arXiv:2102.00837v1 [cs.LG] 1 Feb 2021

MACHINE LEARNING PIPELINE FOR BATTERY STATE OF HEALTH ESTIMATION

Darius Roman
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
dvr1@hw.ac.uk

Saurabh Saxena
Center for Advanced Life Cycle Engineering
University of Maryland
College Park, MD 20742, USA
saxenas@umd.edu

Valentin Robu
CWI, National Research Institute for
Mathematics and Computer Science
Amsterdam, The Netherlands
v.robul@cwi.nl

David Flynn
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
dflynn@hw.ac.uk

Michael Pecht
Center for Advanced Life Cycle Engineering
University of Maryland
College Park, MD 20742, USA
pecht@umd.edu

Note:
Pre-print version of the article in Nature Machine Intelligence

Tuesday 2nd February, 2021

ABSTRACT

Lithium-ion batteries are ubiquitous in modern day applications ranging from portable electronics to electric vehicles. Irrespective of the application, reliable real-time estimation of battery state of health (SOH) by on-board computers is crucial to the safe operation of the battery, ultimately safeguarding asset integrity. In this paper, we design and evaluate a machine learning pipeline for estimation of battery capacity fade — a metric of battery health — on 179 cells cycled under various conditions. The pipeline estimates battery SOH with an associated confidence interval by using two parametric and two non-parametric algorithms. Using segments of charge voltage and current curves, the pipeline engineers 30 features, performs automatic feature selection and calibrates the algorithms. When deployed on cells operated under the fast-charging protocol, the best model achieves a root mean squared percent error of 0.45%. This work provides insights into the design of scalable data-driven models for battery SOH estimation, emphasising the value of confidence bounds around the prediction. The pipeline methodology combines experimental data with machine learning modelling and can be generalized to other critical components that require real-time estimation of SOH.

1 Introduction

Rechargeable Li-ion batteries play a crucial role in many modern-day applications ranging from portable electronics and medical devices, to renewable energy integration in power grids and electric vehicles. The steep decrease in the price of lithium-ion based battery storage by 73% from 2010 to 2016, to an all-time low of \$273/kWh in 2017 [1]

opened up a significant energy storage market evaluated at $65 billion in 2017 [2]. Irrespective of the application, Li-ion batteries degrade with time. With ageing, cells exhibit a loss of capacity and an increase in impedance. The rate of degradation is influenced by the dynamic operating conditions, including varying charge/discharge rates, different voltage operation limits and temperature fluctuations. The ability to estimate degradation in real-time irrespective of the various failure mechanisms and their degradation paths is crucial for safe and effective battery management systems [3]. Battery state of health can be used to predict battery's expected lifetime, however, the feasibility of online state of health estimation via direct measurement of chemical reaction parameters inside batteries remains limited [4].

State of health (SOH) is a parameter that quantifies the general condition of a battery and its ability to deliver the specified performance, measured as capacity or impedance, when compared to its unused state. This work focuses on the battery capacity as the health indicator due to its correlation to the energy storage capability of batteries and its direct impact on the remaining run time and life of the batteries. Capacity fade estimation has received considerable research interest from industry and academia [4], [5], [6], [7] and a number of methods have been proposed. The current approaches to capacity fade estimation involves parameter estimation using either of the following modelling types, equivalent circuit models (ECMs) [8], [9], [10], electrochemical models [11], [12], [13], or data-driven models [14], [15], [16], [17], [18], [19]. Electrochemical models approximate the chemical processes that take place inside a battery cell during operation. This type of modeling requires detailed cell specifications, such as electrode materials and electrolyte chemistry. The method typically deploys complex partial differential equations, leading to significant requirements of both memory and computational power. ECMs, on the other hand, employ circuit components with empirical nonlinear parameters [9]. Compared to electrochemical models, ECMs use fewer inputs, considerably reducing the number of parameters required to be learnt over time, however, they have limited accuracy and robustness due to assumptions in battery behavior [8]. Furthermore, in order to determine ECM model parameters, such as the ohmic resistance and the parallel resistor-capacitor impedance, at different state of charge values, pulse discharging [20] and electrochemical impedance spectroscopy is typically necessary [10], [21], [22], however such measurements are not a viable solution for online applications.

Conversely, the data-driven approach displays a series of advantages such as a chemistry-agnostic modelling capability and an ability to analyse a wide range of degradation mechanisms and operating conditions, including rare loading events often overlooked by simplified models or physics-based simulations. To date, numerous studies have employed machine learning tools for the analysis of battery SOH estimation. Several studies [23], [24], [25], [26], [27] showed that incremental capacity (IC) and differential voltage (DV) curves, a method developed for use in cell aging mechanism analysis [23], can also be used for offline and online capacity fade estimation. However, the approach has several drawbacks linked to obtaining the IC and DV curves that substantially reduce its practicality. The differentiation of the capacity-voltage curve to obtain the IC curve amplifies noise and propagates it into the algorithm. Additionally, both curves must cover a sufficient voltage range for the method to work and, for obtaining a high curve fidelity, it is restricted to low charge current rates(1/5 to 1/25 C-rate) [28], [29], [30]. Unless a low current value is used during charging protocol and the key part of the capacity-voltage curve is recorded, such that specific peak points in the IC curve are captured, the method is impractical for online deployment.

An alternative is to train an algorithm on the raw voltage-time data curve, eliminating the need for differentiation [31], [32]. Notably, Richardson et al. [32] operated on sections of the voltage-time data itself by first smoothing the curve using a Savitsky-Golay filter and then used equispaced voltage values as the input to a Gaussian process regression (GPR) algorithm. However, GPR is considerably slow to train due to its computational cost of learning governed by the kernel function [33], making it unsuitable for online deployment. The high computational complexity, also severely limits its scalability to incorporate bigger datasets. Additionally, the algorithm is sensitive to the section of the voltage-time curve used as input to the GPR. Other Bayesian-based methods, such as the relevance vector machine (RVM) [34], have also been used to estimate battery capacity fade. Unfortunately, RVM also suffers from slow training, particularly when compared to frequentist-based algorithms [35]. Shen et al. [33] presented options for accelerating GPR, however, they compromise accuracy. In contrast to [32] where the constant current part of the charging profile was used, Wang et al. [36] used the constant voltage section to estimate capacity fade using support vector regression (SVR). Although SVR is faster than GPR, it lacks the ability to estimate uncertainty. This inability to estimate uncertainty stemmed from various sources is a major limiting factor when discussing complex dynamic systems, such as Li-ion batteries. SOH assessment without corresponding measures of uncertainty associated with the estimation does not provide sufficient information to form a decision or corrective action plan [37].

Previous work [8]-[36] includes limited assessments of SOH uncertainty or none at all. The proposed machine learning pipeline is capable of real-time estimation of battery SOH and associated algorithm uncertainty referred to as battery health and uncertainty management pipeline (BHUMP). BHUMP operates by passing incoming data streams through a hierarchical sequence of processing steps by first engineering features based on segments of raw charge curves. It then performs offline automatic feature selection, augments the dataset with adversarial examples, and estimates battery health and associated uncertainty with the aid of four machine learning algorithms. Uncertainty is quantified based on

calibration error and an adapted accuracy measure, the α-β accuracy zone. There are numerous battery designs [38] and chemistries available [39], therefore the pipeline is deployed on a total of 179 cells, three designs (prismatic, pouch, and cylindrical), two chemistries ($LiFePO_4$, and $LiCoO_2$), three charge protocols (constant current, constant current - constant voltage, and 2-step fast-charge), and various discharge rates.

This paper refines and extensively tests new and improved machine learning algorithms for the capacity fade estimation problem, but also defines metrics for estimating and accurately quantifying uncertainty in ML models used in battery research. BHUMP provides battery researchers with a scalable SOH estimation solution that is adaptable to any cell chemistry and operating condition. BHUMP is more accurate than conventional methods as the battery is ageing, uses a set of engineered features capable of capturing battery intrinsic degradation, and is capable of estimating cell SOH in under 15 minutes at any point in its life-cycle. An accurate SOH method combined with a quantifiable metric for uncertainty propagation that feeds into SOC and run time calculations improves battery performance and ultimately extends cell lifetime.

2 Machine learning pipeline approach

2.1 Pipeline overview

From a machine learning perspective, determining battery capacity fade can be considered a multivariate supervised regression problem. We use a pipeline-based approach, where features are engineered from charge/discharge curves, on which a Bayesian or frequentist model is trained. Additionally, uncertainty is quantified by predicting a distribution mean and an associated standard deviation. Our learning method is divided into two stages, namely, Stage 1: Offline pipeline creation and training and Stage 2: Online SOH estimation. The offline stage ensures feature engineering, training data augmentation, automatic feature selection, algorithm training, and uncertainty calibration. The online stage diagnoses the cell using the trained pipeline under the assumptions that it is given a battery cell of unknown capacity. Supplementary Figure 1 provides a summary of the two stages via a flowchart of the method.

Feature engineering is split into automatic feature generation or extraction through techniques such as neural network auto-encoders [40], [41], and manual feature construction based on domain knowledge [42], [43]. We adopt a domain knowledge-based approach, where we show the algorithm feature choice based on importance to target variable. We also, provide a hypothesis for the underlying physical degradation quantified by the selected segments of the charge curves in Supplementary Note 1. Supplementary Table 1 summarize the attributes recorded during life cycle testing. The pipeline focuses on segments of the charge curves to capture degradation in the electrodes during cycling (Figure 1 illustrates typical extracted segments). The extracted charge-curve segments are further used in the feature engineering process (see Methods for details).

The pipeline creates a total of 30 features, and selects the most relevant features using a random forest based recursive feature elimination with cross-validation (RF-RFE-CV) similar to the one introduced in [44]. Recursive feature elimination generally outperforms other conventional methods [45], [46], hence the adoption here (refer to Methods section for further details). Before training the algorithms, we perform data augmentation by introducing adversarial examples as proposed by Goodfellow et al. [47] in combination with the weight decay algorithm (see Methods). The use of adversarial examples in our datasets was motivated by the need to ameliorate the differences in battery design/chemistry. In addition, training on adversarial data makes the algorithm robust to outliers, prevents overfitting and reduces distribution variance around the estimated mean. Synthetic data generation generated from electrochemical models like the pseudo-two-dimensional model proposed by Doyle et. al. [48] can also be regarded as a data augmentation policy. Such an approach harnesses the potential of both electrochemical and data based models and we believe future work must incorporate synthetic data as well.

The augmented dataset then serves as the training input to four algorithms: random forest (RF) and deep neural network ensemble (dNNe), Bayesian ridge regression (BRR) and Gaussian process regression (GPR). Unlike Bayesian based algorithms, BRR and GPR, frequentist algorithms are unable to quantify uncertainty naturally due to their formulation. To overcome such limitations, we consider two modified ensemble based algorithms: RF with Infinitesimal Jackknife (IJ) based confidence intervals [49] and the ensemble of neural networks as described in [50]. For training of the algorithms a random search approach is used for hyper-parameter tuning [51], with the exception of the deep ensemble where the Adam optimiser is used. We have found that drawing random samples from a uniform distribution works best for BRR and GPR parameters, whereas for RF and dNNe parameters random initialisation gives satisfactory results. In addition, a batch cross-validation method is used during the hyper-parameter tuning, where each batch is represented by one cell. This prevents the over-fitting of the models and mimics online deployment. Machine learning models in engineering require a stringent performance evaluation both from an error and uncertainty quantification perspective. The models are initially re-calibrated followed by an evaluation based on mean absolute percent error, root mean squared error and uncertainty estimation metrics (refer to Methods section for further details).

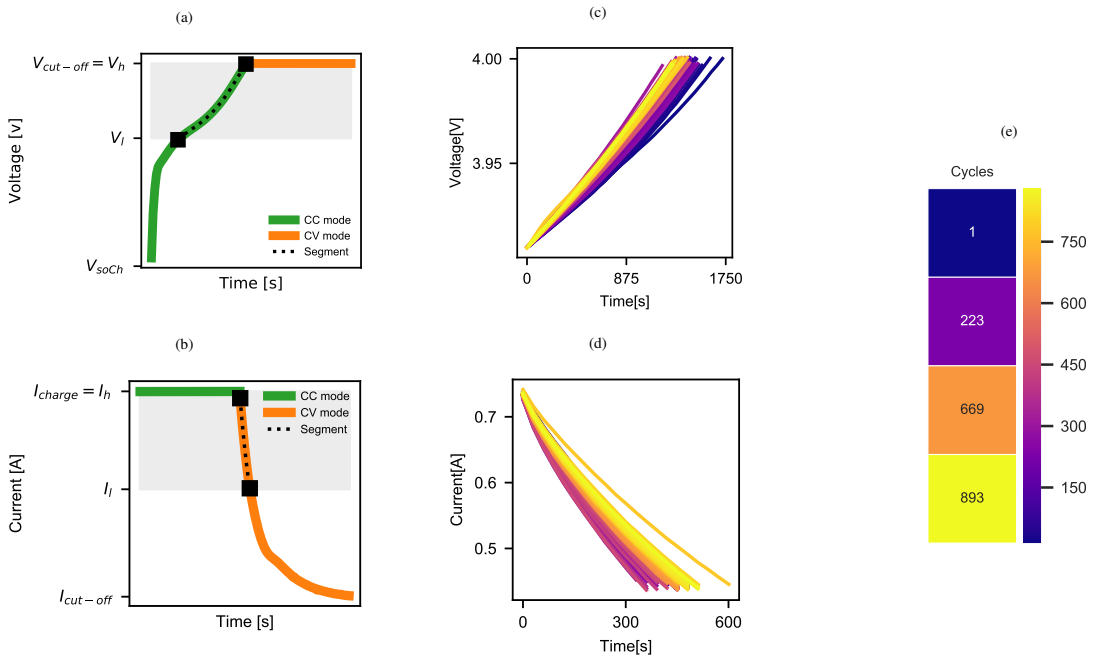

Figure 1: **The CC-CV charge protocol and extracted ageing segment of the curves for a Li-ion pouch cell. a** Voltage during charge protocol. **b** Current during charge protocol, **c** Extracted ageing voltage curve segments corresponding to marked grey area, **d** Extracted ageing current curve segments corresponding to marked grey area, **e** Heatmap of ageing with cycle number. Note: Refer to Methods section for abbreviations.

2.2 Methods

This study developed a pipeline approach for battery SOH estimation, called BHUMP and it incorporates a series of hierarchical steps, feature engineering, feature selection and data augmentation prior to model fitting and tuning as follows.

Feature engineering performs mathematical manipulations of extracted parts of the voltage curve during the constant current charge protocol based on a lower voltage threshold, V_l and an upper voltage threshold, V_h (refer to the grey area of Figure 1a) for all datasets except for cells charged with a 2-step fast-charge protocol. A characteristic to the 2-step fast charge protocol is that the cells can be charged from 0% SOC to 80% SOC with high currents ranging from 3.6 C-rate to 6 C-rate. In this work, due to the nature of the charging method in the 2-step fast charge, we only use the constant-current constant-voltage (CC-CV) charge part of the charging protocol as per the black dotted segments in the grey area observed in Supplementary Figures 2a, 2b. The values of V_h and V_l can be selected based on the intended application and the depth of discharge of the cell. In this work we select V_h to be equal to cut-off voltage, $V_{cut-off}$. Refer to Supplementary Note 2 on how we select V_l. Additional features are developed on extracted segments of the current curve during the constant voltage charge protocol based on two current threshold values, I_h and I_l respectively (see Figure 1b) for all cells except for the 2-step fast charge protocol. We select I_h equal to charge C-rate, while the lower threshold value, I_l, equal to a current drop of 40% from I_h. This allows for sufficient data to be recorded while keeping the diagnostics time to a minimum. For cells cycled with the 2-step fast charge we select the current curve in Supplementary Figure 2b. The obtained segments of voltage and current charge curves are further processed to obtain a plethora of features as described in Supplementary Note 3. Supplementary Table 2 summarises all features generated from processing the curves.

