

Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life

Samuel Greenbank, and David Howey, *Senior Member, IEEE*

Abstract—Lithium-ion cells may experience rapid degradation in later life, especially with more extreme usage protocols. The onset of rapid degradation is called the ‘knee point’, and forecasting it is important for the safe and economically viable use for batteries. We propose a data-driven method that uses automated feature selection to produce inputs for a Gaussian process regression model that estimates changes in battery health, from which the entire capacity fade trajectory, knee point and end of life may be predicted. The feature selection procedure flexibly adapts to varying inputs and prioritises those that impact degradation. For the datasets considered, it was found that calendar time and time spent in specific voltage regions had a strong impact on degradation rate. The approach produced median root mean square errors on capacity estimates under 1%, and also produced median knee point and end of life prediction errors of 2.6% and 1.3% respectively.

Index Terms—Feature selection, machine learning, lithium-ion, degradation, battery

I. INTRODUCTION

Predicting lithium-ion battery degradation during design and operation is a significant challenge, and a large number of techniques for this have been proposed in literature [1], [2]. Machine learning models have recently been applied for forecasting battery state of health, but they remain limited by lack of transparency, and require careful choice of inputs [2]. Battery degradation is typically measured using capacity fade or resistance increase. Degradation is caused by many mechanisms [3], [4], and these may interact in various ways. Degradation mechanisms are influenced by a wide variety of factors, such as calendar time, high power use, low temperature use, and combinations of these [1]. However, a battery end-user can only measure time, current, terminal voltage and cell surface temperature, at best, plus very occasional capacity or resistance through bespoke characterisation tests if they are possible. Consequently, prioritising the mechanisms that drive ageing for a given battery and use case is challenging [1]. For data-driven approaches, this challenge demands careful and flexible feature extraction.

There have been attempts at battery health prognosis using neural networks [5], [6], support vector machines [7]–[9] and Gaussian process regression [10]–[18]. Some previous

researchers have made use of an open source dataset from the NASA AMES research centre [10], [11], [14], [15], and much of this data shows an approximately constant degradation rate. However, lithium-ion cells have sometimes been shown to suffer from the onset of more rapid capacity fade or resistance increase later in life [19]–[21]. This sudden acceleration in ageing is often referred to as a ‘knee point’ [20]. The knee point indicates that significant deterioration has occurred, and manufacturers may wish to replace cells at this point. Beyond the knee point, degradation accelerates rapidly and a cell can be considered to have lost its value for a given application, and there may also be safety concerns [7]. Forecasting it is therefore crucial for understanding the lifetime value of lithium-ion batteries [21]. Previous attempts at data-driven health prediction have tried to estimate the timing of the knee point [7], [24] or the cycle life to 80% capacity [19] respectively. In both cases these were point estimates, rather than predictions based on the full trajectory of health estimates.

Forecasting battery state of health using machine learning approaches usually requires an assumption about the way the batteries are charged and discharged. Often, usage is implicitly considered to be fixed over time, with identical train and test use cases [10]. However, changes in usage can be accommodated, for example by dividing usage into fixed sections of time and summing the health impacts piecewise over the sections. This approach has been successfully applied to capacity forecasting with a Gaussian process regression model [11]. In this case, for the dataset used, a simple manual feature selection exercise identified calendar time and charge throughput as dominant inputs affecting capacity fade, although other features such as time periods spent in extreme current and temperature ranges could also be relevant for other datasets where batteries are used more intensively [11].

For any machine learning model, the input features decide the predictive performance. Several recent battery health publications have used a manually selected small feature set [16]–[18], [25]. There is scope to automate the feature extraction/selection process, and test the model predictions over more general use cases.

We propose a novel automated feature selection approach, illustrated in Fig. 1. The method extracts features from the data, selects a relevant subset of key features, feeds their values into a Gaussian process regression model, and uses the output to produce an estimate of the capacity fade trajectory. The novel contributions of this work, are as follows: First, the feature extraction and selection process is automated and

This research was funded by EPSRC and Siemens. The authors are also grateful to Antti Aitio for providing comments on the manuscript before submission.

Samuel Greenbank and David Howey are from the Battery Intelligence Lab, Department for Engineering Science, University of Oxford, United Kingdom. Corresponding author: david.howey@eng.ox.ac.uk.

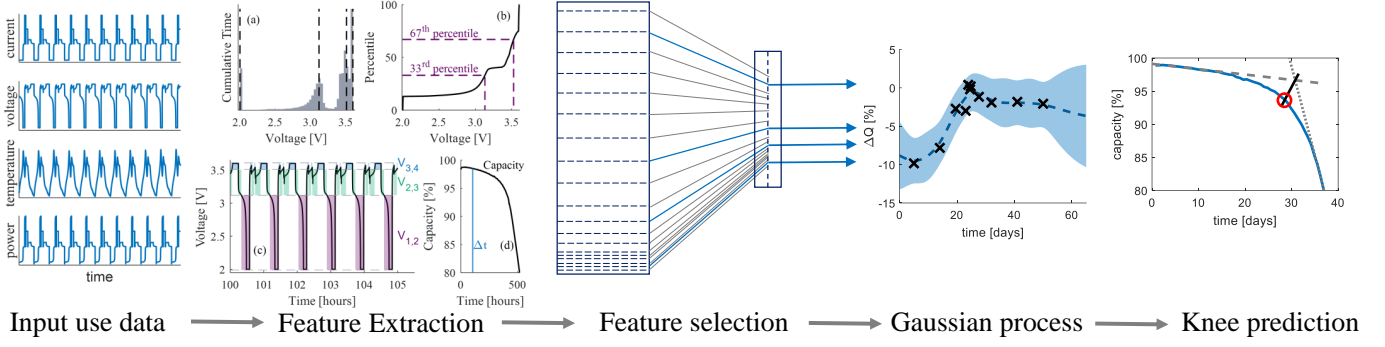


Fig. 1: Workflow of the automated process proposed in this paper.

does not require users to choose beforehand what features they might expect to drive degradation. It is also transparent and easy to interpret. Second, the algorithm automatically chooses features that have reduced correlation with one another, avoiding unnecessary overlap. Third, it is fast and therefore able to deal with large datasets—it scales linearly with the number of input rows and quadratically with the number of features. Fourth, the approach can handle changing battery usage profiles - it does not assume cycling is always the same. Finally, we present (section III-F) comprehensive error metrics using a wide range of test conditions in order to demonstrate the robustness of the approach.

