

Online capacity estimation of lithium-ion batteries with deep long short-term memory networks

Weihan Li^{a,b,*}, Neil Sengupta^{a,b}, Philipp Dechent^{a,b}, David Howey^{e,f}, Anuradha Annaswamy^g, Dirk Uwe Sauer^{a,b,c,d}

^aChair for Electrochemical Energy Conversion and Storage Systems, Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Jaegerstrasse 17/19, 52066 Aachen, Germany

^bJuelich Aachen Research Alliance, JARA-Energy, Germany

^cInstitute for Power Generation and Storage Systems (PGS), E.ON ERC, RWTH Aachen University, Germany

^dHelmholtz Institute Münster (HI MS), IEK-12, Forschungszentrum Jülich, Germany

^eDepartment of Engineering Science, University of Oxford, Oxford, OX1 3PJ, UK

^fThe Faraday Institution, Harwell Campus, Didcot, OX11 0RA, UK

^gDepartment of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract

There is an increasing demand for modern diagnostic systems for batteries under real-world operation, specifically for the estimation of their state of health, for example, via their remaining capacity. The online estimation of the capacity of a cell is challenging due to the dynamic nature of cell aging and the limited variety of inputs available from a cell under operation. The scope of this work is the development of a data-driven capacity estimation model for cells under real-world working conditions with recurrent neural networks having long short-term memory capability. Voltage-time sensor data from the partial constant current phase charging curve is used as input, reflecting input availability in the real world. The network achieves a best-case mean absolute percentage error of 0.76% and is extremely robust while handling input noise. It also has the ability to handle variations in the length of the input time series and can generate a viable estimation even with an incomplete collection of input due to sensor errors. The model validation with several scenarios is done in a local embedded device, highlighting the use case of such models in future battery management systems.

Keywords: battery, lithium-ion, health, capacity, deep learning, LSTM

1. Introduction

Lithium-ion batteries (LIBs) are developing into a widely used technology, especially in the field of electromobility, due to their low associated costs and high energy density. However, as is the case with almost all electrochemical systems, LIBs also undergo performance degradation with time during usage

*Corresponding author. Chair for Electrochemical Energy Conversion and Storage Systems, Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Jaegerstrasse 17/19, 52066, Aachen, Germany
Email address: batteries@isea.rwth-aachen.de, weihan.li@isea.rwth-aachen.de (Weihan Li)

as well as storage, which increases the need for evaluation of the longevity and reliability of the cell under operation. Therefore, appropriate monitoring of the health of LIBs, as a function of the battery management system (BMS), is crucial for both manufacturers and customers of electric vehicles (EVs).

As a metric for the aging status of the LIBs, state of health (SOH) is commonly defined in the industry using one of two indicators, which are remaining cell capacity and internal ohmic resistance. End of life (EOL) is then defined to be the point in the cell's life when the chosen SOH indicator crosses a certain limit, which is usually 80% of the original value in case of remaining capacity, and 200% of the original value in case of internal resistance. In this work, the remaining capacity is chosen as the SOH indicator to be estimated. An accurate SOH estimation is essential for operation, maintenance, and optimization of the cell, but monitoring the aging of a LIB is a non-trivial task due to the complex and non-linear underlying mechanisms that contribute to the aging of the cell. There are several aging mechanisms, as described in [1, 2], such as the formulation of the solid electrolyte interface (SEI), lithium plating during operation, and current collector degradation. These mechanisms are coupled with each other and contribute to different aging modes, which consequently age the cell via complex degradation pathways. Due to this, the battery loses its ability to store energy and deliver power with time.

There are several approaches found in the literature for the estimation of the SOH of a cell. A controlled experimental approach is the first method for SOH estimation, and this can be done using trivial methods such as coulomb counting or more sophisticated approaches such as electrochemical impedance spectroscopy (EIS) [3, 4] and incremental capacity and differential voltage (IC/DV) analysis [5–8]. The major limitation of such methods is that they require a unique current profile, which is hard to obtain in real operational scenarios. For example, IC/DV analysis uses charge-voltage (Q-V) curves typically obtained by passing a very low current through the cell, to simulate equilibrium operation. The curves are then differentiated to get IC and DV plots. The differentiation amplifies the influence of noise, even with added smoothing filters, and the gathering of input data must be done over a wide voltage range, which is challenging in real scenarios. To avoid such limitations, there is a need for methods that can work with sensor data available, usually from a working cell. Particularly, the time series sensor data available from the charging-discharging curves in the normal operation of the battery serves to be ideal for an estimation approach that does not interrupt regular cell operation.

The most widely used online SOH estimation approach in industry is the parameterization of a battery model, usually online in a recursive implementation such as a Kalman filter. Models can be of several types, such as equivalent circuit models [9–13] or electrochemical models [14–21]. Circuit models define an electrical circuit with various components to simulate the cell's electrical dynamics, whereas electrochemical models represent the distribution of lithium-ion concentration and potential within a cell using a set of coupled partial differential equations. The main limitation of these approaches is that their accuracy depends highly on the choice of underlying battery model. Moreover, the identification and tuning of the parameters of

the model so that it captures all the underlying cell degradation mechanisms is a challenging and often computationally expensive task, complicated by the requirement to simultaneously estimate state of charge.

To this end, data-driven supervised learning approaches are becoming increasingly attractive due to their ability to map the input (measurements) to output (SOH) data without necessarily having to choose a physical model. These can range from modified linear regression to Gaussian processes, deep learning and other neural network (NN) approaches [22–35]. For example, Richardson et al. [32], Yang et al. [35], Wang et al. [33] set up different estimation approaches using Gaussian process regression (GPR). This method gives mean capacity estimates as well as probabilistic bounds, using features such as voltage, current, and discharge time, calculated from the charging and discharging curves of the cells. Support vector machines (SVMs) [31] have also produced good results for SOH estimation using similar features. Approaches based on deep neural networks (DNN) [26, 36] employ autoencoders to extract the high-level features from raw charging curve data and pass them to a DNN to obtain the SOH as output. Wu et al. [28] use importance sampling from an extensive input feature set from the charging curves and train a DNN for capacity estimation. Recurrent neural networks (RNNs) have also been used to process an input time sequence and obtain the nominal capacity [23, 24]. The main limitation with the above data-based approaches is that almost all of them require external preprocessing and feature engineering steps, which may require extensive domain-specific knowledge. Moreover, machine learning approaches such as GPR and SVMs often increase substantially in computational complexity with the amount of data to be processed. Thus, they are usually not suitable for placing in small embedded systems associated with a BMS. Approaches using NNs do not have the size limitation but still require the feature extraction step. RNNs would serve as an ideal approach for processing raw time series data, but they are limited by having a short term memory, which leads to the learning gradients vanishing after a few time steps [37]. Therefore, most approaches using RNNs must reduce their input features to a few time steps to train the network effectively.

This paper aims to bridge the aforementioned research gap and proposes a supervised learning approach containing an RNN with long short-term memory (LSTM) cells that can estimate the remaining capacity of a battery under operation. The proposed framework uses the data gathered from cell aging experiments similar to the intended use cases to train the model. The trained model is then served in a local deep learning capable embedded device for validation, to test the usage of such models in future battery systems. The inputs of the network are the voltage and time samples from the partial constant current charging curve of the cell, which is determined to be the most reliably available input without operational interference and requires no further processing or feature engineering. The proposed LSTM-RNN model grants the ability to accept variable lengths of time series data as input, requiring no specific input time size or length. The model validation results show that it is very robust while handling noisy inputs, and is capable of generating viable outputs even with an incomplete input set provided to it. This leads to having feasible SOH estimates even when the input measurement process is interrupted or erroneous.

