Journal of Power Sources 506 (2021) 230024

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

One-shot battery degradation trajectory prediction with deep learning

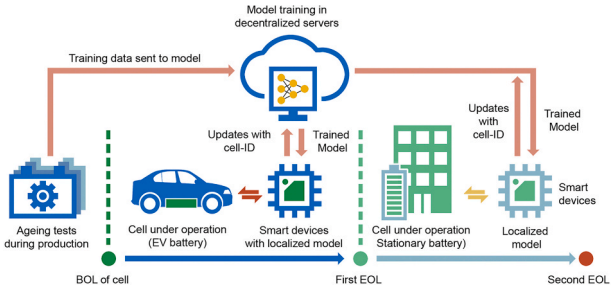
Weihan Li [a,b,*], Neil Sengupta [a], Philipp Dechent [a,b], David Howey [d,e], Anuradha Annaswamy [f], Dirk Uwe Sauer [a,b,c]

[a] Chair for Electrochemical Energy Conversion and Storage Systems, Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Jägerstraße 17-19, 52066, Aachen, Germany
[b] Jülich Aachen Research Alliance, JARA-Energy, Templergraben 55, 52056, Aachen, Germany
[c] Helmholtz Institute Münster (HI MS), IEK-12, Forschungszentrum Jülich, 52425, Jülich, Germany
[d] Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK
[e] The Faraday Institution, Harwell Campus, Didcot, OX11 0RA, UK
[f] Department of Mechanical Engineering, Massachusetts Institute of Technology, 33 Massachusetts Ave, MA, 02139, Cambridge, USA

HIGHLIGHTS

- Deep learning model to predict the battery future degradation pathway in one shot.
- Digital battery passport enabling degradation monitoring in first- and second-life.
- Accurate early-life prediction capability for end-of-life points and knee-points.
- Robust capacity error handling capacility validated by processor-in-the-loop tests.
- Computation burden decreased by 15 times compared with the iterative approaches.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Lithium-ion
Battery
Degradation
Deep learning
Prediction
Knee-point

ABSTRACT

The degradation of batteries is complex and dependent on several internal mechanisms. Variations arising from manufacturing uncertainties and real-world operating conditions make battery lifetime prediction challenging. Here, we introduce a deep learning-based battery health prognostics approach to predict the future degradation trajectory in one shot without iteration or feature extraction. We also predict the end-of-life point and the knee-point. The model correctly learns about intrinsic variability caused by manufacturing differences, and is able to make accurate cell-specific predictions from just 100 cycles of data, and the performance improves over time as more data become available. Validation in an embedded device is demonstrated with the best-case median prediction error over the lifetime being 1.1% with normal data and 1.3% with noisy data. Compared to state-of-the-art approaches, the one-shot approach shows an increase in accuracy as well as in computing speed by up to 15 times. This work further highlights the effectiveness of data-driven approaches in the domain of health prognostics.

* Corresponding author. Chair for Electrochemical Energy Conversion and Storage Systems, Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Jägerstraße 17-19, 52066, Aachen, Germany.
E-mail addresses: weihan.li@isea.rwth-aachen.de (W. Li), batteries@isea.rwth-aachen.de (D.U. Sauer).

https://doi.org/10.1016/j.jpowsour.2021.230024
Received 10 February 2021; Received in revised form 8 April 2021; Accepted 7 May 2021
Available online 10 June 2021

1. Introduction

Lithium-ion batteries (LIBs) are electrochemical systems which are increasingly used in many sectors, especially in electric mobility [1]. Their performance degrades with usage and time, and the degradation can usually be divided with a 'knee-point' into an early life section and then a later life section with accelerated degradation [2]. The end-of-life (EOL) of LIBs is usually defined, e.g. for electric vehicles (EVs), as the point where the remaining capacity falls below 80% of the nominal capacity. Some cells can then also be used in second-life applications with an extended EOL/remaining useful life (RUL). Accurate prediction of the remaining capacity of the cells at an early stage not only benefits maintenance, safety and asset optimization but also serves as a starting point for the technical and economic analysis of possible second-life applications. However, accurate prediction of the future degradation of LIBs is not a trivial task, as the ageing of LIBs is a complex nonlinear process with various internal mechanisms whose dynamics are highly challenging to measure and model accurately.

There has been extensive research in the field of capacity degradation prediction in LIBs. Observation of existing literature reviews, such as [3–5], highlights various approaches for tackling the degradation modelling problem, which can be divided into either mathematical model-based or data-driven methods. Mathematical model-based methods use a physics-based battery model to describe the degradation behaviour of the cells considering battery dynamics. The model can be either an equivalent circuit model (ECM) [6–14], an electrochemical model (EM) [15–21], or an empirical model [22–25]. These battery models are then parameterized via various methods. However, adequate parameterization of such models so that all the underlying degradation mechanics of the battery are captured is an open research topic, and these models are either too complex to be used in online applications with cheap embedded hardware or may not capture the real-life variations in degradation accurately.

Data-driven approaches are a possible solution to the parameterization problem since they don't depend upon a physical model but instead generate the model directly from the data. There have been various approaches using data-driven models, as well as other types of models parameterized using data-driven tools. One approach [26–28] fits EOL (or RUL) as an output using features from early cycling data as an input. Since these models output a point prediction of the EOL or RUL, rather than a prediction of the degradation trajectory, they are usually better suited for broad screening of cells, rather than accurate tracking of degradation. For ongoing tracking of ageing, a more sophisticated approach is to predict the entire degradation curve of the battery iteratively from one or more pre-determined points during the battery life. Several approaches use past capacity series up to a certain point in the life of the cell to develop a parameterized model that is trained using a fixed size rolling input window on the capacity history. This model can then predict the subsequent future capacity points iteratively up to the EOL of the battery. These models can be based on various data-driven approaches such as Gaussian processes [29–34], support vector machines [35–37], relevance vector machines [38–40], and neural networks [41–49]. The main advantage of the iterative prediction approach is that it is flexible and can be used to model changing usage patterns. However, the major disadvantage of these types of models is the associated computing speed and costs, since these models need to run many times to get the prediction of the degradation curve. Moreover, since these models are usually trained with data just from the early parts of the cell's degradation, multiple retraining is needed to update the model as new cell data becomes available over time so that the model can predict the present degradation trends appropriately.

Here, we propose a deep learning model with multiple long short-term memory (LSTM) layers in both encoder and decoder blocks, aiming to bridge the aforementioned research gap by predicting the entire capacity degradation trajectory in one shot. The model was trained using all the lifetime data of the training cells, rather than just point predictions, and therefore it is able to accurately follow trend changes at any point in the cell's life without frequent retraining. The major points of interest in the cell's life can be predicted accurately, such as the points when the remaining capacity of the cell reaches 80% (EOL80), and then 65% of the nominal capacity (EOL65), as well as the knee-point, after which degradation accelerates. This makes the model useful not only for present EV applications but also for second-life applications. Moreover, the model can work with capacity data gathered from any standard SOH estimation model even with estimation errors and can be dynamically updated when new estimates become available. We propose an online prognostics framework that trains models in the cloud and serves trained models to devices present in battery management systems (BMSs) in EVs. This framework enables a health 'cell passport' system, where each cell is individually tagged and the prognostics data from the model for that cell can be carried forward with it over its lifetime. Processor-in-the-loop validation of the model was carried out in a machine learning capable embedded system, demonstrating the viability of the approach not only in the cloud but also onboard future EVs. The model was found to be accurate and robust while handling non-ideal capacity data gathering methods. Finally, the 'one-shot' nature of the trajectory prediction leads to huge gains in computing times, which are almost up to 15 times faster than similar models using iterative point-by-point prediction approaches.

