Mechanical Systems and Signal Processing 184 (2023) 109612

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

# Piecewise-linear modelling with automated feature selection for Li-ion battery end-of-life prognosis

Samuel Greenbank, David A. Howey *

*Battery Intelligence Lab, Department of Engineering, University of Oxford, OX1 3PJ, UK*

ARTICLE INFO

Communicated by J.E. Mottershead

*Keywords:*
Feature selection
Linear model
Piecewise
Lithium-ion
Degradation
Health
Bayes

ABSTRACT

The complex nature of lithium-ion battery degradation has led to many machine learning-based approaches for health forecasting being proposed in the literature. However, machine learning using sophisticated models can be computationally expensive, and although linear models are faster they can also be inflexible. Piecewise-linear models offer a compromise—a fast and flexible alternative that is not as computationally expensive as techniques such as neural networks or Gaussian process regression. Here, a piecewise-linear approach for battery health forecasting, including an automated feature selection step, is compared to a Gaussian process regression model and found to perform equally well in terms of the median error on a training dataset, and indeed somewhat better at the 95th percentile of error. The feature selection process demonstrates the benefit of limiting the correlation between inputs. Further trials found that the piecewise-linear approach was robust to changing input size and availability of training data.

## 1. Introduction

A significant focus of recent battery literature has been on developing data-driven approaches for modelling degradation [1]. Many of the suggested approaches use machine learning (ML) techniques to try to predict current battery state of health (SoH), future SoH, or remaining useful life (RUL). However, training a machine learning model, such as a neural network, can be computationally challenging—either needing extensive quantities of data, and/or scaling poorly as the dataset grows in size. With some approaches (e.g. Gaussian processes), making predictions requires all the training data to be available, posing a challenge for memory requirements and a hindrance for use in control systems [1]. Linear models are a simpler alternative to other ML approaches [2] that have already demonstrated some success in battery applications. Here, we propose a piecewise-linear model for forecasting battery capacity without compromising predictive performance relative to a more complex machine learning approach. This paper is focused on SoH prognosis (i.e., estimate future battery health and lifetime given an assumption of a usage scenario) rather than on SoH diagnosis (i.e., estimate present health given measured data). A full discussion of these two topics is available in the literature, e.g. in Sulzer et al. [3].

Battery degradation is complex, caused by a wide range of possible mechanisms, and compounded by dependence on usage and cell-to-cell manufacturing variability [4–7]. Degradation is usually defined as the change in SoH with usage, either as a fade in the capacity or increase in resistance of a battery cell (or some combination of these metrics, depending on the application). The flexibility of supervised machine learning approaches enables accurate models to be constructed from data for the prediction of battery SoH, RUL or 'knee point', when the health decline begins to accelerate [1,8–13]. Combined with ML

* Corresponding author.
*E-mail address:* david.howey@eng.ox.ac.uk (D.A. Howey).

https://doi.org/10.1016/j.ymssp.2022.109612
Received 15 January 2022; Received in revised form 10 June 2022; Accepted 20 July 2022

## Nomenclature

### Acronyms/Abbreviations

| | |
|---|---|
| EoL | End of life |
| EoL Error | End of life percentage error |
| RMSE Capacity | Root mean square error on capacity predictions |
| RMSE $\Delta Q$ | Root mean square error on change in capacity predictions |
| RUL | Remaining useful life |
| SoH | State of health |

### Mathematical Symbols

| | |
|---|---|
| $\beta_{\text{improv}}$ | Performance threshold in model selection |
| $\beta_l$ | Typical lengthscale of the splitting variable |
| $\Delta Q$ | Change in capacity |
| $\epsilon$ | Noise over target variables |
| $\rho(x)$ | Data density function |
| $\rho_P$ | Pearson's rank correlation |
| $\rho_{P,\max}$ | Maximum shared correlation between input features |
| $\sigma_n$ | Standard deviation of observation noise |
| $a$ | Weighting function used for smoothing |
| $f(X)$ | Function of input array, $X$ |
| $f_{bp}(x)$ | Breakpoint threshold function |
| N | Normal distribution |
| $n_m$ | Number of sub-models |
| $Q$ | Capacity |
| $\hat{t}_{\text{EoL}}$ | Predicted lifetime |
| $t_{\text{EoL}}$ | Observed lifetime |
| $V_{i,j}$ | Proportion of time spent between $i$th and $j$th voltage percentiles |
| $w$ | Coefficients of a linear model |
| $\hat{w}$ | Estimate of coefficients of model |
| $X$ | Input array with a data point per row |
| $y$ | Target variable |
| $*$ | Subscript indicating test data |

### Units

| | |
|---|---|
| Ah | Ampere hours |
| C | C-rate |

models, correlation-based feature selection techniques have also been shown to be useful in establishing the key factors that drive battery ageing [14–16].

Linear correlations have been found between various battery-ageing stimuli/indicators, and SoH. For example, the positions of peaks within and integrals of incremental capacity curves have been found to vary linearly with capacity loss [16–22]. Similarly, the cell thermal response may also exhibit a linear relationship with capacity [23,24]. Other features that may correlate almost linearly with battery health are time in certain voltage regions [16,25–27], internal stresses [19,28] and combinations of impedances [29]. The above-mentioned examples were found in data from controlled experimental scenarios, so may not be universal, but the existence of linear or almost linear relationships between specific ageing features and battery health, across a range of use cases, suggests that this is an avenue worth further exploration.

Having said this, depending on the battery ageing dataset, a very simple linear mapping from a battery ageing feature to SoH may lack sufficient flexibility to accurately capture certain important aspects of degradation. Notably, in some datasets, a "*knee point*" appears in capacity fade trajectories for lithium-ion cells undergoing arduous test protocols, and this manifests as a sudden collapse in capacity [9,30–32]. Such rapid changes in SoH may be associated with input features moving from varying linearly with SoH to having undefined non-linear relationships with SoH [30,33].

