nature
machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-021-00312-3

Machine learning pipeline for battery state-of-health estimation

Darius Roman[1]✉, Saurabh Saxena[2,4], Valentin Robu[1,3,5], Michael Pecht[2] and David Flynn[1]

Lithium-ion batteries are ubiquitous in 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 Article, 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 error of 0.45%. This work provides insights into the design of scalable data-driven models for battery SOH estimation, emphasizing the value of confidence bounds around the prediction. The pipeline methodology combines experimental data with machine learning modelling and could be applied to other critical components that require real-time estimation of SOH.

Rechargeable Li-ion batteries are crucial in many 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% in the period 2010 to 2016, to an all-time low of US$273 per kWh in 2017[1], opened up a substantial energy storage market evaluated at US$65 billion in 2017 (ref. [2]). Whatever 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 SOH can be used to predict a battery's expected lifetime, but the feasibility of online SOH estimation via direct measurement of chemical reaction parameters inside batteries remains limited[4].

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's capacity as its health indicator owing 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 involve parameter estimation using one of the following modelling types: equivalent circuit models[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 modelling requires detailed cell specifications, such as electrode materials and electrolyte chemistry. The method typically deploys complex partial differential equations, leading to substantial requirements of both memory and computational power. Equivalent circuit models, on the other hand, employ circuit components with empirical nonlinear parameters[9]. Compared to electrochemical models, equivalent circuit models use fewer inputs, considerably reducing the number of parameters required to be learned over time, but they have limited accuracy and robustness owing to assumptions in battery behaviour[8]. Furthermore, to determine equivalent circuit 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 has a series of advantages such as chemistry-agnostic modelling capability and the ability to analyse a wide range of degradation mechanisms and operating conditions, including rare loading events that are often overlooked by simplified models or physics-based simulations. Until now, numerous studies have employed machine learning tools for the analysis of battery SOH estimation. Several studies[23,24,25,26,27] showed that the use of incremental capacity and differential voltage curves, a method developed for the analysis of cell ageing mechanisms[23], can also be used for offline and online capacity fade estimation. However, the approach has several drawbacks, linked to obtaining the incremental capacity and differential voltage curves, that substantially reduce its practicality. The differentiation of the capacity–voltage curve to obtain the incremental capacity curve amplifies noise and propagates it into the algorithm. Additionally, both curves must cover a sufficient voltage range for the method to work and, to obtain 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 the charging protocol and the key part of the capacity–voltage curve is recorded,

[1]Smart Systems Group, School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh, UK. [2]Center for Advanced Life Cycle Engineering, University of Maryland, College Park, MD, USA. [3]Algorithmics Group, TU Delft, Delft, the Netherlands. [4]Present address: Argonne National Laboratory, Lemont, USA. [5]Present address: Intelligent and Autonomous Systems Group, CWI, National Research Centre for Mathematics and Computer Science, Amsterdam, the Netherlands. ✉e-mail: dvr1@hw.ac.uk

such that specific peak points in the incremental capacity 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 using equispaced voltage values as the input to a GPR algorithm. However, GPR is very slow to train owing to its computational cost of learning, which is 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[34], have also been used to estimate battery capacity fade. Unfortunately, the relevance vector machine also suffers from slow training, particularly when compared to frequentist-based algorithms[35]. Shen et al.[33] presented options for accelerating GPR, but they compromise accuracy. In contrast to ref. [32], where the CC part of the charging profile was used, Wang et al.[36] used the constant-voltage section to estimate capacity fade using support vector regression. Although support vector regression is faster than GPR, it lacks the ability to estimate uncertainty. This inability to estimate uncertainty stemming 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 a 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 the 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 on the basis of calibration error and an adapted accuracy measure, the $\alpha-\beta$ accuracy zone. There are numerous battery designs[38] and chemistries available[39], so we deployed the pipeline on a total of 179 cells, three designs (prismatic, pouch and cylindrical), two chemistries ($LiFePO_4$ and $LiCoO_2$), three charge protocols (CC, CC−CV and the 2-step fast-charging protocol), 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 machine learning 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 ages, uses a set of engineered features capable of capturing battery intrinsic degradation, and is capable of estimating cell SOH in under 15 min at any point in its life cycle. An accurate SOH method combined with a quantifiable metric for uncertainty propagation that feeds into state of charge (SOC) and run-time calculations improves battery performance and ultimately extends cell lifetime.

Machine learning pipeline approach

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 assumption that the battery cell has unknown capacity. Supplementary Fig. 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 the 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 summarizes the attributes recorded during life-cycle testing. The pipeline focuses on segments of the charge curves to capture degradation in the electrodes during cycling (Fig. 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 ref. [44]. Recursive feature elimination generally outperforms other conventional methods[45,46], hence our adoption here (Methods). 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 (Methods). The use of adversarial examples in our datasets was motivated by the need to ameliorate the differences in battery design or 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 GPR. Unlike the Bayesian-based algorithms BRR and GPR, frequentist algorithms are unable to quantify uncertainty naturally owing 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 ref. [50]. To train the algorithms a random search approach is used for hyper-parameter tuning[51], with the exception of the deep ensemble where the Adam optimizer 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 initialization 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 an uncertainty quantification perspective. The models are initially re-calibrated followed by an evaluation based on mean absolute percentage error, root-mean-squared error and uncertainty estimation metrics (see Methods for further details).