2. Ageing test and dataset

The battery dataset for training the prognostics model can be gathered from experiments in the laboratory or battery management systems in the field. The battery dataset used in this work is from in-house ageing experiments conducted at RWTH Aachen University. The experiments originally aimed to classify a batch of nominally identical 48 Panasonic/Sanyo UR18650E LIB cells based on intrinsic manufacturing variations by finding dependencies between the initial performance of each cell and its cycle lifetime. Compared with the dataset provided in Ref. [28], where cells were tested under different stress factors resulting in large variability in degradation, our dataset here was generated with a large number of cells all under a same ageing scenario, exploring intrinsic cell manufacturing variability and small temperature differences (see Fig. S1 and Fig. S2) in battery packs during operation [50]. As the variability in the degradation trends of the cells in early-life is quite small and only starts to increase greatly from the mid-life of the cells onwards, the accurate prediction of the whole degradation trajectory from early-life data is non-trivial, and even more challenging than in Ref. [28], where high variability of the degradation trends already existed in the early-life. The cells underwent cycling ageing, consisting of characterization tests and cycling tests. A brief analysis of the dataset used here, Fig. 1, highlights several interesting features. Fig. 1a shows a flowchart of the test procedure (see Methods for details). Fig. 1b shows the degradation curves of all the cells in the dataset, highlighting their variability in end-of-life points and knee-points (see Note S1) especially after 700 cycles. The 80% and 65% EOL criteria and knee-points are marked, and we note that predicting down to the 65% limit and predicting the knee-points from the early-stage.

Data is incredibly challenging because of the tiny variations amongst the cells before 500 cycles and the change in the degradation trends over time. This reaffirms the need for an online method that can follow and adapt to individual cell degradation trends. In Fig. 1e and f, the variation in the EOL80 cycle number is indicated with the average EOL80 cycle at 1175 cycles with the variations at that point being within 0.3 Ah. Similarly, the average EOL65 cycle number is found at 1350 cycles with the variation within a range of 0.5 Ah at that cycle number. The average knee-point cycle number of 48 cells, as seen in Fig. 1i, was found to be 1088 cycles, with variations within 0.25 Ah. This highlights the variations present in the cells' degradation trajectories with ageing, at different points of the cell's life, as well as the variations in the three

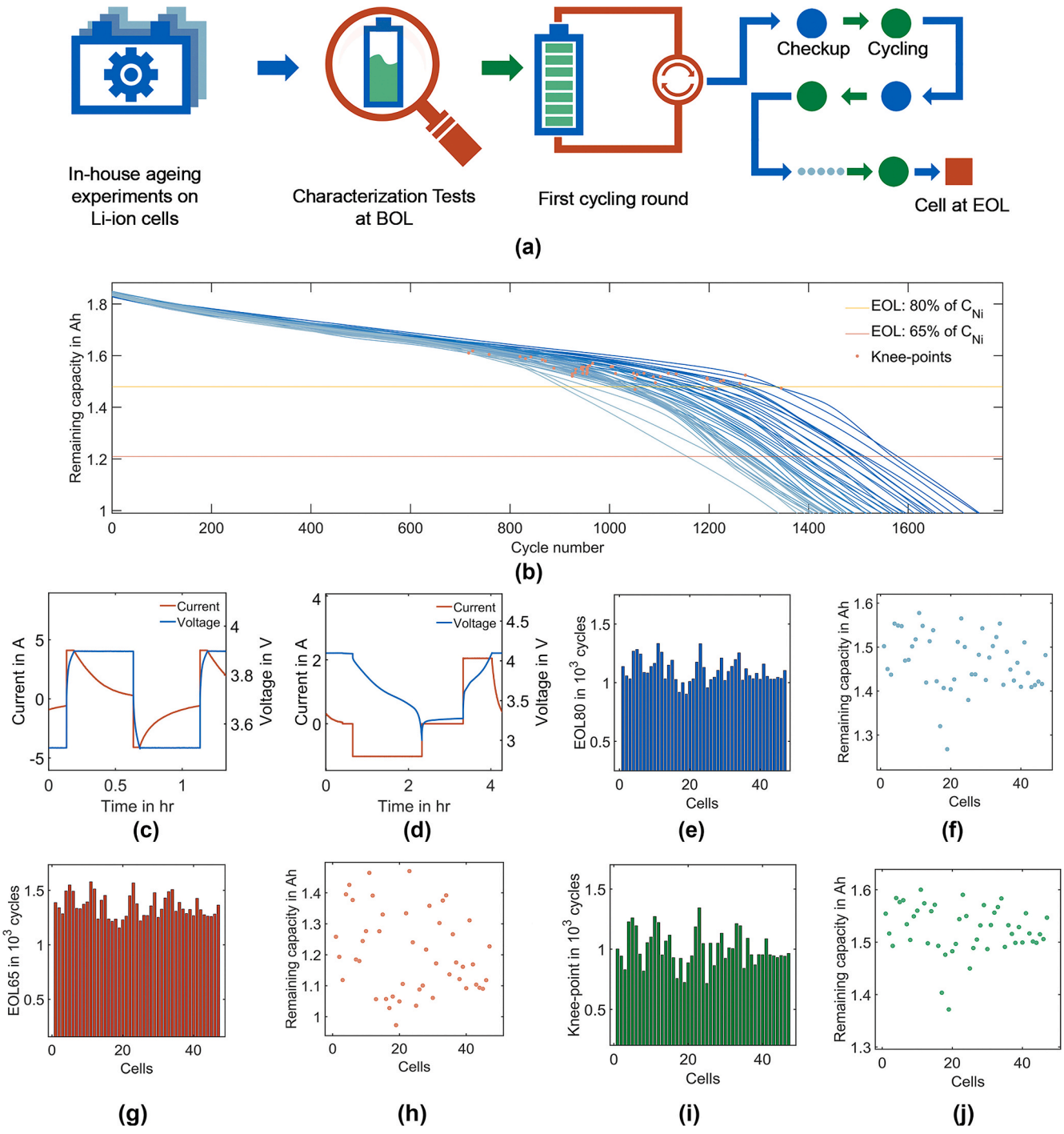


Fig. 1. Ageing experiments used for supervised learning. **(a)** Flowchart of the cyclic ageing process. **(b)** Capacity degradation curves of 48 cells, highlighting the knee-points and the variations in the two different EOL cycle numbers. **(c)** Current and voltage response of one ageing cycle. **(d)** Current and voltage response of one characterization test. **(e)** Variation in the cycle number where EOL80 is reached. **(f)** Variation in the remaining capacity at the average cycle number of reaching the EOL80. **(g)** Variation in the cycle number where EOL65 is reached. **(h)** Variation in the remaining capacity at the average cycle number of reaching the EOL65. **(i)** Variation in the cycle number where the knee-point of capacity degradation is reached. **(j)** Variation in the remaining capacity at the average cycle number of reaching the knee-point.

points of interest in the cell lifetimes.

3. Prognostics framework

Our approach enables the real-time serving of prediction models to BMSs present in EVs or stationary energy storage systems from a cloud server to be then used for local inference. This framework is considered as a part of a 'digital twin' concept for battery systems [10] operating in the field, with battery data being collected for analysis and control over their lifetime. An overview of the framework is given in Fig. 2a, which contains three major steps. The ageing experiments are performed first on several cells of a similar type to the intended use-case, obtaining a dataset for supervised learning. This dataset is then sent to the training server which has computing power for model training and distribution. The server's next task after training is sending the best performing model to embedded devices. The connection is bidirectional since devices also transmit their prediction data and other metrics to the server, which can be used for future model updates. Since each cell's data is sent to the server tagged with a particular ID, a unique 'cell-passport' can be created, which is useful to track performance over the entire lifetime, and also makes providing updates-over-the-air to the models present in the BMSs a trivial task. The cloud connectivity is a key factor that enables the model to be continuously updatable, and the low computation burden of the model also enables it to run locally on BMSs. The model output can be sent to various systems, e.g. supporting energy management, predictive maintenance and full-life cost analysis as needed.

