Data-driven battery state of health diagnostics and prognostics

Samuel Greenbank

Department of Engineering
University of Oxford

This dissertation is submitted for the degree of
Doctor of Philosophy

St. Cross College

March 2022

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and the acknowledgements and publications pages. Parts of the work have been published in journal papers and presented at conferences and seminars. These are specified in the text and referenced as appropriate.

Samuel Greenbank
March 2022

Acknowledgements

It would be beyond remiss of me not to thank David Howey for his help, guidance and efforts in his role as supervisor. This thesis would be of less value than rice paper in a hurricane without him and his oversight. Perhaps thank you is the wrong gesture. I'll say sorry to him here too, just to cover my bases.

Siemens Ltd. provided funding, advice and grounding at various stages. Many of my approaches and ideas were founded on interactions with the various employees of Siemens, and I thank them for their time and patience. EPSRC provided the rest of the funding for which I am extremely grateful.

Other official mentions go to R. Richardson for taking the initial steps with GPs, the groups behind the Severson-2019 and Attia-2020 datasets, P. Dechent for his contributions to the cell variability work, S. Jbabdi for his admirable patience and wisdom, and finally to anyone unfortunate enough to follow a presentation of mine.

As for the others who got me to this point? A. Aitio deserves special mention for his role on the ML desk. V. Grant, R. Lane, A. Lewis-Douglas, T. Raj, J. Reniers, M. Wojtala and others formed an adequate group of colleagues and I wish them a varying amount of luck in the future. There are a whole sequence of friends, teammates and other beings from Oxford University and Oxford Hawks hockey clubs that deserve mentions. As do the various figures within the Vincent's Club, MNTLBC (beer?), 41RR, Marston Kitchen, Lincoln College (2013), Three Spires, and my flat. Sadly, I am limited by space, you'll just have to take it for granted that you would have been listed.

To the pubs, paths, fields, spires, bikes, rivers, and stone walls of Oxford, I want to say thank you for the last 8 years. It's been incredible.

This is the traditional point to thank your parents for not voicing their concerns about not getting a proper job. I would like to add that they always answered the phone and only changed the locks once during the four years.

Abstract

Lithium-ion batteries are increasingly ubiquitous in modern society but the degradation of lithium-ion cells is complex and challenging to predict. Data-driven approaches to estimating and forecasting the state of health of lithium-ion batteries have become increasingly popular in literature due to the growing availability of battery data and the improved flexibility of data-driven approaches relative to physics-based modelling. This thesis begins with a review of health diagnosis and prognosis approaches in battery literature, including a brief introduction to the principles and challenges of using data-driven approaches. The subsequent chapters investigate several of the remaining open questions.

An automated methodology for input feature generation and selection is proposed and thoroughly tested. Gaussian process regression, a well-known non-parametric form of supervised learning, is used to map from the input features to changes in capacity. The inputs are found to be good predictors of degradation and also robust to significantly increased noise and reduced sampling frequencies for the raw cycle data.

Gaussian process regression produces accurate predictions of capacity here. Piecewise, Bayesian linear regression is a faster, more transparent alternative that produces equally accurate end-of-life forecasts. A battery-focussed performance metric is proposed to assess the accuracy of the output probabilistic predictive distributions of both regression models. A proposed adaptation to sparse Gaussian process regression is found to reduce the storage requirements of a Gaussian process battery health model by 98% without significantly impacting predictive performance.

Lithium-ion cells suffer from cell-to-cell variability in ageing rates, thereby posing a challenge to the control of battery packs and experimental design. An experiment finds that around 12 similarly used cells are required to consistently fit a population-level distribution for an empirical model. By contrast, a Gaussian process regression model requires under 5 cells in a training set to consistently fit the hyperparameters, but around 50 cells are required to produce the capacity predictive performance of other models in this thesis.

Publications

Much of the work presented in this thesis has been previously published as joint-author work:

[1] Samuel Greenbank and David Howey, "Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life," *IEEE Transactions on Industrial Informatics*, DOI: 10.1109/TII.2021.3106593, **18**, 2965-2973, 2021.

[2] Samuel Greenbank and David Howey, "Piecewise-linear modelling with feature selection for Li-ion battery end of life prognosis," http://arxiv.org/abs/2104.07576, 2021. [*pre-print, submitted for peer review in January 2022*]

[3] Philipp Dechent, Samuel Greenbank, Felix Hildenbrand, Saad Jbabdi, Dirk Uwe Sauer, and David Howey, "Estimation of Li-ion degradation test sample sizes required to understand cell-to-cell variability," *Batteries & Supercaps*, doi: 10.1002/batt.202100148, 2021.

Other works that were contributed towards, but are not part of this thesis.

[4] Aashutosh Mistry, Ankit Verma, Shashank Sripad, Rebecca E. Ciez, Valentin Sulzer, Ferran Brosa Planella, Robert Timms, Yumin Zhang, Rachel Kurchin, Philipp Dechent, Weihan Li, Samuel Greenbank, Zeeshan Ahmad, Dilip Krishnamurthy, Alexis Fenton, Jr., Kevin Tenny, Prehit Patel, Daniel Juarez Robles, Paul Gasper, Andrew Colclasure, Artem Baskin, Corinne Scown, Venkat Subramanian, Edwin Khoo, Srikanth Allu, David Howey, Steven DeCaluwe, Scott Roberts, Venkatasubramanian Viswanathan, "A Minimal Information Set to Enable Verifiable Theoretical Battery Research," *American Chemical Society Energy Letters*, **6**, 3831-3835, 2021.

[5] Peter M. Attia, Alexander Bills, Ferran Brosa Planella, Philipp Dechent, Goncalo dos Reis, Matthieu Dubarry, Paul Gasper, Richard Gilchrist, Samuel Greenbank, David Howey, Ouyang Liu, Edwin Khoo, Yuliya Preger, Abhishek Soni, Shashank Sripad, Anna G. Stefanopoulou, and Valentin Sulzer, ""Knees" in lithium-ion battery aging trajectories", https://arxiv.org/abs/2201.02891, 2021. [*pre-print, submitted for peer review in January 2022*]

I contributed theory, implementation, analysis and writing for references [1] and [2], all of which was guided, advised, reviewed, and edited by David Howey. Work from references [1] and [2] has also been presented at Advanced Battery Power 2021 (and a poster in 2019), GRC Batteries 2020, NASA Aerospace Battery Workshop 2020, UKES 2019. A summation of much of this thesis was presented as part of the Battery Modelling Webinar Series in September 2021.

Philipp Dechent and I were joint first authors of reference [3] which included data pre-processing, coding, analysis and writing. Felix Hildenbrand contributed software and data side whereas multi-level Bayes was introduced and implemented by Saad Jbabdi. Dirk Uwe Sauer and David Howey were the principle investigators for that study and contributed supervision and review.

I provided a minor contribution to the model checklist in reference [4]. I submitted the piecewise linear model in reference [2] as a model within that piece after an invite to do so from Aashutosh Mistry. I also reviewed and edited Philipp Dechent's contribution of a checklist for the multi-level Bayesian model in reference [3].

Reference [5] is a review of the origins of the knee point and involved significant contributions from many authors. I contributed research, code, and analysis for the knee identification section. Initially submitted for peer-review in January 2022.

Table of Contents

List of Figures

List of Tables

List of Symbols

Acronyms / Abbreviations

ARD automatic relevance determination

cdf cumulative distribution function

EoL end-of-life

IQR interquartile range

MCMC Markov chain Monte Carlo

MLB multi-level Bayes

NLML negative log marginal likelihood

PCA principle component analysis

pdf probability density function

RMSE root mean square error

RPT reference performance test

RPTP repeats per test point (Appendix A only)

RUL remaining useful life

SoH state of health

Datasets

Attia-2020 45 fast-charging 18650 cells

Dechent-2017 21 similarly cycled 18650 cells

Dechent-2020 22 identically cycled, high energy 18650 cells

NASA-2014 28 randomly used 18650 cells

Raj-2020 12 alternately cycled 18650 cells

Sauer-2021 48 identically cycled 18650 cells

Severson-2019 135 fast-charging 18650 cells

Greek Symbols

α size of region in β-score calculation

α_σ fractional tolerance within capturing variability method

β_L fractional lengthscale of the Gaussian moving average

ΔQ changes in capacity

ϵ observed noise

μ_k parameter mean estimate for cell k

μ_p Bayesian population mean estimate

ρ_P Pearson correlation coefficient

ρ_s data density function

$\rho_{P,\max}$ maximum allowed correlation coefficient between input features

$\sigma(x)$ standard deviation of variable x

σ_k^2 parameter variance estimate for cell k

$\sigma_l(\mathbf{x_s})$ lengthscale of the Gaussian moving average

σ_n standard deviation of observed noise

σ_p Bayesian population standard deviation estimate

σ_R standard deviation of noise added to raw data

Σ_w prior variance of parameters $\mathbf{w}$

σ_w prior variance of individual parameter w_k

θ_k parameter set for cell k

θ_p population-level model parameters

Other Symbols

$\text{cov}(x_i, x_j)$ covariance between variables x_i and x_j

Performance Metrics

β-score cumulative probability of a prediction within a specified region

EoL Error percentage error in lifetime

Knee Error percentage error in knee prediction

RMSE ΔQ root mean square error for the changes in capacity

RMSE Capacity root mean square error of capacity

RMSE-Freq root mean square error of frequency

Roman Symbols

$\mathcal{D}$ training set

$\hat{\mathbf{w}}$ posterior mean estimate of parameters for Bayesian linear regression

$\mathbf{x_s}$ vector of observations among the splitting variable

$\mathbf{x}$ model input

D number of input features, length($\mathbf{x}$)= D

$f(\mathbf{x})$ real function

F_s function used for finding break points

f_z pseudo-targets in a sparse Gaussian process model

$I_{n,m}$ feature value for current thresholds n and m

K sub-sample size

$k(\mathbf{x}, \mathbf{x}')$ covariance function of a Gaussian process

k_{RBF} covariance function calculated with an radial basis function kernel

K_{XX} covariance function of training set

M number of data points in the pseudo-inputs.

$m(x)$ mean function of a Gaussian process

N number of training data points

n_c required number of cells

n_m number of sub-models in a piecewise linear regression model

$P_{n,m}$ feature value for power thresholds n and m

Q capacity

R logic function used in feature generation

t time

t_{EoL} time of end-of-life

t_{knee} time of the knee point

t_i time interval i

$T_{n,m}$ feature value for temperature thresholds n and m

$V_{n,m}$ feature value for voltage thresholds n and m.

w parameters of a linear model

x_s splitting variable

x_s^* test points of the splitting variable

y output target value

Z pseudo-inputs in a sparse Gaussian process model

i input feature index

k data point index

Regression Models and Components

ARIMA autogressive integrated moving average

BLR Bayesian linear regression

EXP exponential kernel

GPR Gaussian process regression

GPR-EXP Gaussian process regression with an exponential kernel

GPR-RBF Gaussian process regression with a radial basis function kernel

LinExp linear capacity model followed by exponential decay

LinOne linear capacity model from 100% SoH

LinTwo linear capacity model with variable $Q(t = 0)$

M32 Matérn-3/2 kernel

M52 Matérn-5/2 kernel

NN neural networks

PF particle filter

PLR piecewise linear regression

RBF radial basis function kernel

RF random forest regression

SparseGPR sparse Gaussian process regression

SVM support vector machines

Chapter 1

Introduction

1.1 Lithium-ion battery degradation

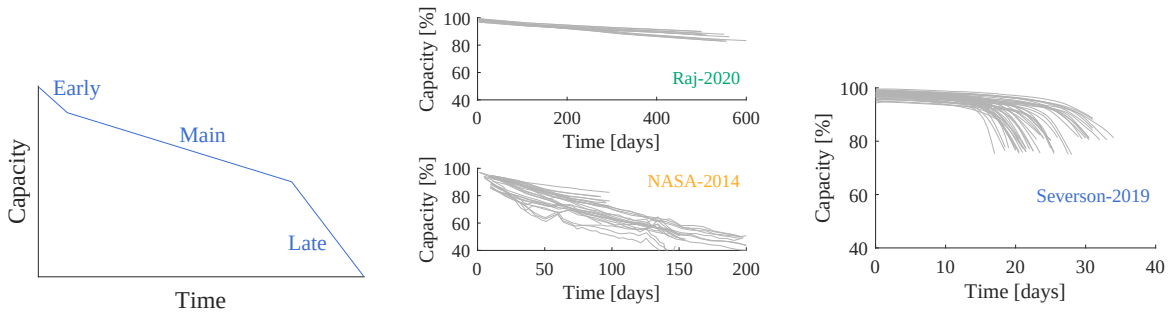
Interest in both electric vehicles and reliable grid applications is increasing the relevance of lithium-ion batteries in the modern world [6]. Lithium-ion batteries suffer from degradation in health through both use, known as cyclic ageing, and rest or storage, known as calendar ageing [7–9]. Safe and confident use of cells is reliant on accurate knowledge of state of health (SoH) and hence degradation [10, 11]. For example, one source of overheating is overcharging, which creates an excess of heat that can eventually lead to thermal runaway [12].

Capacity, alongside internal resistance, is one of the most common measures of the health of a cell [13, 14]. Capacity fade and internal resistance increase are established metrics of degradation, and are each linked with an array of possible degradation mechanisms. The variety of possible degradation mechanisms are linked to a further variety of possible internal and external causes [7, 15]. Manufacturing variability (sometimes referred to as intrinsic variability, i.e. production caused variability or

manufacturing tolerance) further confuses the picture because cells of identical chemistry, form, model and even batch are often found to have differing initial capacities and then will subsequently age at uncorrelated rates [16–18]. In summary, battery degradation is a complex phenomenon and as a result is challenging to predict.

Despite the complexity of the underlying degradation, most capacity fade profiles can be broken down into two or three stages of ageing [19–21], as shown in Fig. 1.1a. There can be an initial, fast change in early capacity. For instance, capacity can decrease rapidly due to solid electrolyte interphase growth [19, 21], but short-term increases have been seen in some laboratory cells [22]. The initial phase is short, and is followed by an extensive phase of slow ageing [19, 20]. The development of a solid electrolyte interphase is a commonly referenced degradation mechanism which usually creates a square root dependence between time and capacity loss [7, 15, 23, 24]. Particle cracking, lithium plating and other mechanisms can also have a slowly increasing impact on SoH [7, 15, 24]. Many cells remain in the slow ageing stage for the remainder of their useful life, but sometimes a final, accelerated ageing phase has been simulated [20] and experimentally demonstrated [16, 19, 22, 23, 25].