Feature selection with recursive feature elimination and cross-validation (RFE-CV) performs selection and subset reduction automatically without requirements of user-based thresholds, such as a maximum number of features to be selected. To suit battery data, we modify the original formulation by replacing the decision function algorithm with a random forest (RF) as opposed to the support vector machine (SVM) used in [44]. The replacement is motivated by RF's ability to deal with unscaled data. We call the resultant modified algorithm RF-RFE-CV. We use 700 decision tree estimators for the random forest algorithm and we set the number of cross-validations equal to the number of batteries in the feature selection dataset (see Supplementary Note 5 for data partition). We perform feature selection for each battery dataset based on a subset of the training data to avoid introducing optimistically biased performance estimates.

Battery SOH is quantified as capacity fade with reference to the first cycle as per equation 1, where C_i represents capacity value at i^{th} cycle and C_1 is the capacity at the first cycle measured by a complete charge-discharge operation.

$$SOH = \frac{C_i}{C_1} \tag{1}$$

The role of the algorithm is to map from inputs $\mathbf{x}$ to target variable y by means of a function $f(\mathbf{x}, \theta)$:

$$y = f(\mathbf{x}, \theta) + \epsilon \tag{2}$$

where θ are the model weights vector and $\epsilon \sim \mathcal{N}(0, \Sigma)$ is a normally distributed noise parameter. Based on the selected algorithm, the function $f(\mathbf{x}, \theta)$ may take different forms based on underlying assumptions of each algorithm. The learned model can then be used to make predictions of capacity given a test vector $\mathbf{x}^*$.

Data augmentation is carried out using the *fast gradient sign method (FGSM)* in combination with the weight decay algorithm (ridge regression). We have found that a Ridge regularised model in combination with the FGSM was able to reduced the confidence interval (CI) around the estimated mean, despite being a simpler model than the original formulation in [47] which was based on a neural network. Given an input $\mathbf{x}$ with a target y and loss $l(\theta, \mathbf{x}, y)$, FGSM generates an adversarial example using:

$$\mathbf{x}_{adv} = \mathbf{x} + \gamma \cdot sign(\nabla_x l(\theta, \mathbf{x}, y)) \tag{3}$$

where γ is a small value such that the max value of the perturbation is bounded and ∇_x is the gradient with respect to $\mathbf{x}$. Because each feature in the dataset has a different range, we set γ to 0.01 or 1% times the range of each feature vector. The adversarial examples are concatenated with the original training data to create a comprehensive training dataset. Note, other methods for data augmentations can also be used such as the ones proposed in [50], [52], [53], [54], however the effect of data augmentation on model performance is beyond the scope of the present work.

The study solves eq. 2 by making use of four algorithms as follows:

Bayesian Ridge regression (BRR) considers a probabilistic model of the regression problem. The algorithm estimates a spherical Gaussian prior over the model weights given by $p(\theta|\lambda) = \mathcal{N}(\theta|0, \lambda^{-1}\mathbf{I}_p)$, where λ^{-1} is the precision. The priors over α (the regulariser) and λ are chosen to be gamma distributions. All parameters, θ, λ and α, are jointly estimated during training as per the implementation in [55]. Posterior inference can be performed in a closed-form because the prior is conjugate. For a complete explanation of the algorithm refer to [56].

Gaussian process regression (GPR) is a nonparametric, Bayesian approach to regression defining a probability distribution over functions rather than random variables, thus eq. 2 is solved by:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{4}$$

where $m(\mathbf{x})$ is the mean and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function. Note, as defined above, GPR does not require learning the parameters of the regression function $f(\mathbf{x}, \theta)$, in a traditional sense. The mean and covariance are defined by:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{5}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x} - m(\mathbf{x}))(f(\mathbf{x}' - m(\mathbf{x}'))] \tag{6}$$

GPR assigns a prior probability to every possible function, where higher probabilities are given to functions that the algorithm considers to be more likely, for example, because they are smoother than other functions. For our implementation, we make use of the standard radial basis kernel (RBF) as detailed in [57], where a mathematical explanation of the algorithm is also given. Other kernel options exist, however, we do not explore the effect of kernel choice on algorithm performance.

Random Forest (RF) is a collection of constructed decision trees who sequentially conduct binary splits of the data to produce a homogeneous subset. For a comprehensive explanation of the algorithm refer to [58]. We adopt a bagging approach where the ensemble members are trained on different bootstrap samples of the training set and we set the number of decision trees in the forest to 1500. The variability of the predictions estimated by the random forest has been investigated based on the study from [49], where the confidence interval's variance has been obtained using the bootstrap replicates used to train the random forest itself.

Deep ensemble of neural networks (dNNe). Ensemble methods combine different regressors into a meta-regressor and we consider an ensemble of deep neural networks as proposed in [50]. Each network in the ensemble incorporates

2 hidden layers with an output of two layers one for the mean, $\mu(x)$ and the other for variance, σ^2 with $\sigma^2 > 0$. We use the negative log-likelihood as a function of the predicted mean and variance for scoring purposes. We also use a feed-forward architecture of 2 densely connected hidden layers. Each layer decreases in size by 50% neurons based on the number of input features. When the input number features is less than 10, we force the network's hidden layers to 4 neurons in the first layer and 3 in the second layer. For example, when 18 input features are considered, the first hidden layer consists of 9 neurons, followed by 4 neurons in the second hidden layer. Each network used in this work has the following parameters: first hidden layer implies a ReLU activation function, followed by a Leaky ReLU for the second hidden layer and a Sigmoid function for the output layer. Additionally, we make use of Adam optimiser with a learning rate of 0.001 and a batch size equal to the number of cycles for each cell in the training set.

All models are evaluated based on mean absolute percent error (MAPE) and root mean squared error (RMSPE).

$$MAPE(y_i^*, y_i) = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i^* - y_i|}{y_i} \tag{7}$$

$$RMSPE(y_i^*, y_i) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i^* - y_i}{y_i}\right)^2} \tag{8}$$

where y_i is the measured capacity value, y_i^* is the estimated capacity value, and n is the total number of samples.

In a regression setting, we obtain probabilistic forecasts using one of the algorithms described above through the estimation of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where μ is the mean estimated capacity and σ^2 is the associated uncertainty quantified as variance. To evaluate the usefulness of predictive uncertainty for decision making, we create reliability diagnostics curves analogous to the work in [59]. To plot calibration curves, we divide each predicted confidence interval in m confidence levels that are monotonically increasing on the interval $[0, 1]$ i.e. $0 < p_1 < p_2 < ... < p_m < 1$. We then compute the empirical probability for each threshold by counting the frequency of true labels in each confidence level p_m. Mathematically this can be summarised as:

$$\hat{p}_m = \frac{|y_n | F_n(y_n) \leq p_m, n = 1, ..., N|}{N} \tag{9}$$

Based on the reliability curve assessment, we then perform re-calibration using isotonic regression [60]. A well-calibrated regressor should lie very close to the ideal diagonal curve, e.g. results Figure 3b. We use the calibration score(C_{score}) as a numerical score to describe the quality of the calibration when referenced to a 90% confidence interval and sharpness (Sh) to describe average standard deviation.

$$C_{score} = \frac{1}{N}\sum_{i=1}^{N}\hat{p}_{m=90\%} \tag{10}$$

Sharpness is calculated as an average of model output variance for each prediction and is given by:

$$Sh = \frac{1}{n}\sum_{i=1}^{n}\sigma_i \tag{11}$$

where i is the sample number and n is the the total number of sampels.

We further propose an assessment of uncertainty prediction via prognostics performance metrics from an engineering point of view, adopted from [61]. First, we introduce the accuracy zone defined by a threshold, α (see Figure 2), which is calculated as a percentage error from the true capacity value, i.e. $y \pm \alpha$. We select an α of $\pm 1.5\%$ (alpha can be adjusted based on intended application). Based on the frequency of predicted values residing in the accuracy zone, we calculate the α-accuracy. Finally, we calculate the average probability mass of the prediction PDF within the α bounds called β, refer to Figure 2. Ideally, β should be one, suggesting that the predicted confidence interval is small and encapsulates the entire α-accuracy zone. Since α summarises the notion of *desired accuracy*, α^+ is the upper bound for estimates above the accuracy zone, and α^- represents low estimates or value residing under the desired accuracy zone. Depending on the application, both or any one of the low or high estimates may be undesirable. We chose to calculate the percentage of early predictions (estimates residing below the true label, the red line in Figure 2), denoted here by PEP, as a measure of algorithm uncertainty in a critical application scenario.

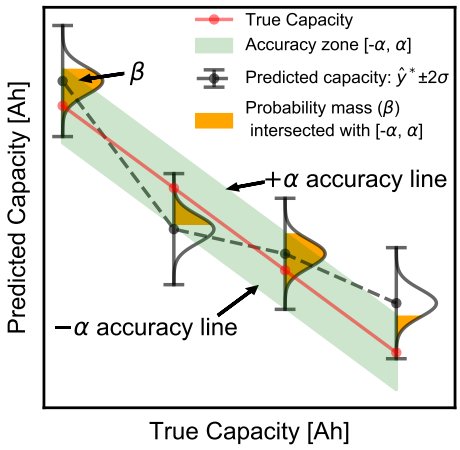

Figure 2: α-accuracy zone and β probability mass illustration.

Group*	I	I	I	I	I	II	III
Dataset	CALCE CS2	CALCE CX2	CALCE PL	NASA 5	NASA11	TRI	Oxford
Manufacturer	Unknown	Unknown	Unknown	LG Chem	LG Chem	A123 Systems	Kokam
Cathode ***	$LiCoO_2$	$LiCoO_2$	$LiCoO_2$	$LiCoO_2$	$LiCoO_2$	$LiFePO_4$	$LiCoO_2$ / $LiNiMnCoO_2$
Form factor	Prismatic	Prismatic	Pouch	18650 Cylindrical	18650 Cylindrical	18650 Cylindrical	Pouch
# cells	6	6	2	8	25	124	8
Charge	CC-CV	CC-CV	CC-CV	CC-CV	CC-CV	Fast-charge	CC
Discharge	2 regimes	2 regimes	1 regime	2 regimes	7 regimes	1 regime	1 regime

*Groups based on charge protocol, **Toyota Research Institute, ***Information from manufacturer, not verified

Table 1: Datasets overview. Note: refer to Supplementary Note 4 for data sources.

3 Dataset

We investigate the performance of BHUMP on a total of 179 Li-ion cells as referenced in Table 1. The cells have been grouped into three categories based on the charging protocol used: constant current - constant voltage (CC-CV) protocol in Group I (47 cells), 2-step fast charge protocol in Group III (8 cells), and constant current (CC) protocol in Group II (124 cells). The separation is important for separate model training and feature selection, as well as model performance assessment on different charge protocols. A detailed explanation of each dataset used can be found in Supplementary Note 4.

4 Algorithm performance

4.0.1 Group I data

Subject to the previously described pipeline steps the feature selection algorithm, RF-RFE-CV chose 18 of the 30 engineered features as the optimum number of attributes for the cells in Group I (refer to Supplementary Figure 8a and Supplementary Table 3). From a threshold point of view, we select a V_h of 4.2V for all batteries in this Group with an associated V_l of 3.9V. Refer to Supplementary Note 5 for train/test partitions.

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
BRR	1.52	2.49	84.49	0.021	70.00	0.57	68.92
GPR	1.49	2.24	92.23	0.025	65.00	0.48	71.76
RF	0.72	0.91	100	0.046	92.00	0.29	95.29
dNNe	0.65	0.92	88.01	0.0082	93.00	0.93	97.71

Table 2: Results for Group I cell no. 38.

We illustrate results for BHUMP when dNNe is considered as base algorithm in Figure 3 (results for all other algorithms are shown in Supplementary Figures 11, 12, 13) for a randomly chosen pouch cell battery, cell no. 38 and summarise algorithm performance on this cell in Table 2. The cell was cycled in full depth of discharge between 4.2V to 2.7V at a discharge C-rate of 0.5C (or 0.55 A) with a CC-CV charging protocol at a current value of 0.5 C-rate. Table 3 summarises each algorithms' performance on cell no. 38. Comparing dNNe in Figure 3a to the other algorithms BRR, GPR, and RF, we show that the resultant confidence interval is considerably smaller (all figures display a confidence level equivalent to a 95% quantile i.e. $\mu \pm 2 \cdot \sigma$). This indicates that the model is sharper, resulting in a high β score (refer to Table 2 for results). Where the predictions are less accurate, such as is the prediction in the first few cycles (see Figure 3a), the error bars capture this variability well. On this battery, dNNe also achieves the best RMSPE and MAPE together with a high calibration score. As per Table 2, the estimates for this cell vary between RMSPE 0.65% to 1.52%, showing that all 4 algorithms can achieve high performance. The same conclusion is not valid for calibration, however. Reliability plots indicate that RF exhibits high variance even after calibration, refer to Supplementary Figure 13.

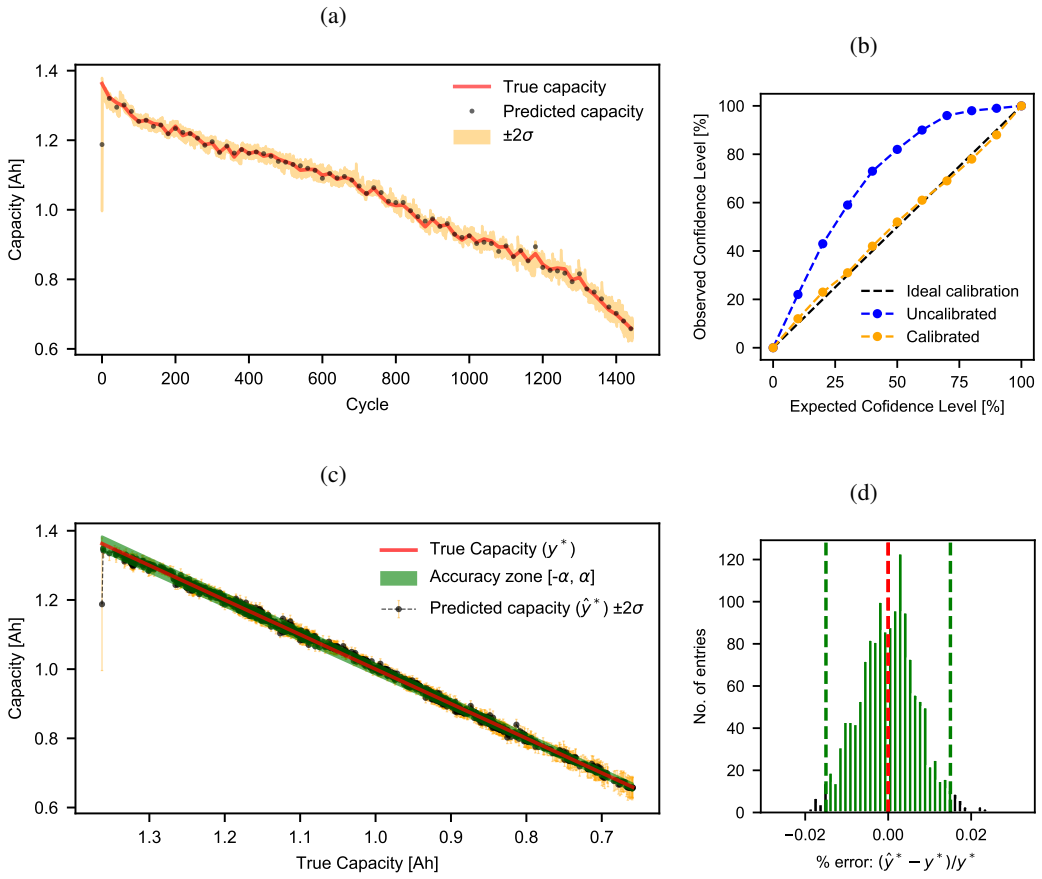

Figure 3: **Prediction results with dNNe Group I cell no. 38.** **a** dNNe prediction as a function of cycle, **b** dNNE calibration results, **c** dNNe actual vs. predicted capacity, **d** Histogram of % error. Note: y^* - true capacity, $\hat{y}^*$ - predicted capacity

When discussing average results across all cells in Group I (Table 3), RF achieves on average a low calibration error of 54.70% possibly due to the method used for estimating the variance, Infinitesimal Jackknife. In practice we prefer a more conservative system, particularly in safety-critical applications. This implies that the number of capacity estimates lower than the true label residing in the α-accuracy zone (Figure 2) should exceed the number of capacity values estimated above it i.e. PEP should be close to 100%. At the same time, too low of a capacity estimate would result in a far too conservative algorithm. However, such behaviour is captured by an increase in RMSPE and thus mitigated for naturally. With reference to Figure 3d together with Table 2 one can conclude that dNNe is conservative, achieving the highest PEP.