II. DATA SOURCES

Open source battery cycling (voltage, current, temperature) and capacity fade data were used for this work [19], [21]. The first dataset [19] consists of 135 lithium iron phosphate/graphite 18650 Li-ion cells (A123) that were cycled in a temperature chamber set at 30 °C. All the cells underwent identical discharge cycles at 4C but had varied fast charging protocols from 3C to 8C [19].

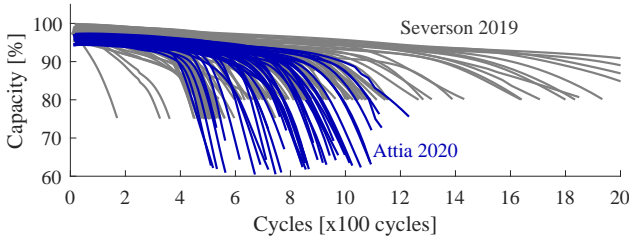


Fig. 2: The data used in references [19] and [21]

The second dataset [21], a follow up to the first, contains 45 cells which were cycled to failure, defined as 80% of the nominal capacity of 1.1 Ah. These cells were the same chemistry, size and manufacturer as those in the first dataset, and were tested at the same temperature setpoint and discharge rate as the previous test, but with a fixed 10-minute charging protocol [21], using a wide range of C-rates from 3C to 8C. Both datasets were non-uniformly sampled with the lowest sampling rate being 0.2 Hz. In our work, the datasets were cleaned before use by removing cells with obvious experimental errors (as identified by the original authors of the dataset),

and a shorter selection of the remaining data, with lifetimes between 15 and 40 days, was chosen. After this, 147 cells worth of data remained available.

III. METHODS

Fig. 1 shows the pipeline from raw measured data through to feature extraction and selection, modelling, and finally capacity forecasting and knee point prediction. Feature extraction and selection are not dependent on battery chemistry, usage or history, therefore this approach should be able to handle a wide variety of different cells in the same manner. Since this is a supervised learning method, regular measurements of state of health (from check up tests) are required. The state of health metric used throughout this paper is the discharge capacity Q [19], [21], but other metrics (such as resistance) could instead be used if desired. Gaussian process regression was used to model the relationship between features and the change in battery health $\Delta Q = Q(t + \Delta t) - Q(t)$, where $Q(t + \Delta t)$ is the measured capacity at time $t + \Delta t$ and $Q(t)$ is the measured capacity at time t . In other words, data was divided into discrete sections, each of time period Δt , each section having a single row of features and a single capacity change ΔQ . The full capacity fade trajectory, $Q(t)$, for a given usage condition can then be constructed by summing all ΔQ over time, in a piecewise linear fashion.

A. Feature Extraction

The aim of the feature extraction and selection process is to reduce the dimensionality of the input data by producing metrics that represent the most important aspects affecting state of health [23]. For example if the raw input data has size $\mathbb{R}^{n \times m}$, where n is the number of time points and m the number of raw data streams, then we aim to have a feature set size $\mathbb{R}^{p \times q}$, where $p \ll n$. (In our case q and m are similar sizes, both relatively small.)

The frequency at which to extract features is a subjective choice, and involves a trade-off between computational complexity versus skipping useful information. If battery capacity is only measured occasionally (e.g. monthly), it would make sense to calculate features using data between capacity measurements, over some time period Δt . Here, the cells were cycled continuously at high rates, resulting in up to 28 cycles per day, and the cycling data was directly used for

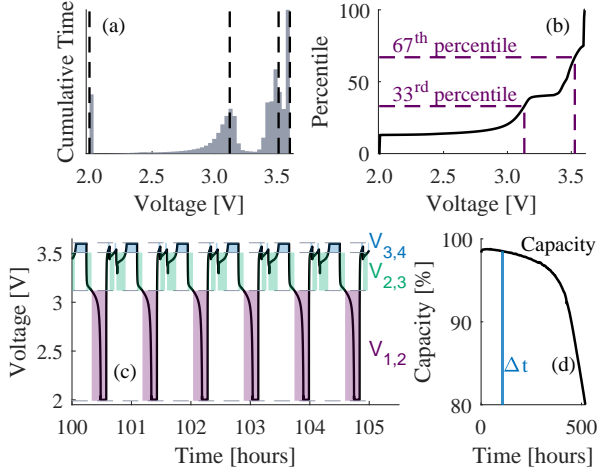


Fig. 3: (a) Histogram and (b) Cumulative histogram, over all voltage values; (c) Example of three voltage features (coloured areas) overlaid on a subset of raw data; (d) Time interval of capacity data shown by (c)

capacity measurements. We therefore could have chosen quite short intervals for Δt , we investigated the impact of different frequencies of feature extraction and found that using a very short interval ($\Delta t = 1$ hour) slowed down the computational speed significantly but did not change overall performance. Using a long interval ($\Delta t = 24$ hours) tended to miss important features such as rapid changes in ageing. Therefore, as a compromise, features were extracted for every $\Delta t = 12$ hours of data, equivalent to taking a health measurement every 9-19th cycle depending on charging protocol. Note, this is specific to the datasets used in this paper and may be different for other data.