Piecewise-linear models, which consist of a number of separate linear models each valid within a certain constrained range of behaviour, offer a compromise between simplicity and flexibility. In the context of batteries, they have been used for state of charge modelling [6,27,34–36] and SoH modelling [2,37,38]. A common approach for piecewise-linear models is to split the linear

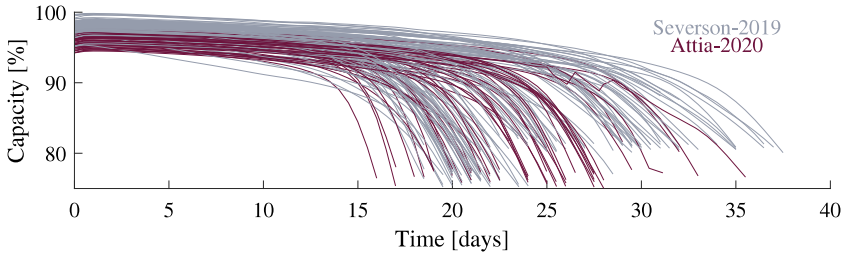


**Fig. 1.** Battery capacity fade measurement data used in this work, from Refs. [10,44].
*Source:* Image adapted from Ref. [15].

sub-models according to the stages of cell life [6,34,36,38]. However since cells degrade at different rates even if used identically, time-based splitting of linear sub-models may give poor performance [39]. Other approaches have separated linear sub-models according to voltage or state of charge regions [35,40,41]. It has even been possible to construct separate linear sub-models according to how a cell is being used [42] or to use local linear regression, where linear models are constructed based on a number of nearest neighbours, to predict capacity loss [43].

Here we propose new approaches for piecewise-linear regression models to forecast battery SoH trajectories up to end-of-life. The approach automates the locations of the boundaries between the linear sub-models and the selection of the input variable used to split the sub-models. The number of linear sub-models is also chosen automatically based on a compromise between complexity and performance. To test the approach, a comparison with a Gaussian process regression machine learning approach was performed. Gaussian processes have been widely explored for battery degradation prediction in the literature [8,11,13–15] and have proven to be a powerful tool, offering a challenging comparison to the piecewise-linear method proposed here. A number of other investigations were also undertaken to assess the resilience of the proposed approach to varying modelling conditions.

## 2. Data sources

This work uses open-source battery cycling and capacity fade data from two datasets [10,44]. The datasets were produced from fast-charging experiments using lithium iron phosphate/graphite 18650 cells, manufactured by A123, all cycled in a temperature chamber set at 30 °C. Capacity estimates were calculated from the repeated 4C discharge cycles. In this paper, capacity is expressed per cent, normalised relative to the nominal cell capacity of 1.1 Ah.

In these datasets, all cells were cycled to failure, defined as 80% of the 1.1 Ah nominal capacity. The first dataset contained 135 cells, with peak charging currents varying between 3.6C and 8C [10]. The second set, with a further 45 cells, had a fixed charging window of 10 min but different paths to full charge [44]. Here, all cells with lifetimes between 15 and 40 days were chosen, to provide a consistent dataset. After these filtering steps, the data from 157 cells was available (Fig. 1) for model fitting and validation. For completeness, the methods were also re-run against the full dataset of 175 cells (with no filtering out of cells first).

## 3. Methods

### 3.1. Data generation and selection

The battery cycling data was reduced from hundreds of millions of rows down to thousands of rows by generating input features from the raw data. For each 12 h time period, the input features were calculated based on cell use during that time period, as described in detail below. The available measured variables from which to generate features were current, voltage, temperature, power, absolute current and absolute power. A capacity measurement was taken at the end of every 12 h time period for which a cell was tested.

This feature generation approach is very similar to the method used in our previous work [15], building also on Richardson et al. [13], and briefly described as follows: The raw data consists of voltage, current and temperature time-series (inputs) and capacity measurements (outputs). The model aims to learn the mapping from inputs to outputs. Rather than learn a direct mapping to absolute values of capacity, the model learns the mapping to change in capacity associated with the input features within a fixed period of time (which is arbitrary, but in this case was 12 h) – as shown in Figures 1 and 2 of Richardson et al. [13]. Capacity fade trajectories over life can then be predicted by adding up these capacity changes cumulatively.

Features can be created from any functions applied to the raw data; in this case features were generated based on time spent within certain input regions, for every fixed 12-hour period. This may be illustrated by a brief example. Consider two batteries, one aged at high temperature between 0%–100% SOC, another at room temperature between 40%–60% SOC. Using temperature and voltage as the key inputs relating to usage, the first step is to compute histograms across all the data. These are then divided into distinct zones, using percentiles. In this example we would expect to have a room temperature region and a high temperature region, and also low voltage (low SOC), medium voltage, and high voltage regions. The number of regions is up to the user, but this can be explored to find a trade-off between flexibility and accuracy. Once the region boundaries have been ascertained from the

**Table 1**
Variable bounds used to define regions to generate input features according to the datasets used in this work.

| Percentile | Current [A] | Voltage [V] | Temperature [°C] | Power [W] | Absolute current [A] | Absolute power [W] |
|---|---|---|---|---|---|---|
| 1st | −4.00 | 2.00 | 29.8 | −12.8 | 0.00 | 0.0 |
| 33rd | −0.45 | 3.12 | 32.5 | −0.9 | 1.00 | 2.8 |
| 67th | 1.00 | 3.51 | 35.0 | 3.4 | 4.00 | 12.4 |
| 99th | 6.00 | 3.60 | 40.6 | 21.3 | 6.00 | 21.3 |

data, these are fixed. The individual feature values are then calculated simply as the proportion of time spent within each region within each fixed time window. In the brief example given here, we would therefore expect the high temperature cycled battery to have non-zero temperature feature values in the high temperature region and zero values of the feature associated with the ambient temperature region. In this way, the feature rows capture the essence of how the battery was cycled within each 12-hour period. This approach gives flexibility to capture different types of cycling patterns e.g. within a single battery test, or between the training and test sets.

For speed, the regions were bounded according to the thresholds derived using the full dataset (rather than just the training set). This was because it was found that the selected thresholds were very consistent across many randomly selected subsets of data. Every variable was split up according to how much time was spent within each region so that a cumulative distribution could be produced from which to draw the bounds. The calculated bounds are given in Table 1. For example, the bounds at the 33rd percentile represent the values of the input variables below which the 157 cells cumulatively spend 33% of their lifetimes. This approach builds on techniques from existing literature where linear relationships have been found between battery ageing and time spent in certain voltage regions [15,16,25–27].

