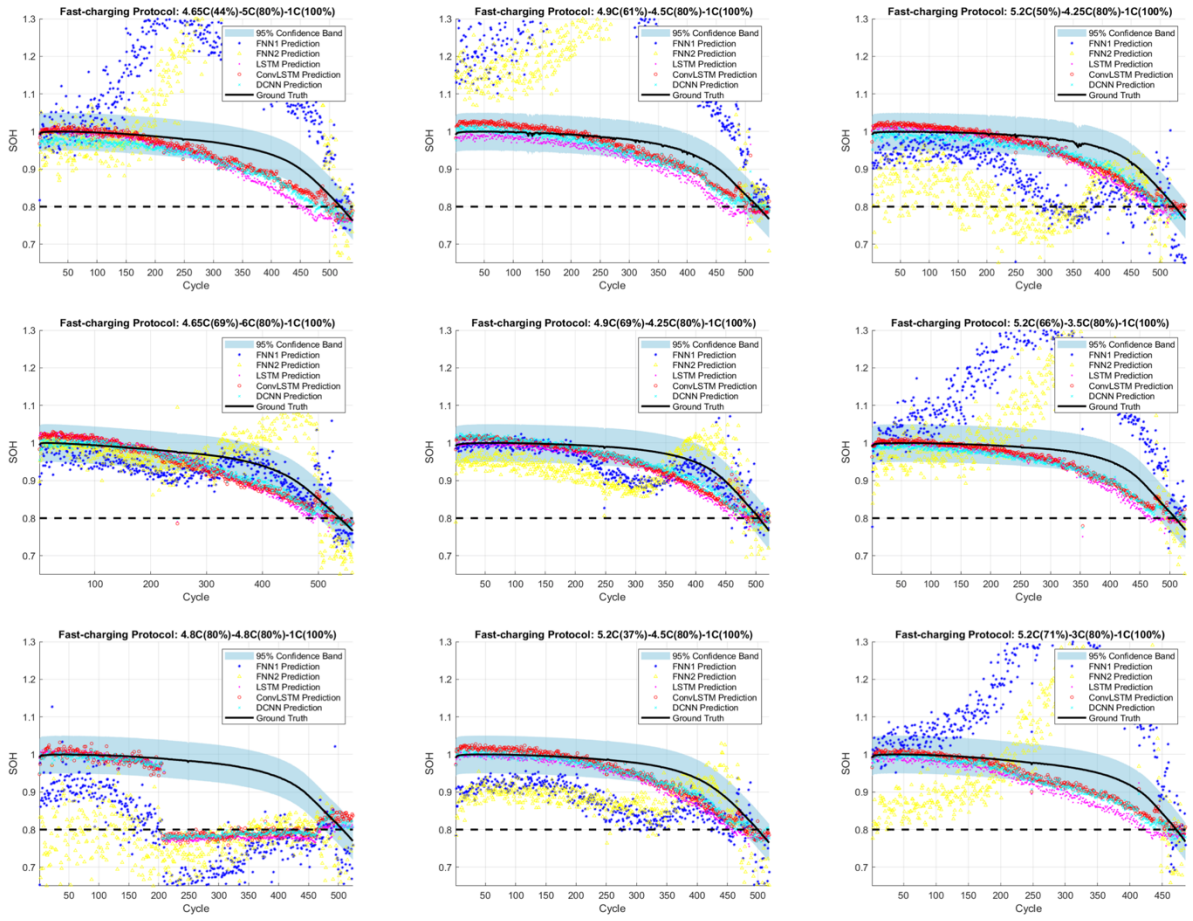


Figure A-7. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

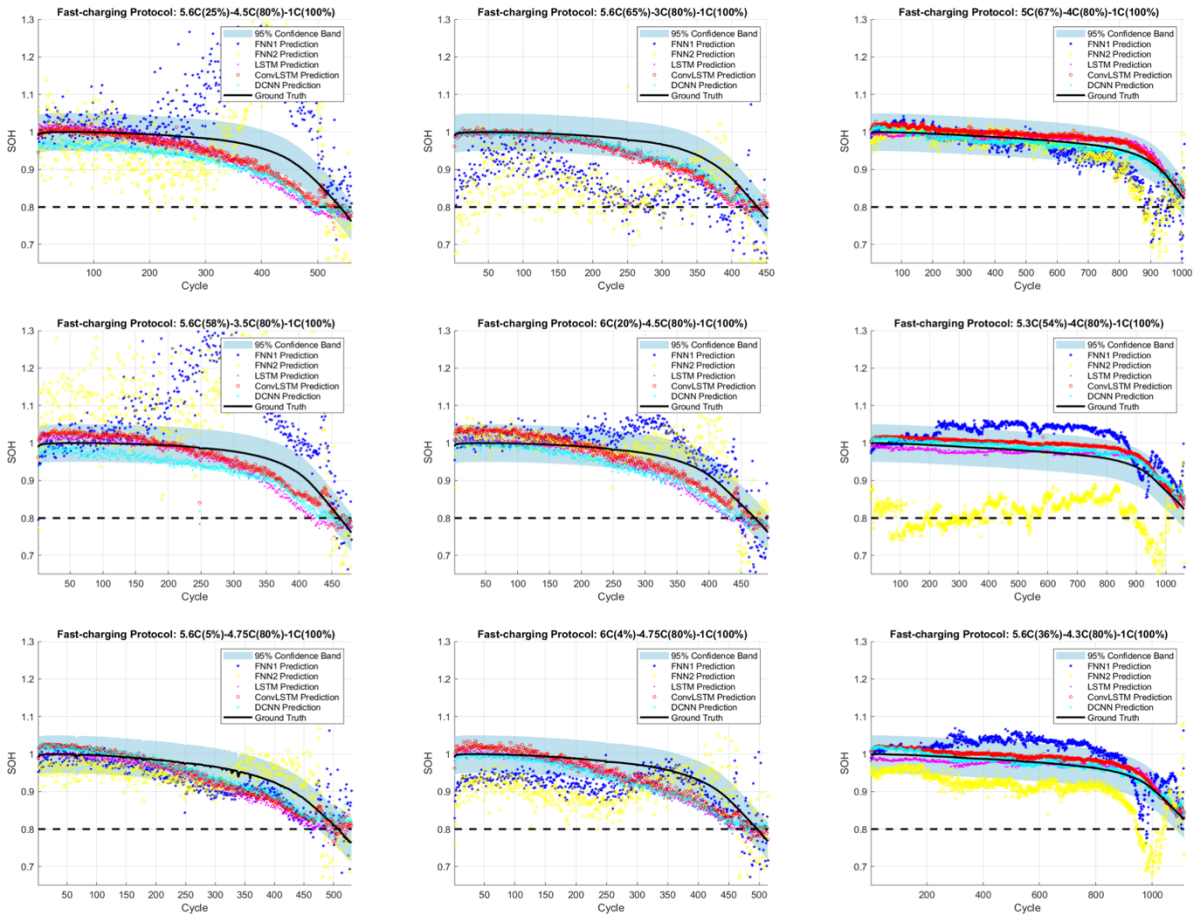