Features can be extracted using any function of the raw data and there is a trade-off between complexity and simplicity [22]. Here we created an automated process based on ‘time spent’ in certain usage regions, based on the training data set. A pseudo-code representation of the feature extraction process is shown in Algorithm 1. Time spent in different regions is a simple health indicator that can easily be understood and acted upon by a user. For the process to be automatic, the health indicators must be extracted based on data calculated from the training set. Hence feature extraction was split into two parts: 1) calculating the variable bounds, 2) extracting the feature data.

Firstly, in order to create the variable bounds, time series of instantaneous absolute current $|I(t)|$, instantaneous power $P(t) = V(t)I(t)$ and instantaneous absolute power $|P(t)| = |V(t)I(t)|$ were calculated. Combined with the measured voltage, current and temperature time series, this gives $m = 6$ basic input data streams. Then, for each of these input data streams, across the entire population of data available, a histogram was generated. Fig. 3(a) shows an example of the histogram and 3(b) the cumulative histogram for the voltage data.

From the cumulative histograms, the values of each input corresponding to the 1st, 33rd, 67th and 99th percentiles were

Percentile	Current [A]	Voltage [V]	Temperature [°C]	Power [W]
1 st	-4.00	2.00	30.0	-12.84
33 rd	-0.53	3.12	32.8	-1.08
67 th	1.00	3.51	35.3	3.43
99 th	6.00	3.60	40.3	21.34

TABLE I: Example feature extraction variable bounds for a training set of 50 cells.

Number	Percentile limits	Label	Voltage Range
1	1 st to 33 rd	$V_{1,2}$	$2.00 \text{ V} < V < 3.12 \text{ V}$
2	1 st to 67 th	$V_{1,3}$	$2.00 \text{ V} < V < 3.51 \text{ V}$
3	1 st to 99 th	$V_{1,4}$	$2.00 \text{ V} < V < 3.60 \text{ V}$
4	33 rd to 67 th	$V_{2,3}$	$3.12 \text{ V} < V < 3.51 \text{ V}$
5	33 rd to 99 th	$V_{2,4}$	$3.12 \text{ V} < V < 3.60 \text{ V}$
6	67 th to 99 th	$V_{3,4}$	$3.51 \text{ V} < V < 3.60 \text{ V}$

TABLE II: Features extracted from voltage, V , profiles are proportions of time spent in specific ranges defined by the four chosen percentiles.

calculated, and each data series was divided into regions, as shown (again using voltage as an example) in Fig. 3(b). Examples of these thresholds are given in Table I and examples of feature types extracted using these thresholds are given in Table II. In addition to features extracted in this way, we also included time and the square root of time (both measured at the point capacity was measured) as features.

The feature data was then calculated for each feature type and for every chunk of data of length Δt , corresponding to the time spent in each different region. This process resulted in the extraction of $q = 74$ different types of features, and reduced $n = 1.08 \times 10^8$ time points in the raw data across $m = 6$ data streams, down to $p = 7386$ rows of features. In section III-B we discuss how q can be reduced substantially further to select only the most relevant ≈ 5 feature types. Typically, the most commonly selected feature was $V_{2,3}$, the proportion of time spent between 3.12 V and 3.51 V, described in equation 1 and shown in grey in Fig. 3(c).

$$V_{2,3}(t_i) = \frac{\text{time between 3.12 V and 3.51 V}}{\Delta t = \text{time between } t_i \text{ and } t_{i-1}} \quad (1)$$

The feature rows associated with the test sets were calculated in the same way, assuming perfect knowledge of the future cell use (but no knowledge of health, which is what the model predicts). Note that the cell usage in testing does not have to be and was not the same as the cell usage during training.

A key advantage of our feature extraction approach over existing methods is that it models battery ageing flexibly as a function of different types of usage (for example, it will naturally account for users who leave the battery at high SOC for long periods of time, or do intensive cycling). Many existing approaches for health modelling are not able to do this because they use only features that we consider to be outputs rather than inputs to a model, in other words they capture instantaneous battery health metrics, but they do not necessarily capture the features that drive ageing. Examples of metrics used in this way include charge or discharge time

between certain voltage limits [9], [16], [25], capacity [16], [18], entropy [25], resistance [12], [16], and incremental capacity analysis and differential voltage analysis based features [7], [13], [19], [23], [25]. Some papers have used charge throughput as an input feature [11], [17], but we found in general this correlates well with time if the usage is consistent throughout life. Temperature-based features may also be of interest [7], [17], [25], although we have generally found voltage-based features to be more important in the datasets considered here.

Algorithm 1: Feature extraction and calculation

Input: 1. current, voltage, temperature time series
Input: 2. health metric, measured every N cycles
 1. calculate absolute current, power, absolute power;
 2. assemble data matrix $[I \ V \ T \ | \ I \ P \ | \ P]$;
 3. **foreach** *column* **do**
 calculate histogram and cumulative histogram;
 calculate 1st, 33rd, 67th, 99th percentiles;
end
 3. **foreach** *variable* **do**
 foreach *percentile* **do**
 feature_name = *variable*_{start,end};
 end
end
 4. **foreach** *battery time series* **do**
 i. record capacity measurement every 12 hours;
 ii. record time and $\sqrt{\text{time}}$ at these points;
 iii. **foreach** *feature_name* **do**
 foreach *time interval* Δt **do**
 a. calculate time spent in range
 *variable*_{start,end};
 b. divide this by duration Δt ;
 end
 end
end
Output: calculated feature data for all cells

B. Feature Selection

The aim of feature selection is to prioritise the types of features that are most important in affecting the battery health, and ignore the less important features. It is not expected that all feature types will affect the battery health equally, and redundant data or overfitting should be avoided. For example, previous work [11] found that calendar time and charge throughput could be particularly significant for battery health prediction.

Battery capacity fade trajectories are often quite smooth, which is unsurprising since loss of battery health is a cumulative process [11], [20]. For the cells used here the profiles can be split into just two or three distinct phases. It is therefore expected that only a small number of features might be required for modelling the state of health.