In many cases, only one or two stages are exhibited. Calenderically aged cells experience steady, shallow ageing for long periods, with the degradation rate determined by temperature and state of charge [20, 26–29], and their capacity has been modelled with simple explicit functions [28, 30–32]. Repeated cycling can also lead to approximately constant degradation rates, as demonstrated by both datasets in Fig. 1.1b [33, 34].



(a) Regions of degradation. (b) Linearly aged cells. (c) Cells showing knee points.

Fig. 1.1 Example lithium-ion battery health profiles. Most profiles can be broken into a small number of stages (a) and many exhibit only linear ageing (b). Arduous use protocols can lead to a collapse in capacity, known as a knee point (c).

The cells shown may have gone on to experience accelerated ageing later on, but that is beyond the scope of the available data.

Arduous use protocols can result in the catastrophic loss of capacity seen in Fig. 1.1c, commonly known as a "knee point". This accelerated ageing has been regularly linked to lithium plating [23, 25, 35, 36] and occasionally to a number of other degradation mechanisms such as pore clogging and uneven compression [15, 37, 38]. Confident prediction of the onset of the knee point has proven difficult; nonetheless, it remains a critical consideration for the safety of any battery system [16, 22, 39].

The uncomplicated profiles provide hope that modelling lithium-ion battery degradation should be achievable, despite the inherent complexity of the underlying processes.

1.2 Modelling degradation

There are a number of techniques available for modelling lithium-ion battery degradation. These are loosely split into *physics-based* techniques (Section 1.2.1) and

data-driven techniques (Section 1.2.2) based on a subjective judgement, but there is overlap in many cases. Degradation modelling typically seeks to address three questions: (1) What is the health now? (2) What will the health be in the future? (3) For how much longer can this cell be used by a customer?

What is the health now?

Finding the current SoH of a cell is known as *diagnosis* or *SoH estimation* in literature. This thesis uses capacity at rate, meaning the capacity when measured at the cycling current, as the health metric, but there are a number of possible metrics that one might wish to know. For example, the open-circuit-voltage capacity is a measure of all the available charge that can be stored in a cell [34] whereas the internal resistance is directly related to the deliverable power from a cell, which can also be estimated directly [40]. One can either estimate the metrics from the cycle data [40, 41] or perform a reference performance test (RPT) and estimate the health from that [42–45]. Note, the references given here are not exhaustive and fuller descriptions can be found in a number of reviews [13, 24, 46–49].

Lithium-ion battery diagnosis has received more focus than SoH prognosis in literature but is not the main focus of this thesis. The work here focusses on the next question, but the techniques presented here are easily adapted for SoH estimation. The techniques in Chapter 2 were adapted to perform SoH estimation, the results of which are found in Appendix B.2.

What will the health be in the future?

Health prognosis or forecasting aims to predict the trajectory of a given health metric through cell life, and can form an essential part of health management [50]. It relies on knowledge of future or intended use. An explicit example is the assumption of continued similar use in *k-step-ahead* approaches [51–54]. As a general rule, approaches aiming to predict the future health trajectory are less certain as the extrapolation length increases, regardless of whether an empirical or data-driven model is used [55, 56].

The models in all subsequent chapters assume knowledge of future use of all test cells. Use prediction is beyond the scope of this thesis but is a significant challenge.

For how much longer can this cell be used safely?

The most common approach to answering this question is remaining useful life (RUL) prediction, i.e. estimating the amount of time or cycles before an end-of-life (EoL) condition is reached [13, 56, 57]. Predicting RUL can be performed either during cell life [58–61] or from a specified early cycle [22, 39, 62–64]. End-of-life conditions vary but are typically taken as a capacity or resistance limit [61].

Beyond the knee point, a cell can be considered to have lost its value for an application and hence can be considered as analagous to EoL but with a flexible location in time-capacity space [65]. There have been attempts to explicitly predict the knee point from early cycles [38, 39, 64], or the equivalent procedure looking at *elbows* in internal resistance rises [66, 67].

1.2.1 Physics-based approaches

Broadly, there are three categories of physics-based approach: electrochemical models, equivalent circuit models and semi-empirical models.

Electrochemical models aim to describe ageing by modelling specific molecular-scale degradation mechanisms. These models, sometimes referred to as physical or mechanistic models, assume degradation occurs through one or more of solid electrolyte interphase growth [68–71], particle cracking [69, 71, 72], lithium plating [73, 74] or loss of active material by other means [69, 70]. Electrochemical models can be extremely precise but rely on detailed information about the cell components and their initial conditions, and require solving partial differential equations, rendering them challenging to apply in the real world [24, 75].

Equivalent circuit modelling is a computationally cheaper alternative with fewer parameters [76]. Equivalent-circuit methods aim to approximate the electrical behaviour of a cell with a relatively small circuit comprised of some form of voltage source plus a number of resistors, capacitors, diodes or transformers [77–81]. Equivalent circuits can be used to estimate states and parameters, such as state of charge and SoH, based on the usage data, using an appropriate parameter estimation algorithm. A popular tool is the Kalman filter, a Bayesian filtering mechanism [77, 82].

Both equivalent-circuit and electrochemical models can be useful tools, but a trained model is usually too specific to be applied to other uses or chemistries or even other nominally identical cells [24].

Finally, semi-empirical methods are a simpler and hence more adaptable alternative [31, 83, 84]. In this case, a model is constructed by specifying the mechanisms and relationships by which cells will degrade before fitting the parameters of that model. One example model used four distinct causes of degradation, such as high temperature, low temperature, and high state of charge, each with a separate physics-based stress factor [31]. Performance of semi-empirical models is a function of the appropriateness of the chosen models [31, 84].

Physics-based approaches provide insight and are interpretable but suffer because they must assume a fixed structure to the degradation model. Battery degradation is complex and there is significant variability between even supposedly identical cases [7, 16]. More flexibility is perhaps required for a successful health model that can be applicable in non-laboratory settings.

1.2.2 Data-driven techniques

Data-driven modelling techniques allow the data to shape the form of the degradation trajectory. These can be loosely divided into three categories: fixed empirical models with adaptable parameters, machine learning models, and deep machine learning models.

Fixed empirical models with adaptable parameters

The first of the data-driven approaches involves using a simpler empirical model than those in the semi-empirical approaches but then adapting the parameters of that model through cell life.

The application of Kalman and particle filtering (PF) techniques to model cell ageing is a good example. The ageing curve can follow a capacity, Q, function as simple as $Q(t) = a + bt^c$ [85–87]. The values of a, b and c, or parameters within equivalent functions, will vary over time. The functionality will approximate the degradation trajectories, but the filtering method aims to improve predictive performance by adapting to the ageing of a specific cell over time [40, 61, 77, 88–94].

These methods rely on a prior assumption of the shape of the capacity profile. For example, it is unlikely that the same capacity function should be used for the cells in Fig. 1.1b as in Fig. 1.1c. This thesis aims instead to present a model that can be applied in a general case.

Adaptive structure models

All models mentioned previously have fixed structures, as do all models mentioned below this section, which adapt parameters to fit the model. Adaptive structure models are more flexible and have been applied for fault detection, but are rare in literature focussed on battery health [95–97]. Fuzzy systems are a good example of a model with an adaptive structure [98], which have been applied to k-step-ahead SoH forecasting [95]. Fuzzy systems, by using fuzzy logic, is a rule-based regression tool that has also

been proposed as an appropriate SoH estimation method that can handle vague and imprecise measurement battery data [99].

Fixed structure models were preferred in this thesis because they are more established in literature. Focus was instead applied to establishing reliable inputs for the fixed structure models and reducing the cost of storing and applying them.

Statistical time-series methods

Here, statistical time-series methods are those that specifically target relationships through time, such as lithium-ion battery degradation, using statistical methods.

One example is the autoregressive integrated moving average, commonly abbreviated to ARIMA [100]. Previous, i.e. known, targets and forecast errors have assigned weights that describe their relevance to the forecast of the next time step. ARIMA models have been used for SoH forecasting [100, 101], but typically need to be combined with other techniques to map battery degradation [13, 101, 102].

Wiener processes and Gamma processes are other examples of time-series modelling that have been applied to SoH forecasting [103–106]. Both Wiener and Gamma processes assume independent incremental changes of some target variable through time, but differ in the assumed shape of the distribution governing those incremental changes. Gamma processes assume a monotonic relationship [105], which is not a strong assumption for battery degradation [22, 33, 107, 108]. In literature, Wiener processes assume a fixed shape of the underlying degradation trend [103, 104], so were deemed insufficiently flexible for use here.

Machine learning approaches

Modelling battery degradation by using machine learning techniques has become increasingly common in literature [13, 55]. The models in this section are classed as *supervised learning* models, which means they are trained based on known input values and output targets. They are also all examples of *regression* models, meaning that the outputs are continuous variables [109–111]. The models referenced here, in general, do not have predefined mean functions although they may have fixed underlying basis functions. Some supervised learning approaches include parameters, i.e. they are formed with fixed basis functions, or they are statistical models, sometimes referred to as non-parametric. All statistical models used here were Bayesian and there is a fuller description of Bayesian approaches in Section 1.5. Alternatively, references [109], [110] & [111] all provide thorough explanations.

Supervised learning techniques take pairs of inputs and outputs to form a training set. The inputs in this work are column vectors $\mathbf{x}$ of length D, with each item representing an individual input feature x_i. The outputs, also known as targets, are scalars, y. The training data, $\mathcal{D}$, is formed of N input-output pairs: $\mathcal{D} = \{\mathbf{x_k}, y_k\}_{k=1}^{N}$. The training data D is used to fit the parameters of the chosen supervised learning model.

The simplest data-driven models are linear models of the form $y = \sum_{i=1}^{D} w_i x_i$ with parameter vector $\mathbf{w}$. Linear models are similar to some empirical models but are classed as supervised learning because the parameters $\mathbf{w}$ are exclusively fit according to how well the model performs on the training data targets. Linear models are fast to

train, easy to store and easily understood. As a result they have been used as part of SoH estimation [112–117], health prognosis [118] and RUL and knee point predictions [22, 39, 113, 119].

The support vector machine (SVM) is a kernel-based method which resembles a linear model in its construction. Instead of using direct inputs, the output is a weighted sum of kernel values [111, 120]. SVM is a *sparse* method because only points outside of a stated tolerance have non-zero weights. A smaller value for that tolerance leads to more non-zero weights and reduced sparsity [111]. Relevance vector machines share a functional form with support vector machines but use Bayesian inference to produce a probabilistic output [121, 122]. Much like linear models, SVMs and RVMs have been used for SoH estimation [112, 123–126], SoH forecasting [127–129] and RUL prediction [127, 130–133].

Random forest (RF) regression has been used in battery literature less than other machine learning techniques. The approach by producing a large number of models through bootstrapping inputs and taking an average of the outputs [134]. Random forest regression reduces the risk of overfitting to a training dataset, a known concern with decision trees, by training many decision trees on randomly chosen subsets of input variables and randomly chosen subsets of data points [111]. The requirement to train multiple decisions trees makes RF models challenging to interpret [111]. They have mostly been used for SoH estimation [134–138].

Gaussian process regression (GPR) is another non-parametric supervised learning regression tool which can be equivalent to a relevant vector machine regression under

specific conditions [139, 140]. GPR has been shown to be a powerful tool where there is good data availability, but, like other statistical approaches, will suffer if forced to extrapolate significantly [55, 59]. Extrapolation can be strengthened with use of an explicit mean function [51, 141]. Again, GPR has been used for SoH estimation [45, 112, 142–147], SoH forecasting [26, 53, 148–153] and RUL prediction [154–157].

After derivations and toy examples later in this chapter, forms of GPR are used extensively in Chapters 2 and 4. Once trained, the hyperparameters offer insight, and the overlying structure of GPR is a simple mapping between inputs and outputs. Nonetheless, GPR relies on good data and so Chapter 2 proposes a method to ensure the availability of good training and test data. Further, Chapter 4 demonstrates an approach to reduce the storage requirements of GPR without significantly impacting predictive performance.

Deep **modelling, no parameters**

One crucial weakness of GPR and RVMs is how they scale with increased input size. Computational complexity rises with the number of training data points cubed, $\mathcal{O}(N^3)$ [110, 158]. Neural networks (NNs) are a far more scalable solution, typically $\mathcal{O}(N)$, and several forms of neural network have been used in literature to model battery health in a similar fashion to the models mentioned above [159–164].

However neural networks can act as 'deep' learning methods, usually meaning that there are multiple hidden layers in the neural network. This requires a very large

amount of data so these methods aim to map lithium-ion battery health based on the cycle data [41–43, 64, 130, 165–170].

Neural network approaches, especially for deep learning, are arguably *black-box* models, i.e. it is extremely difficult to understand the mechanics between inputs and outputs [147, 170]. For this reason, Gaussian process regression and Bayesian linear regression were used in this work because the (hyper)parameters offer a more insightful alternative.

1.3 Input features

Data-driven modelling in the form of supervised regression maps inputs to outputs based on known input-output pairs. In most cases the inputs are features of the raw data, rather than the raw data itself. This section explores the input features used in battery literature and demonstrates the potential impacts of their selection.

1.3.1 Input features in literature

Input features form the entries of column vector $\mathbf{x}$; various sets of features have been used in literature. The deep learning methods use raw cycle data, i.e. the current, voltage and temperature data, directly [41, 42] but summary features are required for a regression tool. Raw data is usually recorded at much higher frequency than any battery health measurements so the data must be summarised such that the number of input data points equals the number of targets.

Time is the most intuitive input feature and has been used in several successful battery health models [26, 129, 144, 149, 155]. Cycle count is a similarly useful input and is effectively equivalent to time in the vast majority of laboratory datasets. For example, the main dataset used in this thesis has a Pearson's rank coefficient of over 0.99 between time and cycle count [22, 107]. Nonetheless, cycle count is a common input too [59, 128, 171]. Charge throughput is a related variable and has been used in regression health models [123, 149, 150, 152]. Time and charge throughput were found to be the best predictors of the largely linear ageing in the NASA-2014 dataset (see Section 1.7.2 for details of NASA-2014) despite the random and varied use protocols [152].

Temperature is a known factor in lithium-ion degradation and it is commonly used as an input feature [13, 28, 31]. Input features based on temperature generally take the form of either explicitly using the cell temperature [26, 45, 123, 135, 136, 148, 149, 152] or measure the temperature response to certain charge/discharge conditions [45, 114, 131, 136, 146]. However the realities of lithium-ion cell testing mean that ambient temperature is controlled and consistent throughout testing [22, 34, 107] plus there can be significant issues with measuring temperature accurately [172].