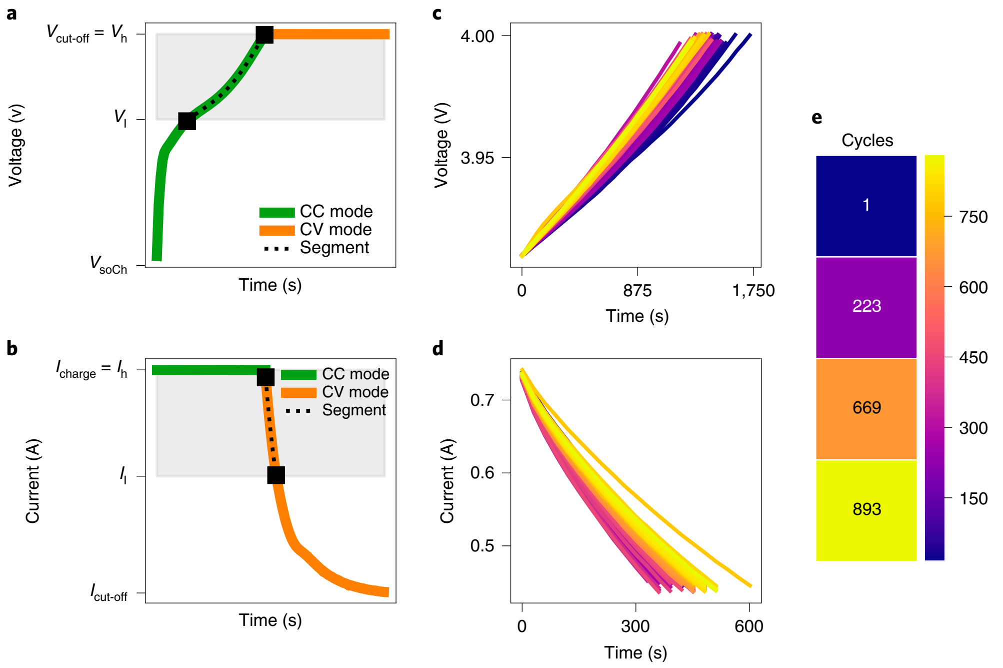

Fig. 1 | The constant current—constant voltage (CC—CV) charge protocol and extracted ageing segment of the curves for a Li-ion pouch cell. a, Voltage during charge protocol. $V_{cut-off}$, cut-off voltage; V_h, upper voltage threshold; V_l, lower voltage threshold; V_{soCH}, start of charge voltage. **b**, Current during charge protocol. I_{charge}, charge current; I_h, upper current threshold; I_l, lower current threshold; $I_{cut-off}$, cut-off current. **c**, Extracted ageing voltage curve segments corresponding to marked grey area in **a**. **d**, Extracted ageing current curve segments corresponding to marked grey area in **b**. **e**, Heatmap of ageing with cycle number. See Methods for definitions of symbols.

Table 1 | Overview of datasets

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
Number of cells	6	6	2	8	25	124	8
Charge	CC—CV	CC—CV	CC—CV	CC—CV	CC—CV	Fast-charging	CC
Discharge	2 regimes	2 regimes	1 regime	2 regimes	7 regimes	1 regime	1 regime

See Supplementary Note 4 for data sources. *Groups based on charge protocol; **Toyota Research Institute; ***Information from manufacturer, not verified.

Dataset

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

Algorithm performance

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 (Supplementary Fig. 8a and Supplementary Table 3). From a threshold point of view, we select an upper voltage threshold V_h of 4.2 V for all batteries in Group I with an associated lower voltage threshold V_l of 3.9 V. See Supplementary Note 5 for training/testing partitions.

We illustrate results for BHUMP when dNNe is considered to be the base algorithm in Fig. 2 (results for all other algorithms are shown in Supplementary Figs. 11, 12 and 13) for a randomly chosen pouch cell battery, cell number 38; we summarize algorithm performance on this cell in Table 2a. The cell was cycled in full depth of discharge between 4.2 V and 2.7 V at a discharge C-rate of 0.5 C (or 0.55 A) with a CC—CV charging protocol at a current value of 0.5 C-rate. Table 2a summarizes each algorithm's performance on cell number 38. Comparing dNNe in Fig. 2a to the other algorithms BRR, GPR and RF, we show that the resulting confidence interval is considerably smaller (all figures display a confidence level equivalent to a 95% quantile, that is $\mu \pm 2\sigma$, where μ is the predicted mean). This indicates that the model is sharper, resulting in a high β score (Table 2a). Where the predictions are less accurate, such as the prediction in the first few cycles (see Fig. 2a), the error bars capture this variability well. On this battery, dNNe also achieves the best mean