Figure A-8. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

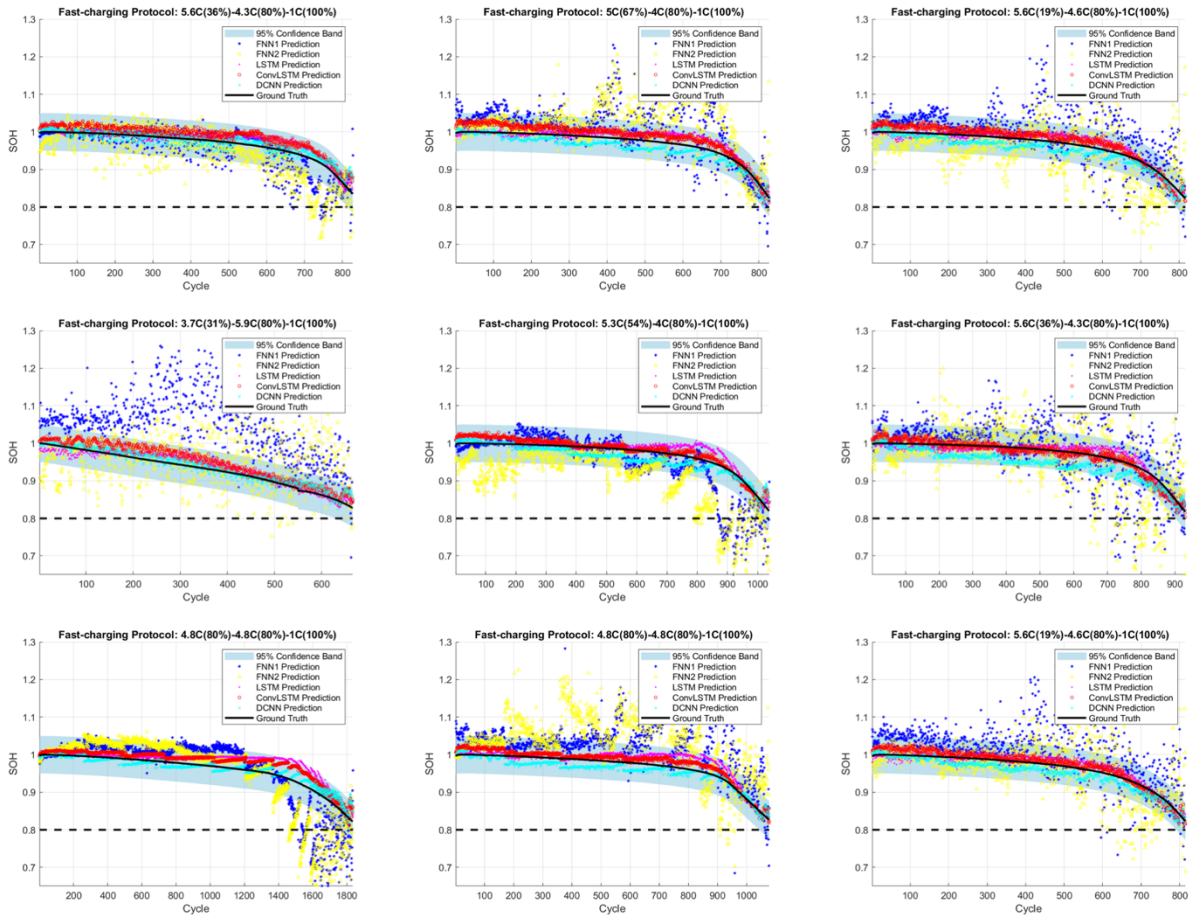


Figure A-9. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

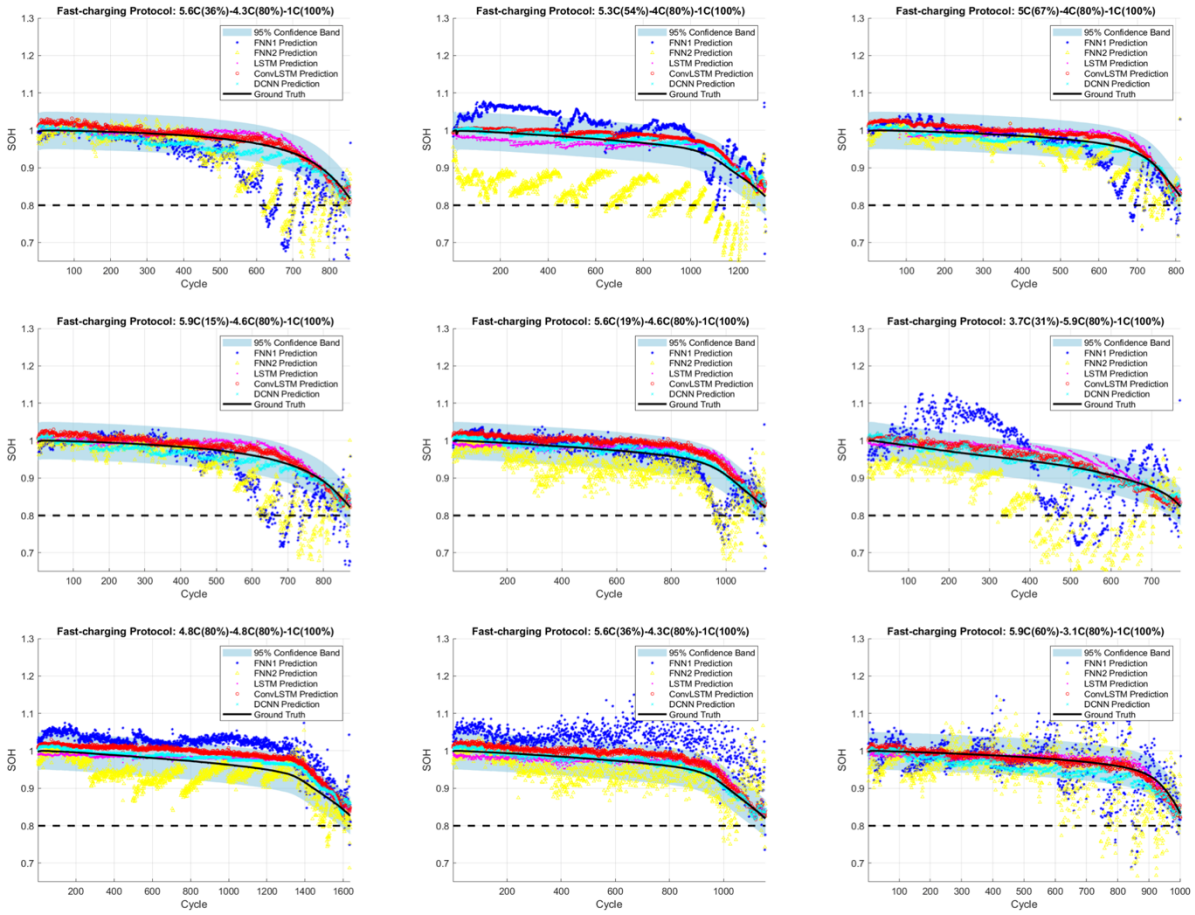


Figure A-10. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