Voltage-based features are among the most common inputs for battery health models, especially for SoH estimation. Constant current (dis-)charging profiles are routinely exploited, as are constant voltage phases if present in the data. Voltage-based variables include voltage difference in a given time interval [45, 125, 146, 154], time difference for a given change in voltage [45, 125, 138, 143, 146, 154, 164, 173], slope of

the voltage curve under constant current [45, 138, 143], and time taken to traverse a specific current interval during constant voltage operation [89, 125, 143, 146, 154, 174]. A further voltage-based input feature is exposure to certain state-of-charge ranges, which can be observed using either voltage, state-of-charge or depth-of-discharge measurements [26, 123, 148, 149].

Current and power are the principal controls for any cell but features of them have only been used in a relatively small number of SoH models in literature [136, 147, 149, 152]. The work in this thesis includes features of current and power as inputs because knowledge of at least one of current or power will be application driven. Current has appeared in SoH estimation models indirectly. These models use the capacity from a subset of a charging cycle as an input to predict the full capacity [113, 116, 133, 137, 146, 148, 151].

Differential voltage analysis and incremental capacity analysis are techniques that use the derivatives of the relationship between voltage and charge throughput during constant (dis-)charging to understand state-of-health. Peak position, peak height, and integrals across specific ranges have all been considered as inputs for degradation modelling [22, 39, 45, 62, 145, 164, 175, 176]. Derivative-based inputs can provide detailed insight but require a high quality of data [7, 13]. Complex properties such as *energy-of-signal* or *sample entropy* have also been used [45, 124, 132, 138, 177] but were similarly omitted here for their demands on the data quality.

1.3.2 Impact of input choices

The data-driven approach presented in Chapter 2 aims to produce input features that were functions of use and adaptable to different use cases. For example, an electric vehicle might experience long periods without (dis-)charging whereas a grid storage system could be in use continuously. The input features are intended to be robust to reduced data quality but need to remain sufficiently detailed to capture key degradation drivers.

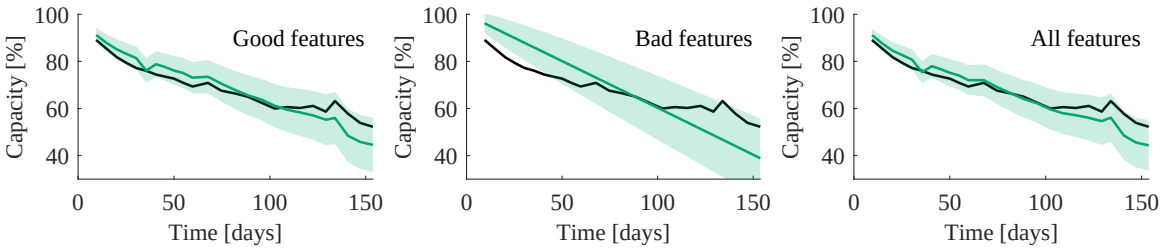


Fig. 1.2 Comparison of predictive performance between a case with good input features, a case with bad input features and the combination of both. Data from NASA-2014.

Input feature selection is the process of defining what elements will be in the input data $\mathbf{x}$ in a regression model. Fig. 1.2 shows the impact of a poor input feature choice on the NASA-2014 data using a data-driven regression model. The features in Fig. 1.2 were all from reference [152] with 'good' and 'bad' being defined as those features previously found to be more or less significant. Fig. 1.2 also neatly demonstrates how even a *bad* data-driven model can produce reasonable results. The specific details behind all models used to produce plots in this thesis are in Appendix A.

Each training cell contributes several rows of the input data array. Unlike the choice of input features, one will usually have no choice of which cells are in the training set

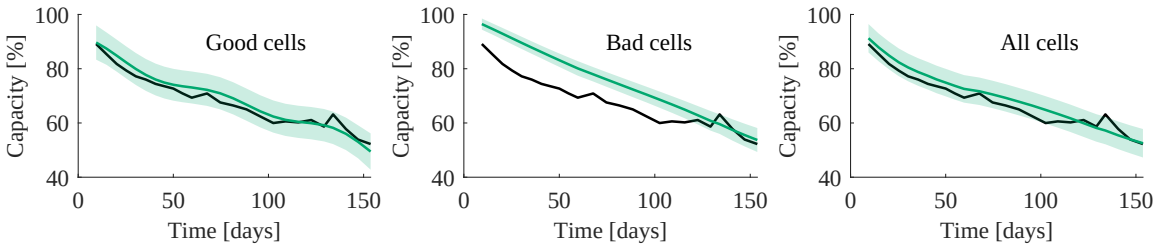


Fig. 1.3 Comparison of the predictive performance for a good choice of training cells, a poor choice and the combination of both. Data from NASA-2014.

but appropriateness of the training cells can be equally impactful. Fig. 1.3 demonstrates the possible impact of poor cell choice where the 'good' and 'bad' cells underwent similar or different use protocols to the test cell respectively. The consequence was a poor prediction when using the 'bad' cells to train the model because they had a different degradation trajectory.

Figs. 1.2 and 1.3 were deliberately constructed as simple toy examples[1]. However they demonstrate the need to create training data that can describe the degradation shape (i.e. input features) and contains sufficiently representative data (i.e. training cells) of the test cases.

All attempts at SoH prognosis in this thesis were performed using a SoH *transition* model so the target variable was changes in cell capacity ΔQ between subsequent time values. The training set then only needs to contain data points from cells of an equivalent age to a given test point to produce a good forecast. This assumes that degradation over some discrete time period is not a function of the cell history, unless past use is included in an input. Transition models reduce the need for well-matched

[1]The *bad features* contained no relevant information for the model, consequently the estimated degradation rate was a constant. The *bad cells* were a set of linearly ageing cells, therefore the predicted capacity was also linear. Both were deliberately chosen to highlight some of the problems associated with input selection.

cells by not requiring identically used training and test cells. Targeting ΔQ has been successful in literature for SoH prognosis [26, 149, 152] but does not eliminate the need to produce descriptive input features. Feature generation and selection are the subjects of Chapter 2.

1.4 Quantifying predictive performance

Most literature cases train SoH models on one set of cell data then test on another set. The predictive performance for the test set is quantified using performance metrics. Quantifying predictive performance can be split into profile-focussed metrics and point-focussed metrics.

Profile-focussed metrics assess the performance of a SoH prediction tool for estimating capacity (or another health metric) throughout life, by measuring how closely the observed capacity trajectory matches the predicted trajectory. These are usually expressed as either (root) mean square error (RMSE) in capacity [123, 152, 159] or mean absolute error (MAE) in capacity [44, 169] or both [32, 45, 134, 138, 149, 177]. Transition models can also use a further profile-focussed metric based on the same calculation, but for ΔQ [26, 149, 152]. This thesis will use RMSE, defined in equation 1.1 for the error between y_{meas} and y_{pred}, as the standard profile-focussed performance metric because it shares units with the capacity data and it puts a higher weight on

Reference	MAE	RMSE	EoL	Approach	Data	Remarks
[22]			9.1	Linear	Severson-2019	
[39]			9.4	Linear	Severson-2019	knee point
[42]	0.9			Deep NN	own data	
[44]				Deep GPR	NASA-2014	
[45]	1.8	0.6		GPR	NASA-2014	4 cells, $D \geq 9$
[45]	0.9	0.2		GPR	Severson-2019	RMSE<MAE
[60]			5.0	NN	own data	approx.
[61]			9.0	PF	own data	approx.
[138]	0.4	1.0		RF	NASA-2014 + others	
[138]	0.1	0.1		RF	Severson-2019	
[149]	1.1	1.2		GPR	own data	
[152]		3.3		GPR	NASA-2014	

Table 1.1 Summarized literature predictive performances. MAE and RMSE have units of % capacity. EoL values include RUL prediction accuracy, and knee point error when indicated, all in units of %.

poor predictions than MAE[2].

$$\text{RMSE}(y_{\text{meas}}, y_{\text{pred}}) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_{\text{meas}} - y_{\text{pred}})^2} \tag{1.1}$$

Point-focussed metrics quantify the predictive performance when targeting a specific point, for example EoL, RUL or knee point. All of these are in units of time or cycles depending on the dataset but the accuracy can be quoted as a percentage [22, 39, 60, 178]. The error, calculated in equation 1.2 for points p_{meas} and p_{pred}, will be summarised using the absolute values but Chapter 2 included an example case where

[2]For example, $x_{\text{meas}} = [0, 0, 0, 0]$, $x_{\text{pred}} = [1, 1, 1, 5]$, RMSE = 2.6, MAE = 2.0.

the true values can also be of interest.

$$\text{Error}(p_{\text{meas}}, p_{\text{pred}}) = 100\% \times \frac{p_{\text{pred}} - p_{\text{meas}}}{p_{\text{meas}}} = 100\% \times \left(\frac{p_{\text{pred}}}{p_{\text{meas}}} - 1 \right) \tag{1.2}$$

Some example performances from literature are included in Table 1.1. A subjective judgement of typical good performance (i.e. average of best values) has been applied where summary statistics couldn't be easily extracted from the works. The values were achieved on a variety of datasets so the table can only be used as an order of magnitude performance comparison.

Credible intervals, a measure of uncertainty, are easily extracted in cases where Bayesian methods were used. The only metric used commonly in previous literature is the calibration score which is the proportion of forecasts for which the observed values lie within two standard deviations of the predicted mean [26, 138, 149, 152]. Further discussion and analysis of credible intervals, calibration scores, and other, less common methods is provided in Chapter 4.

1.5 Gaussian process regression

Gaussian process regression (GPR) is a popular machine learning tool in battery health literature [45, 61, 143, 179]. It is flexible, probabilistic and non-parametric [109] and not as opaque as a neural network because the hyperparameters provide insight into the behaviour of the model [149, 180]. The majority of approaches in subsequent chapters use GPR for the machine learning stage. A derivation of the basic principle is below.

GPR Derivation

This derivation is intended to provide the basis for the following chapters in this thesis and is adapted from Rasmussen and Williams, *"Gaussian processes for machine learning"*, 2006, where one will find a fuller explanation [109].

The target variable y is assumed to be a function of inputs $f(\mathbf{x})$ plus noise, ϵ. The noise has variance σ_n^2:

$$y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{1.3}$$

The function is defined as a Gaussian process, i.e. it is *"a collection of random variables, any finite number of which have a joint Gausssian distribution"* [109]. In short, this means we describe $f(\mathbf{x})$ with a mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$. The function can then be written:

$$f(\mathbf{x}) \sim \mathrm{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{1.4}$$

Gaussian process regression is a Bayesian method and so the model must be treated according to Bayes' rule:

$$\text{Bayes' rule:} \quad p(A|B)p(B) = p(B|A)p(A) \tag{1.5}$$

$$\text{Bayes' rule for GPR:} \quad p(\mathbf{f}|\mathbf{y}, X)p(\mathbf{y}|X) = p(\mathbf{y}|\mathbf{f}, X)p(\mathbf{f}|X) \tag{1.6}$$

where the input array X is formed of rows of input vectors $\mathbf{x}^T$, and $\mathbf{f}$ is shorthand for $f(X)$. $\mathbf{y}$ and $\mathbf{f}$ are column vectors of the same length.

Equation 1.6 shows how Bayes' rule is applied for GPR. The input array X is unnecessary in the expression but is included for clarity. The first term of the right side, $p(\mathbf{y}|\mathbf{f}, X)$, is known as the *likelihood* and it represents the probability density of the observed target values given the model and inputs. The likelihood is expressed $p(\mathbf{y}|\mathbf{f}, X) = \mathcal{N}(\mathbf{f}, \sigma_n^2)$.

The other term on the right side is the *prior* which is the probability density of the function values given the data. Here, the prior is assumed to have a zero mean and covariance defined by inputs X: $p(\mathbf{f}|X) = \mathrm{N}(\mathbf{0}, K(X, X))$, where $K(X, X)$ is the covariance matrix containing the covariance between each input value. The second term on the left side $p(y|X)$ is known as the *marginal likelihood* and is independent of the function values. As a result, equation 1.6 is more commonly expressed as purely a function of the likelihood and prior with the *posterior* as the subject:

$$p(\mathbf{f}|\mathbf{y}, X) \propto p(\mathbf{y}|\mathbf{f}, X)p(\mathbf{f}|X) \tag{1.7}$$

The model is specified by the noise σ_n and any hyperparameters used to calculate covariance $K(X, X)$. The hyperparameters are fit by minimising the negative log marginal likelihood (NLML), $-\log p(\mathbf{y}|X)$, which is fully expressed as:

$$-\log p(\mathbf{y}|X) = \frac{1}{2}\mathbf{y}^T \left(K_{XX} + \sigma_n^2 I\right)^{-1} \mathbf{y} + \frac{1}{2}\log \left|K_{XX} + \sigma_n^2 I\right| + \frac{N}{2}\log 2\pi \tag{1.8}$$

where $K_{XX} = K(X, X)$.

After the NLML has been minimised, the final predictive posterior distribution over predictions $\mathbf{f}_*$ given test input points X_* can be written as:

$$p(\mathbf{f}_*|X, \mathbf{y}, X_*) = \mathcal{N}\left(m(\mathbf{f}_*), \text{cov}(\mathbf{f}_*)\right) \tag{1.9}$$

$$m(\mathbf{f}_*) = K(X_*, X)\left(K_{XX} + \sigma_n^2 I\right)^{-1}\mathbf{y} \tag{1.10}$$

$$\text{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)\left(K_{XX} + \sigma_n^2 I\right)^{-1} K(X_*, X)^T \tag{1.11}$$

Covariance functions

Covariance functions express how similar one data point is to another. They are the principal design choice within a GPR model. There are a large number of possible covariance functions available but the four that were used in this work were the radial basis function (RBF, a.k.a. squared exponential), Matérn-5/2 (M52), Matérn-3/2 (M32) and exponential (EXP). They were chosen because they are all functions of the distance between two points, $r = |\mathbf{x} - \mathbf{x}'|$, and vary in the assumed smoothness with RBF being smoothest and EXP at the opposite extreme [181]. The four functions are described in equations 1.12, 1.13, 1.14 and 1.15. Fig. 1.4 demonstrates how there is a limited difference between the shapes of the four covariance functions but example sample

functions can appear distinct.

$$\text{RBF:} \quad k_{\text{RBF}}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2\sigma_l^2}\right) \tag{1.12}$$

$$\text{M52:} \quad k_{\text{M52}}(r) = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right) \tag{1.13}$$

$$\text{M52:} \quad k_{\text{M32}}(r) = \sigma_f^2 \left(1 + \frac{\sqrt{3}r}{\sigma_l}\right) \exp\left(-\frac{\sqrt{3}r}{\sigma_l}\right) \tag{1.14}$$

$$\text{EXP:} \quad k_{\text{EXP}}(r) = \sigma_f^2 \exp\left(-\frac{r}{\sigma_l}\right) \tag{1.15}$$

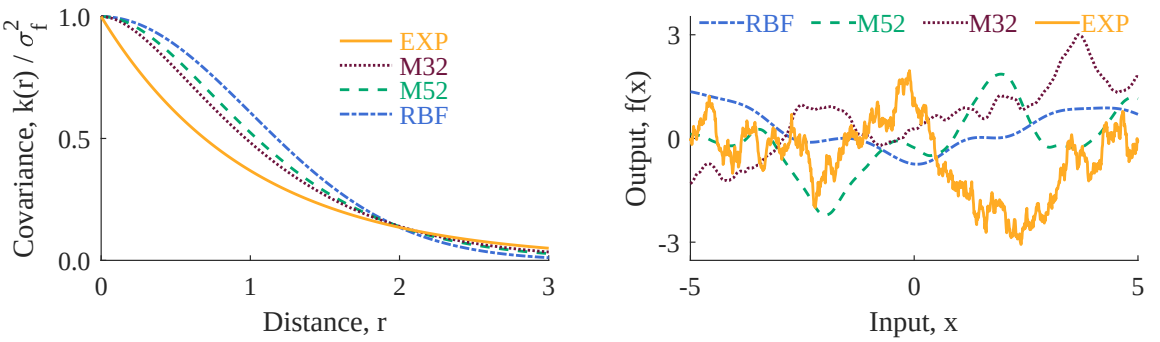


Fig. 1.4 The function shape (left) and an example sample (right) from four stationary covariance functions. Hyperparameters were set at $\sigma_f = 1$ and $\sigma_l = 1$ for all covariance functions.