2. Dataset

2.1. Aging experiments

The motivation for this work is to develop a SOH estimation method that can identify cell-specific degradation of equivalently manufactured cells and give accurate estimates of the remaining capacity of a cell. The dataset used in this work was gathered from in-house experiments at ISEA, RWTH Aachen University, on 48 Sanyo/Panasonic UR18650E cells manufactured in the same batch and labeled the same grade by the manufacturer, as per [38]. The specifications of the cell are given in Table 1. The cells were aged using one specific profile, well beyond their industry-standard EOL, to analyze the variations that arise in aging paths over time. The experiment consisted of cyclic aging interspersed with characterization tests. Each cell had, on average, 17 characterization tests spread across around 2400 cycles over its life, and each cycling aging period contains, on average, 160 cycles.

Category	Specification
Manufacturer	Sanyo/Panasonic
Type	Cylindrical
Chemistry	Graphite/NMC
Nominal Capacity	1.85 Ah
Weight	45.5 g

Table 1: Cell specifications.

The current and voltage response of one cycling aging period is shown in Fig. 1(a) and Fig. 1(b), respectively. Cycling of the cells was conducted at a constant temperature of 25 °C. Each cycle consists of discharging to 3.5 V and charging to 3.9 V with a constant current of 4 A in constant current (CC) mode, followed by a constant voltage (CV) phase until the minimum cut-off current of 40 mA was reached. The focus of this work is the development of a deep learning approach for remaining capacity estimation from available sensor data of battery operation, and this dataset serves as the first proof of concept with one temperature and one fixed profile for battery aging. Expansion of aging profile types and temperatures is a future activity. The current and voltage responses of the characterization test are shown in Fig. 1(c) and Fig. 1(d), respectively. The characterization tests were conducted once at the beginning of life (BOL), and then periodically until EOL was reached. The maximum discharge capacity for different current C-rates was measured first, followed by the OCV curve. Finally, the pulse resistances under different current rates for three different SOCs were determined.

Fig. 2(a) shows the degradation curve of one cell from the dataset, and it is clear that the trend is not linear but changes at a certain point of a cell’s lifetime. This ‘knee point’ is also variable for different cells, as

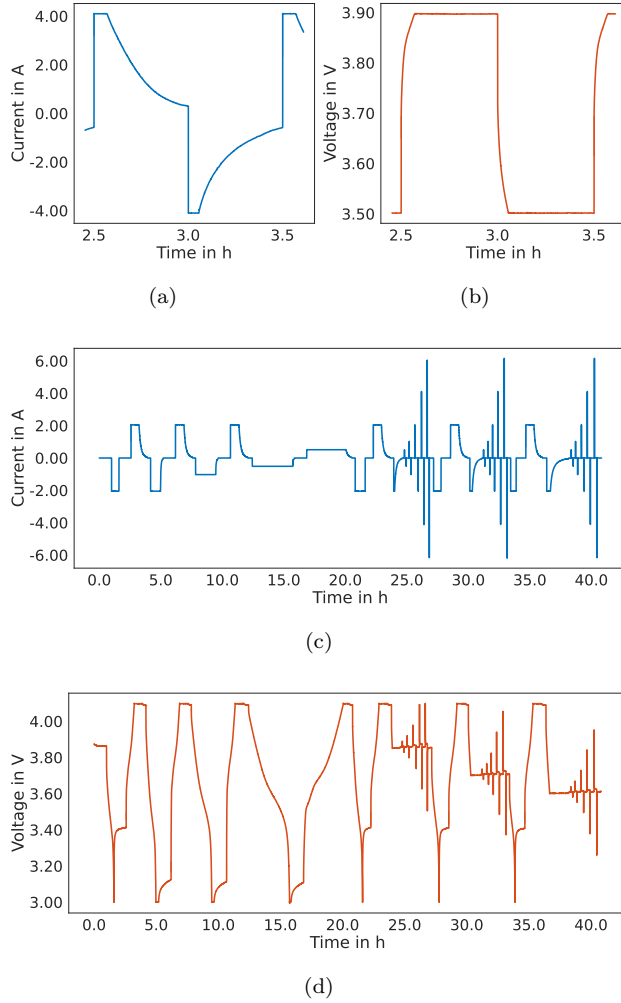


Figure 1: Measured data. (a) Current and (b) voltage response during cycling. (c) Current and (d) voltage response during a characterization test.

seen in Fig. 2(b), which shows the degradation curves of all 48 cells in the dataset, highlighting the dynamic variation that equivalent cells undergo while being aged. Consequently, the cycle number at which the cells reach their respective EOL also widely varies, as seen in Fig. 2(c), with a range of almost 500 cycles between the cells with the longest and shortest lives, respectively. Fig. 2(d), (e), and (f) present correlations of the capacity of the cells at the start, midpoint, and end of their lives, with the final EOL cycle number. An important observation from these figures is that the correlation between EOL and remaining capacity is highly unclear at the BOL. This highlights the need for a SOH estimation approach that can capture the variations observed throughout the cell's life, to give an accurate estimation of the cell's SOH at any point.

After obtaining the raw data, the capacity measurements were interpolated to create the supervised learning dataset, using the piecewise cubic hermite interpolating polynomial (PCHIP) function in MATLAB.

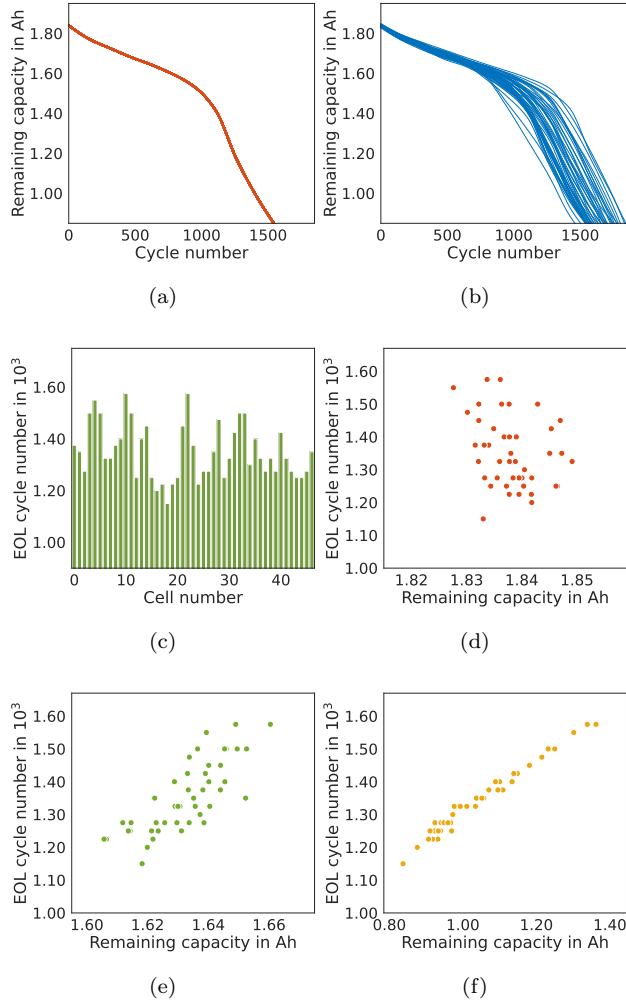


Figure 2: Analysis of the battery dataset. (a) Degradation trend over the lifetime of a single cell. (b) Degradation curves of 48 cells. (c) Variation in the EOL cycle number of the 48 cells. EOL cycle number as a function of (d) initial, (e) mid-life and (f) late-life maximum capacity of the 48 cells, to highlight the variations in correlation.