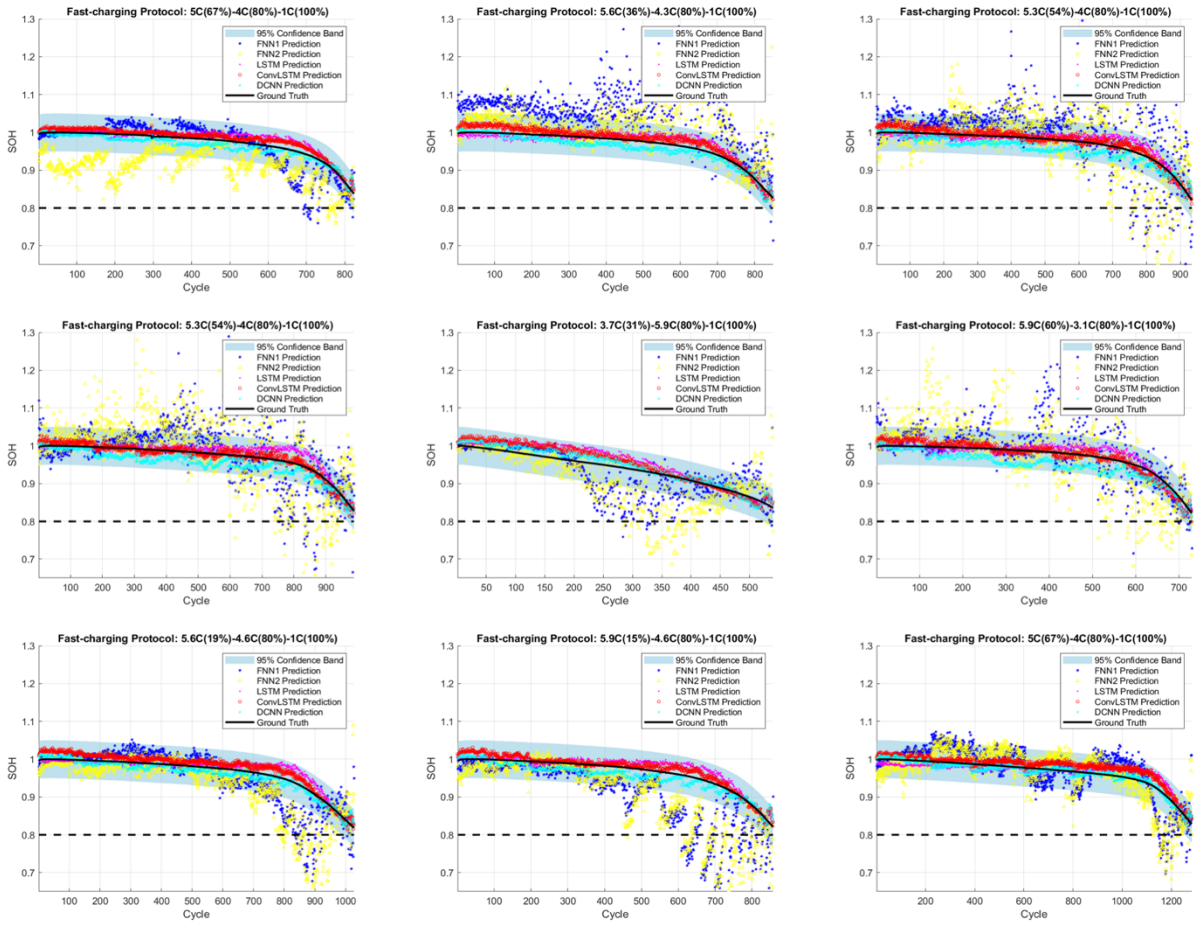


Figure A-11. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols

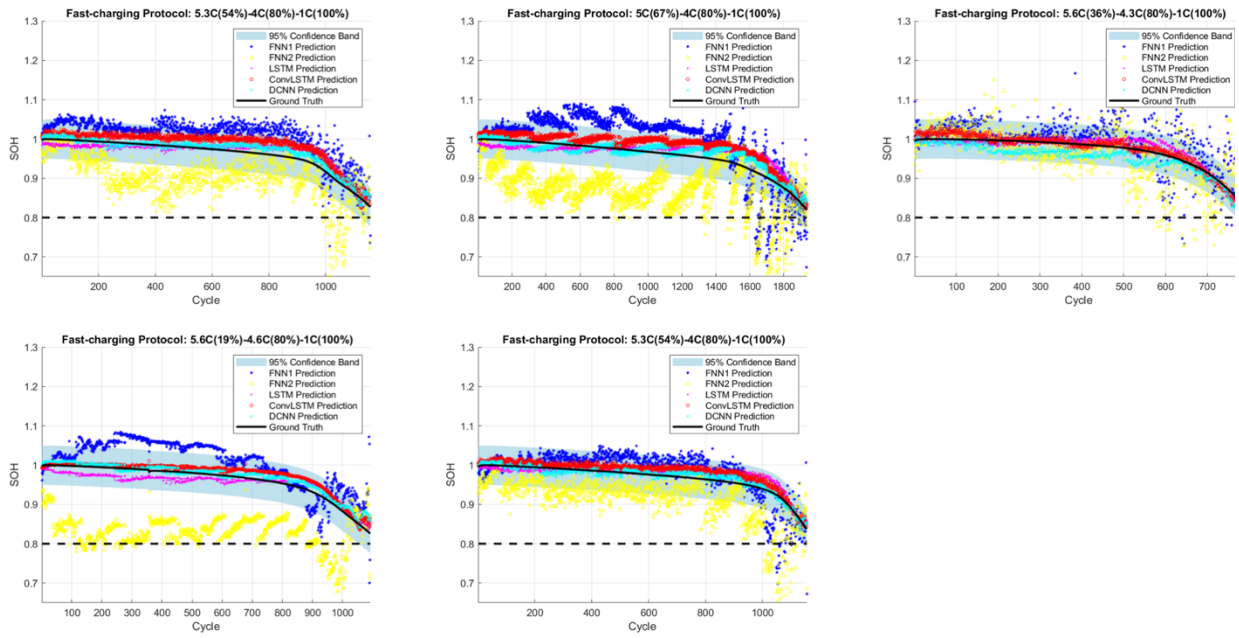


Figure A-12. The SOH estimation obtained from different deep learning algorithm for the selected test data with different charging protocols