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
BRR	4.65	5.54	89.16	0.104	25.76	0.25	36.57
GPR	3.70	4.51	83.62	0.089	32.04	0.29	60.07
RF	2.17	2.70	54.70	0.093	35.94	0.36	65.47
dNNe	3.30	4.26	86.28	0.043	32.14	0.58	63.26

Table 3: Average results over Group I dataset.

Overall, despite RF achieving the lowest average RMSPE and MAPE (Table 3) it does not output well-calibrated predictions, nor it displays a high sharpness value. At the expense of 1.13% in MAPE and 1.56% in RMSPE, the dNNe outputs a well-calibrated model, on average being less than 4% under the ideal calibration score.

4.0.2 Group II data

Group II dataset is the largest dataset incorporating a total of 124 cells. While the dataset exhibits a high variance in charge profiles, it does not have any variation in discharge conditions (all cells in the dataset are discharged at 4 C-rate). This, in turn, showcases the effect of the charge profile on the estimation accuracy of the 4 algorithms. Training is performed on features engineered based on the CC-CV curve obtained after the cell reaches 80% SOC (refer to Supplementary Figures 2a and 2b). Refer to Supplementary Note 5 for train/test partitions. RF-RFE-CV selects a total of 5 features (Supplementary Figure 8b and Supplementary Table 4) out of a total of 30 engineered features. We believe this is caused by the fact that the dataset only incorporates one discharge profile as well as just a single battery type.

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
BRR	0.72	0.90	65.49	0.005	89.00	98.00	20.70
GPR	1.23	1.63	69.94	0.011	65.00	85.00	22.16
RF	0.23	0.43	87.42	0.002	98.00	100	42.81
dNNe	0.34	0.48	71.31	0.002	98.00	100	31.50

Table 4: Results for Group II cell no. 1.

Figure 4 illustrate BHUMP performance with a dNNe as base algorithm for cell no. 1, whilst Suplementary Figures 14, 15, 16 summarise results for all other algorithms. The cell has undergone fast charge profile of 3.6 C-rate to 80% SOC, beyond which the cell is charged with CC of 1C followed by the CV charging. The reason cell 1 was selected in this case was to illustrate the performance of the algorithms when there is a high number of outliers in capacity data (Figure 4a). With reference to Table 4, RF achieves lowest error and highest scores as well as a good calibration compared to all other algorithms. On this particular cell, dNNe achieves the second best performance, however it does not output a well calibrated model, despite showing a good average calibration score as per Table 5.

Average results of the 4 algorithms are concisely summarised in Table 5. All models are able to estimate the SOH with less than 2% RMPSE; this underlines the fact that the models are not affected by the fast-charge section of the charging protocol. RF achieves the highest accuracy with a low sharpness value and high percentages for all other metrics except for calibration where it exhibits over-confidence. In terms of calibration error, dNNe achieves the closest score to a 90% confidence interval with 91.02%. dNNe is also the second-best performing algorithm achieving good scores across all metrics as summarised in Table 4. In comparison, the two Bayesian-based algorithms exhibit a higher percentage error as well as higher sharpness values. However, they tend to be more conservative, averaging a PEP over 60%.

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
BRR	0.45	0.76	91.72	0.005	97.31	99.19	62.86
GPR	1.00	1.91	93.14	0.012	90.43	83.74	63.21
RF	0.11	0.14	79.72	0.001	99.84	99.96	58.77
dNNe	0.23	0.45	91.02	0.002	99.53	99.50	53.41

Table 5: Average results over Group II dataset

In conclusion, from an accuracy and sharpness perspective, the best performing algorithm on dataset Group II is RF, whilst the poorest performance is achieved by GPR. When it comes to uncertainty metrics, and in particular calibration, RF exhibits over-confidence with a C_{score} of 79.72%. Such behaviour is also identified in Group I dataset where RF was, in fact, difficult to calibrate despite the rich dataset. A more reliable calibration score is achieved by dNNe at the expense of a loss of 0.12% in MAPE and 0.31% in RMSPE (refer to Table 5).

4.0.3 Group III data

On Group III we emphasise on the suitability of BHUMP to battery state of health estimation for automotive applications. Group III includes 8 Kokham 740 mAh batteries that have been dynamically cycled under the ARTEMIS [62] dynamic driving profile, followed by characterisation cycles. Each characterisation cycle consists of low rate CC charge and discharge cycles, repeated every 100 cycles. We use the characterisation cycles for diagnostics purposes to derive

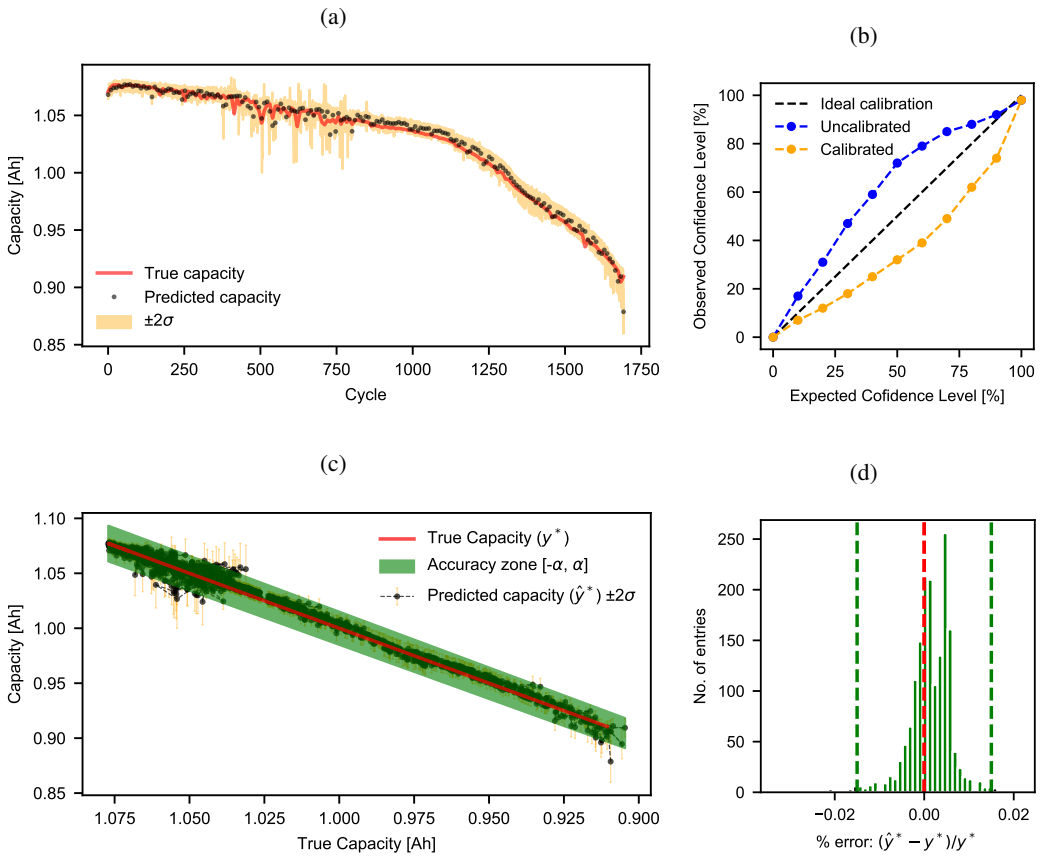

Figure 4: **Prediction results with dNNe Group II cell no. 1.** **a** dNNe prediction as a function of cycle, **b** dNNE calibration results, **c** dNNe actual vs. predicted capacity, **d** Histogram of % error. Note: y^* - true capacity, $\hat{y}^*$ - predicted capacity

features and estimate battery health. This dataset incorporates the lowest variability both in terms of input feature values and capacity degradation values due to the identical charge-discharge conditions. This, in turn, affects feature selection as BHUMP only selects 5 out of the 18 engineered features (note charge protocol does not include CV part of the charge, hence 12 features are missing) as shown in Supplementary Figure 8c and Supplementary Table 5. We keep the same threshold values as in Group I cells for the CC part of the curves, namely a V_h of 4.2V and a V_l of 3.9V on which feature are engineered. Refer to Supplementary Note 5 for train/test partitions.

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
BRR	0.11	0.15	95.55	0.89	100	100	31.11
GPR	0.16	0.19	71.11	1.21	100	100	15.55
RF	0.17	0.21	97.77	2.01	100	100	24.44
dNNe	0.20	0.25	100.00	2.93	100	100	6.67

Table 6: Results for Group III cell no. 5.

For visualisation purposes, we illustrate results for the randomly selected cell no. 5 for dNNe in Figure 5 and Supplementary Figures 17, 18, 19 for all other algorithms. It is clear, from Table 6 that performance on cell 5 is dominated by BRR based on all measures of accuracy and uncertainty quantification. However, all algorithms deployed on cell no. 5 (Table 6) achieve a MAPE and RMSPE smaller than the proposed accuracy zone threshold α of $\pm 1.5\%$.

Average results are summarised in Table 7. In terms of accuracy measures, on average, BRR outperforms all other methods, including the dNNe. As argued in [63] linear regression outperforms considerably more complex algorithms, including NNs when dealing with small sample size that exhibits little variance. Despite the low error, BRR does not achieve a good calibration score as opposed to dNNE. dNNe is the second-best performing algorithm in terms of

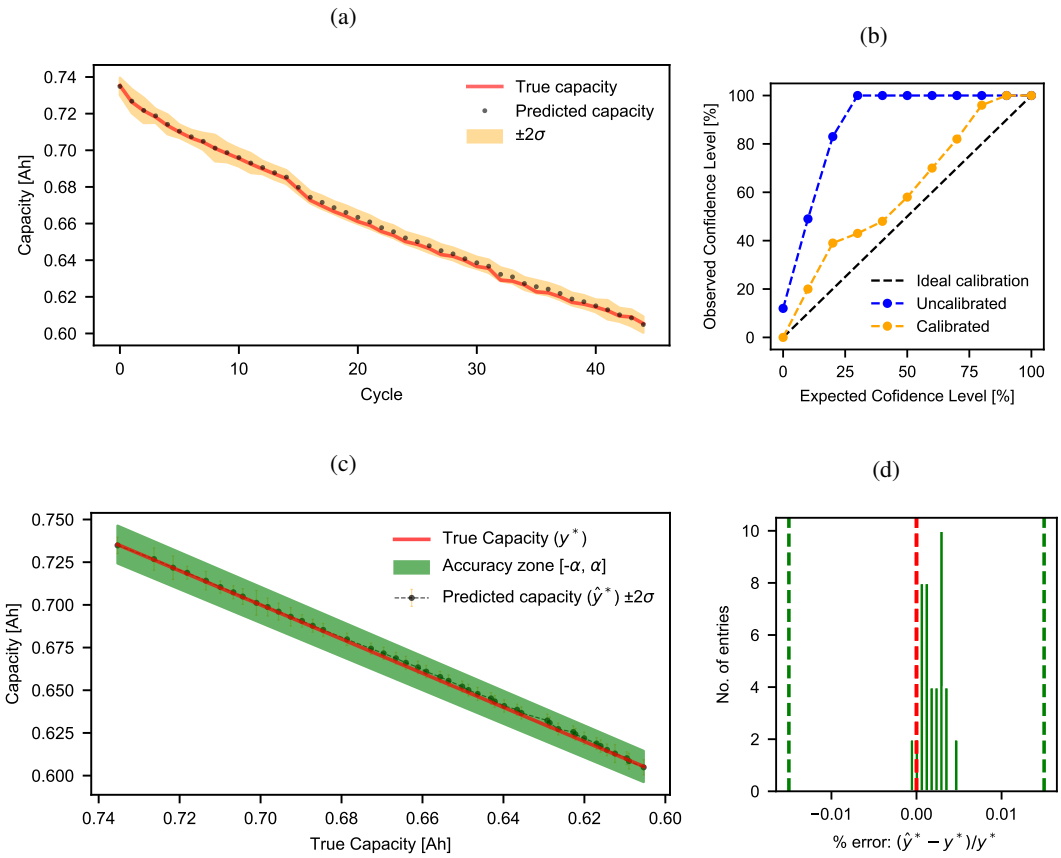

Figure 5: **Prediction results with dNNe Group III cell no. 5. a** dNNe prediction as a function of cycle, **b** dNNE calibration results, **c** dNNe actual vs. predicted capacity, **d** Histogram of % error. Note: y^* - true capacity, $\hat{y}^*$ - predicted capacity

accuracy (MAPE and RMSPE). It also exhibits adequate results for all other metrics, including PEP where it scores the highest.

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
BRR	0.26	0.32	68.11	1.20	100	100	23.54
GPR	0.52	0.65	42.42	2.37	90.50	97.25	23.22
RF	0.36	0.44	72.62	2.16	88.5	100	25.44
dNNe	0.30	0.39	91.17	2.01	98.25	99.75	27.95

Table 7: Average results over Group III dataset

In conclusion, when considering average results over all 4 test cells as referenced in Table 7, dNNe achieves second-best accuracy while attaining the best calibration score of 91.17%.

5 Discussion on practical applicability of BHUMP

BHUMP can complement battery management systems (BMS), for both SOC and SOH estimation, and replace the traditional ECMs altogether. While conventional approaches rely on measuring the capacity in static conditions such as full charge-discharge, BHUMP can estimate capacity fade from sections of the charge profile, accommodating for partial discharge scenarios or various operating conditions such as random or high discharge rates. We succinctly summarised in the results section, BHUMP can estimate capacity fade under fast charging protocol (Group II data) as well as random discharge (Group III data cycled under ARTEMIS driving protocol) typical to the operation of an EV battery pack. Future work could further extend to other charge-discharge protocols and open-source datasets such as the one in [19].

Temperature variations during charging could further introduce uncertainty into the measurement of charge curves and propagate it into the estimation algorithm. Possible mitigation includes the use of temperature as an input when training BHUMP or considering additionally in-situ or operando sensory information such as optical and digital images or X-ray [64] such that the algorithm learns the correlation between temperature, generated features and SOH indicator. Due to such variations, SOH assessment without corresponding measures of algorithm uncertainty does not provide sufficient information to form a decision or corrective action. In addition to inherent algorithm bias, dataset variability also seems to affect the prediction error. To accommodate for such variations in the data BHUMP introduces 30 engineered features and makes use of an unsupervised feature selection algorithm (RF-RFE-CV). Given a training dataset RF-RFE-CV selects a subset of input features, indicating that features must be selected based on intended application, battery design and charge protocol. Despite such dataset variations, we think that deep learning has the potential to exceed it in the future as it requires little tuning from the user and can take advantage of parallelisation and an increasing amount of computational capabilities by deployment on graphics processing units (GPU) and modern data storage solutions. In addition, when training data consists of limited samples or training data is not relevant to the intended application, transfer learning can be used to reduce prediction errors. New hardware, architectures and learning algorithms that are currently being developed for neural network implementation will only accelerate this process, allowing for active learning techniques to be used when deployed onboard a vehicle. More concretely, BHMUP with dNNe as the base algorithm can incorporate transfer learning when trained on a particular cell design and re-trained on a reduced sample set for a different cell design. Additionally, BHUMP can also incorporate active learning as data becomes available when deployed online on different cell design, chemistry or operating temperature.