From this process, there were 36 possible input features calculated for each cell considering all variables over all variable ranges. Another 36 were also added by including how each variable changed between time intervals, in addition to the final input feature being calendar time, giving 73 features in total. These were calculated for each 12 h interval of data, for each cell's entire life. The output variable was the change in capacity $\Delta Q$ over each 12 h time interval, with capacity estimated from the 4C discharge cycle at each time period.

In order to select the most important input features and reduce the total number of features required, the input features which correlated most strongly with the changes in capacity, $\Delta Q$, were prioritised. Pearson's rank, $\rho_P$, was used to quantify the degree of correlation. To ensure that input features covered a wider range of variability, selected features were not allowed to share a correlation coefficient more than $\rho_{P,\max} = 0.85$.

Features are labelled here according to the raw variable used (e.g. $V$ for voltage) and the percentiles which define the thresholds. For example, the most important feature impacting ageing was commonly found to be $V_{2,3}$, i.e. the proportion of time spent between the second and third variable bounds of voltage within each 12-hour window—the 33rd percentile to the 67th percentile, equivalent to between 3.12 V and 3.51 V. Five features were down-selected and used as inputs for the piecewise-linear model. The most commonly selected input features were, in priority order, $V_{2,3}$, $V_{1,2}$, experimental time, $|P|_{1,2}$ (absolute power), and $|I|_{1,2}$ (absolute current). Later in this paper (Section 3.6) there is also a sensitivity analysis that assesses the required number of input features in more detail.

The code developed in this work is available online from the following repository: https://github.com/Battery-Intelligence-Lab/piecewise-linear-rul

### 3.2. Piecewise-linear model breakpoints

A piecewise-linear model consists of a number of separate linear sub-models joined together, each respectively explaining a subsection of the overall behaviour of the data. To construct the model one must therefore find, from the training data, the optimal breakpoints between the linear sub-models.

As discussed in the previous section, the input features were prioritised according to their correlation with capacity fade. Consequently, the feature with the highest correlation was used to establish the breakpoints for the piecewise-linear model. First, a monotonic function $\Delta Q = \Delta Q(x)$, where $x$ is the highest priority input feature, was created directly from the data using a weighted moving average, shown as a black line in Fig. 2(a). The second derivative of this function (weighted by the data density) was used to find the points of maximum curvature, see Fig. 2(b), and between these points, linear sub-models were fitted.

The weights, $a$, were calculated from the squared exponential of the distance between the $i$th value being evaluated, $x_i$, and the $x$-values at other data points, $x_j$:

$$a_{i,j} = a(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{\beta_l^2}\right), \tag{1}$$

where

$$\beta_l = \frac{\max(x) - \min(x)}{10}.$$

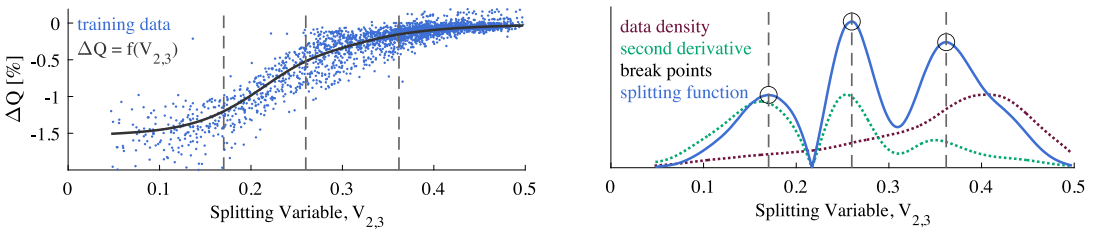


**Fig. 2.** Example calculation of three breakpoints using the best correlating feature, $V_{2,3}$. Left to right (a) $\Delta Q(V_{2,3})$, (b) resultant splitting points.

The function $f_{\Delta Q}(x)$ was then calculated at each point $x_i$ using all $n$ output data points $\Delta Q$ in the training set (Eq. (2)),

$$f_{\Delta Q}(x_i) = \sum_{j=1}^{n} \frac{a_{i,j}\Delta Q(x_j)}{a_{i,j}}. \tag{2}$$

However, this moving average may have large values for the second derivative at the extreme values of input feature $x$ where there are fewer relevant data points. To alleviate this, a data density function, $\rho = \rho(x_i)$, was calculated (Eq. (3)) and multiplied by the second derivative of $f_{\Delta Q}$ (calculated numerically through simple differencing) so that changing gradients in regions with lots of data points were prioritised, as follows,

$$\rho(x_i) = \frac{1}{n}\sum_{j=1}^{n} H\left(\|x_i - x_j\| < \sigma_l\right), \tag{3}$$

where $H$ is a logical function returning 1 when the condition is true and $\sigma_l$ is a lengthscale parameter which was set equal to one-tenth of the range of the splitting variable. We investigated the impact of varying this value and found that results were relatively insensitive to this. The final expression for the breakpoint selection function, $f_{bp}$, was therefore

$$f_{bp}(x_i) = \rho(x_i) \times \left[\frac{d^2 f_{\Delta Q}}{dx^2}\right]_{x=x_i}.$$

The process is illustrated in Fig. 2(b), where the values at the dotted lines are multiplied together to produce the final function for breakpoint selection. Maxima in that function matched, or were near to, the significant changes of gradient in the training set.

Other approaches to piecewise model splitting were considered (details are in Appendix A). K-means clustering and Matlab's *fminsearch* functions were both implemented as a comparison. K-means was initially used over the full input data, but that gave too high a weighting to features not correlated with the output. K-means was found to be most successful by using just the first two high priority features, at which point the results are extremely similar to using curvature, as explained above. Matlab's *fminsearch* allowed for completely free breakpoints selection across the range of the first selected feature.