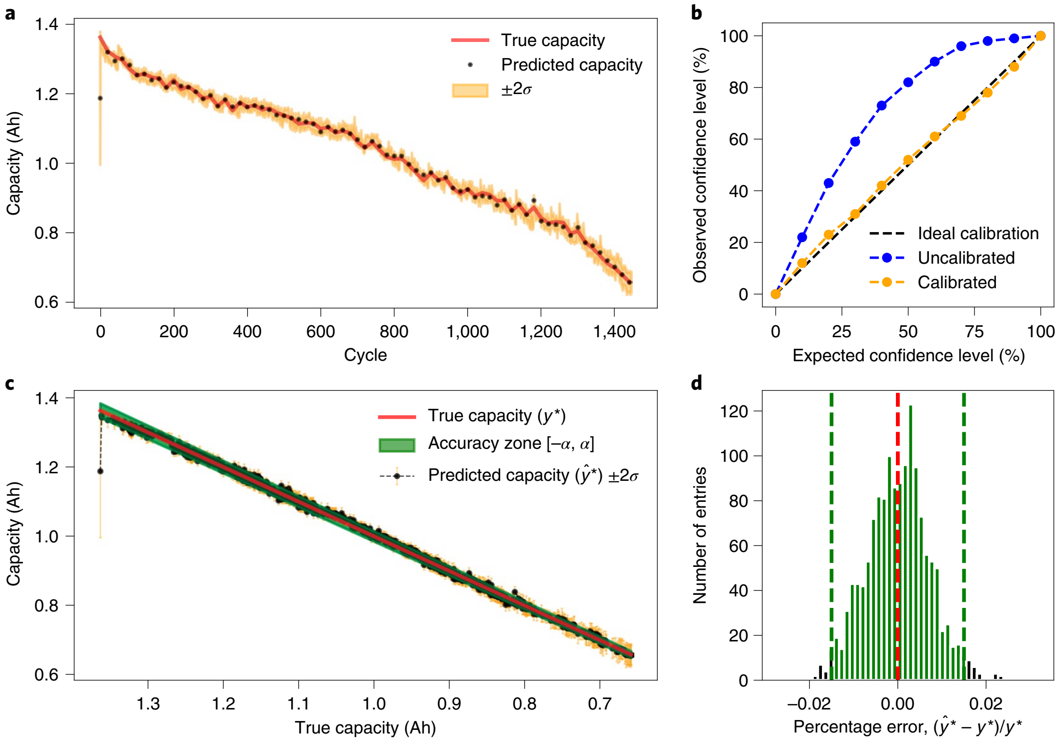

Fig. 2 | Prediction results with dNNe Group I cell number 38. a, dNNe prediction as a function of cycle. **b**, dNNE calibration results. **c**, dNNe actual versus predicted capacity. **d**, Histogram of percentage error.

Table 2 | Results for Group I cell no. 38 and average results over Group I dataset

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
a Results for Group I cell no. 38							
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
b Average results over Group I dataset							
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

absolute percentage error (MAPE) and root-mean-squared error (RMSPE) together with a high calibration score. As in Table 2a, the estimates for this cell vary between RMSPE 0.65% to 1.52%, showing that all four 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; see Supplementary Fig. 13.

When discussing average results across all cells in Group I (Table 2b), RF achieves, on average, a low calibration error of 54.70%, possibly owing to the method used for estimating the variance: infinitesimal jackknife (IJ). 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 (Fig. 3) should exceed the number of capacity values estimated above it; that is, PEP should be close to 100%. At the same time, too low a capacity estimate would result in a far too conservative algorithm. However, such behaviour is captured by an increase in RMSPE and thus is mitigated naturally. With reference to Fig. 2c, together with Table 2a, we can conclude that dNNe is conservative, achieving the highest PEP.

Overall, despite RF achieving the lowest average RMSPE and MAPE (Table 2b) it does not output well-calibrated predictions, nor does it display a high sharpness value. At the expense of 1.13% in MAPE and 1.56% in RMSPE, the dNNe outputs a well-calibrated model, with on average less than 4% below the ideal calibration score.

Group II data. The 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 four algorithms. Training is performed on features

Table 3 | Results for Group II cell no. 1 and average results over Group II dataset

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
a Results for Group II cell no. 1							
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
b Average results over Group II dataset							
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

engineered based on the CC–CV curve obtained after the cell reaches 80% SOC (refer to Supplementary Fig. 2a,b). See Supplementary Note 5 for training/testing partitions. RF-RFE-CV selects a total of five features (Supplementary Fig. 8b and Supplementary Table 4) out of a total of 30 engineered features. We believe this is caused by the fact that the dataset incorporates only one discharge profile as well as just a single battery type.

Figure 4 illustrates BHUMP performance with a dNNe as the base algorithm for cell number 1, whereas Supplementary Figs. 14, 15, 16 summarize results for all other algorithms. The cell has undergone a fast-charging profile of 3.6 C-rate to 80% SOC, beyond which the cell is charged with CC of 1 C followed by the CV charging. We selected cell 1 to illustrate the performance of the algorithms when there is a high number of outliers in capacity data (Fig. 4a). With reference to Table 3a, RF achieves the lowest error and the highest scores as well as a good calibration compared to all other algorithms. On this particular cell, dNNe achieves the second-best performance, but it does not output a well-calibrated model, despite showing a good average calibration score; see Table 3b.

Average results of the four algorithms are concisely summarized in Table 3b. 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-charging 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 summarized in Table 3b. 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%.

In conclusion, from an accuracy and sharpness perspective, the best-performing algorithm on dataset Group II is RF, whereas 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 the 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 (Table 3b).

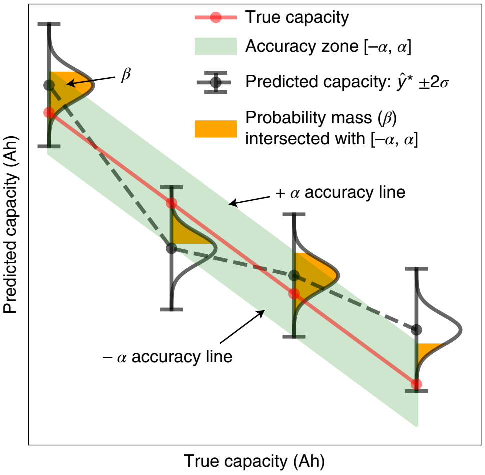

Fig. 3 | Illustration of the α-accuracy zone and probability mass β.