Subsequently, bad input cycles due to missing data, clearly erroneous measurement, smaller than minimum sample sizes of input, and other factors, were removed from the dataset before training. Cycle data down to nearly 65% of the initial nominal capacity of the cells were then selected for training. This was done to test the viability of the model for batteries in second life operations and to improve the tolerance of the model well beyond the industry-accepted EOL of 80% of the initial nominal capacity. The other available dataset rarely includes the aging data after 80%. Accurate capacity estimation until the extended EOL criteria is also more challenging since the nonlinearities in the cell’s capacity degradation pathway increase as the EOL criteria shifts downwards, and the point of acceleration in aging is encountered in this region. Successfully performing this task leads to a robust model for real-world scenarios.

2.2. Choice of input to the model

The voltage and time samples from the raw partial charging curve of the CC-CV charging operation of the cell are used as inputs to the model. This sensor data is relatively easy to obtain from a vehicle during charging, and charging is typically a uniform process over a cell's lifetime since it usually happens over similar periods of time at a charging station. Discharge curves, however, are dependent on vehicle behavior and can differ significantly with distance traveled, traffic trends, driving behavior, and other factors. The focus is on the CC phase of charging because the nature of charging management of the cells in a battery pack may lead to incomplete CV phases for some cells and extended CV phases for others. The entire pack is shifted to CV mode to prevent overcharging as soon as one cell reaches the cut-off voltage, and therefore, many cells in a pack may have different voltage levels in CV mode. The CC phase of the charging operation is clearly more uniform among all cells in a pack, to reliably gather input data. The choice of input data window must be balanced between the availability of data and the information contained in the input. Thus the input window must be chosen in such a way that it is obtainable throughout the cell's lifetime while maximizing the information captured. The type of data available for training the model also affects the window choice, and our chosen window is not the only possible alternative, but may differ with different training datasets and use cases. Commonly, charging of the vehicle may begin from any value of SOC of the battery but typically will go to 100% before being discontinued. Thus, it is clear that the upper part of the CC phase of the charging curve will be reliably available over the cell's lifetime, and therefore is the choice of input to train our model.

Fig. 3 presents an analysis of the inputs chosen for network training and operation. Fig. 3(a) shows the data window for the gathering of the voltage and time samples (starting at 3.65 V and ending at 3.89 V). The manufacturer specified full-cycle voltage range for the cell is 2.75 V to 4.20 V, and the cell aging was done using partial charge-discharge cycles with the voltage range of 3.5 V to 3.9 V. Our chosen window amounts to 60% of the aging voltage range, and 16.5% of the manufacturer specified full charge range. Fig. 3(b) shows the comparison plot of the partial charging curves of the cell as the cell ages. It is clear that there is a wide variation in the shape of the charging curves, which supports the use of such input for the estimation of the nominal capacity at any point of the cell's life. Fig. 3(c) highlights the need for a network capable of handling variable lengths of input sequences since the constant current charging time and the number of samples clearly vary with the age of the cell.

A network operating with raw sensor data does not need the manual extraction of features that may be unreliable to obtain when the raw curves change in shape and size throughout life, making this model more generally applicable. The correlation plots between the number of input samples and the total charging time, with the maximum capacity of the cell at that time point, are also given in Fig. 3(d) and (e), respectively, again highlighting the different variations.

Subsequently, for the creation of the supervised learning dataset for network training, the inputs were first

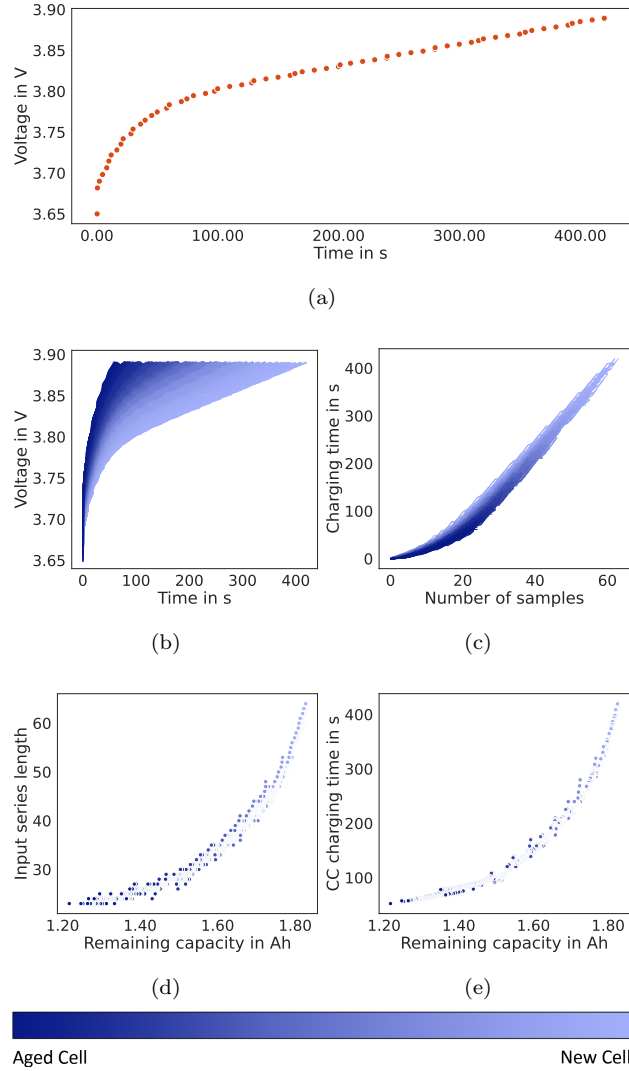


Figure 3: Analysis of the chosen input data window. (a) The selected window of time-voltage curve samples, from 3.65 V to 3.89 V. (b) Time-voltage curves over the cell's lifetime. (c) Input size (number of samples) vs. charging time curves over the cell's lifetime. (d) Input size as a function of capacity, over the cell's lifetime. (e) Constant current charging time as a function of capacity, over the cell's lifetime.

normalized to have values between 0 and 1. This normalization was done by dividing the input matrices with a constant value exceeding the maximum range of the respective inputs in the dataset. After normalization, the input matrices were leading zero-padded to fulfill the requirement of uniform dimensional matrices in the TensorFlow training environment. However, this is not a model-side requirement and the trained model after being implemented, can accept inputs of any shape. Finally, the entire dataset was divided into two parts, with 40 cells being used to train the model, and eight cells kept aside to serve as the blind testing dataset. Additionally, a random 20% of the training data was used for model validation checkpoints in each