Principle component analysis (PCA) is a common tool for dimensionality reduction [22]. It was not used here because it would produce features which are linear combinations of

inputs, whereas for simplicity, we wished to directly rank individual features, rather than combinations of features.

The feature selection method proposed here examines the similarity between features and changes in health. There are many methods available to measure similarity, such as correlation methods [7], [18], [23], [25], [26] and covariance functions [22], [27]. We found most techniques had comparable performance, hence we chose to use the absolute value of Pearson's correlation coefficient, shown in equation 2 for features f_i and f_j . Here, cov is covariance, σ is standard deviation, and results are stored in similarity matrix S .

$$S_{i,j}(f_i, f_j) = \left\| \frac{cov(f_i, f_j)}{\sigma(f_i)\sigma(f_j)} \right\| \quad (2)$$

Elements of S are bounded between 0 and 1, with higher values indicating stronger similarity. Figure 4 gives an example of what a coloured version of S might look like.

time	1	0.59	0.09	0.58	0.17	0.28	0.69
V _{2,3}	0.59	1	0.04	0.88	0.44	0.16	0.85
T _{1,4}	0.09	0.04	1	0.12	0.38	0.43	0.12
V _{1,2}	0.58	0.88	0.12	1	0.18	0.01	0.79
I _{2,3}	0.17	0.44	0.38	0.18	1	0.86	0.11
P _{2,4}	0.28	0.16	0.43	0.01	0.86	1	0.09
	time	V _{2,3}	T _{1,4}	V _{1,2}	I _{2,3}	P _{2,4}	ΔQ

Fig. 4: Example similarity matrix with 6 features and ΔQ . Data was taken from a random subset of cells and feature labels use the scheme of Table I, with T, I and P as temperature, current and power respectively. This does not represent the full array of possible inputs used here.

Feature selection can now be performed; the simplest approach would be to select the features that correlate best with ΔQ . However, that will likely lead to many redundant features being included. The health prediction model would likely perform equally with one of these features as with 10 of them, but with unnecessary increased computational complexity [22], [27]. Therefore a step is included that removes redundant features, applying an upper limit on similarity across features. From the example of Fig. 4, one expects to select V_{2,3}, then reject V_{1,2} for being too similar, then select time as the second input feature. The impact of varying the maximum allowed shared correlation was investigated to verify its impact.

The selection process, which shares similarities with explicit orthogonalization [28] and simplifies wrapper-based methods for SoH estimation [25], can be summarised as follows: (1) Find the feature correlating best with the change in capacity in the training set. (2) Remove all features which share a correlation coefficient greater than 0.85. (3) Repeat the previous steps until the required number of features is obtained. (The number of features required is chosen by the user, and is a trade-off between accuracy and complexity.) In the example of Fig. 4, an appropriate selection might be features V_{2,3}, V_{1,2}

and time, since these are similar to the required output ΔQ . Features $T_{1,4}$, $I_{2,3}$ and $P_{2,4}$ would be poor selections.

The effectiveness of the feature extraction and selection process may be demonstrated by considering how the features span the training and test datasets. If the process has been successful, there should be a strong overlap between train and test data as a function of battery health changes and the features being considered. An example of a successful outcome may be seen in Fig. 5.

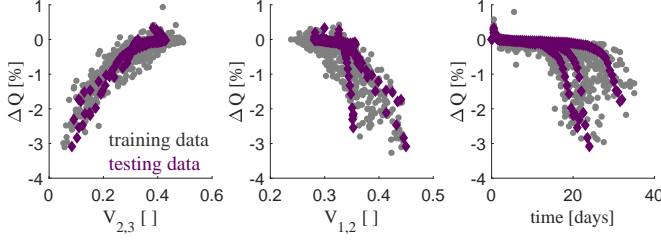


Fig. 5: Example showing first three features selected for one training set plotted against change in capacity, ΔQ . Purple test data overlaps grey training data which leads to successful forecasting.

C. Gaussian Process Regression

We selected Gaussian process (GP) regression to map from the selected input features to capacity transitions ΔQ since it is a flexible approach that makes very few assumptions on the function to be fitted [10], [11], [13], [25]. The dataset was split into training and test sections. The GP hyperparameters were fitted to the training data in the standard way using maximum marginal likelihood estimation, with Matlab's `fitrgp` function. The test datasets were used to quantify the model performance. A Matérn 5/2 covariance function, shown previously to work well for this application [11], [27], was used. For two input data points, x_i and x_j the Matérn 5/2 covariance is given by:

$$r(x_i, x_j) = \sqrt{\sum_k \frac{(x_{i,k} - x_{j,k})^2}{\sigma_{l,k}^2}} \quad (3)$$

$$\kappa_{m52}(r) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2 \right) \exp(-\sqrt{5}r) \quad (4)$$

The two hyperparameters σ_f and σ_l represent the magnitude and lengthscale of the covariance. The covariance function used automatic relevance determination, which allows the length-scale hyperparameter, σ_l , to have a different value for each input feature, k [13].

The test set inputs, unseen by the algorithm in terms of fitting, was used on the trained GP model to produce capacity transition test predictions and their associated $\pm 2\sigma$ credible intervals.

In summary, the feature selection process produced a set of inputs which correlated with the output ΔQ . Together, the features formed an approximate linear model, which the GP then improved on to produce an accurate non-linear model of capacity fade.

D. Knee Point Identification

The capacity transition forecasts were summed over time to form predicted capacity fade profiles, starting with the initial value of capacity which was assumed to be known. From this, various techniques can be used to locate the knee point position [7], [24]. Here we chose to fit the early and late life capacity fade gradients using linear regression, then calculated their angle bisector and found the intersection of this with the capacity fade curve, see Fig. 6.

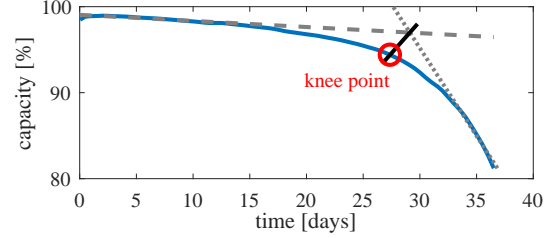


Fig. 6: Knee point calculation.