Group III data. For Group III we emphasize the suitability of BHUMP to battery SOH estimation for automotive applications. Group III includes 8 Kokham 740 mAh batteries that have been dynamically cycled under the ARTEMIS [52] dynamic driving profile, followed by characterization cycles. Each characterization cycle consists of low-rate CC charge and discharge cycles, repeated every 100 cycles. We use the characterization cycles for diagnostics purposes to derive features and estimate battery health. This dataset incorporates the lowest variability both in terms of input feature values and capacity degradation values owing to the identical charge/discharge conditions. This, in turn, affects feature selection because BHUMP selects only five of the 18 engineered features (note that the charge protocol does not include the CV part of the charge, hence 12 features are missing) as shown in Supplementary Fig. 8c and Supplementary Table 5. We keep the same threshold values as in the Group I cells for the CC part of the curves, namely a V_h of 4.2 V and a V_l of 3.9 V on which features are engineered. See Supplementary Note 5 for training/testing partitions.

For visualization purposes, we illustrate results for the randomly selected cell number 5 for dNNe in Fig. 5 and Supplementary Figs. 17, 18 and 19 for all other algorithms. It is clear from Table 4a that performance on cell 5 is dominated by BRR, by all measures of accuracy and uncertainty quantification. However, all algorithms deployed on cell number 5 (Table 4a) achieve a MAPE and RMSPE smaller than the proposed accuracy zone threshold α of ±1.5%.

Average results are summarized in Table 4b. In terms of accuracy measures, on average, BRR outperforms all other methods, including the dNNe. As argued in ref. [53], linear regression outperforms considerably more complex algorithms, including neural networks when dealing with small sample sizes exhibiting little variance. Despite the low error, BRR does not achieve a good calibration

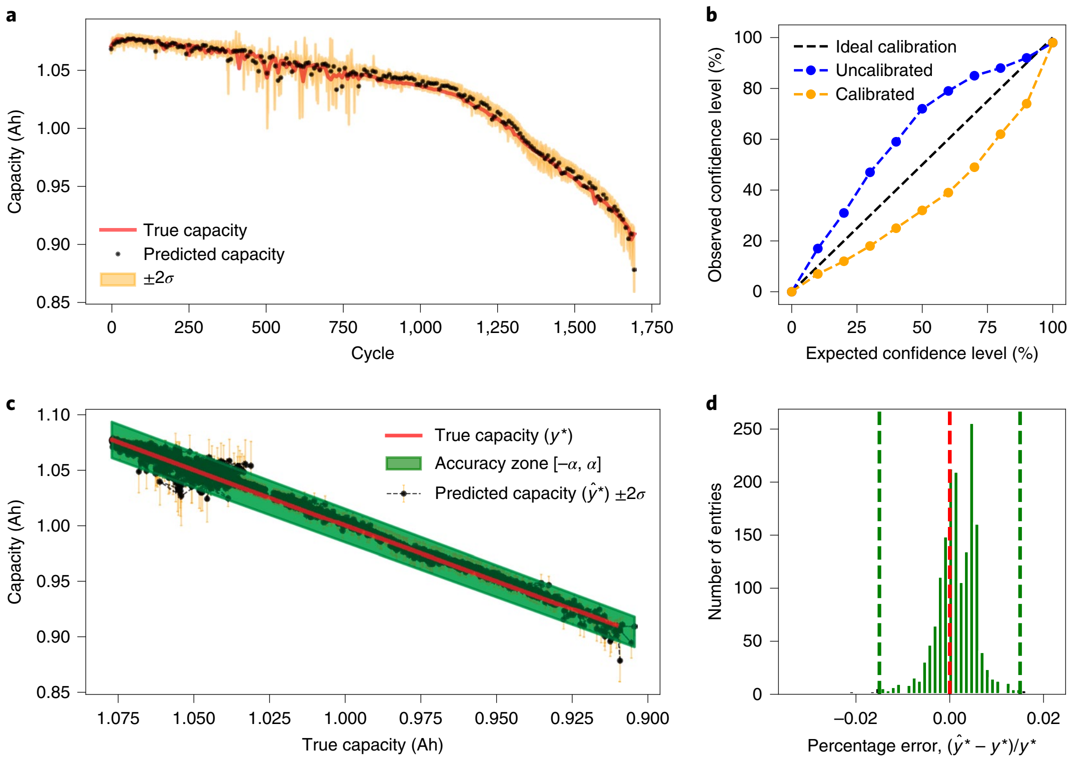

Fig. 4 | Prediction results with dNNe Group II cell number 1. a, dNNe prediction as a function of cycle. **b**, dNNE calibration results. **c**, dNNe actual versus predicted capacity; **d**, Histogram of percentage error.

Table 4 | Results for Group III cell no. 5 and average results over Group III dataset

	MAPE	RMSPE	C_{score}	Sh	α-accuracy	β	PEP
a Results for Group III cell no. 5							
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
b Average results over Group III dataset							
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

score as opposed to dNNE. dNNe is the second-best-performing algorithm in terms of accuracy (MAPE and RMSPE). It also exhibits adequate results for all other metrics, including PEP where it scores the highest.