### 3.3. Linear regression of sub-models

The sub-model for each partition was fitted using Bayesian linear regression. The target variable $y$ was $\Delta Q$ in each time step. This was assumed to be a linear function of input feature set $X$ with associated noise, $\epsilon$,

$$y = f(X) + \epsilon, \quad \epsilon \sim \mathrm{N}(0, \sigma_n^2).$$

Bayesian linear regression involves fitting the parameter vector $w$ to create model $f(X) = Xw$ based on the posterior distribution over the parameters. The input matrix $X$ has a column for every input and a row for every data point. The model parameters $w$ were assigned a zero-mean Gaussian prior and covariance $\Sigma_w$,

$$w \sim \mathrm{N}\left(0, \Sigma_w\right).$$

Here, $\Sigma_w$ was assumed to be a diagonal matrix with a constant variance, $\sigma_w^2 = 10^2$, with this value set by observations of the coefficients in early trials. Varying the prior parameter uncertainty $\sigma_w$ was found to have no impact on performance within an order of magnitude of $\sigma_w = 10$, probably because of the large amount of measurement data. This leads to a mean estimate of the parameters $w$ as a function of the input variables $X$, output target $\Delta Q$ and estimates of observation noise $\sigma_n$ and covariance $\Sigma_w$ [45,46],

$$\hat{w} = \sigma_n^{-2}\left(\sigma_n^{-2}X^T X + \Sigma_w^{-1}\right)^{-1} X^T y. \tag{4}$$

Predictions of capacity loss are produced by multiplying the test set input matrix, $X_*$, by the parameter estimates,

$$y_* = \Delta Q_* = X_* \hat{w}. \tag{5}$$

**Table 2**
Piecewise model selection by choosing the smallest number of models $n_m$ within threshold $\beta_{\text{improv}}$ of the peak performance.

| $n_m$ | RMSE $\Delta Q$ [%] | $\leq 1 + \beta_{\text{improv}}$ | Selection |
|---|---|---|---|
| 1 | 0.325 | | |
| 2 | 0.213 | | |
| 3 | 0.201 | | |
| 4 | 0.192 | 0.192 | $n_m = 4$ |
| 5 | 0.192 | 0.192 | |
| 6 | 0.192 | 0.192 | |
| 7 | 0.210 | | |
| 8–10 | n/a | | |

### 3.4. Piecewise-linear model construction

The number, $n_m$, of sub-models used in the final piecewise model was calculated by a compromise between predictive performance and complexity. The procedure is depicted in Table 2. Piecewise-linear models each having different numbers of sub-models were trained on the training set up to some maximum number of sub-models, taken as 10 in the work here. The selected $n_m$ was the minimum model size that gave an accuracy below the optimal root mean squared error (RMSE) $\Delta Q$ score multiplied by a threshold $(1 + \beta_{\text{improv}})$. For most models here, $\beta_{\text{improv}} = 0.01$ was chosen as a threshold.

The piecewise-linear model produces a series of $\Delta Q$ estimates over time. Capacity forecasts are then calculated over full cycle life of each test cell by summing the predicted $\Delta Q$ estimates in order, starting from assumed initial capacity value.

### 3.5. Performance metrics

To evaluate the predictive accuracy of the model, three performance metrics were calculated for each forecasted capacity profile. Firstly, the root mean squared error (RMSE) between observed and predicted $\Delta Q$ provided a measure of the performance of the model. In addition to this, the capacity profile forecast quality was calculated by using the root mean square error on the capacity, RMSE Capacity, calculated in % capacity.

Another metric used to assess performance was the lifetime prediction accuracy, with end of life (EoL) point defined as the time when 80% nominal capacity was reached. The percentage difference between observed EoL, $t_{\text{EoL}}$, and predicted, $\hat{t}_{\text{EoL}}$, was taken as the error to create the metric $\text{PE}_{\text{EoL}}$,

$$\text{PE}_{\text{EoL}} = 100\% \times (t_{\text{EoL}} - \hat{t}_{\text{EoL}})/t_{\text{EoL}}.$$

Many trials were performed at all test points here so that the quoted forms of the above metrics are the median and 95th percentiles of the above performance metrics.

### 3.6. Trial setup

The first trial was a comparison in performance of the piecewise-linear model versus a baseline machine learning tool, Gaussian process regression (GPR). The same data and features were used as the inputs for the piecewise-linear and GPR models, and performance was evaluated using the metrics described above. There were 200 repeats of the trial with each repeat using 50 training cells and 107 test cells.

In addition to this, a sensitivity analysis was conducted to test the piecewise-linear model and feature selection approach whilst varying assumptions on maximum feature similarity $\rho_{P,max}$, improvement threshold $\beta_{\text{improv}}$, number of input features, and number of sub-models. These assumptions were tested one at a time whilst keeping all other parameters constant at the default values used in the larger scale trial previously described. These default values were $\rho_{P,max} = 0.85$, $\beta_{\text{improv}} = 0.01$, $\max(n_m) = 10$ sub-models, 5 input features and 50 training cells. This trial used 20 repeats at each test point each with 107 test cells.

Third, we calculated the EoL error as a function of the number of training cells used, between 5 and 100. In this case we used K-means and *fminsearch* to select linear sub-model splitting points, to provide a thorough comparison with the curvature-based piecewise splitting techniques. Apart from the splitting technique, the parameters of this trial was identical to the first trial.

Finally, to test the behaviour of the algorithm using all available data, the code was re-run with all 175 cells, i.e. including outliers with very long and short lives.

## 4. Results

Fig. 3 shows the results of the piecewise-linear model trials calculated with the performance metrics previously described. The distribution of the results suggests that the piecewise-linear approach produces accurate forecasted capacity profiles. The median RMSE $\Delta Q$ of 0.17% capacity for the capacity transition model represents an accurate fit. The performance when calculating capacity

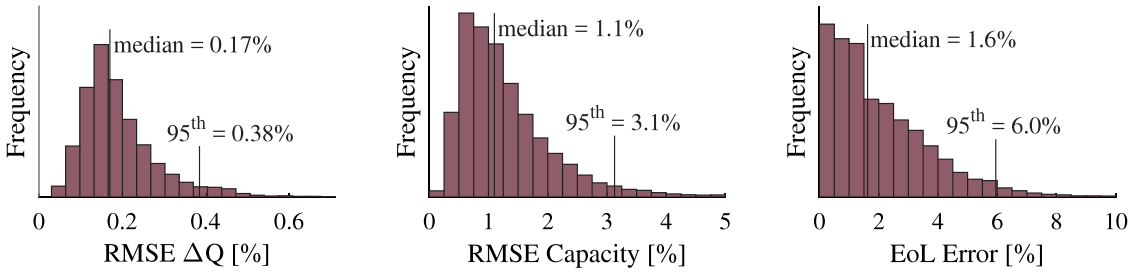


**Fig. 3.** Piecewise-linear model performance in terms of RMSE on capacity changes in each time step, capacity over the whole trajectory, and end of life.