E. Evaluation Metrics

The performance of the approach was evaluated by assessing the accuracy of the predicted capacity profiles of the test set using various metrics. The first is the root mean square error, $RMSE_Q$, of the predictions. This is an effective metric for predictive performance, but it could be significantly influenced by a single, poor ΔQ prediction. Therefore a second metric to consider is the root mean square error of the transitions, $RMSE_{\Delta Q}$, calculated the same way, but with only the capacity transition data.

The third capacity-based performance metric was the end of life percentage error, $PE(t_{EoL}) = 100(\hat{t}_{EoL} - t_{EoL})/t_{EoL}$, which measures the percentage difference in time between the predicted \hat{t}_{EoL} and observed end of life, t_{EoL} .

The fourth metric is the knee point prediction accuracy, which may be evaluated in the same way as the end of life error but using the knee point position in time, $PE(t_{knee})$.

For all evaluation metrics, the median value and 95th percentile are reported. Mean errors are also quoted, for comparative purposes.

F. Trial Setup

The feature selection technique was tested in two ways: a large-scale test, and a limit testing experiment. The large-scale test quantified the general performance of the approach using the standard k-fold cross-validation technique [22], using 20 randomly sampled datasets, each with 100 training cells and 47 test cells. In the limited data test, a subset of the trial data was selected to investigate the impacts of missing data on performance, particularly late life data.

For the large-scale test, temperature, voltage and current data from 147 cells was used, with lifetimes ranging from 15 to 40 days, taken from 4 different batches [19], [21]. Health measurements were calculated for each cell every 12 hours from discharge capacity measurements, and smoothed using a

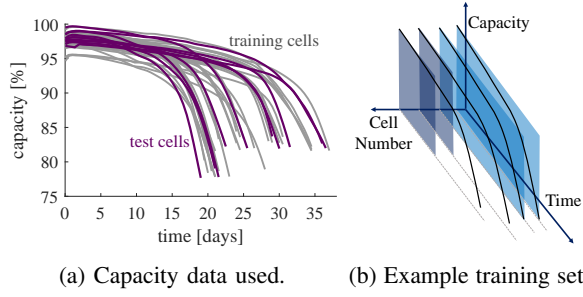


Fig. 7: Trial set up for the limited late life data test. Capacity profiles shown with an example train/test split. The shaded regions in the training data show data is either taken from only early life (dark blue) or the full life (light blue).

moving average. In each trial, 30 test cells and 100 training cells were randomly selected. This whole process was repeated 20 times, giving 600 estimates of the knee point and end of life. Training and test datasets were completely separate. For the purpose of testing, the entire capacity trajectory was forecasted, from day 1, and the knee point and end of life was calculated from that.

The limited late-life data test used a 40 cell subset of the data in reference [19]. In all instances, there were 10 testing cells and 30 training cells, with all cells and an example train/test splitting shown in Fig. 7a. The training sets were used to extract and select 5 features. However data was removed from a random set of cells, leaving between 3 and 30 cells with full life data, schematically shown in Fig. 7b. This replicates a real-life scenario where limited full life data is available.

A short investigation into the impact of varying the maximum shared correlation among the input features was performed. For speed, it used a fixed set of features, but still randomly selected cells for the training and test sets. There were 20 repeats at each of 16 test points, ranging from 1 to 0.7, with 50 cells used for each of training and testing.

Finally, a further test was conducted looking at the relationship between the number of features and the predictive performance. Tests using 1, 2, 3, 4, 5 and 10 features were performed using the large-scale test process detailed above. Further to this, a test with only time is used as the input, and no other features, was used to create a baseline performance for comparison.

IV. RESULTS

A. Feature prioritisation

The most commonly selected features using the process and datasets described above were found to be $V_{2,3}$ and $V_{1,2}$. $V_{2,3}$ was selected every time while $V_{1,2}$ was selected second in 19 trials and removed in the other. Next most commonly selected was calendar time which was selected 18 times.

The length-scale hyperparameters of the GP provide an estimate of how relevant the down-selected features are for predicting capacity transitions. For example, the calendar time feature typically returned a length-scale of around 7 days. Since the data covers a range of 0 to 40 days, this suggests that this input is relevant to capacity fade.

	Mean	Median	95%
$RMSE_{\Delta Q}$ [%]	0.17	0.13	0.39
$RMSE_Q$ [%]	1.1	0.83	3.1
$PE(t_{EoL})$ [%]	2.0	1.3	5.6
$PE(t_{knee})$ [%]	4.2	2.6	14

TABLE III: Summary results from the large scale trial

Error	Mean	Median	95%
EoL [days]	0.49	0.29	1.5
Knee [days]	0.74	0.45	2.48

TABLE IV: Summary results from the large scale test, in units of time (days)

B. Kneepoint and end of life predictions

Fig. 8 shows histograms of prediction accuracy against test set, using the metrics previously introduced. The majority of the root mean square errors on capacity were very small. Table III summarises these results numerically. The median value was 0.83%, while 95% of profiles returned $RMSE_Q < 3.1\%$ capacity (continuous black lines, Fig. 8). The reduction in RMSE frequency at very small values of $RMSE_Q$ (below 0.3%) suggests that the approach avoids overfitting. The capacity transition forecasts exhibited a median RMSE of 0.13% capacity. This, combined with the small $RMSE_Q$ results, leads to good performance for the knee point and end of life forecasting.

Fig. 9 shows a scatter plot of predicted verses observed end of life forecasts. The median absolute end of life forecast percentage error was 1.3%, with a 95th percentile at 5.6% and a mean value of 2.0%.

Knee points were estimated from the capacity fade predictions and compared to the observations. The median absolute value of the knee point position error in time was 2.6%, extending to 14% once 95% of the results are accounted for.