6 Conclusion

The two widely adopted modelling techniques for online battery state of health (SOH) estimation are equivalent circuit models and electrochemical models. However, when deployed online, the trade-off between accuracy and computational efficiency is difficult to achieve. This paper introduced an alternative, machine learning-based solution called battery health and uncertainty management pipeline (BHUMP). The pipeline provides a set of benefits over conventional methods including adaptability to the charging protocols and the discharge current rates, and prediction without knowledge of battery design, chemistry, and operating temperature.

The paper explores four algorithms: Bayesian ridge regression (BRR), Gaussian process regression (GPR), random forest (RF), and a deep ensemble of neural networks (dNNe), as the base algorithm for BHUMP. All algorithms are assessed on error values and the ability to quantify uncertainty. Results indicate that the lowest error achieved depends on the charging protocol adopted. The lowest error was achieved by RF for constant current - constant voltage protocol and fast charge protocol, and BRR for the constant-current protocol. When considering uncertainty assessment metrics, however, RF is hard to calibrate and is overly optimistic in its predictions. At the expense of an average increase in MAPE of 0.43% and RMSPE of 0.97%, dNNe, generally achieves a better calibration score, consistently achieving the second-lowest error irrespective of charge protocol. On the fast-charging protocol, the best dNNe model achieved a RMSPE of 0.45% with a calibration score of 91.02% when referenced to a 90% confidence interval.

Overall, our work highlights the value of coupling machine learning tools with charge curve segments in capturing battery degradation in under 15 minutes. Moreover, we argue that despite achieving low errors, any algorithm must undergo uncertainty quantification checks before deployment in the field. Finally, we show how the use of machine learning pipelines can achieve a computationally efficient and accurate solution for cell SOH estimation. We envision machine learning pipelines to be a standard technique used in designing and implementing battery management systems of the future.

Data availability

The datasets used in this study are available at:

- Group 1:

  `https://web.calce.umd.edu/batteries/data.htm`

  `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/`

- Group 2:

  `https://data.matr.io/1/projects/5c48dd2bc625d700019f3204`

- Group 3:

  `https://ora.ox.ac.uk/objects/uuid:03ba4b01-cfed-46d3-9b1a-7d4a7bdf6fac`

Code availability

Code for the data processing is available from the corresponding authors upon request. Code for the modelling work is available at: `http://doi.org/10.5281/zenodo.4390152`

References

[1] Claire Curry. Lithium-ion battery costs and market. *Bloomberg New Energy Finance*, 5, 2017.

[2] Wolfgang Bernhart. Challenges and opportunities in lithium-ion battery supply. In *Future Lithium-ion Batteries*, pages 316–334. Royal Society of Chemistry, 2019.

[3] Gae-Won You, Sangdo Park, and Dukjin Oh. Diagnosis of electric vehicle batteries using recurrent neural networks. *IEEE Transactions on Industrial Electronics*, 64(6):4885–4893, 2017.

[4] Anthony Barré, Benjamin Deguilhem, Sébastien Grolleau, Mathias Gérard, Frédéric Suard, and Delphine Riu. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 241:680–689, 2013.

[5] Jingliang Zhang and Jay Lee. A review on prognostics and health monitoring of li-ion battery. *Journal of Power Sources*, 196(15):6007–6014, 2011.

[6] Alexander Farmann, Wladislaw Waag, Andrea Marongiu, and Dirk Uwe Sauer. Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles. *Journal of Power Sources*, 281:114–130, 2015.

[7] Mohammad A Hannan, MS Hossain Lipu, Aini Hussain, and Azah Mohamed. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renewable and Sustainable Energy Reviews*, 78:834–854, 2017.

[8] Xiaosong Hu, Shengbo Li, and Huei Peng. A comparative study of equivalent circuit models for li-ion batteries. *Journal of Power Sources*, 198:359–367, 2012.

[9] Tianheng Feng, Lin Yang, Xiaowei Zhao, Huidong Zhang, and Jiaxi Qiang. Online identification of lithium-ion battery parameters based on an improved equivalent-circuit model and its implementation on battery state-of-power prediction. *Journal of Power Sources*, 281:192–203, 2015.

[10] D Andre, M Meiler, K Steiner, H Walz, T Soczka-Guth, and DU Sauer. Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. ii: Modelling. *Journal of Power Sources*, 196(12):5349–5356, 2011.

[11] Matthew J Daigle and Chetan Shrikant Kulkarni. Electrochemistry-based battery modeling for prognostics. 2013.

[12] Brian Bole, Chetan S Kulkarni, and Matthew Daigle. Adaptation of an electrochemistry-based li-ion battery model to account for deterioration observed under randomized use. Technical report, SGT, Inc. Moffett Field United States, 2014.

[13] Githin K Prasad and Christopher D Rahn. Model based identification of aging parameters in lithium ion batteries. *Journal of power sources*, 232:79–85, 2013.

[14] Kristen A Severson, Peter M Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H Chen, Muratahan Aykol, Patrick K Herring, Dimitrios Fraggedakis, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383, 2019.

[15] Bhaskar Saha, Kai Goebel, Scott Poll, and Jon Christophersen. Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on instrumentation and measurement*, 58(2):291–296, 2008.

[16] Kai Goebel, Bhaskar Saha, Abhinav Saxena, Jose R Celaya, and Jon P Christophersen. Prognostics in battery health management. *IEEE instrumentation & measurement magazine*, 11(4):33–40, 2008.

[17] Xiaosong Hu, Jiuchun Jiang, Dongpu Cao, and Bo Egardt. Battery health prognosis for electric vehicles using sample entropy and sparse bayesian predictive modeling. *IEEE Transactions on Industrial Electronics*, 63(4):2645–2656, 2015.

[18] Verena Klass, Mårten Behm, and Göran Lindbergh. A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation. *Journal of Power Sources*, 270:262–272, 2014.

[19] Peter M Attia, Aditya Grover, Norman Jin, Kristen A Severson, Todor M Markov, Yang-Hung Liao, Michael H Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature*, 578(7795):397–402, 2020.

[20] Martin Coleman, William Gerard Hurley, and Chin Kwan Lee. An improved battery characterization method using a two-pulse load test. *IEEE Transactions on energy conversion*, 23(2):708–713, 2008.

[21] Wladislaw Waag, Stefan Käbitz, and Dirk Uwe Sauer. Experimental investigation of the lithium-ion battery impedance characteristic at various conditions and aging states and its influence on the application. *Applied energy*, 102:885–897, 2013.

[22] Uwe Tröltzsch, Olfa Kanoun, and Hans-Rolf Tränkler. Characterizing aging effects of lithium ion batteries by impedance spectroscopy. *Electrochimica Acta*, 51(8-9):1664–1672, 2006.

[23] Christoph R Birkl, Matthew R Roberts, Euan McTurk, Peter G Bruce, and David A Howey. Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, 341:373–386, 2017.

[24] Y. Li, S. Zhong, Q. Zhong, and K. Shi. Lithium-ion battery state of health monitoring based on ensemble learning. *IEEE Access*, 7:8754–8762, 2019.

[25] Yi Li, Changfu Zou, Maitane Berecibar, Elise Nanini-Maury, Jonathan C-W Chan, Peter van den Bossche, Joeri Van Mierlo, and Noshin Omar. Random forest regression for online capacity estimation of lithium-ion batteries. *Applied energy*, 232:197–210, 2018.

[26] Bingxiang Sun, Pengbo Ren, Minming Gong, Xingzhen Zhou, and Jingji Bian. Soh estimation for li-ion batteries based on features of ic curves and multi-output gaussian process regression method. *DEStech Transactions on Environment, Energy and Earth Sciences*, (iceee), 2018.

[27] Xuning Feng, Caihao Weng, Xiangming He, Xuebing Han, Languang Lu, Dongsheng Ren, and Minggao Ouyang. Online state-of-health estimation for li-ion battery using partial charging segment based on support vector machine. *IEEE Transactions on Vehicular Technology*, 68(9):8583–8592, 2019.

[28] Yi Li, Mohamed Abdel-Monem, Rahul Gopalakrishnan, Maitane Berecibar, Elise Nanini-Maury, Noshin Omar, Peter van den Bossche, and Joeri 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*, 373:40–53, 2018.

[29] Matthieu Dubarry, Vojtech Svoboda, Ruey Hwu, and Bor Yann Liaw. Incremental capacity analysis and close-to-equilibrium ocv measurements to quantify capacity fade in commercial rechargeable lithium batteries. *Electrochemical and Solid State Letters*, 9(10):A454, 2006.

[30] Caihao Weng, Yujia Cui, Jing Sun, and Huei Peng. On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. *Journal of Power Sources*, 235:36–44, 2013.

[31] Duo Yang, Xu Zhang, Rui Pan, Yujie Wang, and Zonghai Chen. A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *Journal of Power Sources*, 384:387–395, 2018.

[32] Robert R Richardson, Christoph R Birkl, Michael A Osborne, and David A Howey. Gaussian process regression for in situ capacity estimation of lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, 15(1):127–138, 2018.

[33] Yirong Shen, Matthias Seeger, and Andrew Y Ng. Fast gaussian process regression using kd-trees. In *Advances in neural information processing systems*, pages 1225–1232, 2006.

[34] Bhaskar Saha, Scott Poll, Kai Goebel, and Jon Christophersen. An integrated approach to battery health monitoring using bayesian regression and state estimation. In *2007 IEEE Autotestcon*, pages 646–653. Ieee, 2007.

[35] David Ben-Shimon and Armin Shmilovici. Accelerating the relevance vector machine via data partitioning. *Foundations of Computing and Decision Sciences*, 31(1):27–42, 2006.

[36] Zengkai Wang, Shengkui Zeng, Jianbin Guo, and Taichun Qin. Remaining capacity estimation of lithium-ion batteries based on the constant voltage charging profile. *PLoS ONE 13(7): e0200169.*, 2018.

[37] Stephen J Engel, Barbara J Gilmartin, Kenneth Bongort, and Andrew Hess. Prognostics, the real issues involved with predicting life remaining. In *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*, volume 6, pages 457–469. IEEE, 2000.

[38] Ekaterina Pomerantseva, Francesco Bonaccorso, Xinliang Feng, Yi Cui, and Yury Gogotsi. Energy storage: The future enabled by nanomaterials. *Science*, 366(6468), 2019.

[39] Zhi Wei Seh, Yongming Sun, Qianfan Zhang, and Yi Cui. Designing high-energy lithium–sulfur batteries. *Chemical Society Reviews*, 45(20):5605–5634, 2016.

[40] Guifang Liu, Huaiqian Bao, and Baokun Han. A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis. *Hindawi Mathematical Problems in Engineering Volume 2018, Article ID 5105709, 10 pages*, 2018.

[41] J. M. Kanter and K. Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 10 2015.

[42] Nick Williard, Wei He, Michael Osterman, and Michael Pecht. Comparative analysis of features for determining state of health in lithium-ion batteries. *International Journal of Prognostics and Health Management, ISSN 2153-2648*, 2013.

[43] Yang Zhang and Bo Guo. Online capacity estimation of lithium-ion batteries based on novel feature extraction and adaptive multi-kernel relevance vector machine. *Energies*, 2015.

[44] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[45] Burcu F Darst, Kristen C Malecki, and Corinne D Engelman. Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC genetics*, 19(1):65, 2018.

[46] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678, 2017.

[47] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[48] Marc Doyle, Thomas F Fuller, and John Newman. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical society*, 140(6):1526, 1993.

[49] Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

[50] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[51] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[52] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

[53] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[54] Yash Sharma and Pin-Yu Chen. Attacking the madry defense model with l_1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[56] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[57] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[58] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[59] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*, 2018.

[60] Nilotpal Chakravarti. Isotonic median regression: a linear programming approach. *Mathematics of operations research*, 14(2):303–308, 1989.

[61] Abhinav Saxena, Jose Celaya, Edward Balaban, Kai Goebel, Bhaskar Saha, Sankalita Saha, and Mark Schwabacher. Metrics for evaluating performance of prognostic techniques. In *2008 International Conference on Prognostics and Health Management*, pages 1–17. IEEE, 2008.

[62] Michel André. The artemis european driving cycles for measuring car pollutant emissions. *Science of the total Environment*, 334:73–84, 2004.

[63] Ina S Markham and Terry R Rakes. The effect of sample size and variability of data on the comparative performance of artificial neural networks and regression. *Computers & operations research*, 25(4):251–263, 1998.

[64] Albertus D Handoko, Fengxia Wei, Boon Siang Yeo, Zhi Wei Seh, et al. Understanding heterogeneous electrocatalytic carbon dioxide reduction through operando techniques. *Nature Catalysis*, 1(12):922–934, 2018.

[65] M Rosa Palacín. Understanding ageing in li-ion batteries: a chemical issue. *Chemical Society Reviews*, 47(13):4924–4933, 2018.

[66] Qianqian Liu, Chunyu Du, Bin Shen, Pengjian Zuo, Xinqun Cheng, Yulin Ma, Geping Yin, and Yunzhi Gao. Understanding undesirable anode lithium plating issues in lithium-ion batteries. *RSC Advances*, 6(91):88683–88700, 2016.

[67] Chaofeng Liu, Zachary G Neale, and Guozhong Cao. Understanding electrochemical potentials of cathode materials in rechargeable batteries. *Materials Today*, 19(2):109–123, 2016.

[68] D Aurbach, E Zinigrad, H Teller, and P Dan. Factors which limit the cycle life of rechargeable lithium (metal) batteries. *Journal of The Electrochemical Society*, 147(4):1274–1279, 2000.

[69] Peter Keil and Andreas Jossen. Charging protocols for lithium-ion batteries and their impact on cycle life—an experimental study with different 18650 high-power cells. *Journal of Energy Storage*, 6:125–141, 2016.

[70] Sheng S Zhang, Kang Xu, and TR Jow. Study of the charging process of a licoo2-based li-ion battery. *Journal of Power Sources*, 160(2):1349–1354, 2006.

[71] J Zhou and PHL Notten. Studies on the degradation of li-ion batteries by the use of microreference electrodes. *Journal of power Sources*, 177(2):553–560, 2008.

[72] Akram Eddahech, Olivier Briat, and Jean-Michel Vinassa. Determination of lithium-ion battery state-of-health based on constant-voltage charge phase. *Journal of Power Sources*, 258:218 – 227, 2014.

[73] Zengkai Wang, Shengkui Zeng, Jianbin Guo, and Taichun Qin. State of health estimation of lithium-ion batteries based on the constant voltage charging curve. *Energy*, 167:661 – 669, 2019.

[74] Abhinav Saxena, José R Celaya, Indranil Roychoudhury, Sankalita Saha, Bhaskar Saha, and Kai Goebel. Designing data-driven battery prognostic approaches for variable loading profiles: Some lessons learned. In *European conference of prognostics and health management society*, pages 72–732, 2012.

[75] Abdel Aziz Taha and Allan Hanbury. An efficient algorithm for calculating the exact hausdorff distance. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2153–2163, 2015.

[76] Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *Journal of Mathematical Imaging and Vision*, 27(3):203–216, 2007.

[77] Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless seth fails. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 661–670. IEEE, 2014.

[78] Thomas Eiter and Heikki Mannila. Computing discrete frechet distance. 1994.

[79] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[80] Yves Dupain, Teturo Kamae, and Michel Mendés. Can one measure the temperature of a curve? *Archive for Rational Mechanics and Analysis*, 94(2):155–163, 1986.

[81] RR Moore, AJ Van Der Poorten, et al. On the thermodynamics of curves and other curlicues. In *Conference on Geometry and Physics, Canberra*, pages 82–109. Conference on Geometry and Physics, Canberra, 1989.