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

Discussion on practical applicability of BHUMP

BHUMP can complement battery management systems, for both SOC and SOH estimation, and replace the traditional equivalent circuit models 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 partial discharge scenarios or various operating conditions such as random or high discharge rates. BHUMP can estimate capacity fade under the fast-charging protocol (Group II data) as well as random discharge (Group III data cycled under ARTEMIS driving protocol), typical of the operation of an electric vehicle battery pack. Future work could further extend to other charge/discharge protocols and open-source datasets such as in ref. [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 also considering in situ or operando sensory information such as optical and digital images or X-ray data[54] such that the algorithm learns the correlation between temperature, generated features and SOH indicator. Such variations mean that SOH assessment

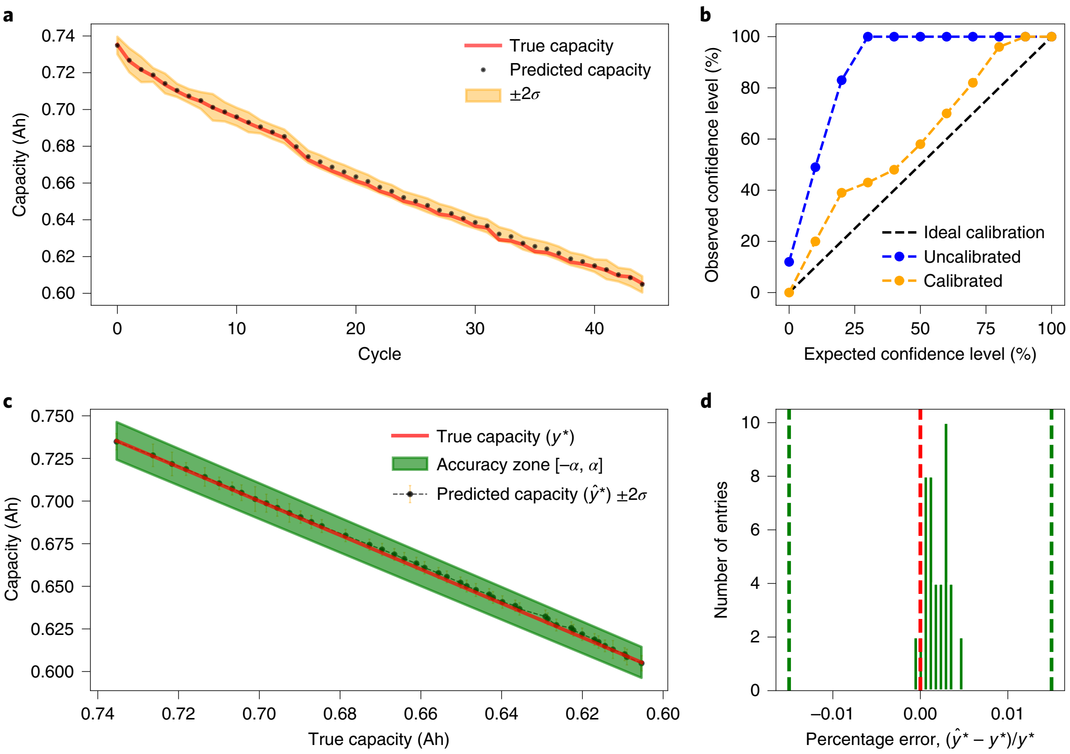

Fig. 5 | Prediction results with dNNe Group III cell no. 5. a, dNNe prediction as a function of cycle. **b**, dNNE calibration results. **c**, dNNe actual versus predicted capacity. **d**, Histogram of percentage error.

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 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 on the basis of the intended application, battery design and charge protocol. Despite such dataset variations, we think that deep learning has the potential to exceed other algorithms in the future as deep learning requires little tuning from the user and can take advantage of parallelization and an increasing amount of computational capabilities by deployment on graphics processing units 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 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.

Conclusion

The two widely adopted modelling techniques for online battery SOH estimation are equivalent circuit models and electrochemical models. However, when deployed online, a satisfactory trade-off between accuracy and computational efficiency is difficult to achieve. Here we have 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—BRR, GPR, RF and 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 the CC–CV protocol and the fast-charging protocol, and BRR for the CC 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 the charging protocol. For 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 for estimating battery degradation in under 15 min. 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 that machine learning pipelines will become a standard technique used in designing and implementing battery management systems of the future.

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.

Feature engineering. Feature engineering performs mathematical manipulations of extracted parts of the voltage curve during the CC charge protocol based on a lower voltage threshold V_l and an upper voltage threshold V_h (refer to the grey area of Fig. 1a) for all datasets except for cells charged with a 2-step fast-charging protocol. A characteristic of the 2-step fast-charging 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, owing to the nature of the charging method in the 2-step fast-charging protocol, we use only the CC−CV charge part of the charging protocol as per the black dotted segments in the grey area observed in Supplementary Fig. 2a,b. The values of V_h and V_l can be selected according to the intended application and the depth of discharge of the cell. In this work we select V_h to be equal to the cut-off voltage, $V_{\text{cut-off}}$. See Supplementary Note 2 on how we select V_l. Additional features are developed on extracted segments of the current curve during the CV charge protocol based on two current threshold values, I_h and I_l respectively (see Fig. 1b) for all cells except for the 2-step fast-charging protocol. We select I_h to be equal to the 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-charging protocol we select the current curve in Supplementary Fig. 2b. The obtained segments of the voltage and current charge curves are further processed to obtain a plethora of features, as described in Supplementary Note 3. Supplementary Table 2 summarizes all features generated from processing the curves.

Feature selection. Feature selection with recursive feature elimination and cross-validation (RFE-CV) performs selection and subset reduction automatically without the 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 an RF as opposed to the support vector machine used in ref. [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 RF 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 the capacity value at the ith cycle and C_1 is the capacity at the first cycle measured by a complete charge/discharge operation.