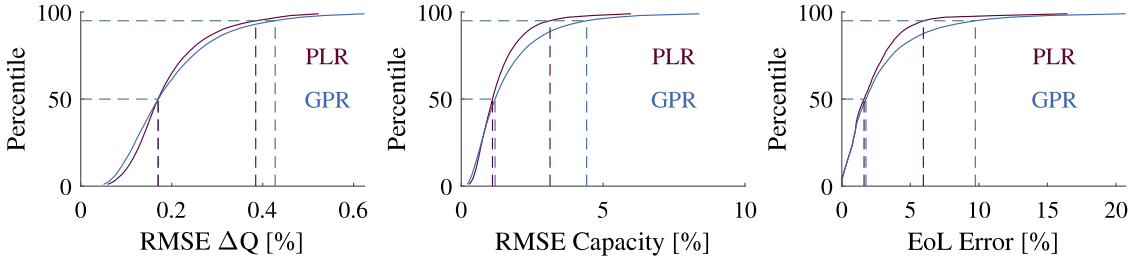


**Fig. 4.** Comparison between piecewise-linear modelling and GPR for capacity forecasting.

**Table 3**
Comparison of performance between full and reduced datasets.

| | Reduced dataset (157 cells) | | Full dataset (175 cells) | |
|---|---|---|---|---|
| Performance metric | median | 95th percentile | median | 95th percentile |
| RMSE $\Delta Q$ [%] | 0.17 | 0.38 | 0.18 | 0.47 |
| RMSE Capacity [%] | 1.1 | 3.1 | 1.5 | 4.6 |
| EoL Error [%] | 1.6 | 6.0 | 1.5 | 10.0 |

trajectories, quantified by RMSE Capacity, was also typically accurate, with a median error value of 1.1% capacity. Finally, the predicted lifetimes of the cells were accurate to within 1.6% of the observed lifetime in half the test cases.

The median performance of the piecewise-linear modelling approach was extremely similar to that of the GPR approach. The 95th percentiles of error were slightly improved by using piecewise-linear modelling relative to GPR in Fig. 4, although the difference is small. The piecewise-linear approach used between 3 and 5 linear sub-models to map the capacity loss based on the data used here, in 79% of cases. In all cases, the input feature used to calculate the thresholds for the breakpoints between the linear sub-models was $V_{2,3}$ which is the proportion of time spent in the mid-range of voltages, between 3.12 V and 3.51 V.

Median lifetime predictive performance was found to be unaffected by significant changes in modelling assumptions, Fig. 5. Any number of sub-models $n_m$ greater than 1 appeared equally accurate. The threshold $\beta_{\text{improv}}$ could be raised to 0.5 without impacting performance. The input feature selection procedure worked most accurately when maximum shared correlation between features was $0.6 < \rho_{P,max} < 0.9$. On the other hand, the 95th percentiles of prediction error varied more as a function of the modelling assumptions. For example, increasing the number of input features reduced the end of life estimation error up to 3 features, above which the improvements were small.

The three sub-model splitting methods presented in Fig. 6 gave very similar results, although using *fminsearch* was found to be comparatively more computationally expensive with bigger training sets. All three methods produced good performance with data from 20 or more training cells. There was only a small improvement in accuracy by increasing the training set to 100 cells' data.

The histogram of sub-model breakpoints, Fig. 7(a), shows how there were generally common breakpoints chosen between the linear sub-models. The most common was at $V_{2,3} \approx 0.37$, which roughly corresponds to the change from slow linear ageing to faster degradation, Fig. 2.

As mentioned, to explore the performance of the algorithm using all available data (including outliers), the code was re-run with all 175 cells. The results are similar to the results already shown, but slightly weaker. They are summarised in Table 3.

## 5. Discussion

The piecewise-linear model for battery capacity fade produced accurate capacity forecasts and accurate lifetime predictions. For the same sets of input features, piecewise-linear modelling matched Gaussian process regression with respect to median error performance, and outperformed GPR for the 95th percentile of performance. This may be because the piecewise-linear approach

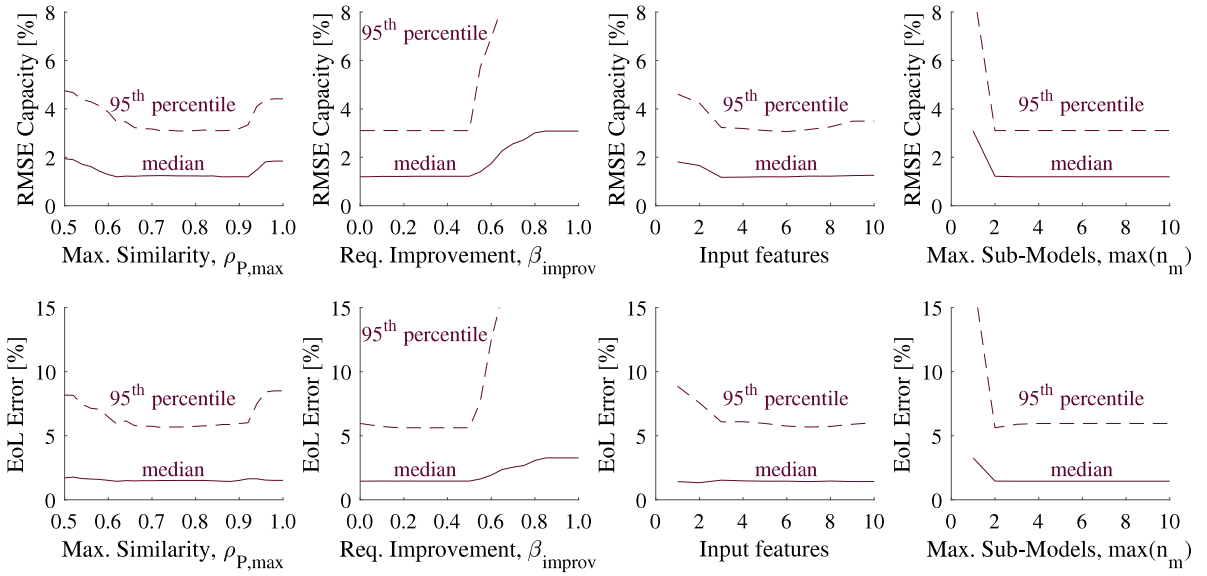


**Fig. 5.** Median and 95th percentile of the capacity and end of life predictive error of the piecewise-linear approach against the maximum correlation among input features, sub-model performance threshold, the number of input features and the maximum number of sub-models.