The differences between the stationary kernels are small, but are visible in Fig. 1.4. All four return higher covariance values for shorter distances, with the key distinction being how fast or slow the covariance reduces in the $r < 0.5$ region of Fig. 1.4. These stationary kernels have been used in GPR models for SoH forecasting in literature, both in isolation [152] or combined with linear kernels [26, 149].

All four covariance functions are defined by the magnitude σ_f and lengthscale σ_l. Consequently a GPR model with one of the above covariance functions only requires 3 hyperparameters to be fit, σ_f, σ_l and σ_n.

1.5.1 Adaptations and Improvements

The majority of models used here have more than one input dimension. However, not all inputs are equally relevant to predict the target variable. Automatic relevance determination (ARD) is used to adjust the lengthscale for each input feature [182, 183]. ARD has been used for battery degradation models because it provides the flexibility to map the inherent but unobservable complexity of lithium-ion battery ageing [26, 149, 152, 184–186]. Without ARD one would be assuming explicit knowledge of the relative relevance between input features and the target variable, usually ΔQ here.

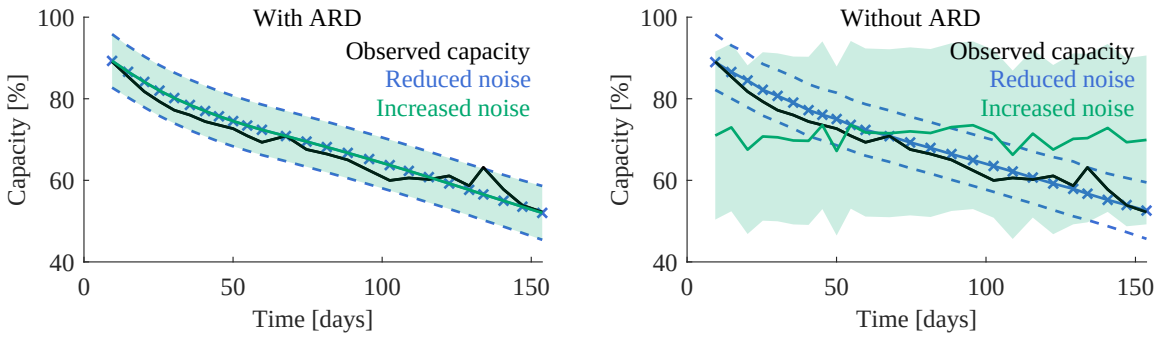


Fig. 1.5 Demonstration of the impact of using ARD by comparing two training sets with an extra noisy input, either much larger than the other input data or much smaller. ARD is shown to avoid the influence of the noisy input in both cases. Data from NASA-2014.

Fig. 1.5 demonstrates the consequence of using ARD by comparing two inputs for a cell from NASA-2014. An input feature of Gaussian noise was added to the input data, in one case of a size much larger than the other inputs, in the other much smaller. The performance is equivalent between the two when ARD is used, but significantly weakened without. With ARD, the lengthscales for the noisy input became very large, but small for the important input features. The relative values of the lengthscales can

be used in analysis of a model, potentially providing incisive information for a user. All GPR models used here make use of ARD without further adaptation.

Unfortunately, the inversion in equation 1.8 means that computational complexity increases as $\mathcal{O}(N^3)$ [100, 187]. Furthermore, the full training set must be retained in order to calculate the predictive posterior [188]. A solution to the scaling problem is to use sparse Gaussian process regression (SparseGPR), where a small number, M, of pseudo-inputs, Z, are used to approximate the posterior distribution when calculated using the full training data set X.

Mathematics of SparseGPR

The following derivation of SparseGPR is based on reference [189], but references [188] and [190] are also good resources for further reading.

Consider pseudo-inputs Z of size $M \times D$ with pseudo-targets $\mathbf{f_z} = f(Z)$ of size $M \times 1$. Each row of Z represents a single pseudo-input corresponding to the elements of $\mathbf{f_z}$. The targets are not observed variables so are denoted with $\mathbf{f_z}$ not $\mathbf{y_z}$. Similarly, there is no noise variance on these targets. The noise hyperparameter σ_n is still calculated based on the real training data. These assumptions produce the likelihood for the training set, X, shown in equation 1.16. For simplicity, let $K_Z = K(Z, Z)$, $K_X = K(X, X)$ and $K_{ZX} = K(Z, X)$.

$$p(\mathbf{y}|X, Z, \mathbf{f_z}) = \mathcal{N}(\mathbf{y}|K_{ZX}^T K_Z^{-1} \mathbf{f_z}, \Lambda + \sigma_n^2 \mathrm{I}) \tag{1.16}$$

where $\Lambda = \text{diag}(\lambda)$, $\lambda_i = K(\mathbf{x_i}, \mathbf{x_i}) - K(\mathbf{x_i}, Z)K_Z^{-1}K(Z, \mathbf{x_i})$.

Equation 1.16 is the data likelihood. The pseudo-targets $\mathbf{f_z}$ can be integrated out explicitly instead of maximising equation 1.16 [191]. First, a Gaussian prior is placed over the pseudo-targets $\mathbf{f_z}$ with zero mean.

$$p(\mathbf{f_z}|Z) = \mathcal{N}(\mathbf{0}, K_Z) \tag{1.17}$$

The resultant likelihood is independent of the targets and sometimes referred to as the *approximate* likelihood [192]. For simplicity, here it is expressed as an explicit calculation[3]:

$$-\log\left[p(\mathbf{y}|X, Z, \mathbf{0})\right] = -\log\left[\mathcal{N}\left(\mathbf{y}|\mathbf{0}, K_{ZX}^T K_Z^{-1} K_{ZX} + \sigma_n^2 I\right)\right] \tag{1.18}$$

The pseudo-target posterior distribution, specifying the pseudo-targets as a function of the training data and the fixed pseudo-inputs, can then be expressed:

$$\begin{aligned} p(\mathbf{f_z}|\mathbf{y}, X, Z) =& \mathcal{N}(\mathbf{f_z}|K_Z Q_Z^{-1} K_{ZX}(\Lambda + \sigma_n^2 \mathrm{I})^{-1}\mathbf{y}, K_Z Q_Z^{-1} K_Z) \\ Q_Z =& K_Z + K_{ZX}(\Lambda + \sigma_n^2 \mathrm{I})^{-1} K_{ZX}^T \end{aligned} \tag{1.19}$$

[3]The form of equation 1.18 depends on the sparse approximation used [193]. Here, the chosen form is known as deterministic training conditional and is the default for SheffieldML's GPy, the chosen tool for implementing sparse GPR [192–194].

The predictive posterior is formed without any dependence on the pseudo-targets by integrating through all $\mathbf{f_z}$,

$$\begin{aligned} p(\mathbf{y}_*|\mathbf{y}, X, X_*) =& \int p(\mathbf{y}_*|X_*, Z, \mathbf{f_Z}) p(\mathbf{f_z}|\mathbf{y}, X, Z) d\mathbf{f_Z} \\ =& \mathcal{N}\left(K_{*Z} Q_Z^{-1} K_{ZX} (\Lambda + \sigma_n^2 \mathrm{I})^{-1} \mathbf{y}, K_* - K_{*Z}(K_Z^{-1} - Q_Z^{-1}) K_{*Z}^T + \sigma_n^2 I\right) \end{aligned} \tag{1.20}$$

where $K_* = K(X_*, X_*)$, $K_{*Z} = K(X_*, Z)$.

It is important to note that the matrix inversion in equation 1.20, $(\Lambda + \sigma_n^2 \mathrm{I})^{-1}$, is not a speed concern because the array is diagonal. This reduces the scaling factor to $\mathcal{O}(M^2 N)$ which is significant, especially when $M \ll N$ [188, 195]. Sparse GPR has previously been used for on-board state of charge estimation [196], where there is a need for rapid calculation, and SoH prediction [113, 141]. Its potential to further economise a GPR model is explored in Chapter 4, including a proposal to economise the storage of a SparseGPR model by only retaining the pseudo-input and pseudo-target pairs.

The GPR models in Chapters 2 and 3 were standard GPR unless computational limits were reached. Matlab's *fitrgp* uses a standard GPR approach until $N > 2,000$, above which a sparse approximation with $M = 2,000$ is applied [197]. SheffieldML's GPy was configured to use a similar approximation when used as a standard GPR tool in Chapter 4, but set at $N = M = 200$ for speed, and without impacting performance [194]. A standard input dataset in models here was 50 training cells, roughly $N = 2,500$ data points, so these conditions applied in the majority of cases.

1.6 Research focus and contributions

There remain a number of challenges regarding the use of data-driven approaches for battery SoH prognosis. This thesis includes a number of investigations, all aimed at tackling those challenges.

As previously mentioned, prediction of cell use is beyond the scope of this thesis. Use prediction remains a significant challenge and so any health prediction must acknowledge that uncertainty. Lithium-ion batteries are also used in a large variety of ways, and there can be significant issues regarding the collection and recording of raw field data. The first challenge that this thesis tackles is the production of applicable input features, featured in Chapter 2. The input features need to be predictable, therefore inputs relying on exact values of voltage or current are ignored here. Equally, applicability will require that the input features are robust to reduced data quality and also flexible to different use cases.

The principal regression tool used in this thesis is GPR. Like other machine learning methods, GPR can be slow to train with large datasets and it is difficult to interpret. Replacing the flexible machine learning techniques with a faster, more transparent modelling approach would be of significant value in the field of SoH prognosis. Chapter 3 proposes a piecewise linear regression approach that aims to replace GPR without impeding predictive performance.

Both GPR and piecewise linear regression are Bayesian techniques that produce predictive posteriors. The credible intervals about all predictions are ignored in Chapters 2 and 3 but they form a crucial part of any forecast. Battery literature lacks

comprehensive performance metrics for predictive distributions. A further challenge in battery SoH prognosis is that producing predictive distributions has significant storage requirements. Chapter 4 includes a proposed solution to both problems, used in conjunction in Section 4.3.

Cell-to-cell variability is known to be a significant factor in lithium-ion battery degradation [16, 18]. All battery health models must capture cell-to-cell variability, but quantifying cell-to-cell variability is challenging. Chapter 4 also includes an attempt to quantify cell-to-cell variability by estimating the sample size required to consistently fit an empirical model.

1.7 Available datasets

1.7.1 Severson-2019 and Attia-2020

The vast majority of models in this thesis use the data from Severson-2019 and Attia-2020. The first contains cycle data for 130 cells, 124 of which were used in reference [22]. The cells were A123 lithium iron phosphate cathode/graphite anode 18650 cells with a nominal capacity of 1.1 Ah. The data is part of a fast-charging investigation under an extreme use case. All discharge cycles were at a C-rate of 4C (A C-rate of 4C means that the current would fully (dis-)charge a cell in 1/4 hours, or 15 minutes) with a constant voltage step and there were varying charging protocols, as specified in the supplementary material of ref. [22]. The data was collected over 3 batches on a 48 channel Arbin LBT potentiostat and the thermal chamber was held at 30 °C using

forced convection. The charging protocols used currents in steps that ranged between 3C and 8C, but each cell underwent the same protocol throughout its life. The cells in Severson-2019 and Attia-2020 underwent use protocols that were significantly more severe than the other datasets mentioned in this section.

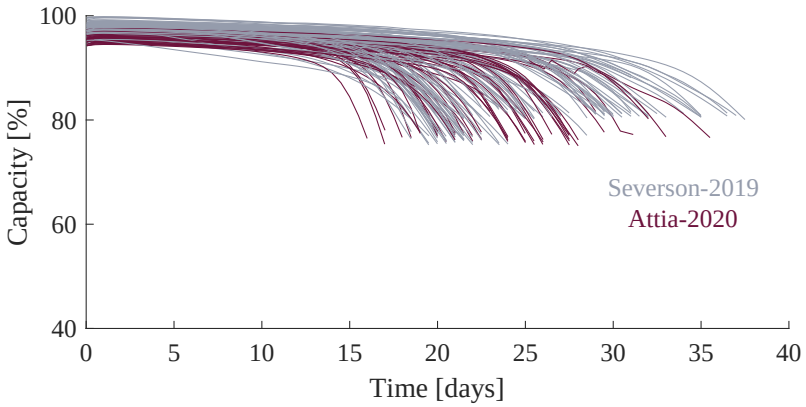


Fig. 1.6 Capacity profiles from Severson-2019 and Attia-2020 datasets.

Attia-2020 was a second contribution to the series of work looking at fast-charging protocols [107]. The full dataset includes 235 cells, but only 45 were cycled to failure, defined as 80% of the 1.1 Ah nominal capacity for both works. The 45 cells were used on a reduced range of charging protocols. The currents were still staggered, but all charges had a duration of 10 minutes followed by a constant voltage phase. All other conditions were nominally identical to those in Severson-2019.

These cells were the principal dataset used here because they exhibit the knee point. If a data-driven model can accurately forecast capacity or ΔQ through the changing degradation rate then there is strong evidence that the method is detecting an effective signal. As a comparison, it is difficult to assess the benefit of different

modelling approaches using NASA-2014 or Raj-2020 because a simple linear model would be only slightly weaker.