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

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

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

where $\boldsymbol{\theta}$ is the model weights vector and $\epsilon \sim \mathcal{N}(0, \Sigma)$ is a normally distributed noise parameter. Based on the selected algorithm, the function $f(\mathbf{x}, \boldsymbol{\theta})$ 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 $\mathbf{x}^{\star}$.

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

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

where γ is a small value such that the maximum 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. We note that other methods for data augmentations can also be used, such as the ones proposed in refs. [50,55–57]; however, the effect of data augmentation on model performance is beyond the scope of the present work.

The study solves equation (2) by making use of four algorithms as follows:

BRR. BRR considers a probabilistic model of the regression problem. The algorithm estimates a spherical Gaussian prior over the model weights given by $p(\boldsymbol{\theta}|\lambda) = \mathcal{N}(\boldsymbol{\theta}|0, \lambda^{-1}\mathbf{I}_p)$, where λ^{-1} is the precision. The priors over α (the regularizer) and λ are chosen to be gamma distributions. All parameters—$\boldsymbol{\theta}$, λ and α—are jointly estimated during training as per the implementation in ref. [58]. Posterior inference can be performed in a closed form because the prior is conjugate. For a complete explanation of the algorithm refer to ref. [59].

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

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

where $m(\mathbf{x})$ is the mean and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function. Please note that, as defined above, GPR does not require learning the parameters of the regression function $f(\mathbf{x}, \boldsymbol{\theta})$, 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, these may be smoother than other functions. For our implementation, we make use of the standard radial basis kernel as detailed in ref. [60], where a mathematical explanation of the algorithm is also given. Other kernel options exist, but we do not explore the effect of kernel choice on algorithm performance here.

RF. RF is a collection of constructed decision trees that sequentially conduct binary splits of the data to produce a homogeneous subset. For a comprehensive explanation of the algorithm refer to ref. [61]. 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 1,500. The variability of the predictions estimated by the RF has been investigated based on ref. [49], where the confidence interval's variance was obtained using the bootstrap replicates used to train the RF itself.

dNNe. Ensemble methods combine different regressors into a meta-regressor and we consider an ensemble of deep neural networks as proposed in ref. [50]. Each network in the ensemble incorporates two 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 two densely connected hidden layers. Each layer decreases in size by 50% of neurons, based on the number of input features. When the input number features is less than ten, we force the network's hidden layers to be four neurons in the first layer and three in the second layer. For example, when 18 input features are considered, the first hidden layer consists of nine neurons, followed by four neurons in the second hidden layer. Each network used in this work has the following parameters: the 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 an Adam optimizer 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 MAPE and RMSPE.

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

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

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 ref. [62]. To plot calibration curves, we divide each predicted confidence interval in m confidence levels that are monotonically increasing on the interval [0, 1]; that is, $0 < p_1 < p_2 < ... < p_m < 1$. 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 summarized as:

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

From the reliability curve assessment, we then perform re-calibration using isotonic regression [63]. A well-calibrated regressor should lie very close to the ideal diagonal curve; see the results in Fig. 2b. 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.

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

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

$$\mathrm{Sh} = \frac{1}{n}\sum\nolimits_{i=1}^{n} \sigma_i \quad (11)$$

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

We further propose an assessment of uncertainty prediction via prognostics performance metrics from an engineering point of view, adopted from ref. [64]. First, we introduce the accuracy zone defined by a threshold, α (see Fig. 5), which is calculated as a percentage error from the true capacity value; that is, $y \pm \alpha$. We select an α of ±1.5% (α can be adjusted according to the intended application). Using 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; see Fig. 5. Ideally, β should be 1, suggesting that the predicted confidence interval is small and encapsulates the entire α-accuracy zone. Since α summarizes the notion of desired accuracy, α^{+} is the upper bound for estimates above the accuracy zone, and α^{-} represents low estimates or the value residing under the desired accuracy zone. Depending on the application, both or any 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 Fig. 5), denoted here by PEP, as a measure of the algorithm's uncertainty in a critical application scenario.

Data availability

The datasets used in this study are available at, for Group 1, https://web.calce.umd.edu/batteries/data.htm and https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/, for Group 2, https://data.matr.io/1/projects/5c48dd2bc625d700019f3204, and for 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 https://doi.org/10.5281/zenodo.4390152.