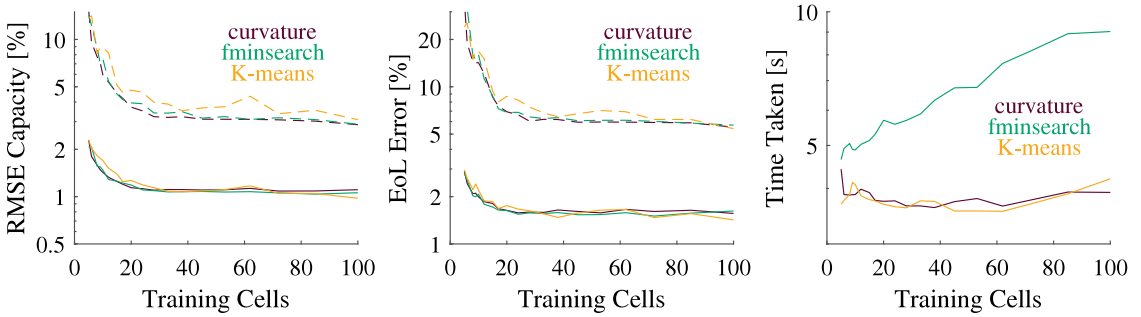


**Fig. 6.** Median and 95th percentile of the capacity and end of life errors against the number of training cells, plus computational time. Results shown for three sub-model splitting techniques: curvature (purple), K-means (yellow) and *fminsearch* (green).

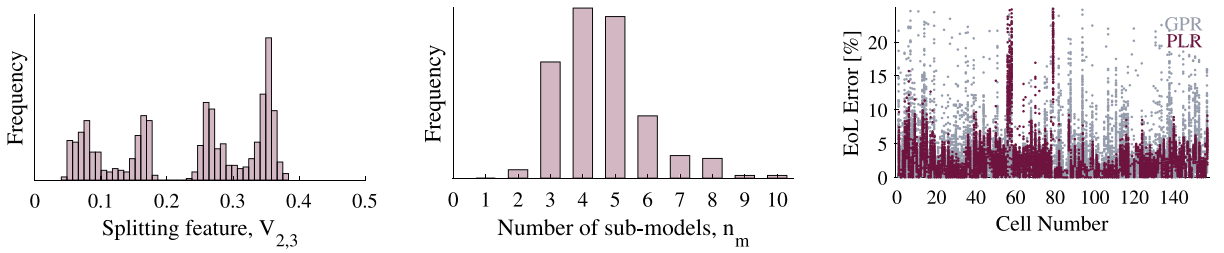


**Fig. 7.** Analysis of the piecewise-linear approach and its results. Left to right: (a) Histogram of breakpoints chosen between sub-models, (b) Number of sub-models used in the large trial in Fig. 3, (c) EoL errors for Gaussian process regression and piecewise-linear regression, sorted by test cell number.

appears less susceptible to small patterns in the training data, potentially reducing overfitting. Using over 8 input features produced weaker performance, Fig. 5, suggesting that the model was beginning to overfit the data in that case.

The distinction between the two data-driven approaches (piecewise-linear and GPR) appears to be related to a small number of outliers. Fig. 7(c) directly compares the end of life error results of the two approaches, grouping them by test cell. The GPR results for the majority of cells were distributed more widely than the piecewise-linear model results. The same effect was seen at higher error percentiles in Fig. 4. Most data-driven approaches struggle to extrapolate beyond their training data, but linear approaches

could diverge more slowly in this case, thus reducing the impact of a substandard model at extremes. With a typical value of 4 linear sub-models used, the piecewise-linear approach was sufficiently flexible to map the trajectories of these rapidly degrading cells. The uniformly-selected primary input feature $V_{2,3}$ was used to create the breakpoints between the linear models. The distribution of those breakpoints, Fig. 7(a), shows that $V_{2,3} \approx 0.37$ was the most popular breakpoint. That point corresponds to the end of purely linear ageing, approximately halfway through cell life—over 50% of all data points in the full dataset were above that value.

Median predictive error performance was consistent across a large range of testing conditions, Fig. 5. Successful prognosis, especially for cells further from the mean behaviour, requires more input data, as demonstrated by the improving 95th percentiles in error when more cells, inputs, models and input distinctiveness were introduced. There was a notable improvement when the maximum allowed correlation between inputs was reduced below 0.90, thereby giving more generality to the model. That improvement suggests that the maximum shared correlation constraint in the feature selection process is an important factor in optimising performance.

According to Fig. 5, 3 input features and 2 linear sub-models were required for successful health modelling of the majority of the cells in the dataset used here. All three piecewise sub-model splitting techniques produced good performance from as few as 20 training cells' data, although those training cells need to represent a reasonable range of lifetimes.

There are limitations to using a piecewise-linear model. If the relationship between the feature and the capacity fade is highly nonlinear, this approach will fail to capture it accurately. Second, the proposed model may perform poorly if the cells under consideration have undergone a wide variety of very different degradation trajectories. The input feature generation and subsequent regression methods both assume that there are significant similarities between the cells being trained and those being tested. A wider set of conditions might challenge that assumption.