Fig. 9 suggests that there is a reasonably consistent performance across all lifetimes. Table IV presents the same results, but with respect to time. The knee point forecasts were less accurate than the end of life forecasts. This was caused by a slight over-prediction of late life gradient, impacting the knee point calculation but not significantly altering the end of life.

Adding more features as inputs tended to improve the predictive performance of the model in terms of the median error, Table V. However the 95th percentile error did not significantly improve with more than 3 features plus time. Poor results were evident when only using time as the input; other

features	$RMSE_Q$ [%]		$PE(t_{EoL})$ [%]		$PE(t_{knee})$ [%]	
	median	95 th	median	95 th	median	95 th
10	0.54	2.6	0.92	4.5	1.7	13
5	0.83	3.1	1.3	5.6	2.6	14
4	0.95	3.2	1.3	6.2	3.2	13
3	0.96	3.1	1.5	6.2	3.6	13
2	1.3	5.1	1.6	9.0	5.1	16
1	1.8	5.0	2.1	11	7.2	17
time	3.5	12	9.9	32	22	39

TABLE V: Results varying the number of features (plus time). The bottom row is a trial using only time as an input.

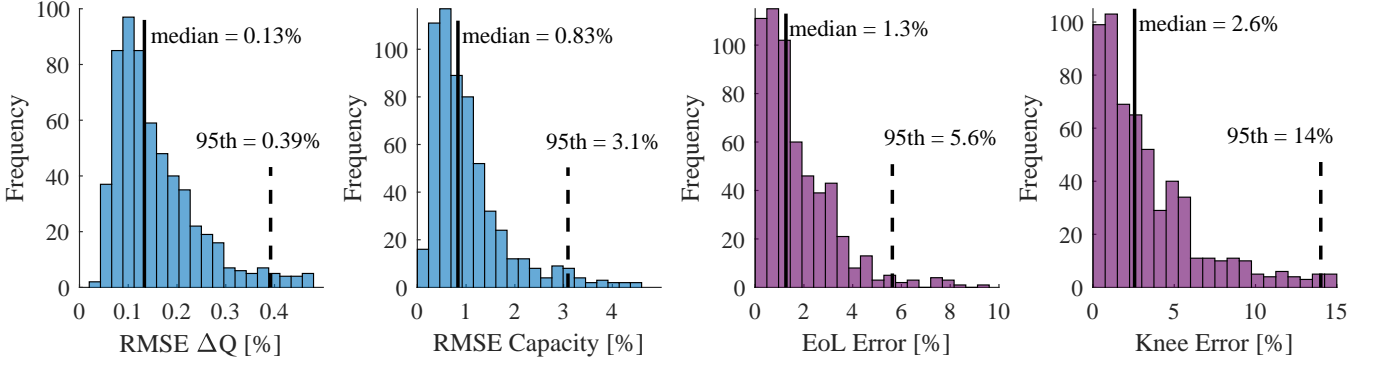


Fig. 8: Histograms of the results for $\text{RMSE}_{\Delta Q}$, RMSE_Q , EoL percentage error and knee point percentage error, with medians and 95th percentiles shown.

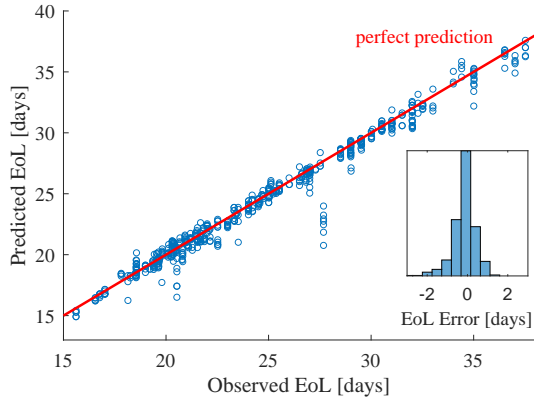


Fig. 9: Scatter plot of observed end of life versus the prediction.

features are also needed for accurate health prognosis.

To test the impact of lower sampling rate on the algorithm, we reduced the raw data to one data point every 100 seconds and found that the algorithm performance was unchanged.

The results of the limited data test are presented in Fig. 10. The median results were accurate using 15 cells worth of late life data, and were relatively consistent from training sets having as little as 6 cells with late life data. The 95th percentiles were variable, but were around 4%, 10% and 20% for the RMSE capacity, end of life percentage error and knee point percentage error respectively.

Finally, we investigated the impact of changing the maximum shared correlation allowed between features, Fig. 11. Elsewhere in this work, the maximum value was 0.85, but here it was varied between 1 and 0.7. As seen, there was a distinct improvement in RMSE capacity under a value of 0.95, while a steadier improvement also occurred for end of life and knee point performance.

V. DISCUSSION

The consistent selection and subsequent performance with the voltage-based features $V_{2,3}$ and $V_{1,2}$ strongly suggests that time spent in specific state of charge ranges had an impact on degradation rate. This is to be expected given the physics of battery degradation [3], since side reactions, such as growth

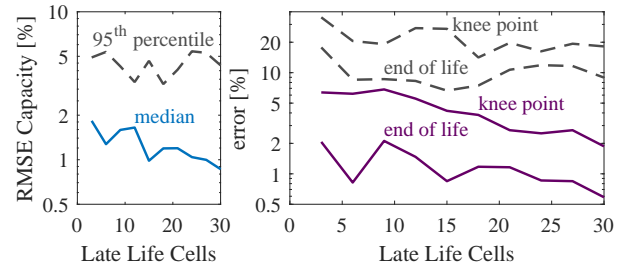


Fig. 10: Comparison of the end of life and knee point errors for the limited data test. The dashed lines represent the 95th percentiles while the solid purple lines represent the median performance across the 40 cells.