The model's input is the capacity time series until the present time point. No other features or feature selection process is required. The model then outputs the future capacity time series of the cell until the EOL. The capacity of the cells can be measured directly with capacity tests in the training dataset, or estimated with any standard SOH estimation models [51–53] onboard a battery management system. Fig. 2b shows the choice of input, as well as the sampling process of the input data that was undertaken before feeding to the model. The input approach is an increasing window of the remaining capacity array, from the beginning of life (BOL) of the cell, to the present time point. This

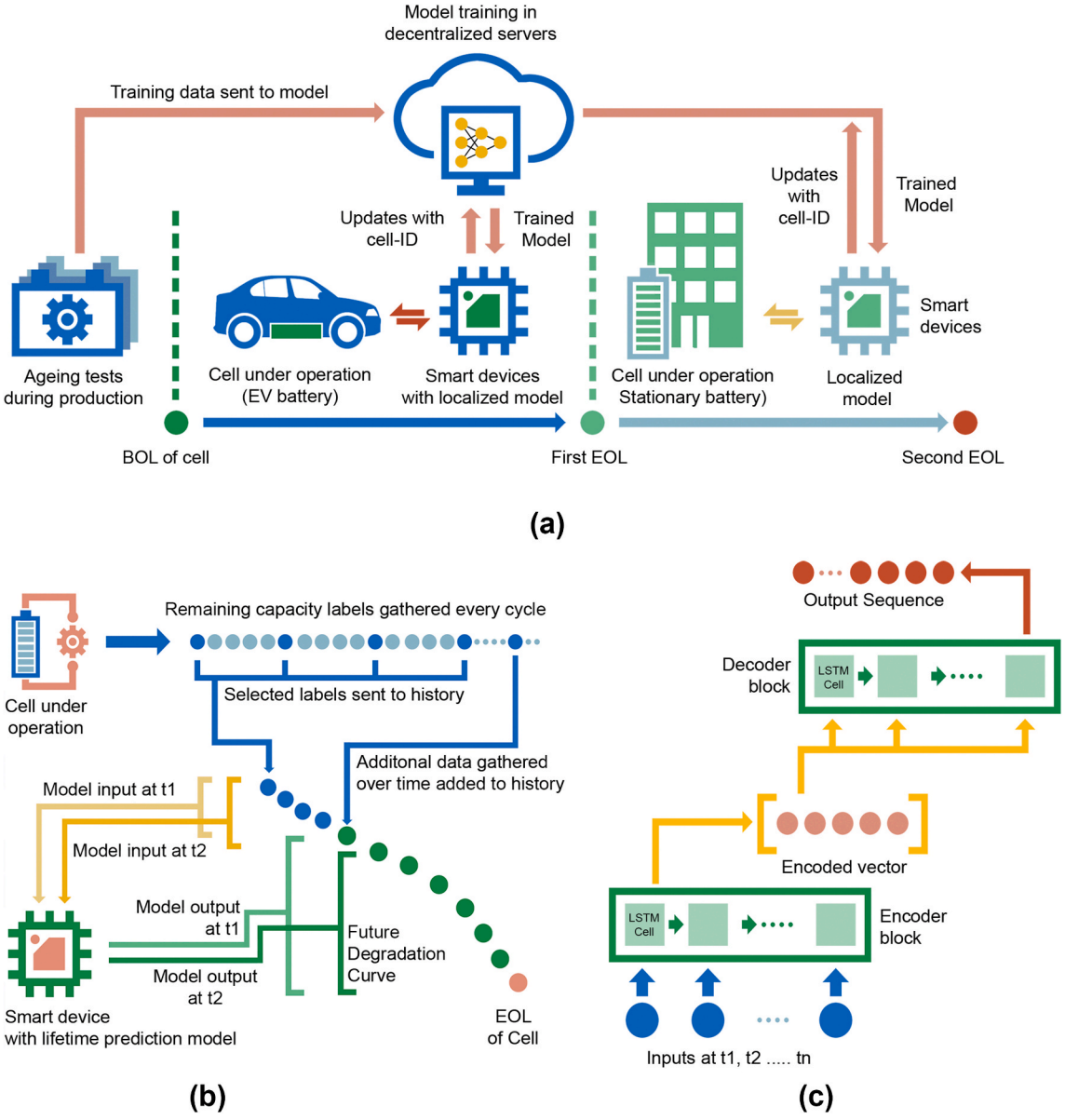


Fig. 2. **(a)** The overview of the proposed framework for the online prediction of battery capacity degradation for cells. The three main sections to the framework are namely, the collection of training data, model training in the cloud, and the operation of the model in both the first and second life of the batteries. **(b)** Choice of input to the model and the approach of increasing the input window with the time. **(c)** The general architecture of the sequence-to-sequence neural network.

input series increases in size as more historical data becomes available, and the model uses the increased information to generate improved predictions. Since both the input and output series lengths are generally constant in LSTM networks, a normal LSTM is insufficient to perform this task. However, RNN-based sequence-to-sequence (S2S) models [54] are specially designed for sequence prediction problems. The S2S approach has a better performance compared with other RNN architectures and performs especially well in predicting sequences of variable length [54], which is the target use case of this work since battery degradation history can be viewed as a variable-length sequence. Fig. 2c highlights the general architecture of the S2S model, which requires two multi-layer LSTM architectures working together. The encoder encodes the input sequence into a static embedding vector, which is then fed to the decoder which decodes the embedding vector and processes it to provide the output sequence desired. The model performance was validated with processor-in-the-loop tests with both normal and noisy capacity data, as shown in Fig. 3. The mean absolute percentage error (MAPE) during curve prediction and absolute cycle number errors for the knee-point and the two EOLs are used to evaluate the prediction performance of the models, as defined in the 'Evaluation criteria' section.

4. Results and discussion

4.1. Model performance

Model validation was undertaken on five random cells that were kept aside for blind testing. Validation was first undertaken with normal input data, serving as the nominal operation scenario. Subsequently, a validation with noisy inputs was done to test the robustness of the model facing capacity estimation errors, which are common in BMSs. The noisy inputs were generated by adding a zero-mean white Gaussian noise (AWGN) with a standard variance (σ) of 0.005 to the capacity inputs. Table S1 highlights the results obtained from the validation of the model in the best- and worst-cases, for all the above-mentioned metrics, along with the maximum computation times, for both the normal and the noisy input scenarios.

From the results, the model is able to predict the capacity degradation trajectory accurately. The predictions are sufficiently accurate even with input data only up to the first 100 cycles of the cell, which roughly amounts to 7% of the cell's total lifetime, assuming an average cycle life of 1500 cycles for the cell type used in the dataset. The initial error at 100 cycles and mean error in the worst-case cell is 4.2% and 3.1%, respectively, which can be considered to be a good performance, being below 5% error. In the best case, the initial and mean errors are 1.8% and 1.2%, respectively, both values well under the 5% error band. The median of the errors is also satisfactory with the best-case median MAPE being 1.1%, and worst-case 3.1%. Fig. 4a shows the predictive ability of the model in the best-case, with increasing input size, as more capacity degradation history is available to the model. Similarly, Fig. 4b shows the curves from the worst-case cell.

The adaptive ability of the model is demonstrated, with the predictions getting more accurate as the model receives more input data. The metrics indicate similar performance for all the highlighted points of interest, with a trend of larger errors at the beginning of life corresponding to minimum input data to the model and improving prediction quality as more data becomes available. The trend is more evident in the worst-cases rather than the best-cases, with some metrics being extremely accurate even in the early stages in the best-case cells and generally more fluctuations evident over a reduced error-band. This is also confirmed by Fig. 4c, which shows the progression of the MAPE of the predicted curves over the cell's lifetime for the best- and worst-cases, respectively (see Fig. S5 for all test cells). The spike of inaccuracy at the very end of the error progression can be attributed to the model having very few meaningful points to predict and, therefore, the errors between those few points being amplified. It is also observed that the most inaccurate prediction doesn't occur at the first prediction point, but a short distance from the beginning of life. The MAPE of the predictions follow a decreasing trend with a wavelike progression, and it is hypothesized that this occurs because the model tries to minimize the average error of the whole curve by predicting an averagely correct curve when input is low. As the input increases, the model updates the average curve to fit more closely to the data, which leads at first to bigger errors since not all parts of the curve are improved simultaneously, which is then followed by a significant decrease in the overall error.

Fig. 4d and e shows histograms of the errors in prediction for all the curves predicted over the lifetime of the best- and worst-case cell, with the median and the 95th percentile error marked. Fig. 4f, g, and 4h show the progression of error like Fig. 4c, but for the three highlighted metrics of interest, namely the EOL80 error, EOL65 error and the knee-point error, respectively (see Fig. S6 for all test cells). Finally, Fig. 4i, j and 4k show the initial, mean and maximum error values for the three metrics for all five cells. The model is shown to have good performance at all the points of interest, as is clear from the metrics analysis.