## 6. Conclusions

In this work, a combined feature selection and piecewise-linear modelling approach to predict battery capacity fade was detailed and tested. Under the testing conditions used here, the combined approach produced median RMSE capacity of 1.1% and median lifetime error of 1.6%. On the data considered here, the piecewise-linear model performed comparably to a Gaussian process regression model when given the same input features, while outperforming the machine learning method at the 95th percentile. The approach was robust to reduced number of input features and reduced training set size, and to limitations being imposed on the piecewise-linear model construction. The feature selection step was shown to give accurate performance by avoiding input features that correlated too well with each other.

The ability of data-driven piecewise-linear models of battery capacity fade to adapt to wider ranges of use than available in the datasets considered here remains unclear. Users of these approaches must always be careful to have reliable and appropriate training data for their application. The work here combines easily understood inputs with simple modelling to construct a flexible and fast battery degradation model with minimal memory requirements for predictions. The model produces accurate capacity forecasts that hold up to and after the acceleration of battery degradation in later life.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: David Howey and Samuel Greenbank report financial support was provided by Engineering and Physical Sciences Research Council and by Siemens AG. David Howey reports a relationship with Habitat Energy that includes: consulting or advisory. David Howey reports a relationship with Brill Power that includes: equity or stocks.

## Acknowledgements

This work was funded by EPSRC, UK and Siemens Ltd., through an industrial CASE award. For the purpose of Open Access, the authors apply a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

## Appendix A. Alternative splitting approaches

The two alternative approaches to finding the breakpoints in the linear piece-wise model were first using the *fminsearch* Matlab function and second using K-means clustering. The method using the *fminsearch* Matlab function represented completely free selection of the breakpoint positions. The objective function to minimise was error metric RMSE $\Delta Q$ over the entire training set, and a limit was included such that breakpoints must be fit in size order, to avoid overlapping. K-means clustering was performed using the Matlab function *kmeans*. A small trial found that performance was best when using K-means with the first two selected input features, instead of the full training dataset.

## Appendix B. Gaussian process regression

Gaussian process regression (GPR) is a non-parametric probabilistic approach to regression. It has been used in battery health and lifetime prediction previously [1,8,13–15]. GPR was used here as a comparison to the piecewise-linear model, performing the same mapping between the five automatically selected inputs and the output—changes in capacity, $\Delta Q$. The rest of the approach was identical to the piecewise-linear approach. The choice of kernel function for GPR was a radial basis function (a.k.a. squared exponential), a commonly used stationary covariance function (Eq. (7)) [45]. Automatic relevance determination allowed for a different length-scale hyperparameter, $\sigma_l$, for each input (Eq. (6)).

$$r(x_i, x_j) = \sqrt{\sum_k \frac{\left(x_{i,k} - x_{j,k}\right)}{\sigma_{l,k}^2}} \tag{6}$$

$$\kappa_{rbf}(r) = \sigma_f^2 \exp\left(-r^2\right) \tag{7}$$

The GPR model was the same as the one used in Ref. [15], but with 50 training cells and the radial basis function kernel.

## References


[1] Y. Li, K. Liu, M. Foley, A. Zulke, M. Berecibar, E. Nanini-Maury, J. Van Mierlo, H.E. Hoster, Data-driven health estimation and lifetime prediction of lithium-ion batteries: a review, Renew. Sustain. Energy Rev. 113 (2019).

[2] B. Xu, J. Zhao, T. Zheng, E. Litvinov, D.S. Kirschen, Factoring the cycle aging cost of batteries participating in electricity markets, IEEE Trans. Power Syst. 33 (2018) 2248–2259.

[3] V. Sulzer, P. Mohtat, A. Aitio, S. Lee, Y.T. yeh, F. Steinbacher, M.U. Khan, J.W. Lee, J.B. Siegel, A.G. Stefanopoulou, D.A. Howey, The challenge and opportunity of battery lifetime prediction from field data, Joule 5 (2021) 1934–1955.

[4] C.R. Birkl, M.R. Roberts, E. McTurk, P.G. Bruce, D.A. Howey, Degradation diagnostics for lithium-ion cells, J. Power Sources 341 (2017) 373–386.

[5] J. Reniers, G. Mulder, D. Howey, Review and performance comparison of mechanical-chemical degradation models for lithium-ion batteries, J. Electrochem. Soc. 166 (2019) A3189–A3200.

[6] C. Fleischer, W. Waag, H. Heyn, D.U. Sauer, On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: part 1. Requirements, critical review of methods and modeling, J. Power Sources 260 (2014) 276–291.

[7] P. Dechent, S. Greenbank, F. Hildenbrand, S. Jbabdi, D.U. Sauer, D.A. Howey, Estimation of Li-ion degradation test sample sizes required to understand cell-to-cell variability, Batter. Supercaps 4 (2021) 1821–1829.

[8] M. Lucu, E. Martinez-Laserna, I. Gandiaga, K. Liu, H. Camblong, W. Widanage, J. Marco, Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data - part B: cycling operation, J. Energy Storage 30 (2020).

[9] P. Fermín-Cueto, E. McTurk, M. Allerhand, E. Medina-Lopez, M.F. Anjos, J. Sylvester, G. dos Reis, Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells, Energy AI 1 (2020).

[10] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R.D. Braatz, Data-driven prediction of battery cycle life before capacity degradation, Nat. Energy 4 (2019) 383–391.

[11] K. Goebel, B. Saha, A. Saxena, J. Celaya, J. Christophersen, Prognostics in battery health management, IEEE Instrum. Meas. Mag. 11 (2008) 33–40.

[12] J. Liu, A. Saxena, K. Goebel, B. Saha, 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.

[13] R. Richardson, M. Osborne, D. Howey, Battery health prediction under generalized conditions using a Gaussian process transition model, J. Energy Storage 23 (2019) 320–328.

[14] X. Hu, Y. Che, X. Lin, S. Onori, Battery health prediction using fusion-based feature selection and machine learning, IEEE Trans. Transp. Electr. (2020).

[15] S. Greenbank, D. Howey, Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life, IEEE Trans. Ind. Inf. (2021) early access.

[16] Y. Li, D. Stroe, Y. Cheng, H. Sheng, X. Sui, R. Teodorescu, On the feature selection for battery state of health estimation based on charging–discharging profiles, J. Energy Storage 33 (2021).