Received: 18 May 2020; Accepted: 28 January 2021;
Published online: 5 April 2021

References

1. Curry, C. Lithium-ion battery costs and market: squeezed margins seek technology improvements & new business models. *Bloomberg New Energy Finance* https://data.bloomberglp.com/bnef/sites/14/2017/07/BNEF-Lithium-ion-battery-costs-and-market.pdf (5 July 2017).
2. Bernhart, W. Challenges and opportunities in lithium-ion battery supply. In *Future Lithium-ion Batteries* 316−334 (Royal Society of Chemistry, 2019).
3. You, G.-W., Park, S. & Oh, D. Diagnosis of electric vehicle batteries using recurrent neural networks. *IEEE Trans. Indust. Electron.* **64**, 4885–4893 (2017).
4. Barré, A. et al. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *J. Power Sources* **241**, 680–689 (2013).
5. Zhang, J. & Lee, J. A review on prognostics and health monitoring of li-ion battery. *J. Power Sources* **196**, 6007–6014 (2011).
6. Farmann, A., Waag, W., Marongiu, A. & Sauer, D. U. Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles. *J. Power Sources* **281**, 114–130 (2015).
7. Hannan, M. A., Lipu, M. H., Hussain, A. & Mohamed, A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: challenges and recommendations. *Renew. Sustain. Energy Rev.* **78**, 834–854 (2017).
8. Hu, X., Li, S. & Peng, H. A comparative study of equivalent circuit models for Li-ion batteries. *J. Power Sources* **198**, 359–367 (2012).
9. Feng, T., Yang, L., Zhao, X., Zhang, H. & Qiang, J. Online identification of lithium-ion battery parameters based on an improved equivalent-circuit model and its implementation on battery state-of-power prediction. *J. Power Sources* **281**, 192–203 (2015).
10. Andre, D. et al. Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. II: Modelling. *J. Power Sources* **196**, 5349–5356 (2011).
11. Daigle, M. J. & Kulkarni, C. S. Electrochemistry-based battery modeling for prognostics. In *Ann. Conf. Prognostics and Health Management Society* 040 (PHM, 2013).
12. Bole, B., Kulkarni, C. S. & Daigle, M. Adaptation of an electrochemistry-based li-ion battery model to account fordeterioration observed under randomized use. In *Proc. Ann. Conf. Prognostics and Health Management Society* (PHM, 2014).
13. Prasad, G. K. & Rahn, C. D. Model based identification of aging parameters in lithium ion batteries. *J. Power Sources* **232**, 79–85 (2013).
14. Severson, K. A. et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy* **4**, 383−391 (2019).
15. Saha, B., Goebel, K., Poll, S. & Christophersen, J. Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Trans. Instrum. Measure.* **58**, 291–296 (2008).
16. Goebel, K., Saha, B., Saxena, A., Celaya, J. R. & Christophersen, J. P. Prognostics in battery health management. *IEEE Instrum. Measure. Mag.* **11**, 33–40 (2008).
17. Hu, X., Jiang, J., Cao, D. & Egardt, B. Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling. *IEEE Trans. Indust. Electron.* **63**, 2645–2656 (2015).
18. Klass, V., Behm, M. & Lindbergh, G. A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation. *J. Power Sources* **270**, 262–272 (2014).
19. Attia, P. M. et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature* **578**, 397–402 (2020).
20. Coleman, M., Hurley, W. G. & Lee, C. K. An improved battery characterization method using a two-pulse load test. *IEEE Trans. Energy Conv.* **23**, 708–713 (2008).
21. Waag, W., Käbitz, S. & Sauer, D. U. Experimental investigation of the lithium-ion battery impedance characteristic at various conditions and aging states and its influence on the application. *Appl. Energy* **102**, 885–897 (2013).
22. Tröltzsch, U., Kanoun, O. & Tränkler, H.-R. Characterizing aging effects of lithium ion batteries by impedance spectroscopy. *Electrochim. Acta* **51**, 1664–1672 (2006).
23. Birkl, C. R., Roberts, M. R., McTurk, E., Bruce, P. G. & Howey, D. A. Degradation diagnostics for lithium ion cells. *J. Power Sources* **341**, 373–386 (2017).
24. Li, Y., Zhong, S., Zhong, Q. & Shi, K. Lithium-ion battery state of health monitoring based on ensemble learning. *IEEE Access* **7**, 8754–8762 (2019).
25. Li, Y. et al. Random forest regression for online capacity estimation of lithium-ion batteries. *Appl. Energy* **232**, 197–210 (2018).
26. Sun, B., Ren, P., Gong, M., Zhou, X. & Bian, J. SOH estimation for Li-ion batteries based on features of IC curves and multi-output Gaussian process regression method. *DEStech Trans. Environ. Energy Earth Sci.* https://doi.org/10.12783/dteees/iceee2018/27789 (2018).
27. Feng, X. et al. Online state-of-health estimation for Li-ion battery using partial charging segment based on support vector machine. *IEEE Trans. Vehic. Technol.* **68**, 8583–8592 (2019).
28. Li, Y. et al. A quick on-line state of health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter. *J. Power Sources* **373**, 40–53 (2018).
29. Dubarry, M., Svoboda, V., Hwu, R. & Liaw, B. Y. Incremental capacity analysis and close-to-equilibrium ocv measurements to quantify capacity fade in commercial rechargeable lithium batteries. *Electrochem. Solid State Lett.* **9**, A454 (2006).
30. Weng, C., Cui, Y., Sun, J. & Peng, H. On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. *J. Power Sources* **235**, 36–44 (2013).
31. Yang, D., Zhang, X., Pan, R., Wang, Y. & Chen, Z. A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *J. Power Sources* **384**, 387–395 (2018).
32. Richardson, R. R., Birkl, C. R., Osborne, M. A. & Howey, D. A. Gaussian process regression for in situ capacity estimation of lithium-ion batteries. *IEEE Trans. Indust. Inform.* **15**, 127–138 (2018).
33. Shen, Y., Seeger, M. & Ng, A. Y. Fast Gaussian process regression using KD-trees. In *Adv. Neural Information Processing Systems (NIPS)* 1225−1232 (2006).
34. Saha, B., Poll, S., Goebel, K. & Christophersen, J. An integrated approach to battery health monitoring using Bayesian regression and state estimation. In *2007 IEEE Autotestcon* 646–653 (IEEE, 2007).
35. Ben-Shimon, D. & Shmilovici, A. Accelerating the relevance vector machine via data partitioning. *Found. Comput. Decision Sci.* **31**, 27–42 (2006).
36. Wang, Z., Zeng, S., Guo, J. & Qin, T. Remaining capacity estimation of lithium-ion batteries based on the constant voltage charging profile. *PLoS ONE* **13**, e0200169 (2018).
37. Engel, S. J., Gilmartin, B. J., Bongort, K. & Hess, A. Prognostics, the real issues involved with predicting life remaining. In *2000 IEEE Aerospace Conf. Proc.* 00TH8484, Vol. 6, 457−469 (IEEE, 2000).
38. Pomerantseva, E., Bonaccorso, F., Feng, X., Cui, Y. & Gogotsi, Y. Energy storage: the future enabled by nanomaterials. *Science* **366**, eaan8285 (2019).
39. Seh, Z. W., Sun, Y., Zhang, Q. & Cui, Y. Designing high-energy lithium–sulfur batteries. *Chem. Soc. Rev.* **45**, 5605–5634 (2016).
40. Liu, G., Bao, H. & Han, B. A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis. *Hindawi Math. Problems Eng.* **2018**, 5105709 (2018).
41. Kanter, J. M. & Veeramachaneni, K. Deep feature synthesis: towards automating data science endeavors. In *2015 IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)* 1−10 (IEEE, 2015).