Full data repository for Severson-2019 and Attia-2020 found at: https://data.matr.io/1/

1.7.2 NASA-2014

The authors of the NASA-2014 random use dataset deserve significant credit for this good quality, detailed, ergonomically published, and publicly available dataset [33, 198]. The accompanying requested citation, reference [198], has between 70 and 100 citations at time of writing because of the wide ranging utility of this data [199, 200]. The ageing is largely linear and all 28 cells are plotted in Fig. 1.1b.

The dataset contains 28 lithium cobalt oxide cathode/graphite anode 18650 cells undergoing *random use*. The dataset is split into seven groups of four, each group being characterised by a particular usage probability distribution. The nominal capacity was ≈2.2 Ah and most were cycled to below 60% capacity. The use of this dataset was limited in this thesis because the ageing is mostly linear in time and there were insufficient data points to confidently extract insight on the efficacy of data-driven modelling techniques.

Data for NASA-2014 is found here[4]:

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

[4]Scroll down to dataset 11, *Randomized Battery Usage Data Set*

1.7.3 Raj-2020

The Raj-2020 dataset was featured in Fig. 1.1b as an example of linear ageing. It contains 12 cells that were part of a study into path dependency [34, 201]. The cells had nickel cobalt aluminium oxide positive electrodes and graphite negative electrodes, they were manufactured by Panasonic and had a cylindrical 18650 form factor.

The 12 cells were divided into four groups of three. The four groups were effectively two pairs, with each pair of groups having an identical charge and current throughput over long time intervals, but the cycling was performed in a different pattern. The results of the study suggested that path dependency is present in lithium-ion battery degradation [34]. These cells were not widely used because of the limited dataset size, the linear ageing, and the limited extent of the ageing.

Data found here:

https://ora.ox.ac.uk/objects/uuid:de62b5d2-6154-426d-bcbb-30253ddb7d1e

1.7.4 Dechent-2017 and Dechent-2020

Dechent-2017 contained 21 18650 cells with a 1.5 Ah nominal capacity produced by Samsung. They had lithium nickel manganese cobalt oxide positive electrodes and graphite negative electrodes. The cells were continuously cycled over 90% of the cycle depth and current rates varied around a 1C charge and 6C discharge.

Dechent-2020 contained 22 18650 cells with a 3.4 Ah nominal capacity produced by Samsung. They had lithium nickel cobalt aluminium oxide positive electrodes and

graphite negative electrodes. The cells were continuously cycled at C/2 (i.e. equivalent to a full (dis-)charge in 2 hours) over 20% of the cycle depth about 50% state of charge.

The lead author has stated that Dechent-2017 and Dechent-2020 will be publicly available soon [202] but provided the following references too [203, 204]. These datasets were used in the cell-to-cell variability study in Section 4.2.

1.7.5 Sauer-2021

Sauer-2021 is a dataset with 48 identically used, nominally identical Sanyo/Panasonic UR18650E cells with a nominal capacity of 2.05 Ah. These cells have a lithium nickel manganese cobalt oxide positive electrode and graphite negative electrode [16, 205, 206]. Sauer-2021 was used in Chapter 4 as part of the study into cell-to-cell variability and is also the subject of Appendix C.4. Previously published work named this dataset Baumhöfer-2014 after the author and publication date of the first paper to feature the data [3]. Sauer-2021 was used here because the full dataset was released separately in 2021 [206].

1.8 Thesis outline

Chapter 1: Introduction described the variety of methods that have been used to model capacity degradation, with a particular focus on the data-driven approaches. The importance of input data quality was briefly demonstrated on the NASA-2014 dataset.

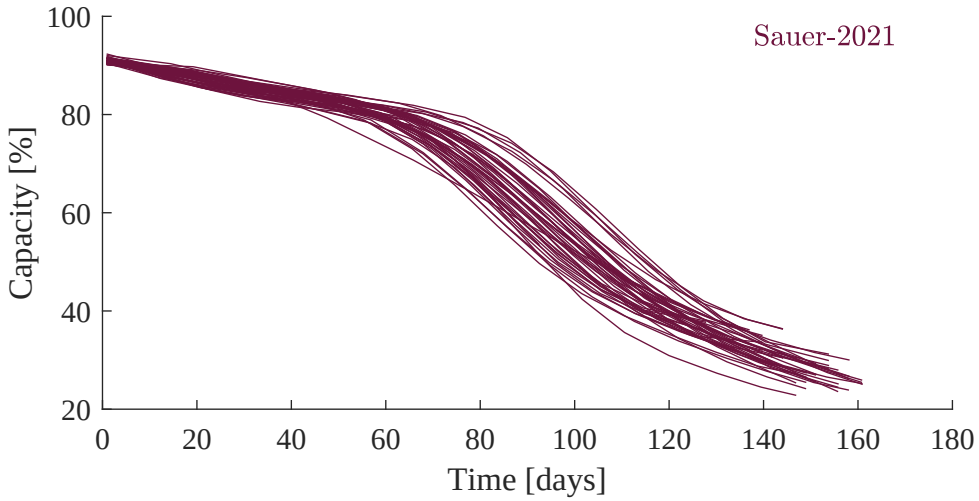


Fig. 1.7 Capacity profiles for Sauer-2021 dataset. Note, the health here is plotted over a wider capacity range than for other datasets.

Chapter 2: Automated input feature generation and selection presents an approach to creating input features for a lithium-ion battery degradation model. Performance is presented as a function of all major model design choices. Gaussian process regression is used in all cases and there is an investigation into the kernel function choice.

Chapter 3: Piecewise linear regression for battery health modelling describes an alternative approach to regression. The advantages that can be gained in speed and insight are demonstrated on the Severson-2019 and Attia-2020 datasets. The limits of the presented modelling are found by varying key controls and using Sauer-2021.

Chapter 4: Capturing degradation uncertainty explores the impact of uncertainty in capacity forecasting. Firstly there is an attempt to quantify cell-to-cell variability using multi-level Bayes and limited empirical models. Second, a credible

interval performance metric is presented and used to compare the methods presented earlier in the thesis.

Chapter 5: Conclusions summarises the contributions and conclusions of the work presented in the previous three chapters. The limitations of the work are discussed and future research areas are suggested.

Appendix A: The models behind the figures and tables contains all the details used to produce plots and tables in the thesis. There are many models and studies presented, these details should allow a reader to reproduce the methods exactly.

Appendix B: Extra trials - Chapter 2 presents the results of any trials used in Chapter 2 in cases where the results were considered to be too large relative to their contribution. A SoH estimation tool is also presented.

Appendix C: Extra trials - Chapter 3 includes the results of several trials used in the production of Chapter 3 but that were considered insufficiently informative for presentation in the main chapter, including applying prognosis methods to the Sauer-2021 dataset.

Appendix D: Quantifying cell-to-cell variability provides a fuller derivation of the multi-level Bayes approach included in Chapter 4 and presents the omitted results.

Chapter 2

Automated input feature generation and selection

The introduction noted that input selection is important for a data-driven model. This chapter focuses on producing an accurate capacity forecasting model through careful input feature extraction and predicting changes in capacity, ΔQ. The process includes producing the features (Section 2.1) and subsequently selecting those to be used in the resultant machine learning model (Section 2.2). Most of this chapter has been published in reference [1].

All of the following feature engineering and degradation modelling is automated, i.e. there is no user input required from collecting the raw cycle data through to predicting the end-of-life for a given cell.

Section 2.1 explains how input features are generated here and in future chapters. Section 2.2 introduces an automated selection methodology that is both fast and effective before Section 2.3 describes how the resultant health model is formed. Sections 2.4, 2.5 and 2.6 propose various trials, present the results and finally discuss the

performance of the proposed approach. Lastly, the chapter is concluded in Section 2.7, which also addresses key areas for improvement.

2.1 Input feature generation

Input features were extracted from raw cycle data, i.e. current I, voltage V, and temperature T as functions of time. The feature generation process used here took over 100 million rows of cycle data and returned under 10,000 rows of feature data for 157 cells from Severson-2019 and Attia-2020 [22, 107]. The input features are intended to describe cell use without requiring knowledge of the use type, the chemistry or any other metadata. This flexibility was achieved by calculating input features based on percentiles of each variable calculated from the training data; percentiles automatically scale with various use cases, voltage limits and environmental conditions. The input features were then taken as the proportion of time spent between given percentile thresholds.

2.1.1 Collecting cycle data

Different datasets will have different measured variables available but most laboratory data includes current and voltage. Some, including Severson-2019 and Attia-2020, also have cell temperature. In the case of the Severson-2019 and Attia-2020 datasets, further inputs were calculated such that the full set of variables included current, voltage, temperature, power, absolute current and absolute power.

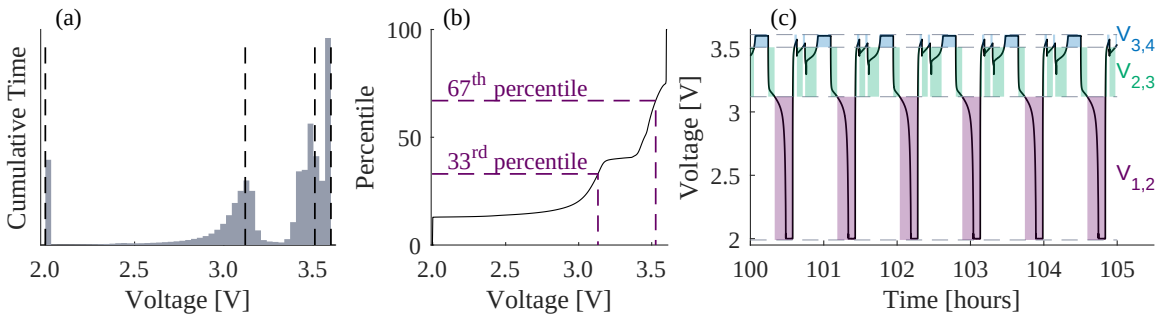


Fig. 2.1 Diagram of the feature generation method. (a) Cell usage was converted into a histogram by counting time spent within small intervals. (b) The 1st, 33rd, 67th and 99th percentiles of each variable were extracted. (c) The feature values were calculated based on the proportion of time spent in regions between those stated percentiles.

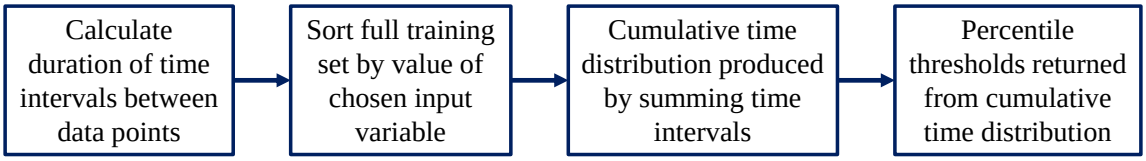


Fig. 2.2 Flowchart summarising the steps between the raw cycle data in the training set and the generated variable thresholds.

2.1.2 Calculating percentiles

The percentiles were calculated once all the raw variables had been calculated and collected. The 1st, 33rd, 67th and 99th percentiles represented the value below which the entire training set spends 1%, 33%, 67% and 99% of time. Fig. 2.1.a shows an example histogram of a training set's voltage data, which was converted to a cumulative distribution for Fig. 2.1.b. The cumulative distribution was then used to calculate the desired percentiles, also shown in Fig. 2.1.b. The 1st and 99th percentiles acted as a minimum and maximum of typical use. The single percentage point means that any unphysical data points will not be relevant to the calculation. The procedure is also shown as a flowchart in Fig. 2.2 for clarity.

Percentile	Current [A]	Voltage [V]	Temperature [°C]	Power [W]	Abs. Current [A]	Abs. Power [W]
1st	-4.00	2.00	30.0	-12.83	0.00	0.00
33rd	-0.38	3.12	32.5	-0.76	0.98	2.67
67th	1.00	3.51	35.0	3.43	4.00	12.35
99th	6.00	3.60	40.6	21.22	6.00	21.22

Table 2.1 Feature thresholds calculated from the combined Severson-2019 and Attia-2020 datasets for all available raw variables. These values were used for input features in future chapters.

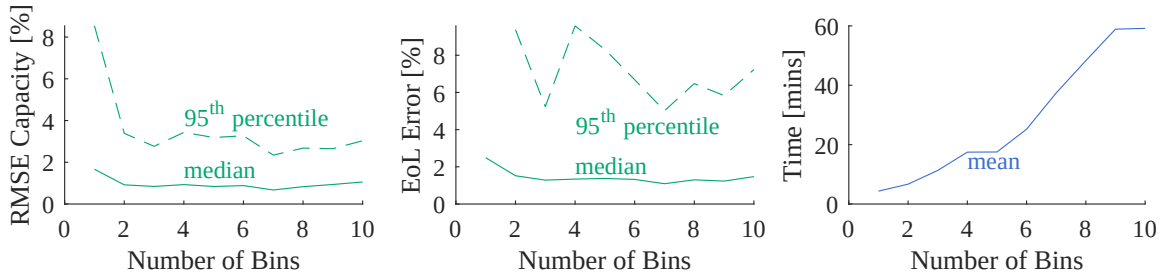


Fig. 2.3 Performance of feature engineering approach as function of number of bins used to calculate features. Trial used only 10 repeats and did not allow any features covering percentile ranges smaller than 20% of the data.

Four percentile threshold points were selected for speed after inspecting a small number of the input data histograms, like that in Fig. 2.1.a. The values of these thresholds for the 157 cells most commonly used here are shown in Table 2.1. Four thresholds creates three bins: 1st to 33rd, 33rd to 67th and 67th to 99th percentiles.

The results shown in Fig. 2.3 demonstrate the impact of increasing the number of percentile thresholds on both predictive performance and the computational time taken. Root mean square errors (RMSE), end-of-life (EoL) error and the use of medians are all explained in section 2.4. There is noticeable noise because only 10 repeats were run, and even then the entire trial lasted around 3 days. However there was no significant performance improvement beyond the 3 bins shown in Table 2.1, but there was a

significant cost in terms of time taken. As with all models used to produce plots in this work, more details of the model can be found in Appendix A.