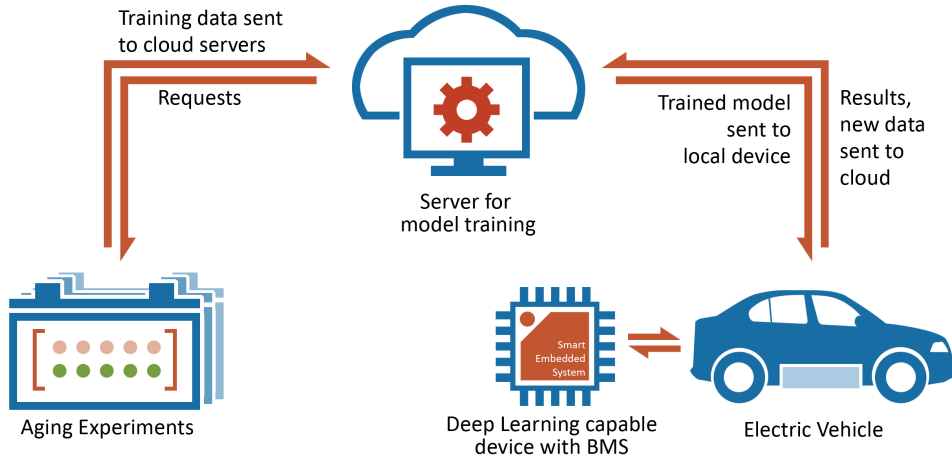


Figure 4: An overview of the model framework.

training epoch.

3. Methodology

3.1. Model framework

The framework of the online capacity estimation model can be split into two major parts, namely the initial training of the model, and the deployment of the trained model to vehicles in operation. Fig. 4 shows an overview of the framework. The experiments to gather aging data, as previously discussed, are the first step, after which the obtained dataset is processed and stored in the cloud to be used for model training. The cloud server offers a bidirectional connection to all inference devices present in the field. Its main task is to train the model with the data provided and serve the best performing model to the devices connected to it. The local devices can also forward their gathered output data, new input samples, and other metrics to the server, which can be used for future improvements of the models. This cloud connectivity feature is a key factor that enables a continuously updatable deep learning model. The hardware present in the cloud server must be sufficiently powerful for model training with huge amounts of gathered data. The local inferences can then be computed on relatively cheap embedded devices within the BMS of EVs, since deep learning models are very computationally cheap to evaluate.

3.2. LSTM-RNN architecture

As mentioned above, this work uses LSTM-RNNs for the estimation of remaining capacity from voltage-time sensor data. Regression methods, e.g. linear models [39] and Gaussian processes [32], need feature extraction from the raw charging data, which requires a lot of domain-specific experiences. Deep learning models have the ability to do efficient feature extraction from the raw data directly. Furthermore, the size

of the input matrix (the number of time-voltage samples) is changing during aging, as shown in Fig. 3, which requires a processing step to have a fixed size of the input matrix for regression methods. In contrast, LSTMs can work with inputs with varying sizes, satisfying the requirements and reducing the processing effort. LSTMs were first proposed by Hochreiter and Schmidhuber in 1997 [40], and gained popularity in the late 2000s, with a simpler version called Gated Recurrent Unit coming out in 2014.

The training of all NNs has its basis on a process called backpropagation. Neural network layers include a non-linear ‘activation function,’ which produces an output for each hidden node in the layers of the NN, after seeing the values of the weights associated with the nodes, and the input. The network learns during training by modifying the values of these node weights so that the activation function outputs values which are matching with the training data provided. This weighting modification is done by the backpropagation process, which takes the output of the NN and generates an error value by comparing it to the labeled data using a cost function termed the ‘loss,’ which is selected manually for each application. The gradients of the error are then propagated backward through all the nodes of the NN, and an ‘optimizer’ function uses this error to optimize the weights of the NN to produce better outputs after every training step. RNNs use a variant of backpropagation called backpropagation through time (BPTT) [41]. This ‘unfolds’ the neural network in time in the forward pass while generating outputs, and calculating the loss for each time step, and then propagates the gradients of the loss with respect to each parameter in the backward pass. The gradients are then used to update the shared parameter vector w across all time steps in the network, using the optimizer. LSTMs were specially developed modifications to standard RNN cells, and they have a hidden memory state along with the cell state at each timestep, which leads them to have the ability for long memory retention.

Fig. 5 shows an overview of the architecture of the LSTM cell. A brief description of the operation of the cell, along with its equations, obtained from [42], is given below, 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. Moreover, h_t and h_{t-1} are the hidden output states of the cell at present time step and the previous time step, respectively, C_{t-1} is the cell’s previous memory state, \tilde{C}_t is the generated new memory, and C_t is the cell’s present memory state. The $(*)$ operator signifies element-wise multiplication of the vectors. There are three gates, namely the input gate i_t , the forget gate f_t and the output gate o_t , and there are two generator circuits, namely the new memory (\tilde{C}_t) generator and the hidden state (h_t) generator.

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

In this paper, the architecture was developed in a Python 3 environment, with TensorFlow 1.14 as the backend and the Keras deep learning library used for layer creation [43, 44]. The LSTM-RNN cell is wrapped in a bidirectional wrapper, which allows the network to process the input timesteps twice, once normally and once in reverse, for additional context and feature recognition while training. The optimizer ‘‘Adam’’ [45], used for training the network, is an adaptive learning method that computes individual learning rates for each parameter in the network. Adam is the current state-of-the-art for training time series neural networks and has been demonstrated to be one of the most efficient training processes, especially when coupled with a momentum parameter, as in [46]. It uses the first and second moments of the error gradients to update the learning rates of each parameter of the network individually, leading to highly efficient training.

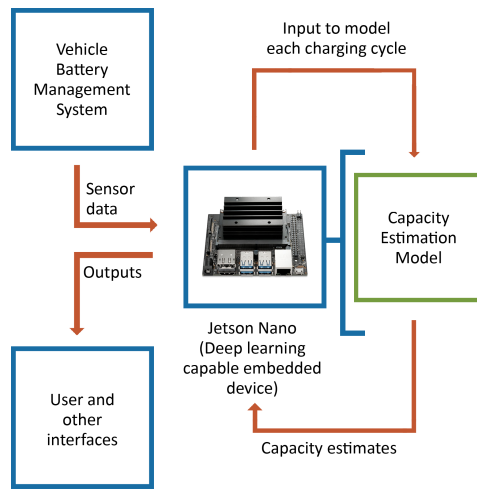
Criteria	Value
Number of epochs	250
Learning rate	0.0001
Validation split	20%
Dropout	30%
Minibatch size	1900 samples
Training loss	Mean absolute error

Table 2: Training Hyperparameters.