Noisy input validation results demonstrate that the model is stable to handle capacity estimation errors without large degradations in performance. The best- and worst-case cells generally remain the same after noise is added, which leads to the conclusion that the model's output depends much more on the features of the data from the cells and the noise is relatively easier to ignore. The performance of the model worsens in every respect, which is to be expected since the input is non-

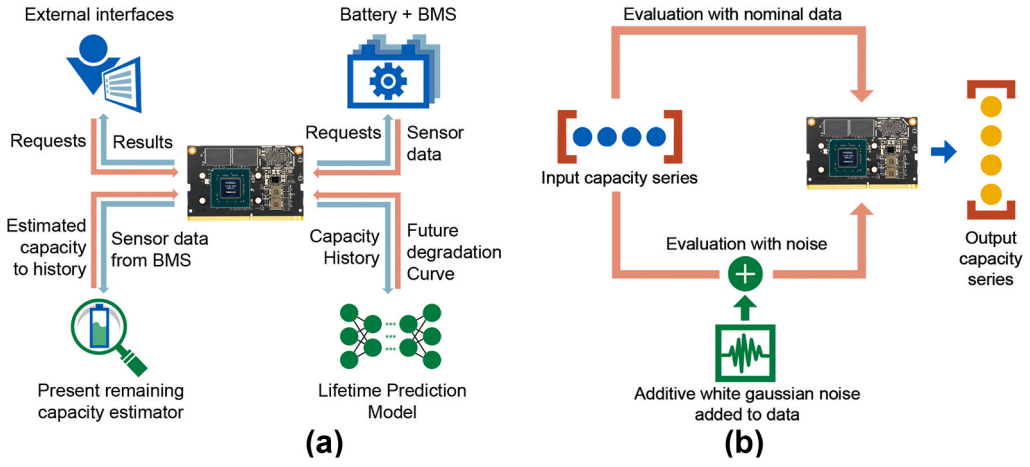


Fig. 3. **(a)** Workflow of the processor-in-the-loop test. **(b)** Two types of evaluation done with the model.

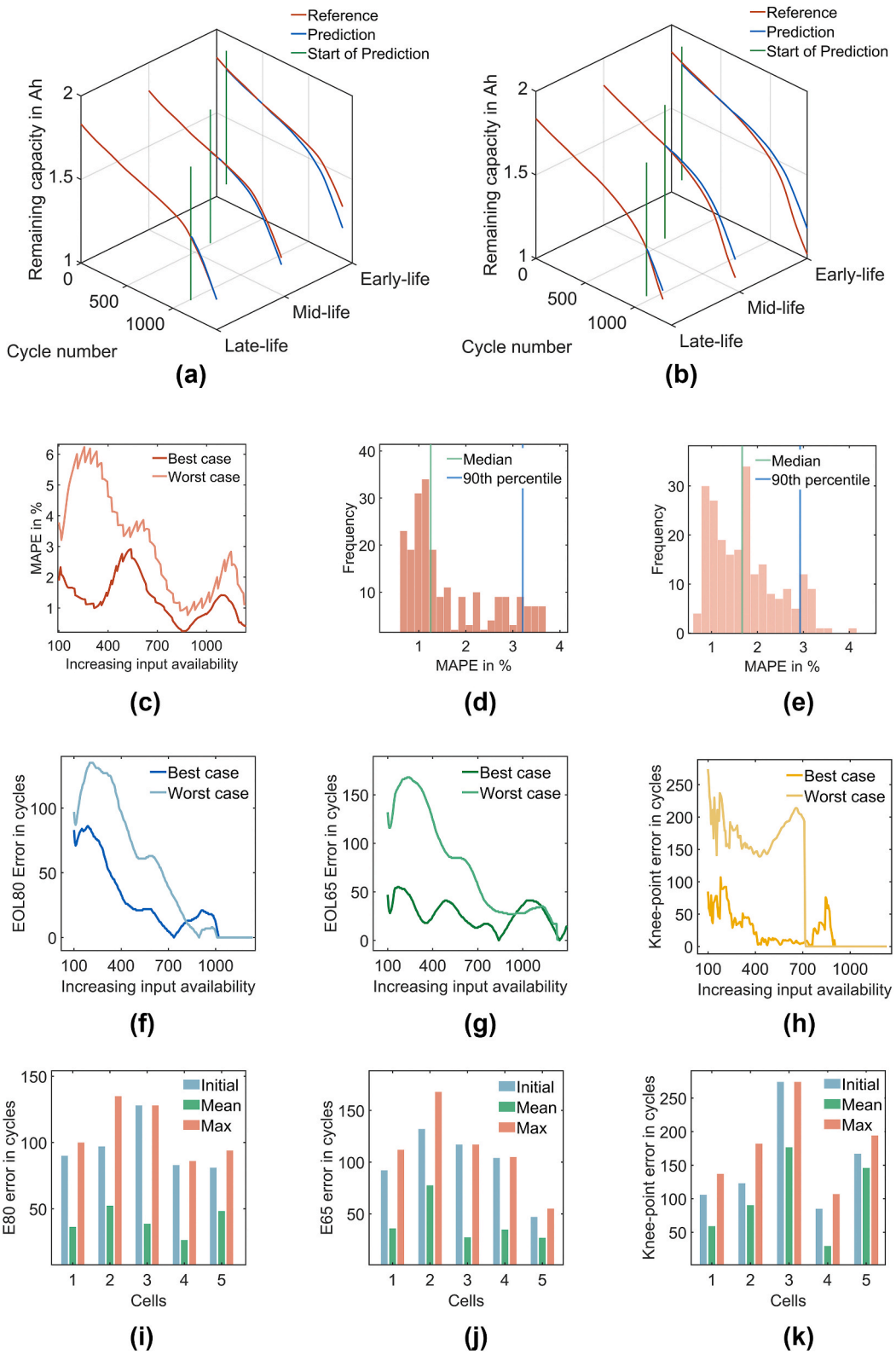


Fig. 4. Validation results for normal input. Firstly, **(a)** and **(b)** the predicted future capacity degradation curves for the best and the worst-case cells, respectively, highlighting prediction quality in three different stages of the cell's life, which corresponds to three different rates of input availability. In the figures, the reference curve is given in red, the prediction curve is in blue, and the green vertical lines show the starting point of the prediction. **(c)** Shows the progression of the MAPE with the availability of input increasing as the cell ages, for both the best and the worst cases of the cell. **(d)** and **(e)** shows the histogram of the curve prediction error for the best and worst-case cell, respectively, with the median and 95th percentile error lines marked. Finally, **(f), (g)** and **(h)** show the best and worst-case progressions for the EOL80, EOL65 and the knee-point errors, respectively, whereas **(i), (j)** and **(k)** show the initial, mean and max error metrics for the three metrics. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

ideal. The initial, mean, and median MAPE error in the worst-case cell is 4.9%, 4.1% and 3.6%, respectively, whereas in the best-case, the initial, mean and median errors are 0.7%, 1.5% and 1.3%, respectively. Interestingly even with noisy input data, the model does not output noisy predictions but outputs the best possible smooth curve of the cell's lifetime similar to the normal input case. A possible rationale is that the labels that the model learns from are smooth in nature, and while predicting, smooth curves are the optimal predictions from the model. A similar set of figures are given for the noisy validation scenario in Fig. 5. As expected, more capacity degradation history available to the model over time leads to a better quality of predictions. Notably, the addition of noise has only a small effect on the mean accuracy of the model over a particular cell's lifetime. However, the major difference is seen in the progression of the error metrics over the cell's lifetime, with fluctuations becoming much more common and individual high error outliers being present in the data, such as the mean curve prediction error progression over the cell's lifetime. This indicates that individual erroneous predictions become more error-prone and the accuracy may fluctuate with the random nature of the noise, but over the cell's lifetime, the noise is handled smoothly.

Finally, the model was compared with a typical LSTM-RNN iterative prediction model (see Methods for details). The results in Fig. 6 show the S2S model as the better performer in predictive ability in both the best- and worst-cases in all metrics, as well as faster than the LSTM model. This is expected since the complexity of the S2S model is almost double of that of the LSTM model, which results in the iterative model taking less training time. However, the iterative nature of the LSTM model requires it to run many times during implementation to provide the full curve output, whereas the S2S model can provide the entire curve in one shot, thereby significantly reducing the computation time, up to almost 15 times in average. The detailed evaluation and comparison can be found in Fig. S7 and Table S2.