[82] Aldo Balestrino, Andrea Caiti, and Emanuele Crisostomi. Generalised entropy of curves for the analysis and classification of dynamical systems. *Entropy*, 11(2):249–270, 2009.

[83] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[84] Christoph R. Birkl. Diagnosis and prognosis of degradation in lithium-ion batteries. 2017.

Supplementary material

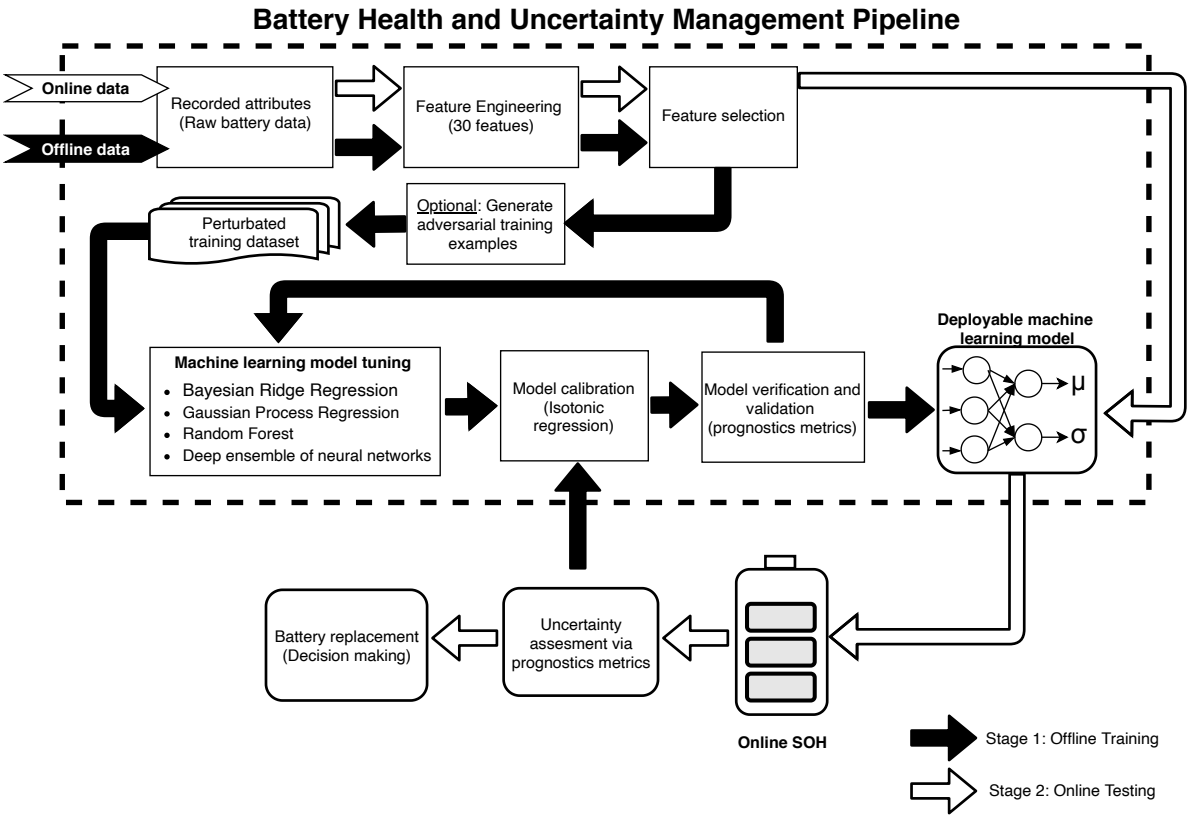

Figure 6: BHUMP flowchart

Supplementary Note 1. Domain explanation of features

Features are generated by mathematical manipulation, involving pattern recognition and information theory principles, of voltage and current charge curves. Any charging protocol finishes with both electrodes materials at their most extreme potential (and most reactive states) [65], namely the highest for the positive electrode and the lowest for the negative electrode. The diffusion of lithium ions inside an electrode is a complex process involving both microscopic and macroscopic processes that can potentially be partially captured by charge curves. During charging two crucial processes occur at the anode side (graphite-based batteries considered here), namely the intercalation of lithium ions into the active material and lithium plating [66], [67], [68]. Due to intercalation kinetics at the anode, cathode deintercalates faster than the anode can intercalate, and thus during charging, the current is the main limiting factor in a graphite-based lithium-ion battery. [66], [69], [68] Consequently, any charging protocol suffers from such limitations. The charging protocols typically go through a constant-current (CC) mode, followed by a constant-voltage (CV) mode, see Supplementary Figure 8 for a typical CC-CV charge protocol.

Zhang et al. [70] investigated the effects of charging protocols in $LiCoO_2$ based batteries by creating a bespoke three-electrode cell. The authors emphasise that lithium-ion plating coexists with the intercalation process in the anode and it occurs in the late period of the CC despite the graphite not being fully lithiated. Similarly, Zhou et al. [71] also mention that kinetically, under high-current charging conditions, the negative electrode can be polarised to such an extent that it's potential drops below 0 V, facilitating lithium metal deposition onto the surface of the electrode particles. It is known that the duration of the CC captures the polarisation phenomenon.[71] Therefore, as the battery ages, the constant current charge time (CCCT) decreases. Upon the start of the CV charging, as the current decreases, the negative electrode slowly recovers to a nominal potential value. The CV mode duration is thus crucial to eliminate the polarisation effect caused during the CC mode allowing for the anode to recover and thus fully charge the battery. With aging, the constant voltage charge time (CVCT) increases as demonstrated in [70] and [42].

A feasibility study of CCCT and constant voltage charge time (CVCT) as proxies for battery state of health was carried out in [42]. CVCT has already been considered as input to SOH methods in the additional studies [72], [73]. To reduce diagnostics time, we only use sections of the charge curves as input to the algorithm. The availability of the entire

charge curve in real-life applications is limited. Hence it is advantageous to design features that could be extracted from segments of such curves. The benefits of the approach are a lower diagnostics time (as little as 15 min) and the possibility of battery SOH estimation even in partial discharge conditions.

During discharge, the process of lithium extraction/insertion happens in reverse from anode to cathode. Since discharge currents vary with usage, we only extract one feature from the discharge curve, namely the pseudo linear resistance as introduced by Saxena et al. [74]. This is due to the instant drop in voltage associated with internal battery impedance on the application of load current. We estimate this resistance as the ratio of the observed voltage drop and the applied load current. It is understood that as the battery degrades the internal resistance of the battery increases, and hence an estimate of this internal resistance can be used as a proxy for battery SOH.[74] We used a lagged version of this feature, i.e. pseudo linear resistance from the previous cycle to estimate the SOH at the end of a charge cycle. For a mathematical explanation of all engineered features in Supplementary Material Table 9 refer to Supplementary Note 3 Feature engineering.

Supplementary Note 2. Voltage threshold values

We first define V_h to be equal to charge cut-off voltage, $V_{cut-off}$, while V_l is defined using the below formula:

$$V_l = V_h - \Delta V \tag{12}$$

where ΔV is a predefined voltage range. The recorded curve between V_l and V_h with each charge as illustrated in Figures 7a, 7a is then normalised on the interval $[0, 1]$ by subtracting the minimum value and dividing by the resulted maximum value. Following the normalisation procedure, we proceed on mathematically deriving the features. This allows for training different batteries types and designs on the same training dataset provided they underwent the same charging protocol. To overcome issues resulting from battery terminal voltage increase after previous discharge cycle and to capture the late period of the CC charging phase (when lithium plating occurs) we make use of a ΔV equal to 0.3V. A high V_l value accommodates for the increase in battery terminal voltage upon removal of load current after each discharge cycle. A behaviour commonly observed with battery ageing as referenced in Supplementary Figure 14. Furthermore, a high V_l threshold reduces the time necessary to record the CC charge curve while accommodating for partial discharge of the battery. Note, ΔV value and corresponding V_l and V_h threshold values could be adjusted based on battery type, application and user behaviour, end of life threshold, data storage capacity and processing power.

Supplementary Note 3. Feature engineering.

Capacity (Q) is calculated based the charge/discharge current (I) and it is given by:

$$Q = \int_{t_0}^{t_{end}} Idt \tag{13}$$

Energy (E) is calculated based on capacity (Q) and voltage (V) given by:

$$E = \int_{t_0}^{t_{end}} V(t) \cdot Idt \tag{14}$$

Attribute	Target variable
Cycle time	Discharge Capacity
Discharge C-rate	
Charge C-rate	
Operational time	
Voltage vs. Time	
Current vs. Time	
Charge times	

Table 8: Parameters recorded during cycling tests.

From pattern recognition domain, three features are derived, signal mean, kurtosis coefficient and skewness coefficient. Skewness coefficient and kurtosis coefficient are calculated based on the following formulas:

$$skewness = \frac{\sum_{i=1}^{n}(x(i) - \bar{x})^3}{(n-1)\sigma_x^3} \tag{15}$$

$$kurtosis = \frac{\sum_{i=1}^{n}(x(i) - \bar{x})^4}{(n-1)\sigma_x^4} \tag{16}$$

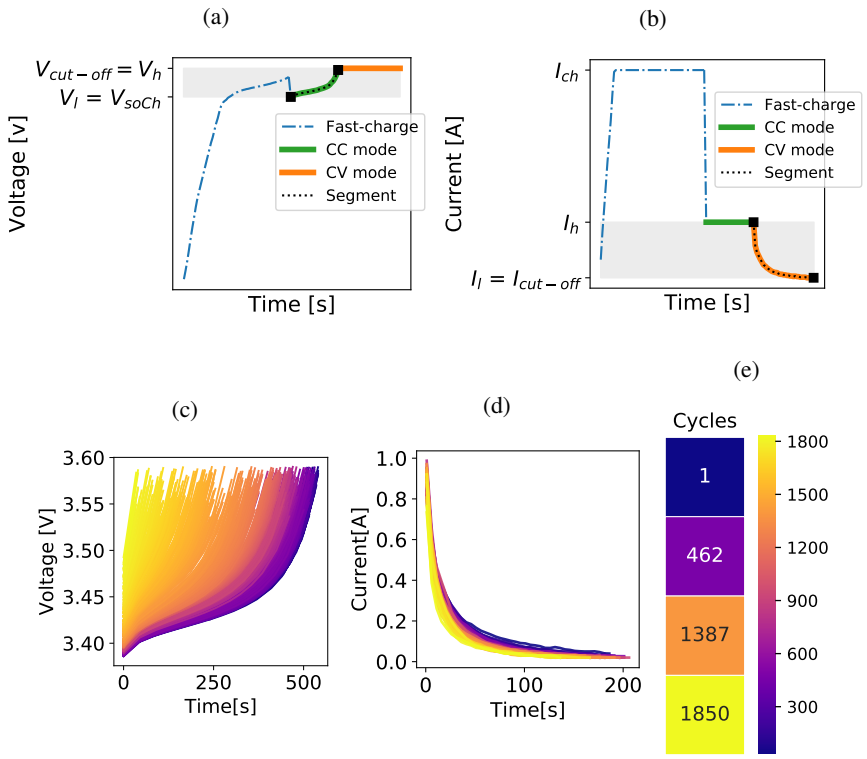

Figure 7: **The 2 step fast-charge protocol and extracted ageing segment of the curves for a Li-ion pouch cell. a** Voltage during charge protcol. **b** Current during charge protocol, **c** Extracted ageing voltage curve segments corresponding to marked grey area, **d** Extracted ageing current curve segments corresponding to marked grey area, **e** Heatmap of ageing with cycle number.

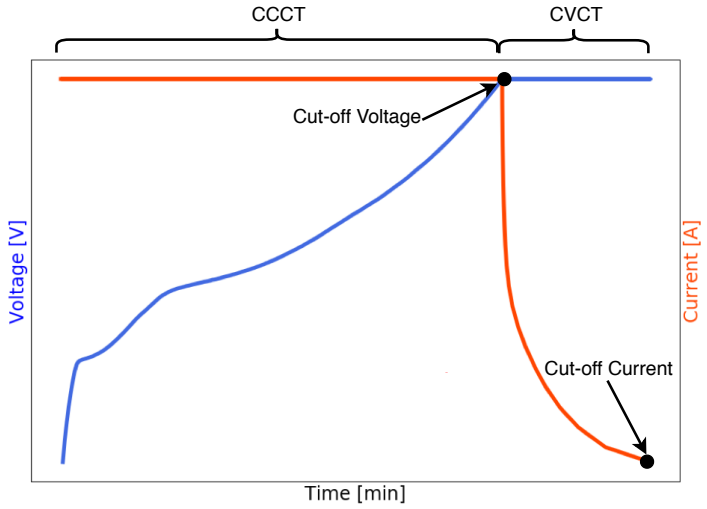

Figure 8: Typical constant current - constant voltage charge protocol. Note: CCCT=constant voltage charge time, CVCT=constant voltage charge time

where $\bar{x}$ and σ_x represent the mean and standard deviation of feature x.

In addition to pattern recognition based features, distance measurements from a predetermined reference curve to CVCC - CVCT curve and CCCV - CCCT have also been considered. We choose here as reference a simple line defined

	Feature	Target variable
Battery specific data	Nominal Capacity [Ah]	Discharge Capacity [Ah]
	Charge Current [A]	
	Discharge Current [A]	
Cumulative (historical) data	Cumulated Discharge Capacity [Ah]	
	Cumulated Discharge Energy [Wh]	
1 Cycle Lagged Data	Lagged Cycle Time [s]	
	Lagged Pseudo Resistance [Ω]	
Instantaneous charge data*	Terminal Voltage @ Start of charge [V]	
	Charge time of CC segment of charge curve [s]	
	Charge time of CV segment of charge curve [s]	
	Mean current during CC segment of the curve [A]	
	Mean voltage during CV segment of the curve [A]	
	Slope of CCCV-CCCT segment of the curve	
	Slope of CVCC-CVCT segment of the curve	
	Energy during CCCV-CCCT segment of the curve [Wh]	
	Energy during CVCC-CVCT segment of the curve [Wh]	
	Energy ratio CCCV-CCCT / CVCC-CVCT segment of the curve	
	Energy Difference between the curve segments (CCCV-CCCT) - (CVCC-CVCT)	
	Entropy of CCCV-CCCT segment of the curve eq 19	
	Entropy of CCCV-CCCT segment of the curve eq 19	
	Shannon entropy of CCCV segment of the curve	
	Shannon entropy of CVCC segment of the curve	
	Skewness coefficient of CCCV-CCCT segment of the curve eq 15	
	Skewness coefficient of CVCC-CVCT segment of the curve eq 15	
	Kurtosis coefficient of CCCV-CCCT segment of the curve eq 16	
	Kurtosis coefficient of CVCC-CVCT segment of the curve eq 16	
	Frechet Distance of CCCV-CCCT segment of the curve eq 18	
	Frechet Distance of CVCC-CVCT segment of the curve eq 18	
	Hausdorff Distance of CCCV-CCCT segment of the curve eq 17	
	Hausdorff Distance of CVCC-CVCT segment of the curve eq 17	

Table 9: Engineered features based on recorded parameters in Table 8. Note: CC = consatnt current, CV = constant voltage, CCCV = constant current charge voltage, CVCC = contant voltage charge current, CCCT = constant current charge time, CVCT = constant voltage charge time

by the equation $y = mx + c$ where y represents current or voltage depending on the curve under scrutiny, and x represents time. An illustration of the two curves and their reference lines are shown in figures 9 and 10. Instead of simple Euclidean distance, we employ here two different measurements, namely Directed Hausdorff (DH) and Frechet (FD) distance. Both methods are well established in various domains and thoroughly explained in [75], [76] and [77]. We only consider here Directed Hausdorff distance from charge curve to reference line and not vice-versa. DH distance between two point sets $A(a_1, a_2)$ and $B(b1, b2)$, where a_1, a_2, b_1, b_2 are 2D coordinates, is calculated as maximum distance between each point $x \epsilon A$ to its nearest neighbour $y \epsilon B$ and is given by:

$$H(A, B) = max_{x \epsilon A}\{min_{y \epsilon B}\{||x, y||\}\} \tag{17}$$

where $||x, y||$ can be any norm, including the Euclidean distance. Note that $H(A, B) \neq H(B, A)$, in other words, DH is not symmetric.