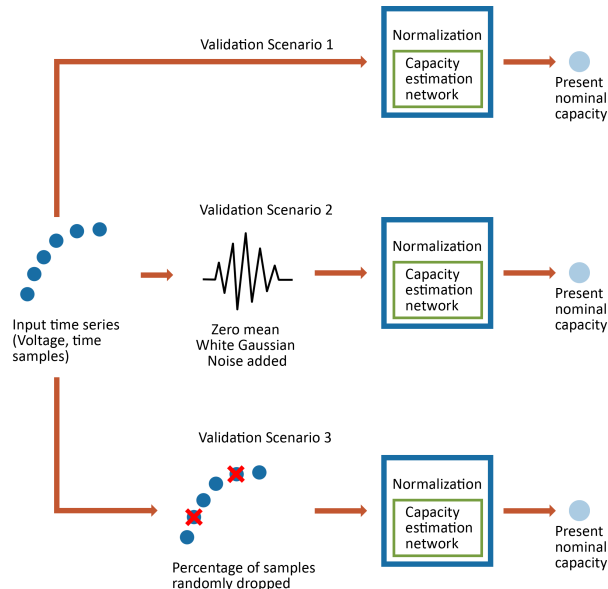
3.3. Network training

Several bidirectionally wrapped LSTM-RNN networks were trained, with varying layer and node dimension sizes, and the most efficient network after observation of performance metrics (defined in section 4) was chosen for implementation. The network loss for learning was chosen to be ‘mean absolute error’ since this loss performs best for regression type models. Other experiment hyper-parameters, such as training length, learning rate limits, and the training-testing split of the training dataset for the network training, are given in Table 2. Regularization was implemented in all networks in the form of dropout layers, which assigns a certain probability for each node to be randomly dropped from being trained in the current training step, increasing the generality of the network further and decreasing the chances of overfitting the network to the training data.

The best performing network after training was found to be a bidirectional LSTM network with four layers and 50 nodes per layer, with the best training loss at 0.94%. A simple RNN cell with the same number of layers and hidden nodes per layer was also trained to have a benchmark for the effectiveness of the LSTM-RNN on the data. The final training loss reached by the simple RNN was 7.8% It is clear that the accuracy is low because the simple RNN does not have the long short-term memory ability of the



(a)



(b)

Figure 6: (a) The role of the processor in the loop. (b) The different validation scenarios performed during the processor-in-the-loop validation.

LSTM-RNN, therefore making it impossible for it to derive the correlations between the charging curve data and capacity when the number of time steps is large, causing the loss of predictive ability. However, the LSTM-RNN does not suffer from this setback and can accurately capture the full range of correlations between the input and output throughout the cell's lifetime.

3.4. Processor-in-the-loop validation

As talked about in the framework section, this model is meant to be usable in small deep learning capable embedded devices, highlighted in Fig. 4, which could serve as the processor for the data fed from the sensors attached to the battery inside an EV in operation. The embedded device used for the processor-in-the-loop (PIL) validation in this paper is a small on-board computer specially designed for machine learning and deep learning applications, the Nvidia Jetson Nano. Its role in the framework is shown in Fig. 6(a). It has a small form factor and a CUDA technology-enabled processing unit, which enables it to perform deep-learning operations, making it ideal for association with an EV’s BMS [47] for prognostics tasks. The PIL validation was done after deploying the trained model into the Nano. The connection environment to an actual BMS to stream sensor data was not realized in this work; rather, the data arrays are fed to the Nano, from which the capacity estimation model takes input cycle by cycle as it would do in the case of real data availability. The objective is to validate the model by simulating various conditions reflecting real operation scenarios and to demonstrate the computing capability and the viability of using the proposed model in such devices in future BMSs.

4. Results and discussion

There are three validation scenarios, as shown in Fig. 6(b), for the capacity estimation model. Validation with ideal inputs, as well with special input condition scenarios, was chosen to test the robustness of the model when performing in real-life operation conditions, where the network may have less than ideal inputs provided to it. Data from eight battery cells was used for validation, as stated before, and this data was not used for the model training process. The cells having the best and worst-case results are shown in the tables below, for all scenarios. The error metrics used are defined as follows:

$$\text{APE} = \left| \frac{\tilde{y} - y}{y} \right| \times 100\% \quad (7)$$

$$\text{MAPE} = \frac{\sum_{i=1}^n \text{APE}_i}{n} \quad (8)$$

$$\text{M-SIGMA} = \frac{\sum_{i=1}^n |\text{APE}_i - \text{MAPE}|}{n} \quad (9)$$

where y and \tilde{y} are the real and the estimated capacity, respectively, and n is the number of data points for each battery cell.

There are two significant evaluation metrics, the first of which is the mean absolute percentage error (MAPE) score of the network on each battery cell. The equation for calculating the MAPE score is given in (8). MAPE highlights the average accuracy of the model’s estimates over the whole life of the battery cell.

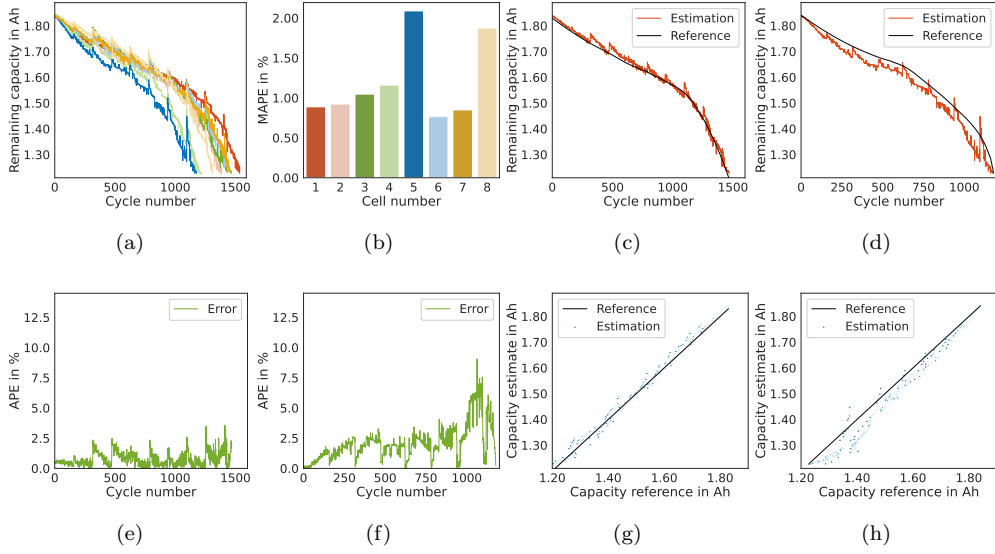


Figure 7: Validation results of scenario 1. The first two plots are for overall metrics with (a) Estimated maximum capacity vs. cycle number plot for all 8 cells and (b) MAPE comparison of all 8 cells. Comparison plots for estimates and references for the (c) best-case cell and (d) worst-case cell are shown in the next two figures. The following two plots show the error in percentage for each charging cycle, over the life of the (e) best-case cell and (f) worst-case cell. Plots of estimates as a function of the reference values to highlight the deviation of the estimates from the references for the (g) best-case cell and (h) worst-case cell, over their lifetime.

The second evaluation metric is the mean deviation from the MAPE, M-SIGMA of the network for each cell, with its equation given in (9). M-SIGMA highlights the consistency of the quality of model estimates by indicating the average deviation of the estimation error from the MAPE, over the whole life of the battery cell. Both scores indicate better performance with lower values. The three validation scenarios, their results, and relevant discussion are given subsequently.

4.1. Capacity estimation from nominal sensor input

As highlighted in section 3.4, the model directly receives the raw sensor data of the partial charging curve without normalization or padding and processes one curve at a time, as would be the use case in a real application. The curves are then dynamically normalized by the model using similar predefined limits, and estimation is done without the need for zero paddings to highlight the ability of the LSTM network to handle input data of varying time steps. The result metrics of the best-case and worst-case cells from the validation set are given in Table 3.