42. Williard, N., He, W., Osterman, M. & Pecht, M. Comparative analysis of features for determining state of health in lithium-ion batteries. *Int. J. Prognostics Health Manage.* **4**, 1.7 (2013).
43. Zhang, Y. & Guo, B. Online capacity estimation of lithium-ion batteries based on novel feature extraction and adaptive multi-kernel relevance vector machine. *Energies* **8**, 12439–12457 (2015).
44. Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning* **46**, 389–422 (2002).
45. Darst, B. F., Malecki, K. C. & Engelman, C. D. Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC Genet.* **19**, 65 (2018).
46. Gregorutti, B., Michel, B. & Saint-Pierre, P. Correlation and variable importance in random forests. *Statist. Comput.* **27**, 659–678 (2017).
47. Goodfellow, I. J., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. Preprint at https://arxiv.org/abs/1412.6572 (2014).
48. Doyle, M., Fuller, T. F. & Newman, J. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *J. Electrochem. Soc.* **140**, 1526 (1993).
49. Wager, S., Hastie, T. & Efron, B. Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. *J. Machine Learning Res.* **15**, 1625–1651 (2014).
50. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Adv. Neural Information Processing Systems (NIPS)* 6402–6413 (Curran Associates, 2017).
51. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *J. Machine Learning Res.* **13**, 281–305 (2012).
52. André, M. The Artemis European driving cycles for measuring car pollutant emissions. *Sci. Total Environ.* **334**, 73–84 (2004).
53. Markham, I. S. & Rakes, T. R. The effect of sample size and variability of data on the comparative performance of artificial neural networks and regression. *Comput. Operations Res.* **25**, 251–263 (1998).
54. Handoko, A. D., Wei, F., Yeo, B. S. & Seh, Z. W. et al. Understanding heterogeneous electrocatalytic carbon dioxide reduction through operando techniques. *Nat. Catal.* **1**, 922–934 (2018).
55. Jagielski, M. et al. Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symp. on Security and Privacy (SP)* 19–35 (IEEE, 2018).
56. Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J. & Hsieh, C.-J. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *Proc. AAAI Conf. Artificial Intelligence* Vol. 32 (AAAI, 2018).
57. Sharma, Y. & Chen, P.-Y. Attacking the Madry defense model with L_1-based adversarial examples. Preprint at https://arxiv.org/abs/1710.10733 (2017).
58. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Machine Learning Res.* **12**, 2825–2830 (2011).
59. Bishop, C. M. *Pattern Recognition And Machine Learning* (Springer, 2006).
60. Rasmussen, C. E. Gaussian processes in machine learning. In *Summer School on Machine Learning* 63–71 (Springer, 2003).
61. Breiman, L. Random forests. *Machine Learning* **45**, 5–32 (2001).
62. Kuleshov, V., Fenner, N. & Ermon, S. Accurate uncertainties for deep learning using calibrated regression. Preprint at https://arxiv.org/abs/1807.00263 (2018).
63. Chakravarti, N. Isotonic median regression: a linear programming approach. *Math. Operations Res.* **14**, 303–308 (1989).
64. Saxena, A. et al. Metrics for evaluating performance of prognostic techniques. In *2008 Int. Conf. on Prognostics and Health Manage.* 1–17 (IEEE, 2008).

Acknowledgements

This work was supported by the Lloyd's Register Foundation (grant number AtRI_100015), The Engineering and Physical Sciences Research Council (EPSRC), the Center for Doctoral Training in Embedded Intelligence, and Baker Hughes (grant number EP/L014998/1). The work was further supported by the EPSRC through the UK National Centre for Energy Systems Integration (CESI) (grant number EP/P001173/1), and by InnovateUK through the Responsive Flexibility (ReFlex) (project reference 104780). We thank the more than 150 companies and organizations that support research activities at the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland annually.

Author contributions

D.R. conceived the study, analysed the experimental data, developed the machine learning pipeline and wrote the paper, while S.S. assisted with experimental data interpretation, problem statement formulation, and feature engineering. V.R. provided technical input for the machine learning method development, while D.F. and M.P. provided input for the battery SOH application. V.R., M.P. and D.F. supervised the work. All authors commented on and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-021-00312-3.

Correspondence and requests for materials should be addressed to D.R.

Peer review information *Nature Machine Intelligence* thanks the anonymous reviewers for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.