The point set A is represented by one of the two charge curves namely, CCCV-CCCT or CVCC-CVCT, whereas B is represented by a line of 30-40 points as shown in 9 and 10.

Frechet distance of two curves A , B has been generally described as the minimal length of a leash required to connect a dog to its owner, as they walk along A or B, respectively, without backtracking. In contrast to distance notions such as the Hausdorff distance, it takes into account the order of the points along the curve, and thus better captures the similarity as perceived by human observers.[77] In mathematical terms, however, the Frechet distance between two curves is defined as:

$$FD(A, B) = min\{max||A(\alpha(t)), B(\beta(t))||\} \tag{18}$$

where $\alpha(t)$ and $\beta(t)$, range over continuous and increasing functions with $\alpha, \beta, t \epsilon [0, 1]$. Again, $||...||$ can be any norm, including Euclidian distance. A more elaborate mathematical explanation is beyond the scope of the present material, however, a thorough mathematical explanation can be found in [78]

The entropy of CVCC-CVCT and CCCV-CCCT curves is also considered as a feature. In information theory, entropy is the average rate at which information is produced by a stochastic source of data [79], whereas in statistical mechanics, entropy is an extensive property of a thermodynamic system. Thermodynamic property of curves has been thoroughly

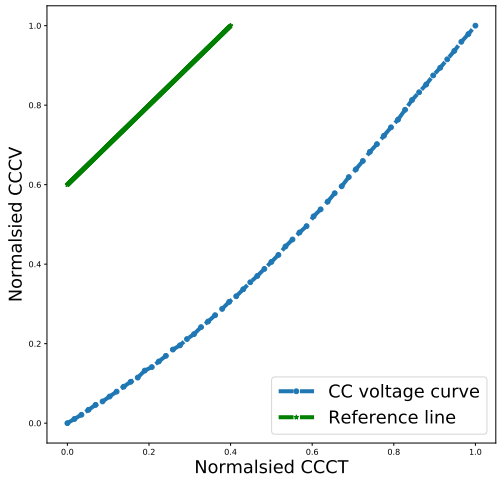

Figure 9: Typical constant current (CC) charge curve with associated reference line of equation $y = mx + c$

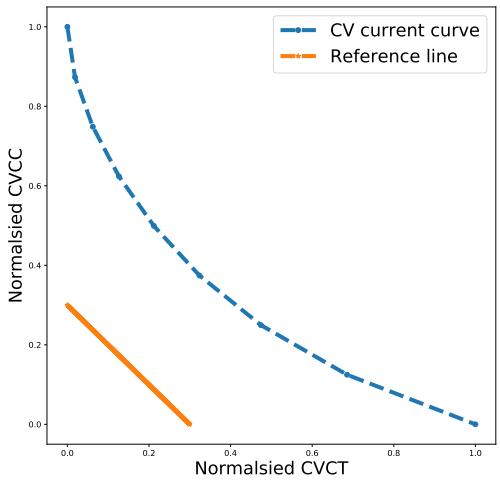

Figure 10: Typical constant voltage (CV) charge curve with associated reference line of equation $y = mx + c$

analysed in [80], [81], [82]. Authors in [82] provide an algorithmic procedure to compute curve entropy, and it has been adopted here with slight modification as follows. Curve entropy (EC) is defined by:

$$EC = \frac{\log_2 \left(\frac{2L}{D}\right)}{\log_2 \left(N - 1\right)} \tag{19}$$

where L is the length of the plane curve, D is the diameter of the smallest hypersphere covering the curve, and $N - 1$ is the number of segments approximating the line. For a thorough mathematical explanation on how all variables have been calculated refer to reference [82].

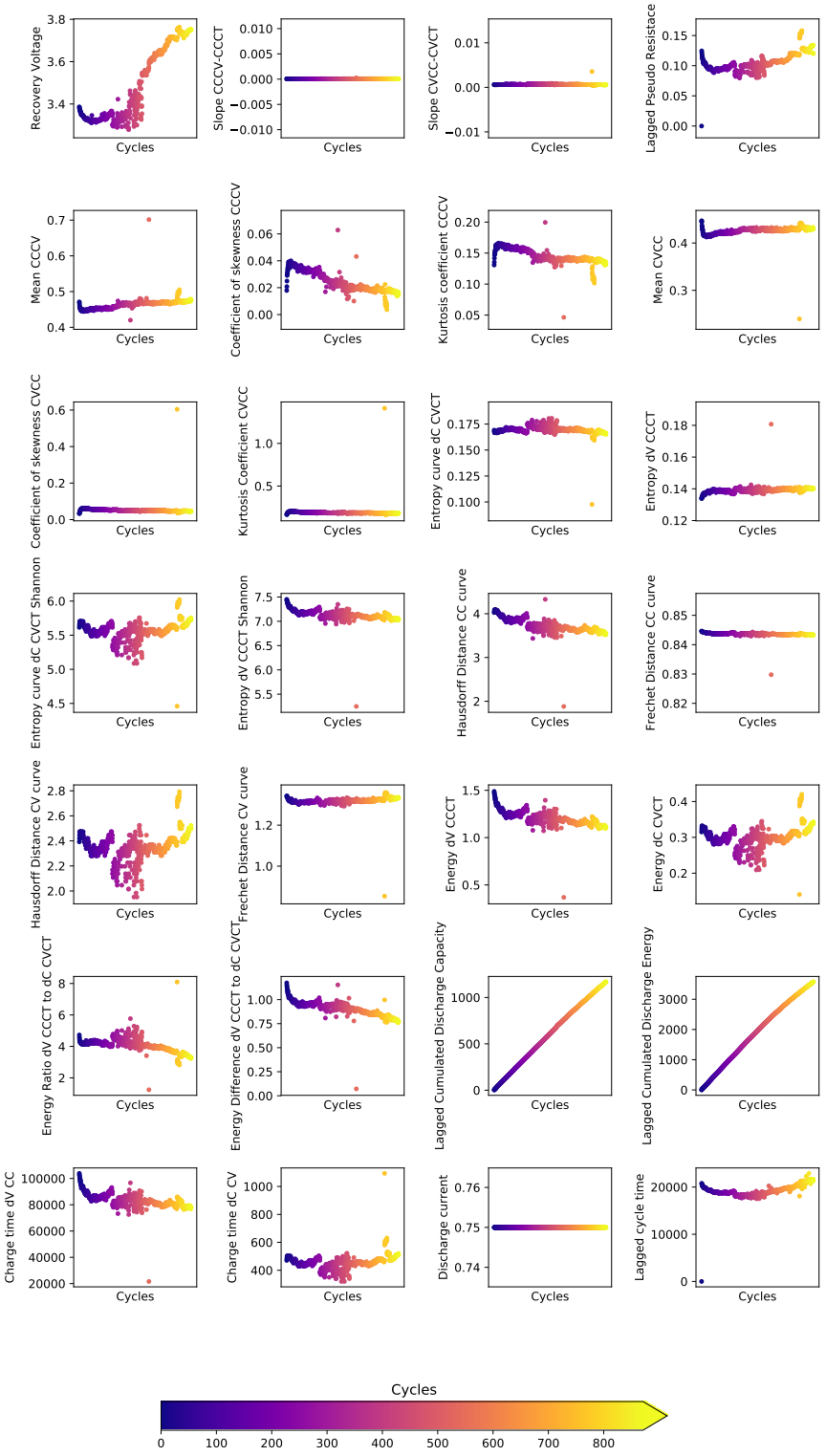

Figure 11: Example visualisation of derived features for Group I datasets cell no. 11.

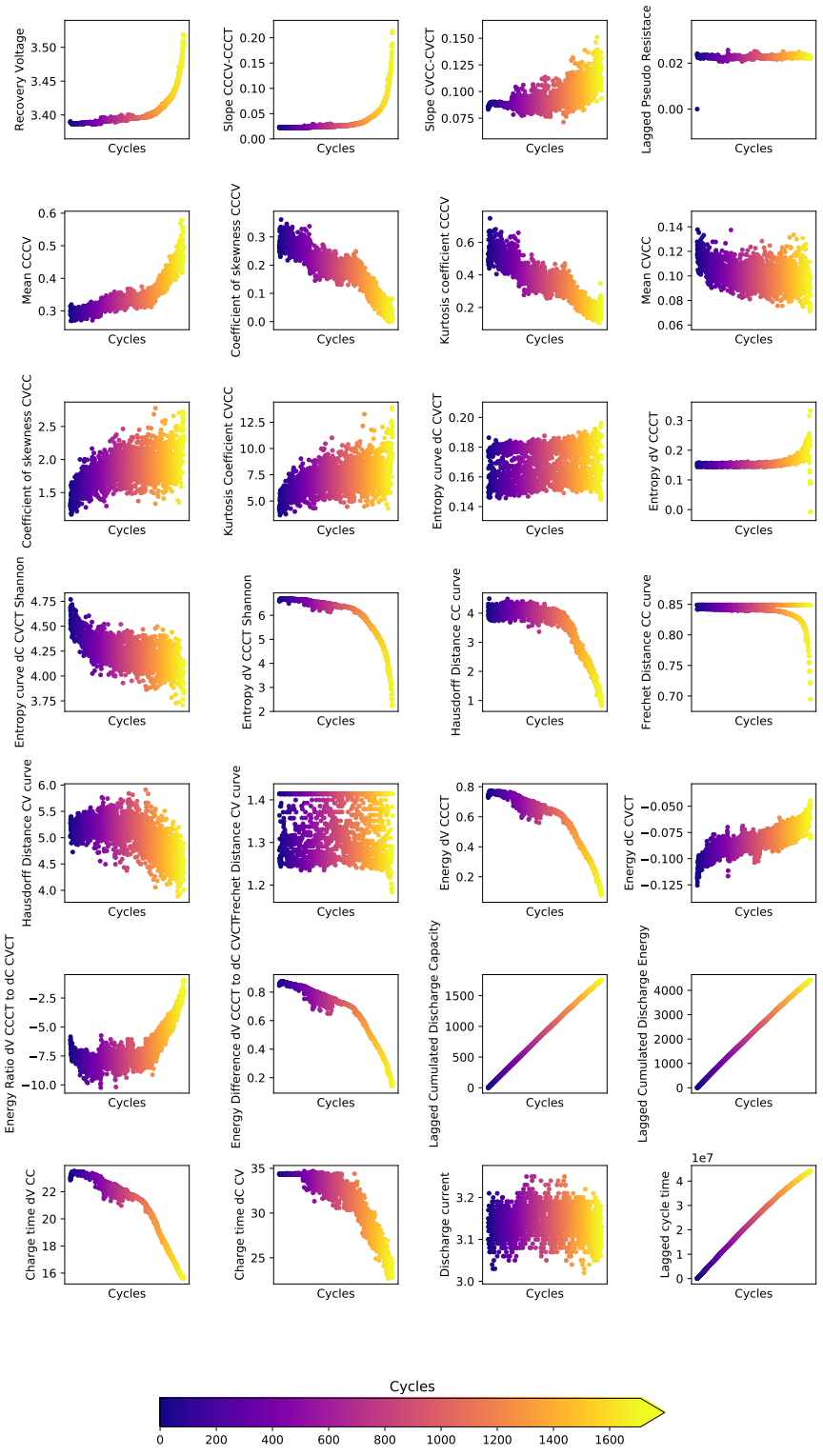

Figure 12: Example visualisation of derived features for Group 2 datasets cell no. 1.

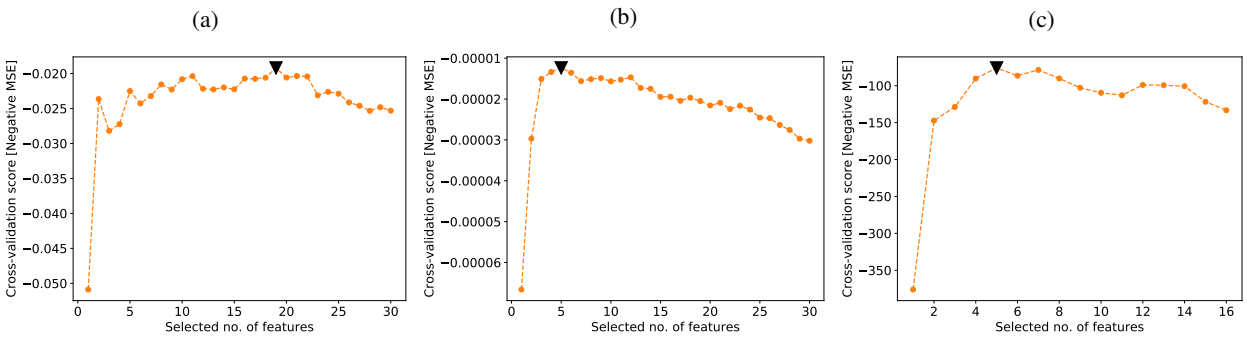

Figure 13: **Automatic feature selection with RF-RFE-CV - Note: black triangle indicates selected no. of features. a** 18 features selected for Group I. **b** 5 features selected for Group 2, **c** 5 features selected for Group 3.

Data type	Feature	Feature no.
Battery specific data	Nominal Capacity [Ah]	1
	Charge Current [A]	2
Cumulative (historical) data	Cumulated Discharge Capacity [Ah]	3
	Cumulated Discharge Energy [Wh]	4
1 Cycle Lagged Data	Lagged Cycle Time [s]	5
	Terminal Voltage @ Start of charge [V]	6
Instantaneous Charge Data	Charge time of CC segment of charge curve [s]	7
	Charge time of CV segment of charge curve [s]	8
	Mean current during CC segment of the curve [A]	9
	Slope of CCCV-CCCT segment of the curve	10
	Slope of CVCC-CVCT segment of the curve	11
	Energy during CCCV-CCCT segment of the curve [Wh]	12
	Energy during CVCC-CVCT segment of the curve [Wh]	13
	Energy ratio CCCV-CCCT / CVCC-CVCT segment of the curve	14
	Energy Difference between curve segments (CCCV-CCCT) - (CVCC-CVCT)	15
	Entropy of CCCV-CCCT segment of the curve based on \ref{}	16
	Shannon entropy of CCCV segment of the curve	17
	Frechet Distance of CCCV-CCCT segment of the curve	18

Table 10: Selected features using RF-RFE-CV for Group I. Note: CC = consatnt current, CV = constant voltage, CCCV = constant current charge voltage, CVCC = contant voltage charge current, CCCT = constant current charge time, CVCT = constant voltage charge time

Data type	Feature	Feature no.
Instantaneous charge data	Energy during CCCV-CCCT segment of the curve [Wh]	1
	Energy Difference between curve segments (CCCV-CCCT) - (CVCC-CVCT)	2
	Hausdorff Distance of CCCV-CCCT segment of the curve	4
	Shannon entropy of CCCV segment of the curve	3
	Frechet Distance of CCCV-CCCT segment of the curve	5

Table 11: Selected features using RF-RFE-CV for Group 2. Note: CC = consatnt current, CV = constant voltage, CCCV = constant current charge voltage, CVCC = contant voltage charge current, CCCT = constant current charge time, CVCT = constant voltage charge time

Supplementary Note 4. Data overview

Irrespective of dataset, input data consistency is ensured by removing outliers in the training data, possibly introduced due to inherent cell variability and measurement errors. The data preprocessing step involves filtering of the raw data based on erroneous capacity measurements by utilizing Random Sample Consensus (RANSAC) algorithm [83]. Training data that contains a significant percentage of gross errors in capacity from one cycle to another is removed as illustrated in the examples of Supplementary Figure 15. Note, test data has not been processed for outliers to simulate a realistic deployment scenario.