Fig. 7 shows the results from validation scenario 1. Fig. 7(a) shows the plot comparing the estimated capacity degradation curves of all eight cells of the testing dataset. This is to show that the model can capture different capacity degradation paths for the different cells, and does not produce a single averaged

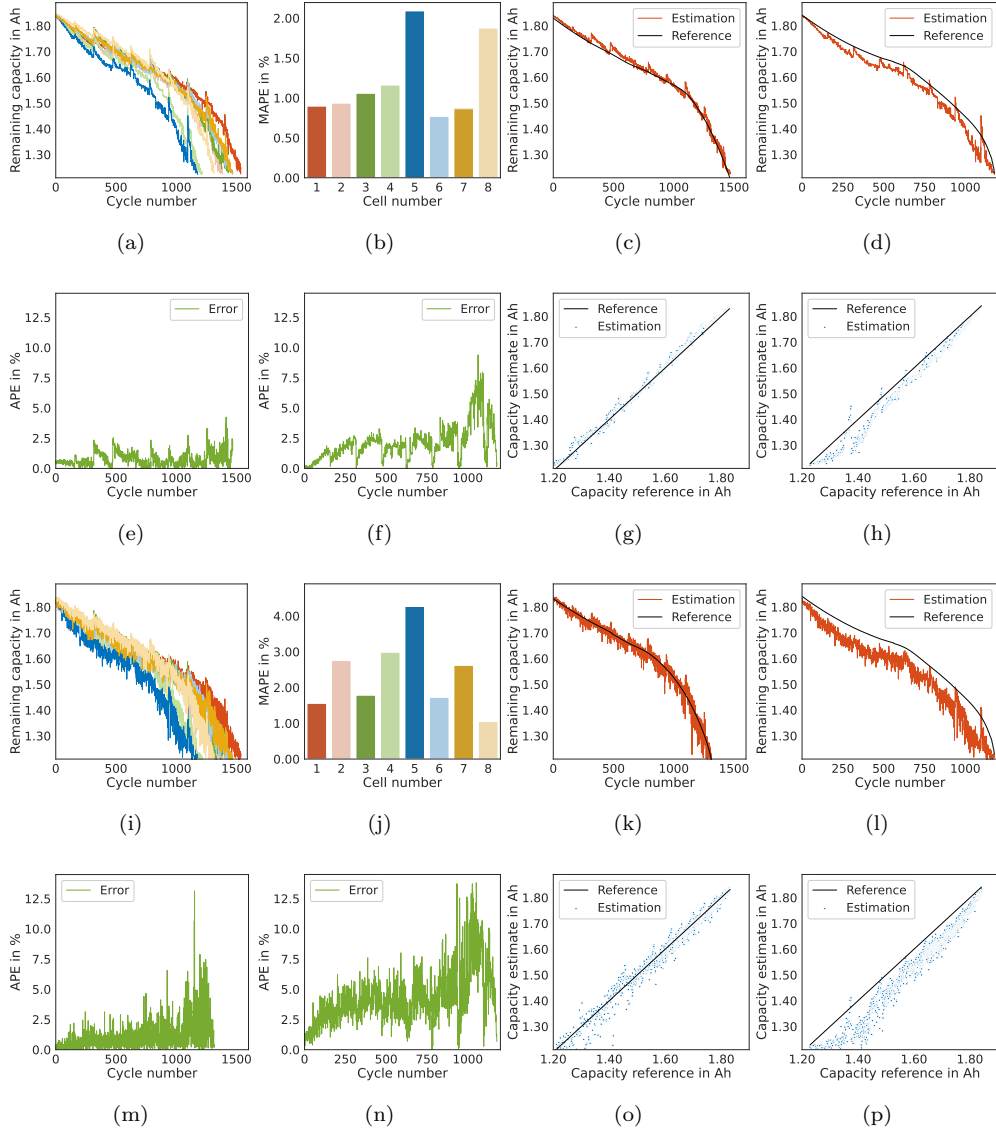


Figure 8: Validation results of scenario 2 (noise added) in rows 1 and 2 and scenario 3 (samples dropped) in rows 3 and 4. The first two figures (a, b, i, j) of row 1 and 3 show the estimated remaining capacity vs. cycle number plot and the MAPE comparison of all 8 cells, respectively. Estimations and references comparison plot for the (c, k) best-case cell and (d, l) worst-case cell are shown for both scenarios. Similarly, the error progression plot over lifetime of (e, m) best-case cell and (f, n) worst-case cell, as well as the deviation of estimations from references plot for the (g, o) best-case cell and (h, p) worst-case cell are also shown.

curve for all the cells in the dataset. In Fig. 7(b), the MAPE of each cell over the entire life is given and can be seen to be within acceptable tolerances, with the mean MAPE of all eight cells being 1.24%, and the best and worst-case cells having a MAPE of 0.76% and 2.08%, respectively. Fig. 7(c) and (d) show the comparison plots of estimated capacity against real capacity from the validation data to highlight the

Criteria	Best-case	Worst-case
MAPE	0.76%	2.08%
M-SIGMA	0.46%	1.02%

Table 3: Validation results of scenario 1.

accuracy of the model in the best and worst cases. In contrast, Fig. 7(e) and (f) show the error values for each cycle in the best and worst-case cell, and lead to the observation that the estimation is more erroneous in the late stages of the cell’s life, which is because the quality of input collected also suffers when the cell is close to it’s EOL. Fig. 7(g) and (h) show the deviations of each estimation from the reference, by plotting the estimated capacity values as a function of the reference capacities with the reference line added for the best-case and worst-case cells. This is a visualization of the model’s M-SIGMA score and serves as an indicator of the consistency of the quality of estimates of the network.

4.2. Capacity estimation from noisy sensor input

This scenario highlights the robustness of the model while dealing with noisy sensor data. The data generation method is like scenario 1, with the addition of zero-mean additive white gaussian noise (AWGN) with a mean standard deviation (σ) of 5% to both the time and the voltage signal, before forwarding it to the model. This level of noise is higher than most industry acceptable sensor tolerances. The result metrics of the best-case and worst-case cells from the validation set are given in Table 4.

Criteria	Best-case	Worst-case
MAPE	0.77%	2.09%
M-SIGMA	0.47%	1.02%

Table 4: Validation results of scenario 2.

The first two rows of Fig. 8 show the result plots of validation scenario 2, i.e., validation when noise is added to the input data. The model proves to be extremely robust when estimating with noisy inputs since the mean MAPE for the entire validation dataset is still well within 3%, with the best-case MAPE being 0.77% and the worst-case MAPE being 2.09%, as seen in Fig. 8 (b). The estimation comparison with reference plots, error progression over the lifetime of the cell plots and the estimation deviation from reference plots are also shown, similar to scenario 1, for the best-case cell in Fig. 8(c), (e), and (g) and the worst-case cell in Fig. 8(d), (f), and (h), respectively.

4.3. Capacity estimation from incomplete sensor input

This evaluation scenario is used to highlight the predictive ability of the model when a percentage of data is randomly selected to be dropped while sampling before feeding the input to the model. The model was validated with 15% of the input samples randomly dropped before being forwarded to the estimation network. The result metrics of the best-case and worst-case cells from the validation set are given in Table 5.