4.2. Applications and outlook

The main applications of the model are degradation analysis and failure predictions for improved maintenance, ensuring safe and reliable operation of battery systems. The model can also be used to analyze battery packs with various topologies by identifying inter-pack cell degradation variations and pointing out the strongest and weakest cells within each battery branch. The noise handling capability of the model reaffirms the application on real-world data, where the capacity estimation may have uncertainties that need to be considered. The 'cell-passport' also adds value to first- and second-life applications from the collection of usage information during the whole life of each cell, introducing exciting developments for battery warranties and insurance. Furthermore, digital certificates can be provided along with the cell-passports for various second-life applications.

Whilst this paper has presented a proof of concept, several challenges still exist in degradation prediction using these methods, and are aimed to be further investigated in the future. First, data generation in the laboratory needs to be designed considering not only the intrinsic manufacturing variations in cells, but also the variabilities in working conditions and dynamics in load profiles in real-world applications. Second, data-driven models should be able to be transferred to new cells and chemistries with limited testing time and a limited number of cells, which is required by industry due to the fast development of new battery materials. Third, synthetic data generation methods based on the limited testing data available might help to reduce the testing time and cost further while increasing the model performance and providing useful ground truth for comparison.

5. Conclusions

In this paper, a one-shot battery degradation trajectory prediction model was proposed for batteries under real-world operations, coupled with a cloud-based prognostics framework. The model is based on sequence-to-sequence learning and is trained and validated on a 48 NMC/graphite cell dataset. Model training requires only measured operational data without the need for additional parameterization or feature engineering. The model provides a 'one-shot' prediction of the entire future degradation trajectory, which decreases the computational burden almost 15 times compared with current iterative prediction approaches. An early prediction capability from as few as 100 cycles is possible not only for degradation prediction but also for the endpoints of both first- and second-life and the degradation knee-point. The performance of the model and its robust noise handling capability is validated by processor-in-the-loop tests with 1.8% mean error in the best-case and 7.8% maximum error in the worst-case. This work primarily serves as a proof of concept for the use of modern deep learning-based architectures in the domain of battery prognostics. In general, the health prognostics framework in this work can not only be applied in batteries with different materials but also further in other energy storage systems, e.g. fuel cells and super-capacitors.

Methods

Data availability

The raw experiment dataset and processed dataset of 48 cells used in this study is publicly available at https://git.rwth-aachen.de/isea/battery-degradation-trajectory-prediction. The raw dataset consists of the data from initial characterization tests (multi-pulse test, capacity test and qOCV test), cycling ageing tests (high-resolution data of current, voltage, capacity, energy and temperature) and regular characterization tests (multi-pulse test, capacity test and qOCV test). The processed dataset consists of analyzed data of 48 cells during ageing tests and the change of cell capacity, resistances at different frequency domains and temperature during ageing are provided.

Code availability

Code for experiment data analysis and pre-processing prior to model training is publicly available at https://git.rwth-aachen.de/isea/battery-degradation-trajectory-prediction. Code for the modelling work is available from the corresponding authors upon request.

Ageing experiment

Table S3gives the cell specifications. One characterization test was undertaken at the BOL for each cell and then repeated periodically between cycling tests until the cells reached their EOL. The experiment was continued well beyond the cell's industry specified EOL of 80% of the nominal capacity, to generate a comprehensive dataset of the progression of the capacity degradation throughout the life of the cell. We highlight two specific EOL criteria, the first being 80% of the nominal capacity, and the second being 65% of the nominal capacity. There were, on average, 17 characterization tests in each cell's life. Out of the 48 available cells, 42 (90%) were kept for model training and five cells (10%) were selected randomly and kept aside for blind testing of the model. One cell had an incomplete capacity degradation curve and was therefore discarded from the training batch. Apart from the capacity checkups under various current rates, a quasi-open-circuit voltage test and hybrid-pulse power characterization test were also carried out in each characterization round (see Fig. S8). In the cycling tests, the cells underwent a 30 min constant current constant voltage (CC-CV) charging-discharging regime, with the CC phase at 2C until the cut-off voltage of 3.9 V, followed by a CV phase until 30 min was reached. Both the voltage-over-time curves in CC phase and current-over-time curves in CV phase highlight the changes during degradation (see Fig. S9 and Fig. S10). There were, on average, 160 charging-discharging cycles in each cycling round.

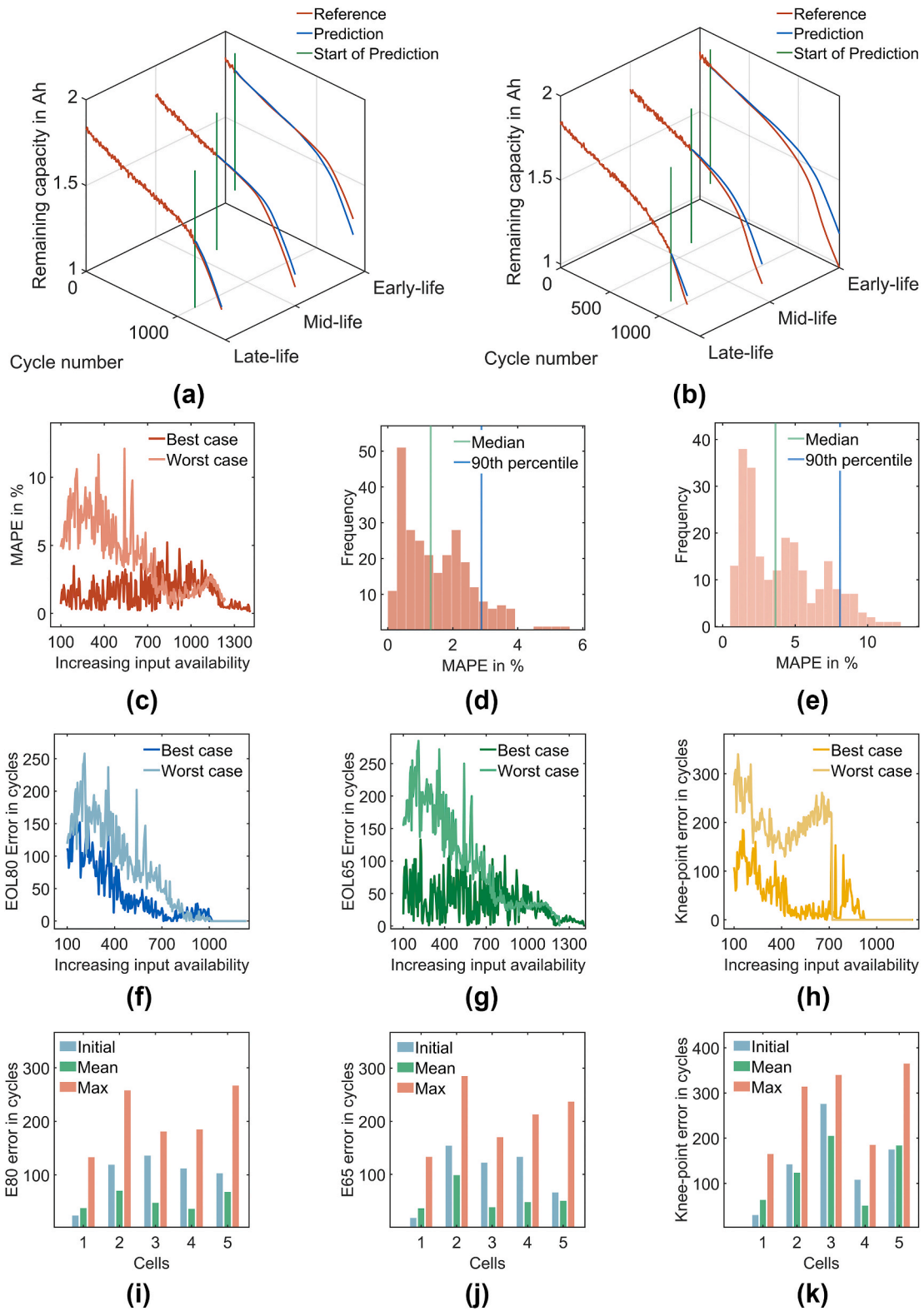


Fig. 5. Validation results for noisy input. Like in the previous figure, **(a)** and **(b)** show the predicted future capacity degradation curves for the best and the worst-case cells, respectively, highlighting prediction quality in three different stages of the cell's life, which corresponds to three different rates of input availability. In the figures, the reference curve is given in red, the prediction curve is in blue, and the green vertical lines show the starting point of the prediction. Like in the previous case, **(c)** shows the progression of the MAPE with the availability of input increasing as the cell ages, for both the best and the worst cases of the cell, **(d)** and **(e)** shows the histogram of the curve prediction error for the best and worst-case cell respectively, with the median and 95th percentile error lines marked, **(f)**, **(g)** and **(h)** show the best and worst-case progressions for the EOL80, EOL65 and the knee-point errors respectively and finally, **(i)**, **(j)** and **(k)** show the initial, mean and max error metrics for the three metrics. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