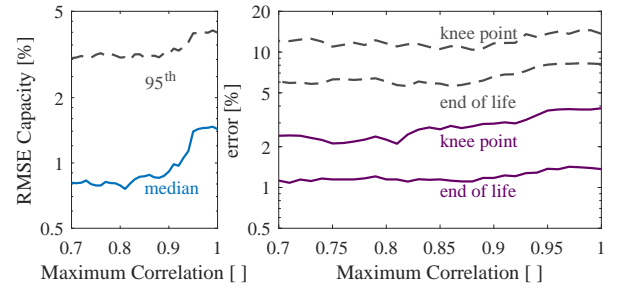


Fig. 11: Comparison of the end of life and knee point errors for the varied maximum shared correlation. The dashed lines represent the 95th percentiles while the solid purple lines represent the median performance across the 1,000 forecasts for each test point.

of the anode solid electrolyte interphase, are dependent on potential. Since the high voltage region feature $V_{3,4}$ and the low voltage region $V_{1,2}$ were found to correlate very closely with the first selected feature, $V_{2,3}$, the latter was generally prioritised. The reason is that as the cells degraded, since voltage limits were reached sooner, greater periods of time were spent at high and low constant voltages during charge and discharge. This in turn reduced the amount of time spent in the voltage mid-range. Therefore the mid-range voltage feature is strongly correlated with ΔQ , because in a single feature it indirectly indicates increased time spent at both high and low

voltages.

Temperature was never returned from the feature selection process despite it being a known factor in battery ageing. All cells were cycled in thermally controlled environments which perhaps meant that a sufficiently wide range of temperatures was not explored to elucidate the dependence of degradation on temperature. Similarly, the automated feature extraction process was found to be robust to significant reductions in data frequency, accuracy and precision in other trials.

Overall, the feature selection approach gave accurate results in the large-scale test, with median percentage errors for end of life predictions of 1.3%. The mean error of 2.0% represented at least a three-fold improvement on previous work published with similar data [7], [19]. This may be due to the use of training data across the entire life of each cell, allowing for variability in usage, rather than point measurements.

The $RMSE_Q$, $RMSE_{\Delta Q}$ and $PE(t_{EoL})$ results suggest that our feature engineering approach produced successful predictions of both capacity and end of life. We speculate that because the relationships between features and capacity changes is generally monotonic and smooth, the linear correlation approach for feature selection works well. The accurate capacity forecasts led to three quarters of knee point estimates lying within a single day of the measured value and 95% of the profiles predicted the knee point within just two and a half days of the observation.

The limited late-life data test produced evidence of the versatility of the approach, even in the face of significantly restricted training data. This makes the technique more viable in the real-world. Nevertheless, the process cannot completely remove the need for comprehensive ageing data to end of life. Data-driven approaches cannot necessarily make accurate predictions outside of the range of their training data.

The clear improvement in accuracy when the allowed maximum correlation between features was reduced shows that a diverse input is crucial for performance. Imposing this condition either avoided overfitting to the training set and/or found input features that would not otherwise be obvious.

There were still a few outlier predictions, and Fig. 9 shows some end of life estimates far from a perfect prediction. Using medians is a more robust measure of overall performance, compared to using mean errors, but because medians ignore the extent of outliers some caution is required [22].

Unfortunately, where there were larger errors between forecasted and observed end of life, this was not matched by having larger credible intervals at these points. The calibration score, the proportion of capacity observations within the predicted 2σ interval [11], was 0.42 for the entire large-scale trial, and for end of life predictions it was up to 0.75. Both values are below the target of 0.95, strongly suggesting the model was overconfident in its uncertainty estimates. Exploring and improving credible intervals provides an interesting avenue for future work, alongside looking at the uncertainty around use prediction. The large-scale of the tests used here allowed for reliable assessment of credible intervals. Without easily accessible large battery data sets, cross-validation represents a strong option to assess uncertainty performance.

The k-fold cross-validation approach for quantifying performance acted as an alternative measure of accuracy instead of credible intervals. The large number of results contained in Fig. 8 showed an estimate of the performance for a typical prediction and a typical poor prediction of health, both useful pieces of information, and usually hidden by the use of a mean average as a summary statistic.

VI. CONCLUSIONS

A combined feature selection and machine learning approach for battery health prediction was proposed and tested, producing knee point forecast errors of 0.45 days or 2.6% across 600 predictions. That success was due to an accurate capacity forecast, with half of all profiles having a root mean square error of under 0.83% capacity when predicting over full lifetimes. A further trial showed that, as might be expected, having more data generally led to better predictive performance. The results also showed that this approach is capable of handling lesser qualities and quantities of data without unduly impacting performance for end of life prediction, despite the changing degradation rates. K-fold cross-validation produced sufficient results to calculate multiple summary statistics. Medians and higher percentiles are an informative pair of measures when used in conjunction and future work should aim to use them, especially for any asymmetric error measures.

Interesting open questions remain regarding the presented procedure and data-driven approaches in general. The number of features required could be further investigated, using a wider range of datasets. Other areas of research are improved credible intervals and tackling datasets without regular health measurements. In a real-world application, the algorithm will still be applicable and be able to extract features relating to battery usage. However, regular measurements of health may not be available in the same way as for a laboratory dataset. Solving this is an open challenge that will require new approaches, for example unsupervised learning, or using a different and more general health metric to train the model.

The feature selection approach using easily understood features provides a level of insight unavailable from black box machine learning techniques. Information on how different features correlate with one another, and which features impact degradation, is extremely useful to a user alongside the accurate forecasts of capacity, end of life and the knee point.