[17] Y. Li, M. Abdel-Monem, R. Gopalakrishnan, M. Berecibar, E. Maury-Nanini, N. Omar, P. van den Bossche, J. Van Mierlo, A quick on-line state of health estimation method for li-ion battery with incremental capacity curves processed by Gaussian filter, J. Power Sources 373 (2018) 40–53.

[18] N. Samad, Y. Kim, J. Siegel, A. Stefanopoulou, Battery capacity fading estimation using a force-based incremental capacity analysis, J. Electrochem. Soc. 163 (2016) A1584–A1594.

[19] A. Bartlett, J. Marcicki, K. Rhodes, G. Rizzoni, State of health estimation in composite electrode lithium-ion cells, J. Power Sources 284 (2015) 642–649.

[20] C. Weng, Y. Cui, J. Sun, H. Peng, On-borad state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression, J. Power Sources 235 (2013) 36–44.

[21] M. Berecibar, F. Devriendt, M. Dubarry, I. Villarreal, N. Omar, W. Verbeke, J. Van Mierlo, Online state of health estimation on NMC cells based on predictive analytics, J. Power Sources 320 (2016) 239–250.

[22] X. Li, J. Wang, L. Wang, D. Chen, Y. Zhang, C. Zhang, A capacity model based on charging process for state of health estimation of lithium ion batteries, Appl. Energy 177 (2016) 537–543.

[23] Y. Wu, A. Jossen, Entropy-induced temperature variation as a new indicator for state of health estimation of lithium-ion cells, Electrochem. Acta 276 (2018) 370–376.

[24] J. Belt, V. Utgikar, I. Bloom, Calendar and PHEV cycle life aging of high-energy, lithium-ion cells containing blended spinel and layered oxide cathodes, J. Power Sources 196 (2011) 10213–10221.

[25] L. Willenberg, P. Dechent, G. Fuchs, M. Teuber, M. Eckert, M. Graff, N. Kürten, D. Uwe Sauer, E. Figgemeier, The development of jelly roll deformation in 18650 lithium-ion batteries at low state of charge, J. Electrochem. Soc. 167 (2020).

[26] B. Gou, Y. Xu, X. Feng, An ensemble learning-based data-driven method for online state-of-health estimation of lithium-ion batteries, IEEE Trans. Transp. Electr. (2020).

[27] P. Mohtat, S. Lee, J. Siegel, A. Stefanopoulou, Towards better estimability of electrode specific state of health: decoding the cell expansion, J. Power Sources 427 (2019) 101–111.

[28] J. Cannarella, C. Arnold, State of health and charge measurements in lithium-ion batteries using mechanical stress, J. Power Sources 269 (2014) 7–14.

[29] B. Saha, S. Poll, K. Goebel, J. Christophersen, An integrated approach to battery health monitoring using bayesian regression and state estimation, IEEE Autotestcon (2007) 646–653.

[30] X.-G. Yang, Y. Leng, G. Zhang, S. Ge, C.-Y. Wang, Modeling of lithium plating induced aging of lithium-ion batteries: Transition from linear to nonlinear aging, J. Power Sources 360 (2017) 28–40.
[31] S. Atalay, M. Sheikh, A. Mariani, Y. Merla, E. Bower, W. Widanage, Theory of battery ageing in a lithium-ion battery: Capacity fade, nonlinear ageing and lifetime prediction, J. Power Sources 478 (2020).
[32] K. Pugalenthi, H. Park, N. Raghavan, Piecewise model-based online prognosis of lithium-ion batteries using particle filters, IEEE Access 8 (2020) 153508–153516.
[33] A. Mandli, A. Kaushik, R. Patil, A. Naha, K. Hariharan, A. Kolake, S. Han, W. Choi, Analysis of the effect of resistance increase on the capacity fade of lithium ion batteries, Int. J. Energy Res. 43 (2019) 2044–2056.
[34] A. Alsharif, M. Das, A piecewise linear time-varying model for modeling the discharge process of a lithium-ion battery, in: IEEE International Conference on Electro Information Technology, 2014, pp. 587–592.
[35] Y. Li, R. Anderson, J. Song, A. Phillips, X. Wang, A nonlinear adaptive observer approach for state of charge estimation of lithium-ion batteries, in: Proceedings of the American Control Conference, 2011, pp. 370–375.
[36] J. Meng, D. Stroe, M. Ricco, G. Luo, R. Teodorescu, A simplified model-based state-of-charge estimation approach for lithium-ion battery with dynamic linear model, IEEE Trans. Ind. Electron. 66 (2019) 7717–7727.
[37] Z.-X. Zhang, X.-S. Si, C.-H. Hu, M.G. Pecht, A prognostic model for stochastic degrading systems with state recovery: application to Li-ion batteries, IEEE Trans. Reliab. 66 (2017) 1293–1308.
[38] R. Xiong, Y. Zhang, J. Wang, H. He, S. Peng, M. Pecht, Lithium-ion battery health prognosis based on a real battery management system used in electric vehicles, IEEE Trans. Veh. Technol. 68 (2019) 4110–4121.
[39] T. Baumhöfer, M. Brühl, S. Rothgang, D.U. Sauer, Production caused variation in capacity aging trend and correlation to initial cell performance, J. Power Sources 247 (2014) 332–338.
[40] M. Farag, M. Fleckenstein, S. Habibi, Continuous piecewise-linear reduced-order electrochemical model for lithium-ion batteries in real-time applications, J. Power Sources 342 (2017) 351–362.
[41] Y. Wen, R. Wolski, C. Krintz, Online prediction of battery lifetime for embedded and mobile devices, in: Third Power-Aware Computer Systems, 2004, pp. 57–72.
[42] A. Chu, A. Allam, A. Cordoba Arenas, G. Rizzoni, S. Onori, Stochastic capacity loss and remaining useful life models for lithium-ion batteries in plug-in hybrid electric vehicles, J. Power Sources 478 (2020).
[43] J. Yu, J. Yang, Y. Wu, D. Tang, J. Dai, Online state-of-health prediction of lithium-ion batteries with limited labeled data, Int. J. Energy Res. 44 (2020) 11345–11360.
[44] 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, W.C. Chueh, Closed-loop optimization of fast-charging protocols for batteries with machine learning, Nature 578 (2020) 397–402.
[45] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.
[46] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.