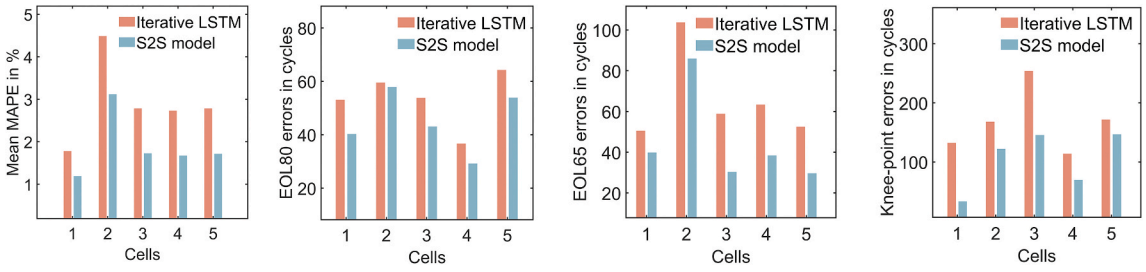


Fig. 6. Comparison results of the LSTM-RNN versus the S2S model.

Data pre-processing

The model proposed in this work provides a 'one-shot' prediction. To make the model training and prediction efficient, we have not used the entire data length, but rather data sampled from the full capacity curves at regular intervals to form both the training input as well as the training output labels for the model. To this end, after the raw dataset was obtained, the capacity checkups were interpolated in MATLAB via the piecewise cubic hermetic interpolating polynomial (PCHIP) function [55]. After interpolation, while creating the training and testing datasets, the input capacity series was sampled in the form of one capacity label every five cycles, whereas the output capacity series was resampled in the form of one capacity label every 50 cycles. The present selection serves as a proof of concept of the proposed method. The input and output data density selection were made to optimize dataset and network sizes and to ensure sufficient data for network learning and output prediction. Other checkup frequencies, such as one capacity label per cycle at the most data-rich case, are also possible and the choice of data density depends on the application and the available hardware.

The supervised learning dataset for each cell was set up to have a minimum amount of input capacity samples, signifying the earliest point in the cell's life that the model can generate predictions. Since this work proposes a model that can obtain predictions very early in the cell's life, the minimum amount of data selected here is data up to 100 cycles of the cell's life. The input size was then increased from that point until up to 90% of the cell's capacity data until its EOL was in the input vector. Since the deep learning model is trained with the entire degradation curve set, no retraining is needed for the same cell type in different lifetimes. Finally, the dataset was divided into a training dataset consisting of 42 cells, and five cells were kept for blind testing the network.

Network architecture and training

As described in Refs. [56,57], the LSTM cell, along with the hidden cell state, has a memory pipeline which allows it to preserve long term temporal correlations. The various gates and valves of every time step of the LSTM network can be seen as a one-layer neural network themselves, which transforms various parts of the input according to their operations. The equations that govern the LSTM-RNN operation, from Ref. [56], are given below, where w_{all} are weight matrices, b_{all} are biases, σ (x) is the sigmoid operator, $\tanh(x)$ is the hyperbolic tangent operator, i_t and $\widetilde{C}_t$ are intermediaries to the new memory $\widetilde{C}_t$ and h_t is the new output state of the cell.

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

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = \tanh(w_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t{*}C_{t-1} + i_t{*}\widetilde{C}_t \tag{4}$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t{*}\tanh(C_t) \tag{6}$$

The structure of the LSTM cell and the various connections of the cell with its sub-networks are given in Fig. S11. The LSTM cell takes the previous memory cell state C_{t-1} , the previous output state h_{t-1}, and the input vector for the current time step x_t as input. The forget gate decides which part of the old memory to forget and pushes the forgetting vector f_t to the cell pipeline. The new memory valve contains two networks, one to decide the influence of the new memory on the current cell state, called the input gate, which pushes the influencing vector i_t, and the other to generate the new memory for the current time step as $\widetilde{C}_t$, called the memory generator. The operations of the forget valve and the new memory valve together update the memory cell state, which is then forwarded to the next time step as C_t. Finally, the output gate generates the current output state vector o_t, which is then updated in accordance with the new memory state C_t to push the new output state h_t, which becomes the output of the current time step.

The network consists of four LSTM layers in both the decoder and encoder, with the encoder layers having the hidden dimension size as the maximum length of the training input series and the decoder layers having the hidden dimension size as the maximum length of the desired output series, as shown in Fig. S12. The network was trained with the learning managed by the Adam optimizer [58] utilizing mean absolute error as the cost function or loss, modified to be able to mask the lagging zero paddings of the target series. The training was divided into two phases, with different learning rates. The first phase has a larger learning rate of 0.0004 to converge to the approximate parameter region within 1000 epochs and the second phase has a lower learning rate of 0.00025 to explore the possibility of minimizing the training validation loss within the region when trained for another period of 1000 epochs while checking for overfitting (see Fig. S13). Additionally, a random 15% of the training data was used as validation data for each epoch. After the first phase, the model had seemingly settled with a training validation loss of around 0.04. The second phase of training proved to be effective in lowering the training validation loss further. The best validation loss achieved by the S2S network while training was 0.0312, after which further improvements were not seen. The training data contained 10497 samples, with a mini-batch processing size of 300 samples. This model was then chosen to perform in the different validation scenarios.

Processor-in-the-loop test

Validation was done by serving the trained model into an embedded system, Jetson Nano [59]. This small embedded hardware validation of the model is significant mainly to demonstrate the standalone 'prediction on edge' capability where the proposed model, after being fitted into a local vehicle can operate perfectly, generate and store results even without constant connectivity or remote access to external servers. The Nano functions as a mini-sized onboard computer with an NVidia CUDA-enabled GPU, which enables it to perform deep learning tasks. The board runs a Linux-based operating system that supports a full-sized TensorFlow environment; therefore, almost all functionalities of architectures designed in other such systems can be ported. In the

'processor-in-the-loop' test [60], data arrays are fed to the Nano's storage, from which it can run the model to validate the various validation scenarios taking the sensor data sequentially following a real operation usage style. The validation in the Nano is done mainly to demonstrate the model accuracy, computing capability and the viability of using the device in future BMSs.

Evaluation criteria

The main output of the model is the prediction of the future degradation curve, which is compared with the true degradation curves of the test cells. The mean absolute percentage error (MAPE) of each curve is calculated, and the progression of the error is shown as more data becomes available to the network to process, with the progression of the cell's lifetime. The mean of the MAPE of all the curves from the entire lifetime of the cells is also taken as a representative metric. The formulae for the calculation of MAPE is given below, where y is the true output data point and $\tilde{y}$ is the predicted output data point of one predicted curve, respectively.

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

$$MAPE_{curve} = \frac{\sum_{j=1}^{n} APE_j}{n} \tag{8}$$

$$MAPE_{cell} = \frac{\sum_{i=1}^{n} MAPE_{curve_i}}{n} \tag{9}$$

Apart from the curve MAPE, absolute errors of three points of interest, namely EOL80, EOL65, and knee-point, are also looked at and compared to get the accuracy of the model's prediction. Since the proposed model aims to have a comprehensive lifetime prediction ability, it is crucial to validate the results of these additional three metrics along with the prediction accuracy.