REFERENCES

- [1] C. R. Birkl, M. R. Roberts, E. McTurk, P. G. Bruce, and D. A. Howey, "Degradation diagnostics for lithium-ion cells," *Journal of Power Sources*, vol. 341, pp. 373-386, 2017.
- [2] Y. Li, K. Liu, M. Foley, A. Zulke, M. Bercibar, E. Nanini-Maury, J. Van Mierlo, and H. E. Hoster, "Data-driven health estimation and lifetime prediction of lithium-ion batteries: a review," *Renewable and Sustainable Energy Reviews*, vol. 113, 2019.
- [3] J. Reniers, G. Mulder, and D. Howey, "Review and Performance Comparison of Mechanical-Chemical Degradation Models for Lithium-Ion Batteries," *Journal of the Electrochemical Society*, vol. 166, pp. A3189-A3200, 2019.
- [4] G. Dong, F. Yang, Z. Wei, J. Wei, and K. Tsui, "Data-Driven Battery Health Prognosis Using Adaptive Brownian Motion Model," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 4736-4746, 2020.

- [5] K. Goebel, B. Saha, J. Liu, A. Saxena, and W. Wang, "An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries," in *Conference of the Prognostics and Health Management Society*, 2010.
- [6] J. Wu, C. Zhang, and Z. Chen, "An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks," *Applied Energy*, vol. 173, pp. 134–140, 2016.
- [7] P. Fermín, E. McTurk, M. Allerhand, E. Medina-Lopez, M. F. Anjos, J. Sylvester, and G. dos Reis, "Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells," *Energy and AI*, vol. 1, in press, 2020.
- [8] M. A. Patil, P. Tagade, K. S. Hariharan, S. M. Kolake, T. Song, T. Yeo, and S. Doo, "A novel multistage support vector machine based approach for Li-ion battery remaining useful life estimation," *Applied Energy*, vol. 159, pp. 285–297, 2015.
- [9] Q. Zhao, X. Qin, H. Zhao, and W. Feng, "A novel prediction method based on the support vector regression for the remaining useful life of lithium-ion batteries," *Microelectronics Reliability*, vol. 85, pp. 99–108, 2015.
- [10] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Gaussian process regression for forecasting battery state of health," *Journal of Power Sources*, vol. 357, pp. 209–219, 2017.
- [11] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Battery health prediction under generalized conditions using a Gaussian process transition model," *Journal of Energy Storage*, vol. 23, pp. 320–328, 2019.
- [12] K. Goebel, B. Saha, A. Saxena, J.R. Celaya, and J.P. Christophersen, "Prognostics in battery health management," *IEEE Instrumentation and Measurement Magazine*, vol. 11, pp. 33, 2008.
- [13] K. Liu, Y. Li, X. Hu, M. Lucu and W. D. Widanage, "Gaussian Process Regression With Automatic Relevance Determination Kernel for Calendar Aging Prediction of Lithium-Ion Batteries," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 3767–3777, 2020.
- [14] X. Li, Z. Wang, and J. Yang, "Prognostic health condition for lithium battery using the partial incremental capacity and Gaussian process regression," *Journal of Power Sources*, vol. 421, pp. 56–67, 2019.
- [15] P. Tagade, K. S. Hariharan, S. Ramachandran, A. Khandelwal, A. Naha, S. Mayya, Kolake, and S. H. Han, "Deep Gaussian process regression for lithium-ion battery health prognosis and degradation mode diagnosis," *Journal of Power Sources*, vol. 445, pp. 227–281, 2020.
- [16] X. Hu, Y. Che, X. Lin, and Z. Deng, "Health prognosis for electric vehicle battery packs: a data-driven approach," *IEEE/ASME Transactions on Mechatronics*, doi: 10.1109/TMECH.2020.2986364, 2020.
- [17] M. Lucu, E. Martinez-Laserna, I. Gandiaga, K. Liu, H. Camblong, W. D. Widanage, and J. Marco, "Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data - part B: cycling operation," *Journal of Energy Storage*, vol. 30, in press, 2020.
- [18] X. Zheng, and X. Deng, "State-of-health prediction for lithium-ion batteries with multiple Gaussian process regression model," *IEEE Access*, vol. 7, pp. 150383–150394 2019.
- [19] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggadakis, M. Z. Bazant, S. J. Harris, W. C. Chueh, and R. D. Braatz, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, pp. 383–391, 2019.
- [20] T. Baumhöfer, M. Brühl, S. Rothgang, and D. U. Sauer, "Production caused variation in capacity aging trend and correlation to initial cell performance," *Journal of Power Sources*, vol. 247, pp. 332–338, 2014.
- [21] P. M. Attia, A. Grover, N. Jin, K. A. Severson, T. M. Markov, Y-H. Liao, M. H. Chen, B. Cheong, N. Perkins, Z. Yang, P. K. Herring, M. Aykol, S. J. Harris, R. D. Braatz, S. Ermon, and W. C. Chueh, "Closed-loop optimization of fast-charging protocols for batteries with machine learning," *Nature*, vol. 578, pp. 397–402, 2020.
- [22] K. P. Murphy, "Machine learning: a probabilistic perspective," MIT Press, 2012.
- [23] Z. Deng, X. Hu, X. Lin, L. Xu, Y. Che, and L. Hu, "General Discharge Voltage Information Enabled Health Evaluation for Lithium-Ion Batteries," *IEEE/ASME Transactions on Mechatronics*, 2020.
- [24] W. Diao, S. Saxena, B. Han, and M. Pecht, "Algorithm to determine the knee point on capacity fade curves of lithium-ion cells," *Energies*, vol. 12, pp. 2910–2919, 2019.
- [25] X. Hu, Y. Che, X. Lin, and S. Onori, "Battery health prediction using fusion-based feature selection and machine learning," *IEEE Transactions on Transportation Electrification*, 2020.
- [26] Z. Deng, X. Hu, X. Lin, Y. Che, L. Xu, and W. Guo, "Data-driven state of charge estimation for lithium-ion battery packs based on Gaussian process regression," *Energy*, vol. 205, 2020.
- [27] C. E. Rasmussen, and C. K. I. Williams, "Gaussian processes for machine learning," MIT Press, 2006.
- [28] B. Lund, and B. Foss, "Parameter ranking by orthogonalization - applied to nonlinear mechanistic models," *Automatica*, vol. 44, pp. 278–281, 2008.