2.1.3 Generating features

The values for the input features were calculated based on the time spent between two thresholds during a given time interval. Time spent in a certain region has been used in literature previously, but this assumes that the intervals are the same across a training and testing set [26, 149, 152]. Alternatively, using the proportion of time spent in a given region is more flexible to varying time intervals between health measurements. Consider voltage V as an example variable. Equation 2.1 shows how the feature value between time t_{i-1} and t_i and between voltage thresholds V_n and V_m ($V_n < V_m$) is calculated,

$$V_{n,m}(t_i) = \frac{\int_{t_{i-1}}^{t_i} R\left(V_n < V(t) \leq V_m\right) dt}{t_i - t_{i-1}}, \quad R(\text{True}) = 1, \quad R(\text{False}) = 0 \tag{2.1}$$

All input features that were calculated by equation 2.1 are listed in Table 2.2 using the variable thresholds in Table 2.1.

Lastly, the difference between consecutive feature values acted as an input feature. The aim was to produce input features that were functions of changing use:

$$\Delta V_{n,m}(t_i) = V_{n,m}(t_i) - V_{n,m}(t_{i-1}) \tag{2.2}$$

Feature label	Variable	Unit	Minimum value	Maximum value
$I_{1,2}$	Current	A	-4.00	-0.38
$I_{1,3}$	Current	A	-4.00	1.00
$I_{1,4}$	Current	A	-4.00	6.00
$I_{2,3}$	Current	A	-0.38	1.00
$I_{2,4}$	Current	A	-0.38	6.00
$I_{3,4}$	Current	A	1.00	6.00
$V_{1,2}$	Voltage	V	2.00	3.12
$V_{1,3}$	Voltage	V	2.00	3.51
$V_{1,4}$	Voltage	V	2.00	3.60
$V_{2,3}$	Voltage	V	3.12	3.51
$V_{2,4}$	Voltage	V	3.12	3.60
$V_{3,4}$	Voltage	V	3.51	3.60
$T_{1,2}$	Temperature	°C	30.0	32.5
$T_{1,3}$	Temperature	°C	30.0	35.0
$T_{1,2}$	Temperature	°C	30.0	40.6
$T_{2,3}$	Temperature	°C	32.5	35.0
$T_{2,4}$	Temperature	°C	32.5	40.6
$T_{3,4}$	Temperature	°C	35.0	40.6
$P_{1,2}$	Power	W	-12.83	-0.76
$P_{1,3}$	Power	W	-12.83	3.43
$P_{1,4}$	Power	W	-12.83	21.22
$P_{2,3}$	Power	W	-0.76	3.43
$P_{2,4}$	Power	W	-0.76	21.22
$P_{3,4}$	Power	W	3.43	21.22
$\|I\|_{1,2}$	Abs. Current	A	0.00	0.98
$\|I\|_{1,3}$	Abs. Current	A	0.00	4.00
$\|I\|_{1,4}$	Abs. Current	A	0.00	6.00
$\|I\|_{2,4}$	Abs. Current	A	0.98	4.00
$\|I\|_{2,4}$	Abs. Current	A	0.98	6.00
$\|I\|_{3,4}$	Abs. Current	A	4.00	6.00
$\|P\|_{1,2}$	Abs. Power	W	0.00	2.67
$\|P\|_{1,3}$	Abs. Power	W	0.00	12.35
$\|P\|_{1,4}$	Abs. Power	W	0.00	21.22
$\|P\|_{2,3}$	Abs. Power	W	2.67	12.35
$\|P\|_{2,4}$	Abs. Power	W	2.67	21.22
$\|P\|_{3,4}$	Abs. Power	W	12.35	21.22

Table 2.2 Detailed list of input features generated here, excluding all the functions of differences through time (e.g. $\Delta V_{2,3}$). The minimum and maximum values are the variable thresholds as calculated for the combined Severson-2019 and Attia-2020 datasets, listed in Table 2.1.

The equivalent feature labels for current, temperature, power, absolute current and absolute power are $I_{n,m}$, $T_{n,m}$, $P_{n,m}$, $|I|_{n,m}$ and $|P|_{n,m}$ respectively. Input features were generated for all $m > n$ so some features were functions of multiple bins.

Other input features could have been included here without difficulty. For example, an input feature describing time spent at very small or zero current could provide insight into the role of calendar ageing. However, the percentile-based approach adapts to different use protocols, meaning that there will always be an approximation of the above calendar ageing input feature where there is also current data. Here, that approximation would be $|I|_{1,2}$, the proportion of time spent between currents of 0 A and 0.98 A.

Table 2.2 shows that some of the generated input features contain duplicated information, potentially rendering them redundant. For example, $V_{1,3}$ is the sum of $V_{1,2}$ and $V_{2,3}$. Section 2.2 describes a technique that automatically selects a set of input features for a regression model which removes any instances of input features with strong correlations. Consequently, the duplicated information was not expected to cause poor fits or overfitting[1].

Capacity was calculated from regular 4C discharge cycles for the Severson-2019 and Attia-2020 datasets. As a result, the cells in Severson-2019 and Attia-2020 have around 1,000 (between 600 and 2,000) individual measures of health over their lifetimes which can be selected from. By contrast, some datasets have convenient reference performance tests (RPT) at sporadic points to determine one or more consistent health

[1]It was possible that $V_{1,3}$ could have been selected with either $V_{1,2}$ or $V_{2,3}$, but it was deemed unlikely that $V_{2,3}$ would be selected with both.

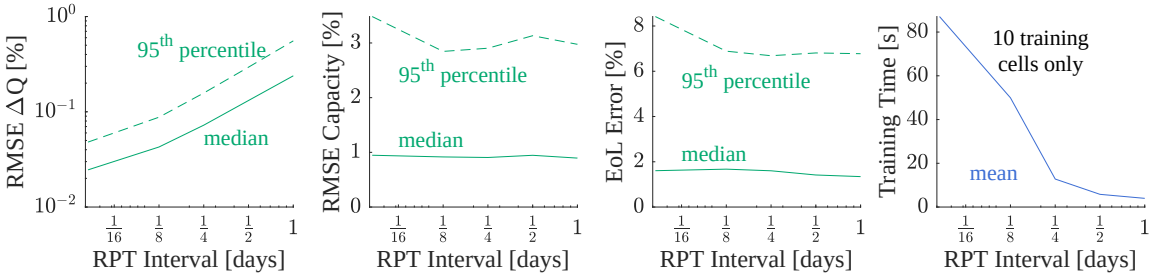


Fig. 2.4 Modelling performance as a function of RPT frequency. 20 repeats per test point with 50 training cells and 107 test cells each time, all from the Severson-2019 and Attia-2020. Features calculated using the thresholds in Table 2.1.

.

metrics, such as datasets NASA-2014 [33] and Raj-2020 [201], where the RPTs neatly divide the data into sections.

Where one has a choice, the decision over the frequency of health measures is a compromise between computation time and detail in the training set. Fig. 2.4 demonstrates that while the fitting of RMSE ΔQ scales with the frequency of capacity measurements because there was more data at higher frequencies, there was very little impact on the accuracy of capacity profiles when the ΔQ predictions were combined. There appeared to be a weakened performance at the 95$^{\text{th}}$ percentile of EoL forecasting where very regular capacity measurements were used, possibly because of overfitting.

Time taken to train the GPR model had the most distinct dependence on the RPT frequency because more frequent RPTs created larger input data arrays. Fig. 2.4 shows the time taken for an example with only 10 training cells in order to save time and to limit the effect of approximation techniques in Matlab's *fitrgp* tool. The other plots used 50 training cells.

Feature generation was the time-limiting process whenever used, even with only 3 bins and RPTs every 12 hours. Consequently, most investigations in this thesis used a single set of features that were generated using all 157 cells with lifetimes between 15 and 40 days from the combined Severson-2019 and Attia-2020 datasets; only this chapter includes trials with automated feature generation. Using the same feature set for investigations dramatically reduced time taken, but introduced an undeserved performance improvement by producing features based on the test data.

In summary, feature generation procedure produced input features based on the time spent in automatically calculated regions of use. Four percentiles were used to calculate those regions as a compromise between speed and detail. The end result was that each cell had 36 features based on the proportions of time, 36 features calculated based on the change in those features over time, and a final input variable consisting of the calendar time. The regression tool then needed to narrow the 73 possible input features down to a sensible quantity.

2.2 Automated feature selection

All degradation prognosis algorithms use some form of feature selection, even if implicit. The process aims to produce input features which will be successful predictors of some target variable. Unless specified, the target variable in models in Chapters 2 and 3 was ΔQ.

In general, measures of correlation are used for computation-based feature selection in battery literature. The simplest of the measures of correlation used is the Pearson correlation coefficient, typically defined for two variables x_i and x_j as [45, 112, 207],

$$\rho_P(x_i, x_j) = \frac{\text{cov}(x_i, x_j)}{\sigma(x_i)\sigma(x_j)} \tag{2.3}$$

The numerator, $\text{cov}(x_i, x_j)$, is the covariance between the two variables and $\sigma(x)$ is the standard deviation of each given variable. Spearman's rank, the Pearson correlation coefficient after the two variables are sorted by the size of the former, has also been successfully applied feature selection for SoH estimation and forecasting [112, 159, 173, 174, 208, 209]. A more complex measure of the relationship between two variables is Grey relational analysis. It is based on Grey system theory and has been used in battery literature with success [143, 164, 175, 210–212].

Regardless of the measure of the correlation or application, the techniques are always applied so that the best input features with the strongest correlation with the target variable are subsequently applied in predictive tools. Correlation-based selection techniques have the advantage of not requiring the model to be trained, thereby making them fast. However selecting the best correlating features could result in needlessly including extremely similar variables in a regression tool which would increase complexity [32, 45].

Alternative feature selection approaches exist. Sensitivity analysis [152], sequence backward search [45, 213] and recursive feature elimination [138, 214, 215] have been

applied in literature. These methods will return an effective and varied set of input features but all of these techniques require one or more SoH models to be trained. Model training is a slow and complex process, so these methods have previously been used to select from an already small number of features [45, 138]. Principle component analysis was considered because the input features would be orthogonal but it was rejected because the input features would be extremely difficult to interpret, and one would still need to select a subset of them.

The method proposed below used the fast, correlation-based methods to automatically select input features from the large array produced in Section 2.1, but discriminated based on shared correlation between input features, to remove very similar features and return an effective but varied set of inputs. The process is similar to sensitivity orthogonalisation, where a subset of input features that have orthogonal impacts on the model outputs instead of the highest sensitivities, but without the need to train a model [216].

The process was as follows: (1) Calculate $|\rho_P|$ between all potential input features and ΔQ, including the correlation coefficients between the potential input features. (2) Select the feature with the strongest correlation with ΔQ. (3) Remove all other features from consideration that correlate too strongly with the selected feature. (4) Repeat steps (2) and (3) until the desired number of features are selected.

Fig. 2.5 presents an example set of automatically selected input features. The first selected input feature, $V_{2,3}$ (i.e. time between voltage thresholds 2 and 3), showed a

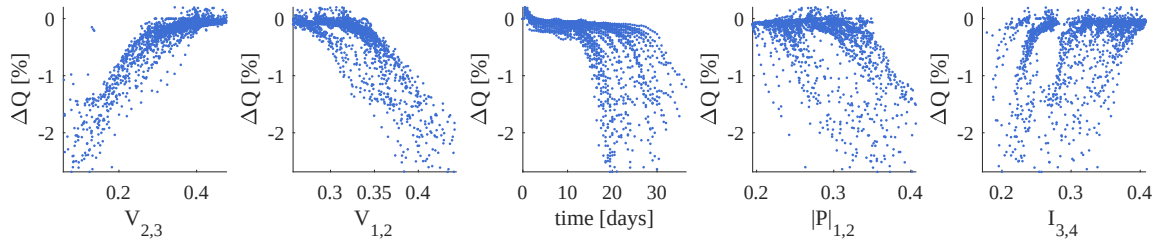


Fig. 2.5 The most commonly selected features in this study. Most to least common selections runs left to right.

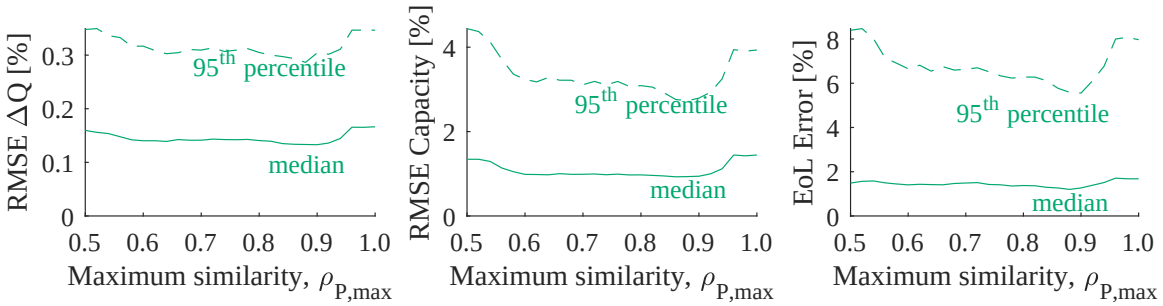


Fig. 2.6 Modelling performance as a function of the maximum shared correlation between input features, $\rho_{P,\max}$.

clear monotonic relationship with ΔQ but that relationship then visibly weakened with each subsequent feature.

Step (3) is critical for the success of the proposed procedure and its purpose was to produce inputs that were not correlated with one another. The cross-correlation condition required a decision of what was a reasonable value of maximum shared correlation, $\rho_{P,\max}$. The majority of models here used $\rho_{P,\max} = 0.85$ but the impact of the $\rho_{P,\max}$ on performance was examined and plotted in Fig. 2.6. There was a distinct improvement at 0.85 compared with selecting the best correlating features ($\rho_{P,\max} = 1.0$) suggesting that the feature selection method was producing a reliable set of input features for the subsequent Gaussian process regression (GPR) model. There

was very little change in performance below $\rho_{P,\max} = 0.90$, suggesting that critical shared correlation coefficients were above that value.

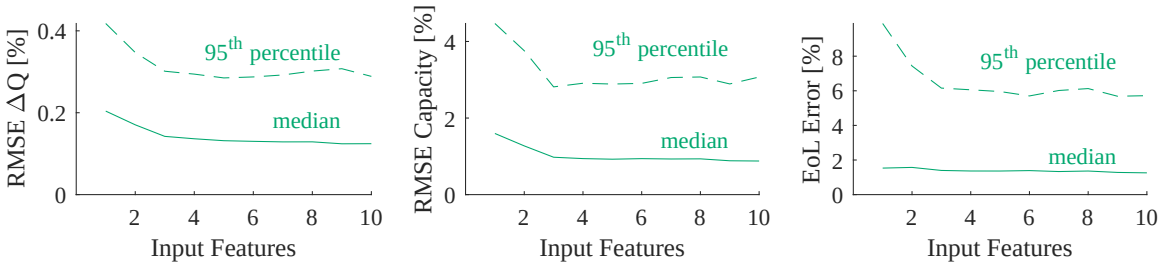


Fig. 2.7 GPR modelling performance as a function of the number of selected input features.