Comparison with current iterative prediction approaches

For comparison, a standard iterative LSTM-RNN model was trained on the same dataset, and the results were compared with our proposed model to serve as a benchmark. The benchmark LSTM consisted of four bidirectional LSTM layers with 90 hidden nodes in each layer and was trained on the dataset for 300 epochs before testing. The training length was determined through experimentation and was fixed when no further improvement to the model was seen upon increment. The iterative LSTM model was also trained on 100% of the training curves for the sake of fair benchmarking since the S2S model was also trained with 100% of the training curve data. The loss used for the training was an unmodified version of the 'mean absolute error' metric available from Keras, and the optimizer used is Adam, similar to the S2S model.

The LSTM network follows an iterative rolling window approach, which is common in the literature [44,47]. The input window is set to 100 cycles, the same as with the S2S model's earliest input window. The key difference between the networks is that while the S2S model has the ability to accept dynamic window sizes and therefore does not need to sacrifice previous historical data while predicting with newer inputs, the LSTM network must roll the input window forward instead of increasing it, to keep a constant window size, which leads to loss of historical data. The comparison was made with three key metrics, the earliest, mean and maximum MAPE of the best- and worst-case cells, and the maximum computation time of a sample on the Jetson Nano (see Table S2).

CRediT authorship contribution statement

Weihan Li: designed the research and performed the modelling, wrote the manuscript with contributions from. **Neil Sengupta:** designed the research and performed the modelling, wrote the manuscript with contributions from. **Philipp Dechent:** performed experiment data management. **David Howey:** all authors reviewed and commented on the manuscript. **Anuradha Annaswamy:** all authors reviewed and commented on the manuscript. **Dirk Uwe Sauer:** supervised the work.

Declaration of competing interest

W.L., N.S., D.U.S. have filed a patent related to this work: DE Application No. 102020210148.5, dated 11 August 2020.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation program under the grant "Electric Vehicle Enhanced Range, Lifetime And Safety Through INGenious battery management" (EVERLASTING-713771). Part of the work was done within the research project "E2Fuels" (03EIV011F) funded by the German Federal Ministry for Economic Affairs and Energy (BMWi). We would like to thank T. Baumhöfer for the design of the ageing experiments and data generation. We also thank S. Greenbank for review and discussions.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jpowsour.2021.230024.

References

[1] R. Sprague, An analysis of current battery technology and electric vehicles, J Undergrad Research UIC 8 (2017), https://doi.org/10.5210/jur.v8i1.7544.
[2] S.J. Harris, D.J. Harris, C. Li, Failure statistics for commercial lithium ion batteries: a study of 24 pouch cells, J. Power Sources 342 (2017) 589–597, https://doi.org/10.1016/j.jpowsour.2016.12.083.
[3] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, K. Dietmayer, Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods, J. Power Sources 239 (2013) 680–688, https://doi.org/10.1016/j.jpowsour.2012.11.146.
[4] L. Liao, F. Kottig, Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction, IEEE Trans. Reliab. 63 (2014) 191–207, https://doi.org/10.1109/TR.2014.2299152.
[5] X. Hu, Le Xu, X. Lin, M. Pecht, Battery lifetime prognostics, Joule 4 (2020) 310–346, https://doi.org/10.1016/j.joule.2019.11.018.
[6] H. He, R. Xiong, H. Guo, Online estimation of model parameters and state-of-charge of LiFePO4 batteries in electric vehicles, Appl. Energy 89 (2012) 413–420, https://doi.org/10.1016/j.apenergy.2011.08.005.
[7] G.L. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs, J. Power Sources 161 (2006) 1369–1384, https://doi.org/10.1016/j.jpowsour.2006.06.004.
[8] J. Remmlinger, M. Buchholz, T. Soczka-Guth, K. Dietmayer, On-board state-of-health monitoring of lithium-ion batteries using linear parameter-varying models, J. Power Sources 239 (2013) 689–695, https://doi.org/10.1016/j.jpowsour.2012.11.102.
[9] C. Fleischer, W. Waag, H.-M. Heyn, D.U. Sauer, On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models part 2. Parameter and state estimation, J. Power Sources 262 (2014) 457–482, https://doi.org/10.1016/j.jpowsour.2014.03.046.
[10] W. Li, M. Rentemeister, J. Badeda, 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 30 (2020) 101557, https://doi.org/10.1016/j.est.2020.101557.
[11] L. Zhang, W. Fan, Z. Wang, W. Li, D.U. Sauer, Battery heating for lithium-ion batteries based on multi-stage alternative currents, Journal of Energy Storage 32 (2020) 101885, https://doi.org/10.1016/j.est.2020.101885.
[12] H. Peng, J. Li, K. Deng, A. Thul, W. Li, L. Lowenstein. An efficient optimum energy management strategy using parallel dynamic programming for a hybrid train powered by fuel-cells and batteries, IEEE, NJ, 2019, pp. 1–7, https://doi.org/10.1109/VPPC46532.2019.8952323.
[13] J. Wu, Z. Wei, W. Li, Y. Wang, Y. Li, Battery thermal- and healthconstrained energy management for hybrid electric bus based on soft actor-critic DRL algorithm, IEEE Trans. Ind. Inf. 1 (2020), https://doi.org/10.1109/TII.2020.3014599.
[14] W. Li, H. Cui, T. Nemeth, J. Jansen, C. Ünlübayir, Z. Wei, et al., Deep reinforcement learning-based energy management of hybrid battery systems in electric vehicles, J. Energy Storage 36 (1) (2021), 102355, https://doi.org/10.1016/j.est.2021.102355.