Data type	Feature	Feature no.
Cumulative (historical) data	Cumulated Discharge Capacity [Ah]	1
	Cumulated Discharge Energy [Wh]	2
1 Cycle Lagged Data	Lagged Cycle Time [s]	3
Instantaneous Charge Data	Capacity during CCCV-CCCT segment of the curve [Ah]	4
	Energy during CCCV-CCCT segment of the curve [Wh]	5

Table 12: Selected features using RF-RFE-CV for Group 3. Note: CC = consatnt current, CV = constant voltage, CCCV = constant current charge voltage, CVCC = contant voltage charge current, CCCT = constant current charge time, CVCT = constant voltage charge time

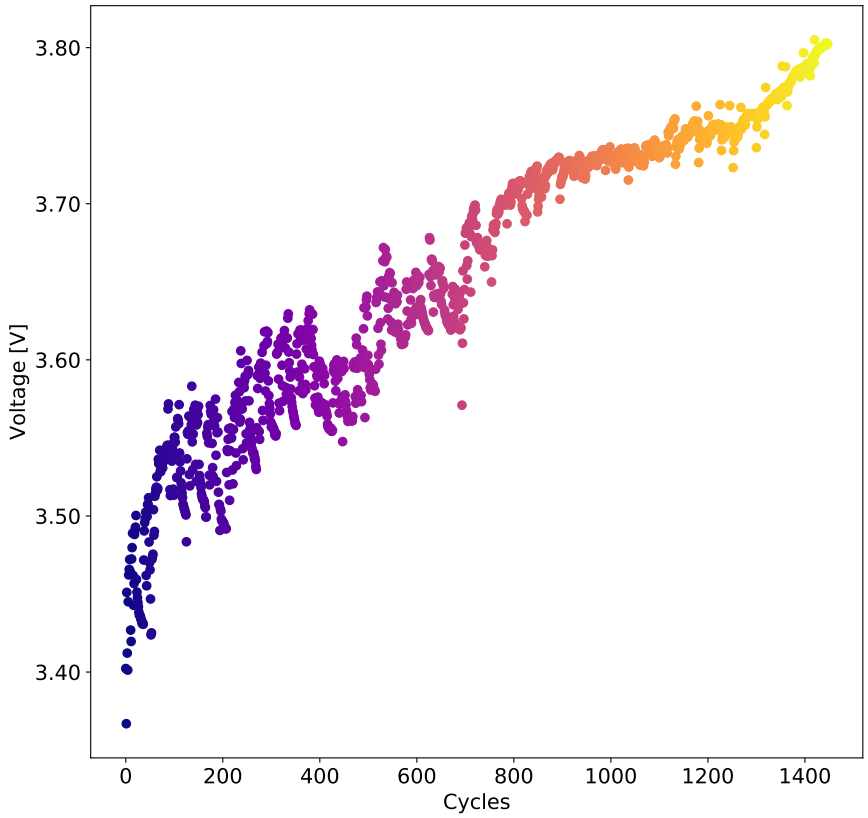

Figure 14: Increase in start of charge voltage between test cycles for a $LiCoO_2$ prismatic battery. The cell underwent a full depth of discharge at a current value of 1 C-rate with constant current - constant voltage charging.

CALCE dataset

Data sourced from CALCE battery group consists of three batteries. For ease of reference, we preserve the original dataset names as per their website `https://web.calce.umd.edu/batteries/data.htm`. All cells in the dataset underwent the same charging profile, the standard CC-CV. The CC phase of charging profile includes a 0.5 C-rate charging current until the voltage reached the cut-off threshold value of 4.2V. The CV top-up phase sustained the previously reached 4.2V until the current dropped to a value of 0.05 C-rate, at which point the charging is complete. Except for batteries in CALCE PL dataset, which were discharged at 1 C-rate, the other two datasets have been discharged at both 0.5 C-rate and 1 C-rate until the battery voltage reached the pre-defined discharge cut-off voltage of 2.7V. A schematic of the charge profile together with a detailed summary of discharge conditions for each battery can be found in Figure 8 and in Supplementary Table 13, respectively.

NASA dataset

NASA data can be retrieved from the public NASA Ames Prognostics Centre of Excellence website `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/` and includes two datasets. The first repository, the battery dataset denoted here by NASA5, includes a mixture of constant discharge current and squared

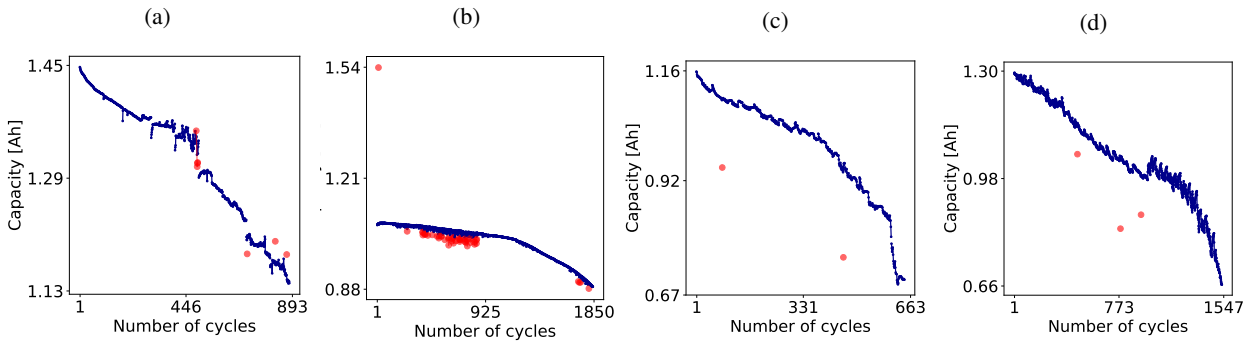

Figure 15: **Training data outlier removal with RANSAC (red denotes outliers, blue denotes inlier). a** Cylindrical A123 LFP/graphite training cell 11 Group II. **b** Pouch LCO cell 2 Group I, **c** Prismatic CS2 LCO training cell 34 Group I, **d** Prismatic CX2 LCO training cell 34 Group I.

wave-based discharge current experiments at different temperatures. The second repository, the randomised battery usage dataset, denoted here by NASA11, includes batteries that are continuously cycled with randomly generated current profiles. The randomised nature of the load profiles is an ideal representation of practical battery usage. Both NASA5 and NASA11 dataset follow the traditional CC-CV charge protocol. CC charging mode was carried at 1.5A until the battery voltage reached 4.2V and then continued in a CV fashion until the charge current dropped to 0.02A, at which point the battery was deemed fully charged. In terms of discharge, NASA5 discharge was carried out at a constant current level of 2A or square wave loading profile of 4A until the battery voltage fell to 2.7V, 2.5V or 2.2V. Whereas, NASA11 undergone a randomised discharge profile of varying duration ranging from 5 minutes to 3 hours as well as varying discharge current values ranging from 0A to 5A. All cells underwent a periodic characterisation test whereby a 2A CC and 0.02A CV current cut-off charge protocol and a 2A constant current discharge was applied. The characterisation test data was used in BHUMP to evaluate battery health as a function of capacity, as opposed to cyclic data. Details of charging and discharging profiles per battery are found in Supplementary Table 14.

TRI dataset

The work supported by Toyota Research Institute in partnership with MIT and Stanford generated a lifecycle battery dataset consisting of 124 cells, available at `https://data.matr.io/1/projects/5c48dd2bc625d700019f3204`. The dataset was used in [14] where more details on battery type, manufacturer and testing equipment can be found. All cells in the dataset were cycled with a total of 72 different fast charging-polices but identically discharged with a current 4 C-rate between 3.6V and 2.0V. The charging protocol included a two-step fast charge between 0% to 80% SOC. The fast charge section is followed by a CC protocol, i.e. a uniform charge current value of 1 C-rate to 3.6V until the voltage reaches the cut-off value of 3.6V, immediately followed by a CV top-up phase until current dropped to 0.02 C-rate. The raw data from each cycle is used as input to BHUMP pipeline. Details regarding the charge profile as well as the cycling regimes for each battery can be found in [14], whilst Figure 7 illustrates the charging protocol and Supplementary Table 16 indicates which cells have been used for training and testing of the algorithms.

Oxford dataset

The Oxford Battery Degradation Dataset and can be accessed at `https://ora.ox.ac.uk/objects/uuid:03ba4b01-cfed-46d3-9b1a-7d4a7bdf6fac`. A comprehensive explanation of the testing method, equipment and battery specific characteristic is found in [84]. The data consists of ageing experiments by repeatedly cycling the cells via a CC charge profile coupled with the ARTEMIS urban drive cycle discharge profile. The CC charge protocol uses a 2 C-rate current to a voltage of 4.2V. The discharge profile voltage range is 4.2V to 2.7V. After every 100 cycles of repeated charge-discharge using the protocol mentioned above, a characterisation test (incorporating a full constant current charge-discharge at C/18.5 (40 mA), repeated every 100 drive cycles.) is carried out. The characterisation test data is used in this work for battery health degradation estimation purposes. Supplementary Table 17 indicates which cells have been used for training/testing the algorithms.

Supplementary Note 5. Data partitioning

Group I

Out of the 47 cells in Group I, we use 23 cells for training (out of this 10 are used for feature selection), 5 cells for calibration and remaining 19 for evaluating the algorithm performance (the cells used during training-testing can be

found in Supplementary Tables 13, 14, 15). Note that the calibration dataset is neither used in training nor testing to prevent overfitting.

Group II

Group II dataset is randomly split into 63 cells for training (out of which 37 cells are used for feature selection), 10 for calibration and the remainder 51 cells for testing, refer to Supplementary Table 16 for cell partition in each dataset.

Group III

Group III dataset is split into 3 cells for training (cells 1 to 3), one cell for calibration (cell no. 4), and the remainder of 4 cells for testing (see Supplementary Table 17 for details).

Cell name	Discharge condition	Dataset
CS2 - 33	0.5 C-rate	Test
CS2 - 34	0.5 C-rate	Train
CS2 - 35	1 C-rate	Train & Feature Selection
CS2 - 36	1 C-rate	Train & Feature Selection
CS2 - 37	1 C-rate	Calibration
CS2 - 38	1 C-rate	Test
CX2 - 33	0.5 C-rate	Test
CX2 - 34	0.5 C-rate	Train
CX2 - 35	0.5 C-rate	Train & Feature Selection
CX2 - 36	0.5 C-rate	Calibration
CX2 - 37	0.5 C-rate	Train & Feature Selection
CX2 - 38	0.5 C-rate	Test
PL - 11	0.5 C-rate	Train
PL - 13	0.5 C-rate	Test

Table 13: Group I: CALCE battery data discharge conditions and train, calibration and test split. For complete details on test conditions access `https://web.calce.umd.edu/batteries/data.htm`.

Cell name	Discharge condition	Dataset
B0005	2A	Train & Feature Selection
B0006	2A	Test
B0007	2A	Train
B0018	2A	Test
B0025	Square wave @ 4A	Test
B0026	Square wave @ 4A	Train & Feature Selection
B0027	Square wave @ 4A	Train
B0028	Square wave @ 4A	Calibration

Table 14: Group I: NASA 5 battery data discharge conditions and train, calibration and test split. For complete details on test conditions access `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/`.

Cell name	Discharge condition	Dataset
RW1	Random Sequence	Train & Feature Selection
RW2	Random Sequence	Train
RW3	Random Sequence	Train
RW4	Random Sequence	Train
RW5	Random Sequence	Test
RW6	Random Sequence	Test
RW7	Random Sequence	Test
RW8	Random Sequence	Test
RW9	Random Sequence	Train & Feature Selection
RW10	Random Sequence	Train
RW11	Random Sequence	Calibration
RW12	Random Sequence	Test
RW13	Random Sequence	Train
RW14	Random Sequence	Train
RW15	Random Sequence	Test
RW16	Random Sequence	Test
RW20	Random Sequence	Train & Feature Selection
RW21	Random Sequence	Train & Feature Selection
RW22	Random Sequence	Train
RW23	Random Sequence	Test
RW24	Random Sequence	Test
RW25	Random Sequence	Train & Feature Selection
RW26	Random Sequence	Train
RW27	Random Sequence	Test
RW28	Random Sequence	Calibration

Table 15: Group I: NASA 11 battery data discharge conditions and train, calibration and test split. Note: batteries are discharged to 3.2V using a randomized sequence of discharging loads between 0.5A and 4A. For complete details on test conditions access `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/`.

Dataset	Cell number	Number of cells
Feature Selection	2, 6, 8, 14, 18, 19, 26, 28, 32, 35, 37, 45, 51, 53, 55, 58, 60, 61, 65, 69, 72, 76, 79, 83, 90, 91, 92, 103, 107, 109, 110, 113, 115, 116, 119, 120, 124	37
Training	2, 3, 6, 8, 9, 13, 14, 16, 18, 19, 20, 21, 23, 25, 26, 28, 32, 35, 37, 42, 45, 46, 50, 51, 53, 55, 56, 58, 60, 61, 63, 64, 65, 66, 69, 72, 73, 76, 79, 83, 84, 86, 88, 90, 91, 92, 94, 95, 98, 100, 103, 105, 106, 107, 109, 110, 113, 115, 116, 118, 119, 120, 124	63
Calibration	7, 12, 22, 48, 54, 59, 68, 77, 82, 108	10
Testing	1, 4, 5, 10, 11, 15, 17, 24, 27, 29, 30, 31, 33, 34, 36, 38, 39, 40, 41, 43, 44, 47, 49, 52, 57, 62, 67, 70, 71, 74, 75, 78, 80, 81, 85, 87, 89, 93, 96, 97, 99, 101, 102, 104, 111, 112, 114, 117, 121, 122, 123	51

Table 16: Group II: TRI dataset splitting for: feature selection, training, calibration and testing. For complete details on test conditions access `https://data.matr.io/1/projects/5c48dd2bc625d700019f3204`.

Dataset	Cell number	Total number of cells
Feature Selection	1, 3	2
Training	1, 2, 3	3
Calibration	4	1
Testing	5, 6, 7, 8	4

Table 17: Group III: Oxford dataset splitting for: feature selection, training, calibration and testing. For complete details on test conditions access `https://ora.ox.ac.uk/objects/uuid:03ba4b01-cfed-46d3-9b1a-7d4a7bdf6fac`.

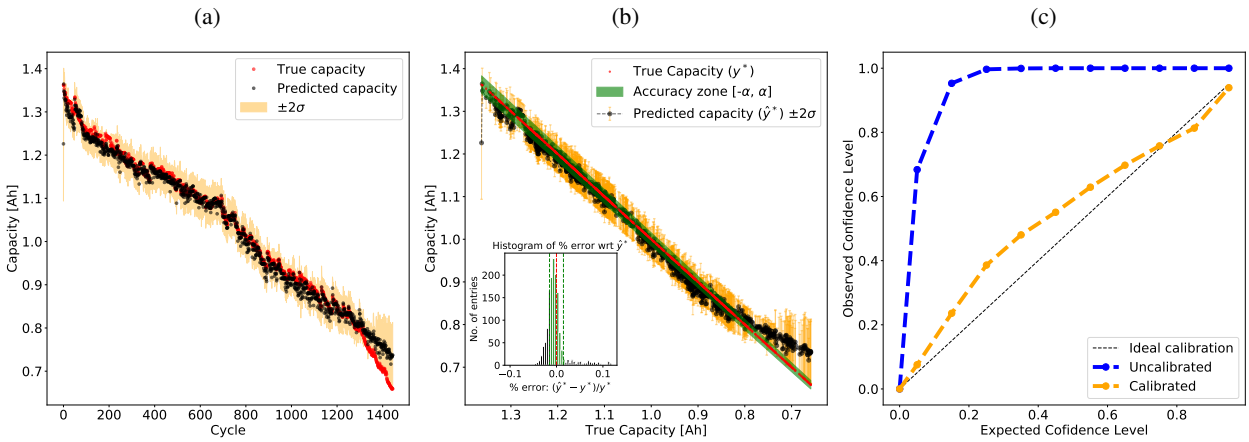

Figure 16: **Prediction results with BRR Group I cell no. 38. a** Prediction as a function of cycle numbers, **b** Actual vs. predicted capacity, **c** Calibration results.