Typically, five input features were selected but performance was consistent with three or more, as shown by the results in Fig. 2.7. A user needs to decide how many input features to select as a compromise between increased information for the subsequent machine learning model and the computational complexity and the risk of overfitting, a decision that will depend on the dataset being studied. The limit of three features is also supported by Fig. 2.5 where the fourth and fifth selections had a very weak relationship with the target variable.

2.3 Degradation model

The work in Chapter 2 used models that mapped to ΔQ as target variable. The capacity profiles were then calculated by assuming knowledge of initial capacity and summing the forecasted changes in capacity, based on assumed usage, for all time steps for which there was data available. The feature generation and selection procedures

Kernel	NLML	RMSE ΔQ [%]		RMSE Capacity [%]		EoL Error [%]	
		median	95^{th}	median	95^{th}	median	95^{th}
RBF	-6543	0.17	0.41	1.2	4.5	1.8	10.0
M52	-6673	0.16	0.36	1.1	4.0	1.6	8.6
M32	-6728	0.15	0.34	1.1	3.5	1.5	7.5
EXP	-6666	0.13	0.31	0.9	2.9	1.4	6.1

Table 2.3 Comparative predictive performance of four stationary kernels on a trial with 100 repeats of 50 training cells. NLML was calculated as in equation 1.8 and using the all available cells in the training data.

can alternatively be used to predict capacity directly, an example of which is in B.2 as part of a study into input noise.

Sections 2.1 and 2.2 discussed the generation and selection of a set of varied but simple input features for use in a GPR model. The choice of kernel function was decided by a trial on the Severson-2019 and Attia-2020 datasets using 50 training cells and the variable thresholds in Table 2.1. Table 2.3 shows that the exponential kernel performed best, with steadily weakening predictive performance as the kernel function became smoother. The performance improvement with decreasingly smooth kernel functions was believed to be caused by the quality of the available data across the majority of the input space.

All models in this chapter use an exponential kernel unless specified. The lowest NLML, across the complete dataset, was for the Matérn-3/2 kernel, but predictive performance was consistently weaker than with an exponential kernel. Limitations of the exponential kernel relative to a radial basis function kernel are explored in Section 4.3.

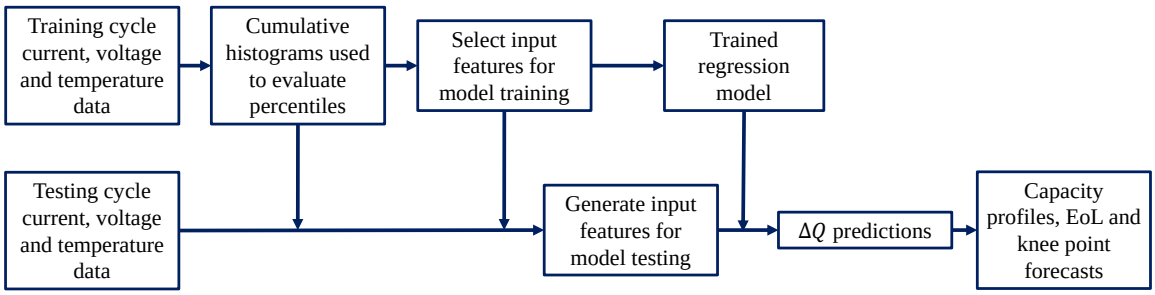


Fig. 2.8 Flowchart summarising the full capacity model proposed in this chapter.

The test set was then required for validating the prediction results. The required input features were calculated based on the forecasted or known use, with thresholds determined by the training set. The test cell use data was assumed known in this thesis, but that assumption is weak more generally.

The test input features were used to calculate ΔQ forecasts using the trained model, and these were then summed to produce capacity profiles through time. The full data pipeline is shown in Fig. 2.8 for clarity.

2.4 Model evaluation

The target variable for the GPR model was ΔQ. Root mean square error of mean ΔQ predictions (RMSE ΔQ, equation 2.4) was used to assess the performance of the model fitting. The capacity model was a function of the sum of the forecasted ΔQ values and was assessed by calculating the root mean square error of mean capacity (RMSE Capacity, equation 2.5). Both RMSE ΔQ and RMSE Capacity have units of

% capacity and are depicted in Fig. 2.9a.

$$\text{RMSE } \Delta Q = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\Delta Q_j^* - \Delta\hat{Q}_j\right)^2} \tag{2.4}$$

$$\text{RMSE Capacity} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(Q_j^* - \hat{Q}_j\right)^2} \tag{2.5}$$

where Q_j^* and ΔQ_j^* are the predicted capacities and changes in capacity at time interval j, and $\hat{Q}_j$ and $\Delta\hat{Q}_j$ are the equivalent measured capacities and changes in capacity.

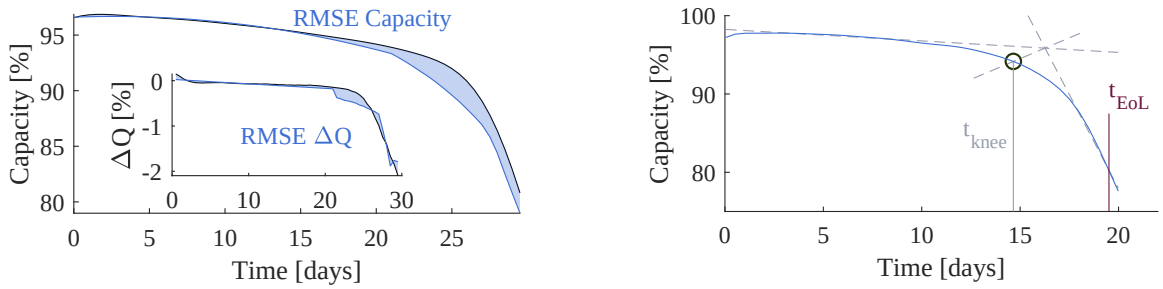
Remaining useful life (RUL) prediction accuracy is one of the most common metrics by which battery health prognosis algorithms are usually compared (see Section 1.2.2). Here, end-of-life time (t_{EoL}) accuracy was used instead because the approach forecasted the full lifetime, not from some single point during cell life[2]. In the majority of cases, the absolute values of EoL error as a percentage of the observed lifetime were used,

$$\text{EoL Error } = \left|100\% \times \left(\frac{t_{\text{EoL}}^*}{\hat{t}_{\text{EoL}}} - 1\right)\right| \tag{2.6}$$

where t_{EoL}^* and $\hat{t}_{\text{EoL}}$ are the predicted and measured EoL times.

The EoL condition was defined for Severson-2019 and Attia-2020 as 80% of the nominal capacity [22, 107], marked in Fig. 2.9b. In some cases the time intervals of $\Delta t = 12$ hours resulted in profiles truncating prior to the 80% condition so the final measured capacity point is taken as the estimate of $\hat{t}_{\text{EoL}}$. A spline through the capacity profile was used to estimate t_{EoL} if data was available past 80% capacity, if not then a

[2] t_{EoL} is the RUL from $t = 0$



(a) Capacity profile performance metrics.

(b) Single point performance metrics.

Fig. 2.9 Performance metrics used for predictions in this work. (a) Profile-focussed metrics RMSE Capacity and RMSE ΔQ were calculated from the difference between observed and predictive profiles. (b) Point-focussed metrics EoL error and knee error were calculated based on error between observed and predicted properties of the profiles.

linear extrapolation of the final 5 data points was used until the profile passed the 80% capacity condition.

In addition to EoL, the time of the knee point t_{knee} is an alternative point-focussed performance metric. All the known identification methods require accurate knowledge of a full capacity profile, including the bisector method shown in Fig. 2.9b. The work here used the method in Fig. 2.9b to calculate knee points from capacity profiles. The calculation starts by bisecting linear extrapolations of early and late life, taken as the first half and last five capacity points respectively, and finding the coordinates where that bisector meets the capacity curve. Other techniques are available for estimating knee points on capacity curves, but none have been found to be advantageous [5]. The error for the knee point calculation was equivalent to that in equation 2.6.

The GPR model produces probabilistic predictions. Methods for assessing the accuracy of the credible intervals are discussed in Chapter 4.

The RMSE ΔQ, RMSE Capacity, EoL error and knee error were calculated for every predicted profile in all prognosis trials in this chapter. Several repeats with at

least 57 test cells were used to produce a large number of performance values. For example, the main trial below used 100 repeats with 100 randomly selected training cells and 57 test cells each to produce 5,700 values of each metric. From these results the medians and 95$^{\text{th}}$ percentiles were drawn as estimates of performance of a typical and a poor forecast, respectively. Percentiles were preferred to mean averages because the performance metric distributions were asymmetric in all cases.

The feature generation process was computationally intensive. The only other trial in this chapter that generated features based on the specific training sets was that in Fig. 2.3. The mean time taken per repeat was shown to be >10 minutes on a standard desktop computer when using three bins. Only 10 repeats were used to make that trial manageable. Increased computing power or sophisticated coding techniques could be used to speed up the feature generation process[3].

Other results, like those shown in Figs. 2.4, 2.6, 2.7 and Table 2.3, used a feature set that had been calculated previously. Figs. 2.6 and 2.7 were generated using the values from Table 2.1. The automated feature selection process in Section 2.2 was investigated in isolation. That trial used 10,000 randomly selected training combinations in under 43 seconds because GPR modelling and feature generation were omitted[4].

Three tests of the robustness of the input features were performed. The first looked at how resilient the feature generation process was to noisy raw data. The independent variable was σ_R, the standard deviation of noise that was added to the raw data prior

[3]For example, the feature generation process is parallelisable.

[4]The full trial took > 4 minutes, most of which was taken up by forming randomly selected training sets.

to feature generation. It was a dimensionless value such that the noise had a standard deviation of the range of the data times σ_R. Consider voltage as an example, with voltage profile $V(t)$, the altered raw data was calculated according to equation .

$$V(t) = V(t) + \mathcal{N}(\mathbf{0}, \sigma_R \times (\max(V) - \min(V))) \tag{2.7}$$

The value of σ_R ranged from 10^{-3} to 10^{1}.

The second robustness test looked at the impact of sampling the raw data at different rates. The test variable was the sample gap, the minimum time between two subsequent raw data points. The raw values from the full dataset were extracted to produce a reduced raw data array.

The last robustness trial aimed to weaken the raw data by increasing the quantization of the raw data. The current and voltage data from Severson-2019 and Attia-2020 were sensitive to 5 significant figures. As a comparison, this trial allowed between 2 and 200 evenly spaced raw data values for each variable[5]. The original raw data was rounded to the nearest of the values prior to the feature generation process.

All three trials performed the feature generation process on all 157 cells in a single batch for speed. The features produced at each test point were stored and the train/test combinations were held constant across a trial.

[5]For example, 5 data points for voltage used $V = 2.70, 2.93, 3.15, 3.38, 3.60$ V

2.5 Results

2.5.1 Input feature generation

The automated feature generation algorithm produced consistent thresholds. The thresholds in Fig. 2.10 are from the main trial in this chapter. That trial had 100 repeats, and most columns for voltage and current in Fig. 2.10 represent a value of exactly 100.

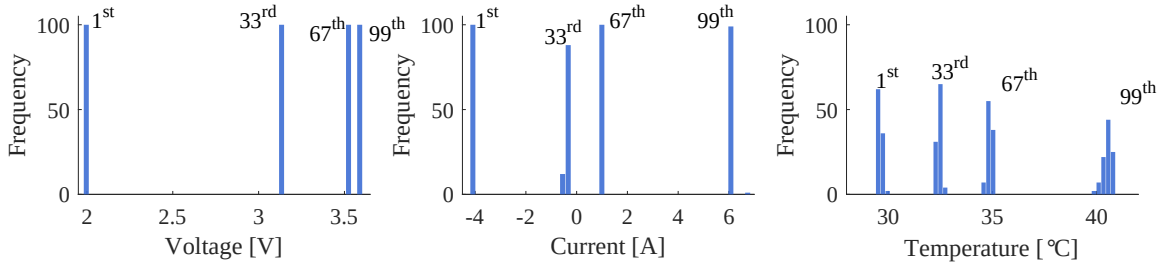


Fig. 2.10 Histograms of the locations of the automatically calculated input variable thresholds. The percentiles are marked on the plots.

The histograms were calculated by using 50 bins to cover the range of each variable. These distributions are therefore very tight. Only temperature demonstrated any distinguishable variability.

2.5.2 Input feature selection

Feature $V_{2,3}$, the proportion of time spent between approximately $V = 3.12$ V and $V = 3.51$ V, was almost universally selected first in all trials. The exceptions were at the low-resolution limits of the robustness trials, although in those cases selection still tended to be dominated by voltage-based features. In the investigation exclusively

looking at feature selection in Table 2.4, $V_{2,3}$ was the first selected feature in all 10,000 cases. The full results are in Appendix B.

	Selection number				
Feature	1	2	3	4	5
$V_{2,3}$	100				
$V_{1,2}$		71			
time		29	71		
$\|P\|_{1,2}$			29	69	
$\|I\|_{1,2}$			1	10	11
$I_{1,3}$				13	41
$I_{3,4}$				5	10
$\|P\|_{2,4}$				2	14
$V_{1,4}$				1	8

Table 2.4 Frequency of selection for the most common features in units of %. The trial contained 10,000 repeats.

As shown in Fig. 2.11, the input features in Table 2.4 were not those with the highest correlations with ΔQ. The best correlating input features were all voltage based. $V_{1,4}$ was the only feature with $\rho_P < 0.70$ when compared with ΔQ. However $V_{2,3}$ was sufficiently better correlated with the target variable, and most other voltage based features shared too high a correlation with it, that in almost all cases $V_{2,3}$ was selected first and there was maximum of one further voltage-based feature. Even then, $V_{1,2}$ was subsequently used in only 71% of cases.

Fig. 2.11 is a representative view of what the algorithm in Section 2.2 saw. It swept from top to bottom, using the values in the left-most column to select features before using the rest of the selected row to remove features. The values in dark-red showed why the performance improvement in Fig. 2.6 began at around $\rho_{P,\max} \approx 0.9$, since the excessive number of voltage-based features were removed.