Criteria	Best-case	Worst-case
MAPE	1.04%	4.26%
M-SIGMA	0.68%	1.66%

Table 5: Validation results of scenario 3.

The third and fourth rows of Fig. 8 show the validation results for scenario 3 similarly, with Fig. 8 (i) showing the degradation curves and Fig. 8(j) showing the MAPE score of the 8 test cells, as before. A key feature of LSTM networks is that they can work with flexible input sizes, and the input window shrinking or expanding does not hinder the model operation. The results show that the model suffers from missed data more than noisy data, which is expected since it is a data-based estimation approach. However, even with this handicap, the model performs appreciably well, as seen in the results. Validation with 15% of the data dropped is carried out, and as can be seen in Fig. 8(j), the mean MAPE is still less than 3%, and the worst-case MAPE is 4.26%, which is still within the 5% error tolerance band. As can be seen from the plots, the model suffers the most from missing data, with the error increasing by more than 100% of the error with a complete input set for the worst case. The best case error, even with 15% of input data missing, is at 1.04%, which is also acceptable. The prediction comparison with reference plots, error progression over the lifetime of the cell plots, and the prediction deviation plots are also shown for the best-case cell in Fig. 8(k), (m), and (o), and for the worst-case cell in Fig. 8(l), (n), and (p), respectively.

5. Conclusions

This work aimed to create a battery remaining capacity estimation model that is suitable for application in electric vehicles under real-world operation, to be associated with future battery management systems with the capability to take advantage of intelligent diagnostics algorithms.

To date, in this domain, no comparable work has been found which implements long short-term memory networks to use raw voltage-time sensor data from partial CC-phase charging curves to obtain remaining capacity estimates. The validation of our model with a market-ready machine learning capable embedded device was also achieved, demonstrating the viability of deep learning-based diagnostics in future battery management systems. The cloud-based training and deployment to localized devices for implementation

makes model operation and updating a seamless process with no changes to the architecture needed even with new data, the model parameters just need to be updated online with new training and then distributed. Several highlights of the proposed model are given below:

- Estimates battery remaining capacity from raw data from constant current charging curves without the requirement of additional processing of the raw voltage-time charging curve samples.
- LSTM-based time series processing architecture implemented, allowing the input charging curves to be variable in time steps and estimation also possible with incomplete sensor data.
- Fast computation times, as measured in an embedded device, at around 2 seconds per sample charging curve. This time is considered negligible, especially when put in relation to the time taken to gather the sensor data per charging cycle of the EV.
- Accuracy is acceptable by industry standards with error (MAPE) in the worst-case being 2.08% and increasing negligibly upon the addition of AWGN noise to the input.
- The model still outputs reasonable predictions with up to 15% missing data, with the worst-case error being 4.26%.

In the future, we aim to expand the scope of this model in various directions, the priority being to expand the model from capacity diagnostics to future prediction of capacity and lifetime, and to make the model more general, by training with different types of data such as common drive cycles, accelerated aging cycles and different types of cells used in various aging scenarios. Exploring further options for increasing training efficiency and looking at upcoming network architectures specialized for such applications can also be added to future goals. This direction will serve to complete the functionality of the model and will require the use of different deep learning-based networks to predict the future degradation path of each cell, based on the past data. Finally, the full integration of the model in an embedded platform by setting up a deep learning-based inference environment within a BMS with application-based prediction and display capacity, along with other metrics of the battery needs to be implemented and evaluated in the future.

Acknowledgment

This work was financially supported within the project EVERLASTING from the Horizon 2020 program by the European Union [grant number EVERLASTING-713771].

References

- [1] J. Vetter, P. Novák, M. R. Wagner, C. Veit, K.-C. Möller, J. O. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, A. Hammouche, Ageing mechanisms in lithium-ion batteries, *Journal of Power Sources* 147 (1-2) (2005) 269–281. doi: 10.1016/j.jpowsour.2005.01.006.

- [2] J. M. Reniers, G. Mulder, D. A. Howey, Review and performance comparison of mechanical-chemical degradation models for lithium-ion batteries, *Journal of The Electrochemical Society* 166 (14) (2019) A3189–A3200. doi:10.1149/2.0281914jes.
- [3] D. Andre, M. Meiler, K. Steiner, C. Wimmer, T. Soczka-Guth, D. U. Sauer, Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. i. experimental investigation, *Journal of Power Sources* 196 (12) (2011) 5334–5341. doi:10.1016/j.jpowsour.2010.12.102.
- [4] M. Galeotti, L. Cinà, C. Giammanco, S. Cordiner, A. Di Carlo, Performance analysis and soh (state of health) evaluation of lithium polymer batteries through electrochemical impedance spectroscopy, *Energy* 89 (2015) 678–686. doi:10.1016/j.energy.2015.05.148.
- [5] Y. Li, M. Abdel-Monem, R. Gopalakrishnan, M. Berecibar, E. Nanini-Maury, N. Omar, P. van den Bossche, J. van Mierlo, A quick on-line state of health estimation method for li-ion battery with incremental capacity curves processed by gaussian filter, *Journal of Power Sources* 373 (2018) 40–53. doi:10.1016/j.jpowsour.2017.10.092.
- [6] C. Pastor-Fernández, K. Uddin, G. H. Chouchelamane, W. D. Widanage, J. Marco, A comparison between electrochemical impedance spectroscopy and incremental capacity-differential voltage as li-ion diagnostic techniques to identify and quantify the effects of degradation modes within battery management systems, *Journal of Power Sources* 360 (2017) 301–318. doi:10.1016/j.jpowsour.2017.03.042.
- [7] M. Lewerenz, A. Marongiu, A. Warnecke, D. U. Sauer, Differential voltage analysis as a tool for analyzing inhomogeneous aging: A case study for lifepo4—graphite cylindrical cells, *Journal of Power Sources* 368 (2017) 57–67. doi:10.1016/j.jpowsour.2017.09.059.
- [8] C. R. Birkl, M. R. Roberts, E. McTurk, P. G. Bruce, D. A. Howey, Degradation diagnostics for lithium ion cells, *Journal of Power Sources* 341 (2017) 373–386. doi:10.1016/j.jpowsour.2016.12.011.
- [9] W. Li, M. Rentemeister, J. Badedá, D. Jöst, D. Schulte, 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* (2020).
- [10] H. He, R. Xiong, H. Guo, Online estimation of model parameters and state-of-charge of lifepo4 batteries in electric vehicles, *Applied Energy* 89 (1) (2012) 413–420. doi:10.1016/j.apenergy.2011.08.005.
- [11] G. L. Plett, Sigma-point kalman filtering for battery management systems of lipb-based hev battery packs, *Journal of Power Sources* 161 (2) (2006) 1369–1384. doi:10.1016/j.jpowsour.2006.06.004.
- [12] Q. Wang, Y. Jiang, Y. Lu, State of health estimation for lithium-ion battery based on d-ukf, *International Journal of Hybrid Information Technology* 8 (7) (2015) 55–70. doi:10.14257/ijhit.2015.8.7.06.
- [13] Y. Zou, X. Hu, H. Ma, 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* 273 (2015) 793–803. doi:10.1016/j.jpowsour.2014.09.146.
- [14] W. Li, D. Cao, D. Jöst, F. Ringbeck, M. Kuipers, F. Frie, D. U. Sauer, Parameter sensitivity analysis of electrochemical model-based battery management systems for lithium-ion batteries, *Applied Energy* (2020).
- [15] W. Li, D. W. Limoge, J. Zhang, D. U. Sauer, A. M. Annaswamy, Estimation of potentials in lithium-ion batteries using neural networks, *IEEE Transactions on Control Systems Technology* (under review).
- [16] D. W. Limoge, A. M. Annaswamy, An adaptive observer design for real-time parameter estimation in lithium-ion batteries, *IEEE Transactions on Control Systems Technology* doi:10.1109/TCST.2018.2885962.
- [17] B. Jenkins, A. Krupadanam, A. M. Annaswamy, Fast adaptive observers for battery management systems, *IEEE Transactions on Control Systems Technology* doi:10.1109/TCST.2019.2891234.
- [18] A. Bartlett, J. Marcicki, S. Onori, G. Rizzoni, X. G. Yang, 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*, IEEE, 2013, pp. 7791–7796. doi:10.1109/cdc.2013.6761126.
- [19] S. J. Moura, M. Krstic, N. A. Chaturvedi, Adaptive pde observer for battery soc/soh estimation, in: *Volume 1*:

- Adaptive Control; Advanced Vehicle Propulsion Systems; Aerospace Systems; Autonomous Systems; Battery Modeling; Biochemical Systems; Control Over Networks; Control Systems Design; Cooperativ, ASME, 2012, pp. 101–110. doi:10.1115/dscc2012-movic2012-8800.
- [20] L. Zheng, L. Zhang, J. Zhu, G. Wang, J. Jiang, Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model, *Applied Energy* 180 (2016) 424–434. doi:10.1016/j.apenergy.2016.08.016.
- [21] C. Zou, C. Manzie, D. Nešić, 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* 335 (2016) 121–130. doi:10.1016/j.jpowsour.2016.10.040.
- [22] N. Yang, J. Feng, Q. Sun, T. Liu, D. Zhong, Online estimation of state-of-health for lithium ion batteries based on charge curves, in: 2016 11th International Conference on Reliability, Maintainability and Safety (ICRMS), IEEE, 2016, pp. 1–8. doi:10.1109/icrms.2016.8050034.
- [23] H. Chaoui, C. C. Ibe-Ekeocha, State of charge and state of health estimation for lithium batteries using recurrent neural networks, *IEEE Transactions on Vehicular Technology* 66 (10) (2017) 8773–8783. doi:10.1109/TVT.2017.2715333.
- [24] A. Eddahech, O. Briat, N. Bertrand, J.-Y. Deléage, 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* 42 (1) (2012) 487–494. doi:10.1016/j.ijepes.2012.04.050.
- [25] H. Chaoui, C. C. Ibe-Ekeocha, H. Gualous, Aging prediction and state of charge estimation of a lifepo 4 battery using input time-delayed neural networks, *Electric Power Systems Research* 146 (2017) 189–197. doi:10.1016/j.epsr.2017.01.032.
- [26] Y. Ding, C. Lu, J. Ma, Li-ion battery health estimation based on multi-layer characteristic fusion and deep learning, in: 2017 IEEE Vehicle Power and Propulsion Conference (VPPC), IEEE, 2017, pp. 1–5. doi:10.1109/vppc.2017.8331058.
- [27] H.-T. Lin, T.-J. Liang, S.-M. Chen, Estimation of battery state of health using probabilistic neural network, *IEEE Transactions on Industrial Informatics* 9 (2) (2013) 679–685. doi:10.1109/tii.2012.2222650.
- [28] J. Wu, C. Zhang, Z. Chen, An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks, *Applied Energy* 173 (2016) 134–140. doi:10.1016/j.apenergy.2016.04.057.
- [29] J. Wu, Y. Wang, X. Zhang, Z. Chen, A novel state of health estimation method of li-ion battery using group method of data handling, *Journal of Power Sources* 327 (2016) 457–464. doi:10.1016/j.jpowsour.2016.07.065.
- [30] C. Hu, G. Jain, C. Schmidt, C. Strief, M. Sullivan, Online estimation of lithium-ion battery capacity using sparse bayesian learning, *Journal of Power Sources* 289 (2015) 105–113. doi:10.1016/j.jpowsour.2015.04.166.
- [31] V. Klass, M. Behm, G. Lindbergh, A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation, *Journal of Power Sources* 270 (2014) 262–272. doi:10.1016/j.jpowsour.2014.07.116.
- [32] R. R. Richardson, C. R. Birkl, M. A. Osborne, D. A. Howey, Gaussian process regression for in situ capacity estimation of lithium-ion batteries, *IEEE Transactions on Industrial Informatics* 15 (1) (2019) 127–138. doi:10.1109/tii.2018.2794997.
- [33] Z. Wang, J. Ma, L. Zhang, State-of-health estimation for lithium-ion batteries based on the multi-island genetic algorithm and the gaussian process regression, *IEEE Access* 5 (2017) 21286–21295. doi:10.1109/access.2017.2759094.
- [34] J. Wei, G. Dong, Z. Chen, Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression, *IEEE Transactions on Industrial Electronics* 65 (7) (2018) 5634–5643. doi:10.1109/tie.2017.2782224.
- [35] D. Yang, X. Zhang, R. Pan, Y. Wang, Z. Chen, A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve, *Journal of Power Sources* 384 (2018) 387–395. doi:10.1016/j.jpowsour.2018.03.015.
- [36] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, L. Zhang, Remaining useful life prediction for lithium-ion battery: A deep learning approach, *IEEE Access* 6 (2018) 50587–50598. doi:10.1109/ACCESS.2018.2858856.

- [37] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International conference on machine learning, 2013, pp. 1310–1318.
- [38] T. Baumhöfer, M. Brühl, S. Rothgang, D. U. Sauer, Production caused variation in capacity aging trend and correlation to initial cell performance, *Journal of Power Sources* 247 (2014) 332–338. doi:10.1016/j.jpowsour.2013.08.108.
- [39] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggadakis, M. Z. Bazant, S. J. Harris, W. C. Chueh, R. D. Braatz, Data-driven prediction of battery cycle life before capacity degradation, *Nature Energy* 4 (5) (2019) 383–391. doi:10.1038/s41560-019-0356-8.
- [40] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
- [41] P. J. Werbos, Backpropagation through time: what it does and how to do it, *Proceedings of the IEEE* 78 (10) (1990) 1550–1560. doi:10.1109/5.58337.
- [42] Olah, Christopher, Understanding lstm networks (2015).
- [43] Keras: a python deep learning library.
URL <https://keras.io/>
- [44] Tensorflow: a machine learning platform.
URL <https://www.tensorflow.org/>
- [45] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2014).
URL <http://arxiv.org/pdf/1412.6980v9>
- [46] Timothy Dozat, Incorporating nesterov momentum into adam.
URL http://cs229.stanford.edu/proj2015/054_report.pdf
- [47] M. Lelie, T. Braun, M. Knips, H. Nordmann, F. Ringbeck, H. Zappen, D. Sauer, Battery management system hardware concepts: An overview, *Applied Sciences* 8 (4) (2018) 534. doi:10.3390/app8040534.