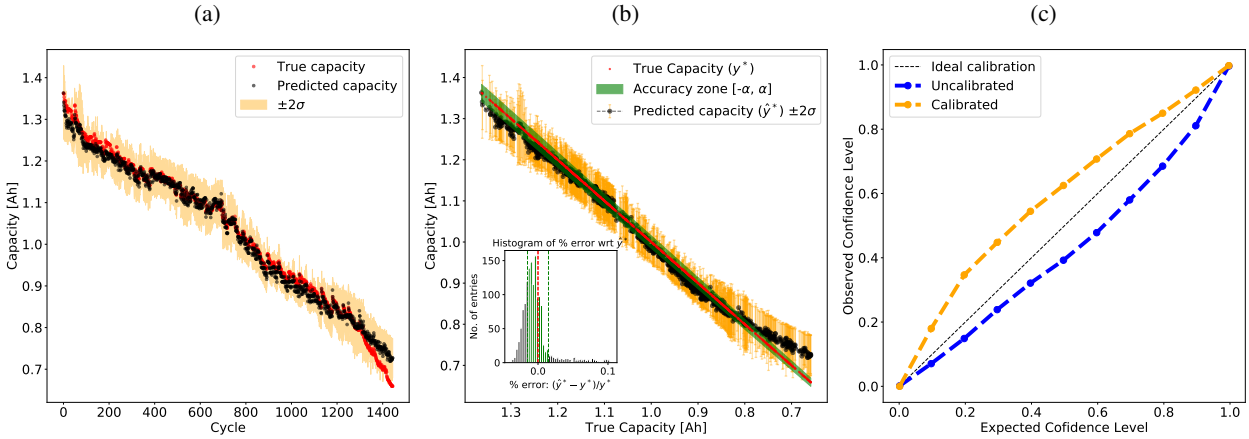

Figure 17: **Prediction results with GPR Group I cell no. 38. a** GPR prediction as a function of cycle numbers, **b** GPR actual vs. predicted capacity, **c** GPR calibration results.

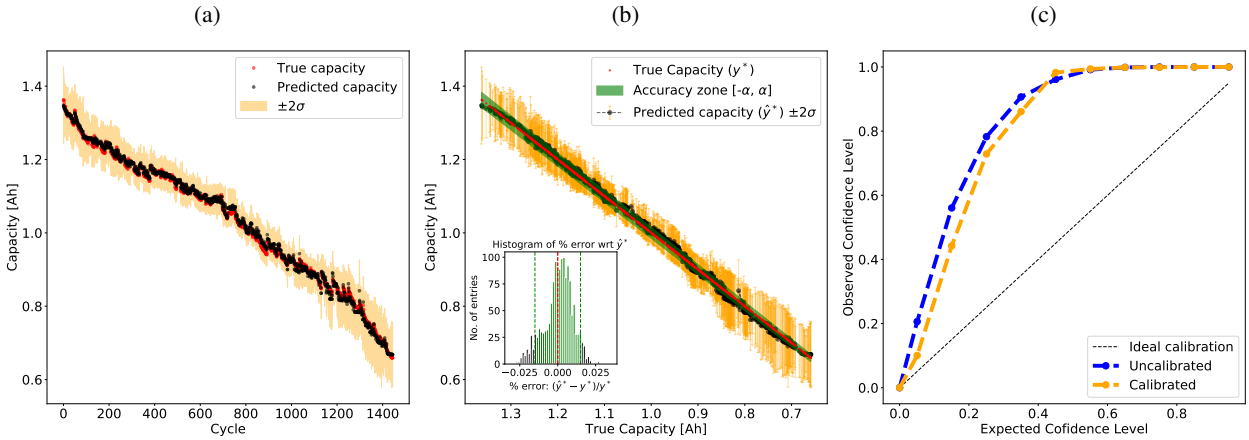

Figure 18: **Prediction results with RF Group I cell no. 38. a** RF prediction as a function of cycle numbers, **b** RF actual vs. predicted capacity, **c** RF calibration results.

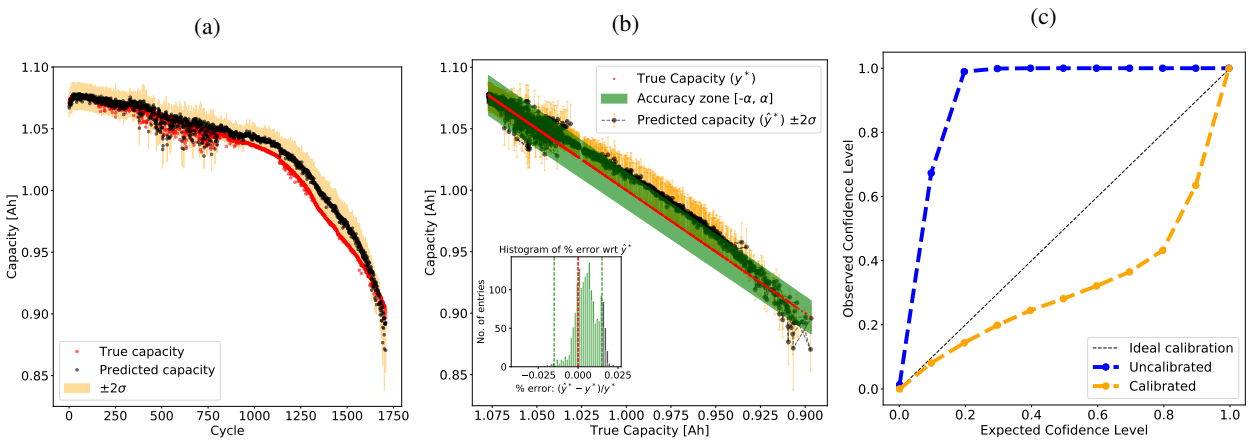

Figure 19: **Prediction results with BRR Group II cell no. 1. a** BRR prediction as a function of cycle numbers, **b** BRR actual vs. predicted capacity, **c** BRR calibration results.

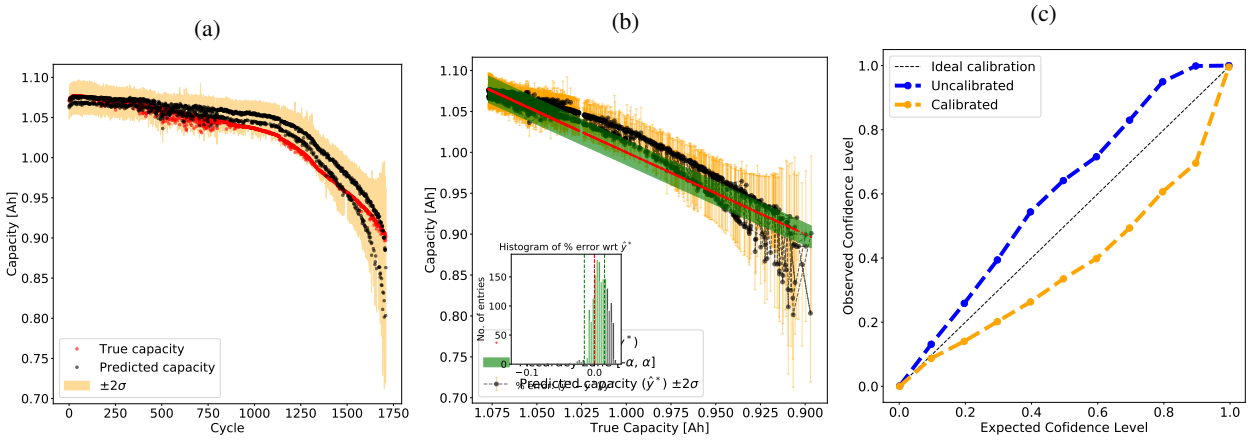

Figure 20: **Prediction results with GPR Group II cell no. 1. a** GPR prediction as a function of cycle numbers, **b** GPR actual vs. predicted capacity, **c** GPR calibration results.

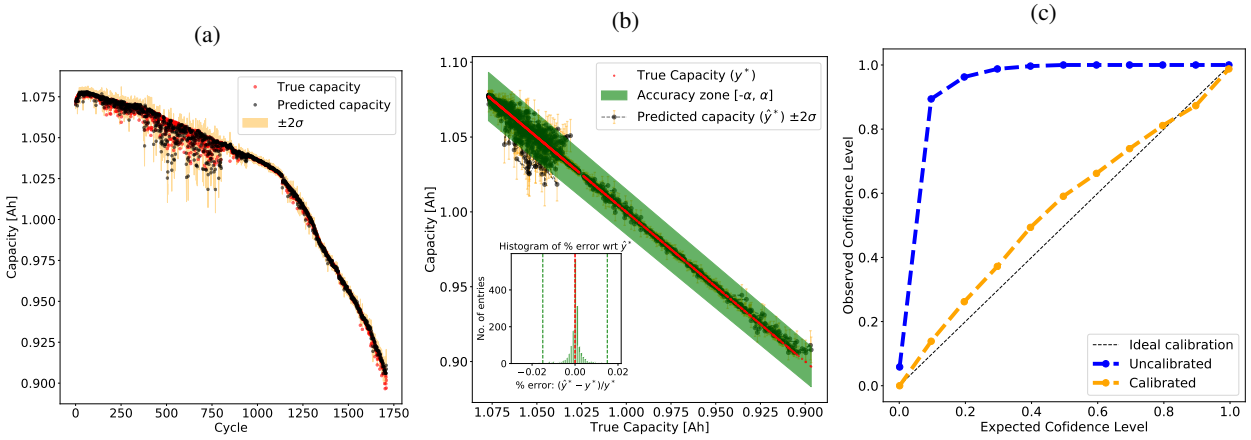

Figure 21: **Prediction results with RF Group II cell no. 1. a** RF prediction as a function of cycle numbers, **b** RF actual vs. predicted capacity, **c** RF calibration results.

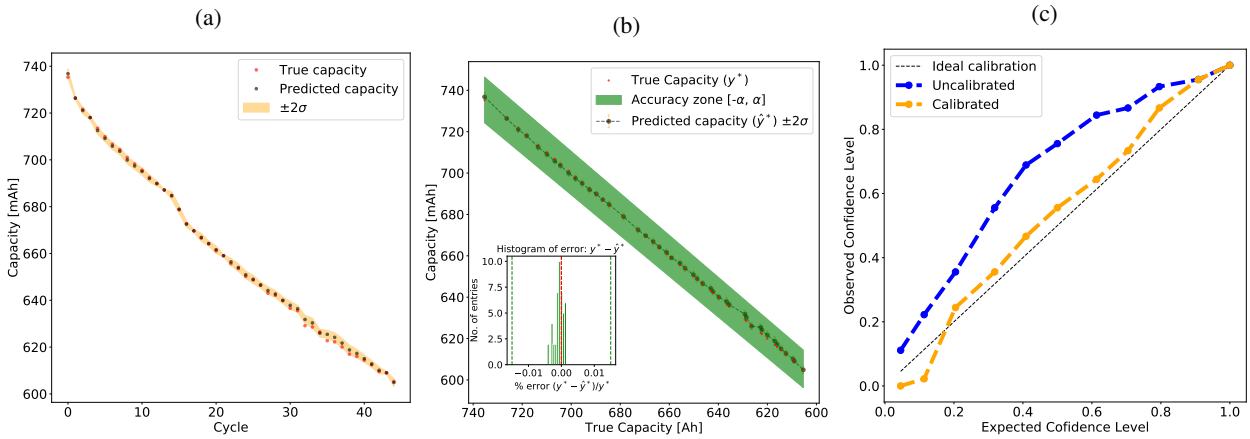

Figure 22: **Prediction results with BRR Group III cell no. 5. a** BRR prediction as a function of cycle numbers, **b** BRR actual vs. predicted capacity, **c** BRR calibration results.

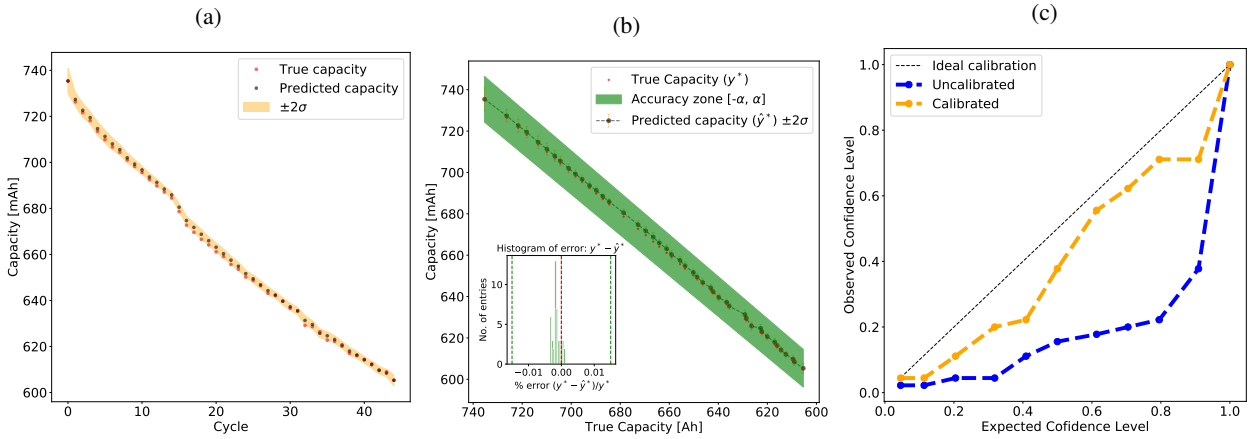

Figure 23: **Prediction results with GPR Group III cell no. 5. a** GPR prediction as a function of cycle numbers, **b** GPR actual vs. predicted capacity, **c** GPR calibration results.

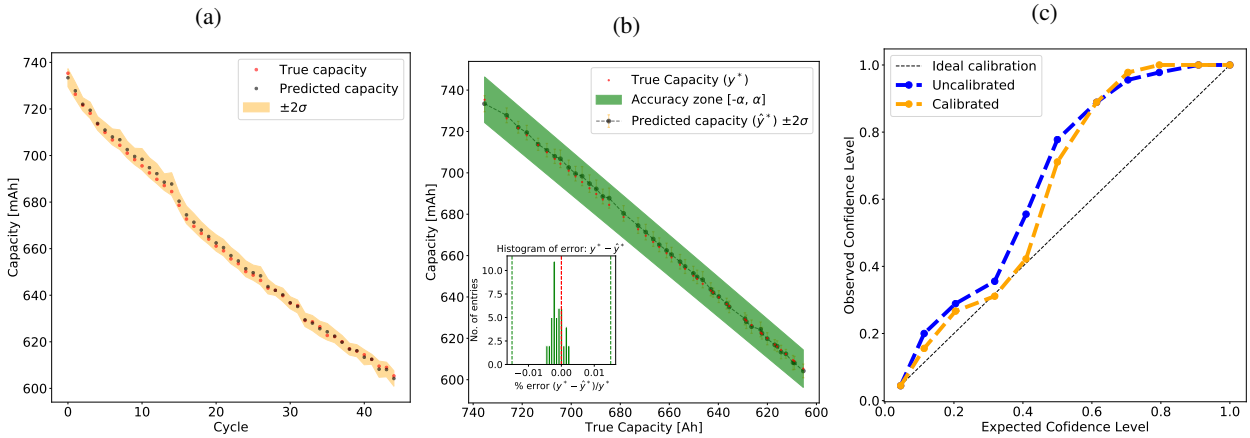

Figure 24: **Prediction results with RF Group III cell no. 5. a** RF prediction as a function of cycle numbers, **b** RF actual vs. predicted capacity, **c** RF calibration results.