[15] N. Lotfi, J. Li, R.G. Landers, J. Park, Li-ion battery state of health estimation based on an improved single particle model, in: American Control Conference (ACC), IEEE, Seattle, WA, USA, 2017, pp. 86–91, 24-May-17 - 26-May-17.
[16] 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, Appl. Energy 269 (2020) 115104, https://doi.org/10.1016/j.apenergy.2020.115104.
[17] F. Ringbeck, M. Garbade, D.U. Sauer, Uncertainty-aware state estimation for electrochemical model-based fast charging control of lithium-ion batteries, J. Power Sources 470 (2020) 228221, https://doi.org/10.1016/j.jpowsour.2020.228221.
[18] W. Li, Y. Fan, F. Ringbeck, D. Jöst, X. Han, M. Ouyang, D.U. Sauer, Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented Kalman filter, J. Power Sources 476 (2020) 228534, https://doi.org/10.1016/j.jpowsour.2020.228534.
[19] J.M. Reniers, G. Mulder, D.A. Howey, Unlocking Extra Value from Grid Batteries Using Advanced Models, 2020.
[20] J.C. Forman, S.J. Moura, J.L. Stein, H.K. Fathy, Genetic identification and Fisher identifiability analysis of the Doyle–Fuller–Newman model from experimental cycling of a LiFePO4 cell, J. Power Sources 210 (2012) 263–275, https://doi.org/10.1016/j.jpowsour.2012.03.009.
[21] W. Li, D. Limoge, J. Zhang, D.U. Sauer, Estimation of potentials in lithium-ion batteries using machine learning models, IEEE Trans. Control Syst. Technol. (2021), https://doi.org/10.1109/TCST.2021.3071643.
[22] J. Schmalstieg, S. Käbitz, M. Ecker, D.U. Sauer, A holistic aging model for Li (NiMnCo)O2 based 18650 lithium-ion batteries, J. Power Sources 257 (2014) 325–334, https://doi.org/10.1016/j.jpowsour.2014.02.012.
[23] I. Bloom, B.W. Cole, J.J. Sohn, S.A. Jones, E.G. Polzin, V.S. Battaglia, G. L. Henriksen, C. Motloch, R. Richardson, T. Unkelhaeuser, D. Ingersoll, H.L. Case, An accelerated calendar and cycle life study of Li-ion cells, J. Power Sources 101 (2001) 238–247, https://doi.org/10.1016/S0378-7753(01)00783-2.
[24] D.U. Sauer, H. Wenzl, Comparison of different approaches for lifetime prediction of electrochemical systems—using lead-acid batteries as example, J. Power Sources 176 (2008) 534–546, https://doi.org/10.1016/j.jpowsour.2007.08.057.
[25] S. Wang, D. Guo, X. Han, L. Lu, K. Sun, W. Li, et al., Impact of battery degradation models on energy management of a grid-connected DC microgrid, Energy 207 (2020), 118228, https://doi.org/10.1016/j.energy.2020.118228.
[26] A.Z. Hinchi, M. Tkiouat (Eds.), A Deep Long-Short-Term-Memory Neural Network for Lithium-Ion Battery Prognostics, 2018.
[27] D. Liu, W. Xie, H. Liao, Y. Peng, An integrated probabilistic approach to lithium-ion battery remaining useful life estimation, IEEE Trans. Instrum. Meas. 64 (2015) 660–670, https://doi.org/10.1109/TIM.2014.2348613.
[28] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R. D. Braatz, Data-driven prediction of battery cycle life before capacity degradation, Nat Energy 4 (2019) 383–391, https://doi.org/10.1038/s41560-019-0356-8.
[29] K. Goebel, B. Saha, A. Saxena, J. Celaya, J. Christophersen, Prognostics in battery health management, IEEE Instrum. Meas. Mag. 11 (2008) 33–40, https://doi.org/10.1109/MIM.2008.4579269.
[30] F. Li, J. Xu, A new prognostics method for state of health estimation of lithium-ion batteries based on a mixture of Gaussian process models and particle filter, Microelectron. Reliab. 55 (2015) 1035–1045, https://doi.org/10.1016/j.microrel.2015.02.025.
[31] R.R. Richardson, M.A. Osborne, D.A. Howey, Gaussian process regression for forecasting battery state of health, J. Power Sources 357 (2017) 209–219, https://doi.org/10.1016/j.jpowsour.2017.05.004.
[32] D. Liu, J. Pang, J. Zhou, Y. Peng, M. Pecht, Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression, Microelectron. Reliab. 53 (2013) 832–839, https://doi.org/10.1016/j.microrel.2013.03.010.
[33] M. Lucu, E. Martinez-Laserna, I. Gandiaga, K. Liu, H. Camblong, W.D. Widanage, J. Marco, Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data – Part A: storage operation, Journal of Energy Storage 30 (2020) 101409, https://doi.org/10.1016/j.est.2020.101409.
[34] X. Hu, J. Jiang, D. Cao, B. Egardt, Battery health prognosis for electric vehicles using sample entropy and sparse bayesian predictive modeling, IEEE Trans. Ind. Electron. (2015) 1, https://doi.org/10.1109/TIE.2015.2461523.
[35] M.A. Patil, P. Tagade, K.S. Hariharan, S.M. Kolake, T. Song, T. Yeo, S. Doo, A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation, Appl. Energy 159 (2015) 285–297, https://doi.org/10.1016/j.apenergy.2015.08.119.
[36] D. Wang, Q. Miao, M. Pecht, Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model, J. Power Sources 239 (2013) 253–264, https://doi.org/10.1016/j.jpowsour.2013.03.129.
[37] Y. Zhang, R. Xiong, H. He, M. Pecht, Validation and verification of a hybrid method for remaining useful life prediction of lithium-ion batteries, J. Clean. Prod. 212 (2019) 240–249, https://doi.org/10.1016/j.jclepro.2018.12.041.
[38] Y. Chang, H. Fang, Y. Zhang, A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery, Appl. Energy 206 (2017) 1564–1578, https://doi.org/10.1016/j.apenergy.2017.09.106.
[39] D. Liu, J. Zhou, D. Pan, Y. Peng, X. Peng, Lithium-ion battery remaining useful life estimation with an optimized Relevance Vector Machine algorithm with incremental learning, Measurement 63 (2015) 143–151, https://doi.org/10.1016/j.measurement.2014.11.031.
[40] X. Zheng, H. Fang, An integrated unscented kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction, Reliab. Eng. Syst. Saf. 144 (2015) 74–82, https://doi.org/10.1016/j.ress.2015.07.013.
[41] 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, https://doi.org/10.1109/ACCESS.2018.2858856.
[42] Y. Zhou, Y. Huang, J. Pang, K. Wang, Remaining useful life prediction for supercapacitor based on long short-term memory neural network, J. Power Sources 440 (2019) 227149, https://doi.org/10.1016/j.jpowsour.2019.227149.
[43] X. Li, L. Zhang, Z. Wang, P. Dong, Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks, Journal of Energy Storage 21 (2019) 510–518, https://doi.org/10.1016/j.est.2018.12.011.
[44] L. Mao, J. Xu, J. Chen, J. Zhao, Y. Wu, F. Yao, A LSTM-STW and GS-LM fusion method for lithium-ion battery RUL prediction based on EEMD, Energies 13 (2020) 2380, https://doi.org/10.3390/en13092380.
[45] C. Wang, N. Lu, S. Wang, Y. Cheng, B. Jiang, Dynamic long short-term memory neural-network- based indirect remaining-useful-life prognosis for satellite lithium-ion battery, Appl. Sci. 8 (2018) 2078, https://doi.org/10.3390/app8112078.
[46] Y. Zhang, R. Xiong, H. He, Z. Liu, A LSTM-RNN method for the lithuim-ion battery remaining useful life prediction, in: 2017 Prognostics and System Health Management Conference (PHM-Harbin), Harbin, China, IEEE, 7/9/2017 - 7/12/2017, pp. 1–4.
[47] Y. Yu, C. Hu, X. Si, J. Zheng, J. Zhang, Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset, Neurocomputing 402 (2020) 134–147, https://doi.org/10.1016/j.neucom.2020.03.041.
[48] Y. Zhang, R. Xiong, H. He, M.G. Pecht, Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries, IEEE Trans. Veh. Technol. 67 (2018) 5695–5705, https://doi.org/10.1109/TVT.2018.2805189.
[49] L. Li, Y. Peng, Y. Song, D. Liu, Lithium-ion battery remaining useful life prognostics using data-driven deep learning algorithm, in: 2018 Prognostics and System Health Management Conference (PHM-Chongqing), IEEE, Chongqing, 10/26/2018 - 10/28/2018, pp. 1094–1100.
[50] T. Baumhöfer, M. Brühl, S. Rothgang, D.U. Sauer, Production caused variation in capacity aging trend and correlation to initial cell performance, J. Power Sources 247 (2014) 332–338, https://doi.org/10.1016/j.jpowsour.2013.08.108.
[51] D. Andre, C. Appel, T. Soczka-Guth, D.U. Sauer, Advanced mathematical methods of SOC and SOH estimation for lithium-ion batteries, J. Power Sources 224 (2013) 20–27, https://doi.org/10.1016/j.jpowsour.2012.10.001.
[52] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, D.U. Sauer, Online capacity estimation of lithium-ion batteries with deep long short-term memory networks, J. Power Sources 482 (2021) 228863, https://doi.org/10.1016/j.jpowsour.2020.228863.
[53] G.L. Plett, Recursive approximate weighted total least squares estimation of battery cell total capacity, J. Power Sources 196 (2011) 2319–2331, https://doi.org/10.1016/j.jpowsour.2010.09.048.
[54] I. Sutskever, O. Vinyals, Q. Le V, Sequence to Sequence Learning with Neural Networks, 10-Sep-14.
[55] Mathworks, Piecewise Cubic Hermetic Interpolating Polynomial (PCHIP) Function. https://uk.mathworks.com/help/matlab/ref/pchip.html#bvjbz1m-2.
[56] Christopher Olah, Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/?source=post_page———37e2f46f1714———————————————, 2015.
[57] Yan Shi, Understanding LSTM and its Diagrams. https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714, 2016.
[58] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 22-Dec-14.
[59] Nvidia Corporation, Jetson Nano Developer Kit. https://developer.nvidia.com/embedded/jetson-nano-developer-kit, 2019.
[60] W. Li, H. Cui, T. Nemeth, J. Jansen, C. Ünlübayir, Z. Wei, et al., Cloud-based health-conscious energy management of hybrid battery systems in electric vehicles with deep reinforcement learning, Appl. Energy 291 (2021), 116977, https://doi.org/10.1016/j.apenergy.2021.116977.