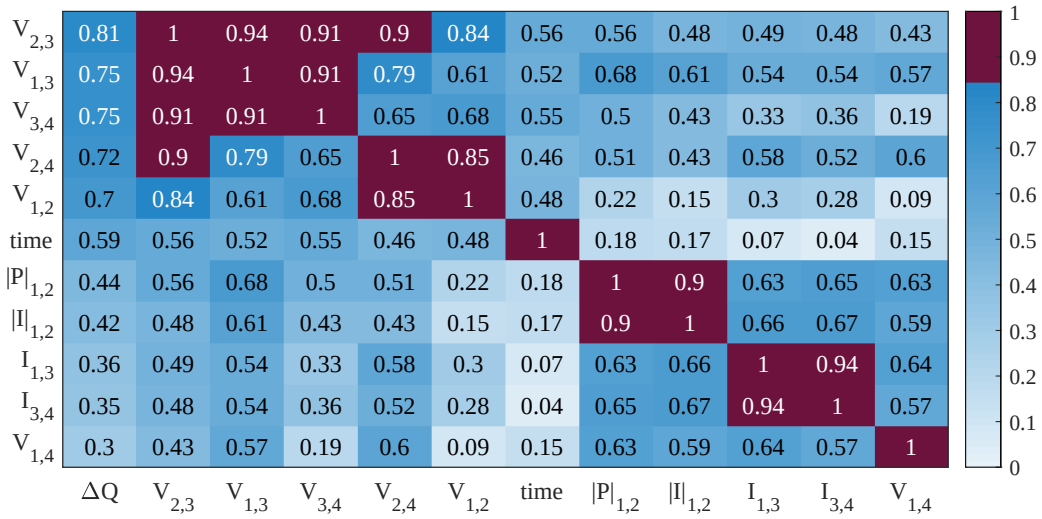


Fig. 2.11 $|\rho_P|$ for the most commonly selected features and those with the strongest correlations. The algorithm moved from top to bottom, recursively selecting the best correlation in the left-most column then removing any features in dark red in that row. All values are Pearson's correlation coefficients.

2.5.3 Hyperparameters

As mentioned in Section 1.2.2, the trained hyperparameters of a GPR model offer insight into the relevance of different inputs. An example GPR model using an exponential kernel and inputs $V_{2,3}$, $V_{1,2}$, experimental time, $|P|_{1,2}$ and I_{1_3} returned lengthscale hyperparameters of 7.78, 2.84, 346, 2.65 and 3.47, respectively. The best correlating input, $V_{2,3}$, was found to have the longest lengthscale hyperparameter among the inputs used for this model, excluding time. This result was found in a number of trials, using both exponential and radial basis function kernels.

2.5.4 Capacity forecasting

The RMSE ΔQ appeared to be small across the 5,700 predicted profiles in Fig. 2.12 with the 95$^{\text{th}}$ percentile of RMSE ΔQ at 0.26% capacity. A tight fitting for ΔQ led to

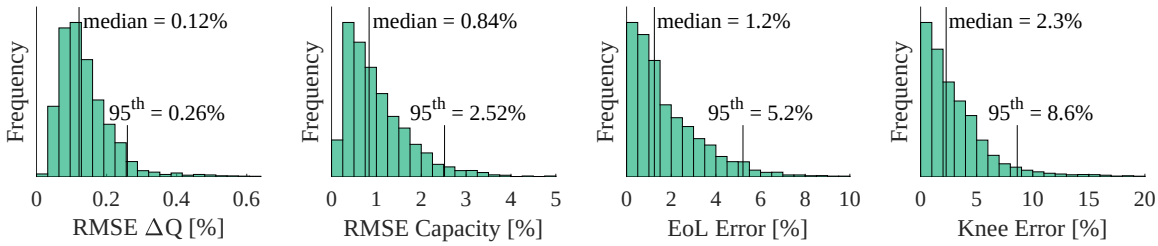


Fig. 2.12 Predictive performance of feature engineering approach with GPR model. Scores for each profile were combined to form histograms and calculate median and 95$^{\text{th}}$ percentiles of performance.

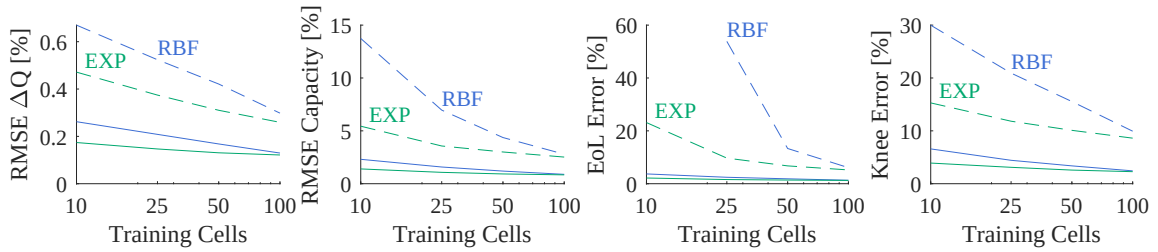


Fig. 2.13 Performance of the automated feature generation and selection combined with GPR model as a function of the number of training cells.

a median RMSE Capacity of only 0.84% capacity, with the 95$^{\text{th}}$ percentile at 2.52% capacity. The median value of 0.84% is comparable to models in the literature, but was higher than some SoH estimation methods in Table 1.1. The mean RMSE Capacity here was 1.06% capacity.

The accuracy of EoL and knee forecasting was notable, especially at the 95$^{\text{th}}$ percentile. Values of 5.2% and 8.6% error respectively are very low, and are similar to the mean accuracy of several models in literature in Table 1.1.

The performance in Fig. 2.12 was achieved with the subset of 157 cells within a limited range of lifetimes, contributing to the good predictive performance. However the median and 95$^{\text{th}}$ percentiles of EoL error only increase to 1.5% and 9.0% when

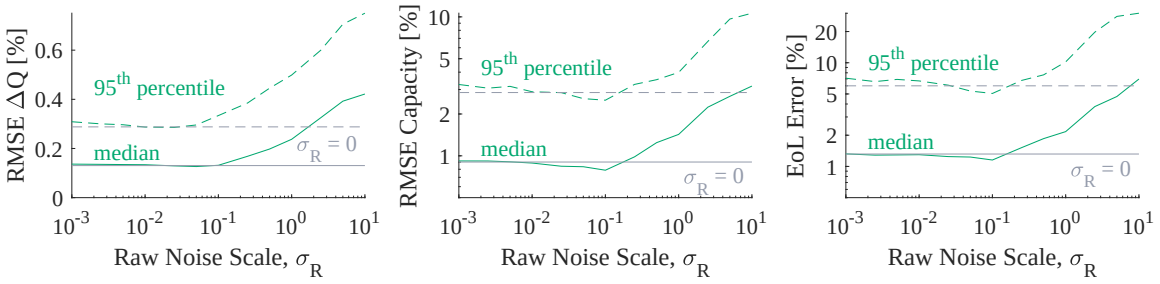


Fig. 2.14 Feature engineering as a function of size of noise added to raw cycle data.

including all 175 available cells from Severson-2019 and Attia-2020, shown in Appendix B.

All performance metrics were a function of the number of cells used to train the model. Fig. 2.13 shows that the full procedure is more resilient when the GPR model uses an exponential kernel. Too high a percentage (6%) of predicted profiles based on 10 training cells had an indiscernible EoL so there was no value recorded in Fig. 2.13 for the RBF kernel.

The robustness trials demonstrated that the technique was able to perform well with significantly reduced data qualities. Adding noise to the raw data of size up to $\sigma_R = 1/10$ maintained the optimum performance for all metrics. The sampling frequency trial found that intervals of 180 seconds between cycle data samples produced optimal predictive performance. Poor performance was found at $\sigma_R > 1$ and a sample interval of over 10 minutes.

There was very little impact on performance shown when varying the quantization of the raw data. Fig. 2.16 shows a consistent median and slowly changing 95^{th} percentile for all performance metrics used.

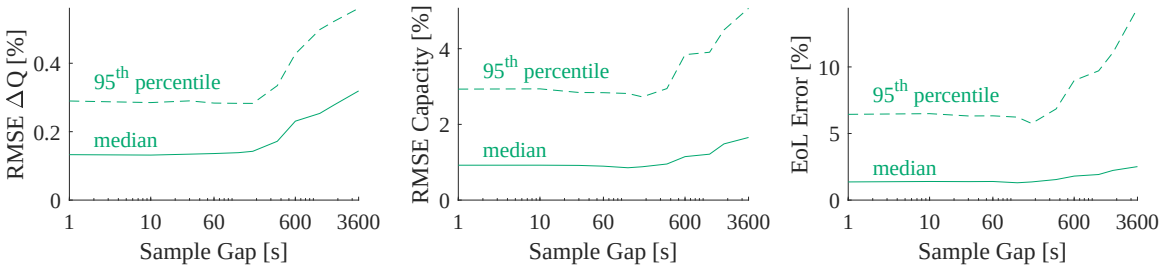


Fig. 2.15 Modelling performance as function of the sampling frequency. Raw data was extracted at regular intervals from the high-frequency raw cycle data to produce the sample gaps.

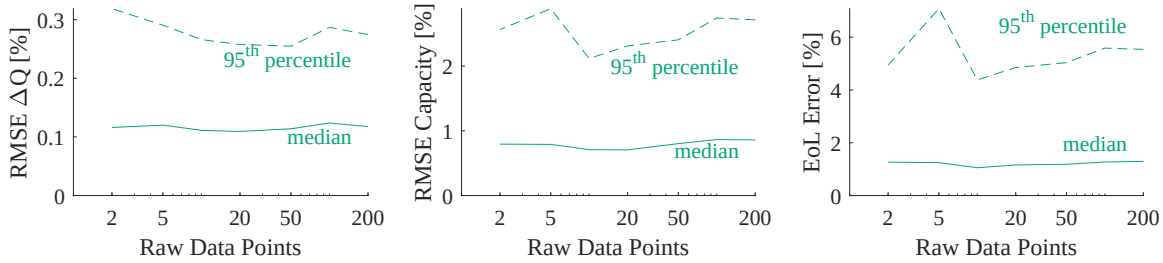


Fig. 2.16 Modelling performance as a function of raw data quantization. The raw data was rounded to fixed values of each variable, the number of possible raw data points forms the x-axes above.

The results thus far consider all predicted profiles and all batteries. Fig. 2.17 shows that performance is not equivalent from one cell to another. There was also a bias, i.e. an underestimation of the times of EoL and knee points among the weaker cells.

2.6 Discussion

The feature generation process produced consistent thresholds in Fig. 2.10. Most trials in this thesis used the thresholds in Table 2.1 in order to reduce the computational cost of each trial, but the consistency of the thresholds suggests that automated feature generation could have been applied more widely without impacting predictive performance.

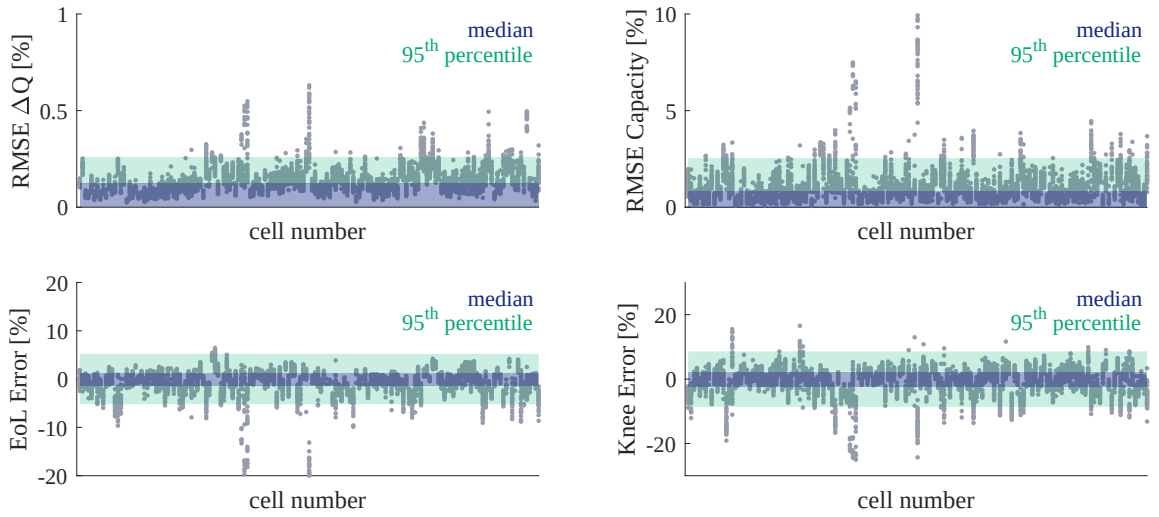


Fig. 2.17 Cell specific performance for all performance metrics. Median and 95$^{\text{th}}$ percentile regions are shaded in blue and green respectively. Each x-axis contains cell numbers from 1 to 157.

The feature selection procedure improved forecasting performance by ensuring a diverse input. However, the resultant input feature arrays included some features which correlated poorly with the ΔQ ($\rho_P \approx 0.4$). These poorly correlated inputs failed to improve predictive performance from 3 features up to 10 in Fig. 2.7. Adding different inputs, i.e. those not drawn from percentiles, might improve this result but one would have to be wary of overfitting.

The most commonly selected feature, $V_{2,3}$, represented the proportion of time spent in the mid-voltage range, typically between 3.12 V and 3.51 V. It was found to correlate negatively with ageing rate suggesting that the less time spent in this mid-voltage region, the faster the degradation. That relationship was presumed to be due to less time in the mid-voltage range being directly caused by more time spent at higher and lower voltages, both known to significantly contribute to degradation [15, 31]. For instance, the growth of solid electrolyte interphase is known to be enhanced at higher

voltages [15, 217], whereas low state of charge has been linked to both lithium plating and solid electrolyte interphase growth [15, 218].

Temperature is known to be related to degradation but was rarely selected as a feature. In the trial with 10,000 selections (Appendix B.3), a temperature-based feature appeared only 41 times and never before the 9^{th} priority selection. There were experimental issues with temperature in Severson-2019 but the controlled conditions may be the reason for temperature's omission. This may not be the case in less controlled environments where cells experience different use and wider varying temperature conditions.

The trained hyperparameters produced the surprising relationship that the best correlating input, $V_{2,3}$, was fit using a longer lengthscale than inputs with significantly weaker correlations. The results in Fig. 2.7 demonstrated that the GPR model performed well with only 3 input features, therefore the first selected input features are providing high quality input data. Instead, the hyperparameter values imply that there is valuable information in the 4^{th} and 5^{th} selected inputs that the GPR model is capable of using, despite how uninformative the less-correlated scatter plots in Fig. 2.5 appeared.

The overall model produced consistently accurate capacity forecasts. Literature models with better RMSE Capacity performance were SoH estimation approaches which included capacity-based input features. End-of-life accuracy of the automated approach represents a significant improvement on the literature, with 95^{th} percentiles of performance lower than the mean performance in literature. Even with all 175 cells,

the 95^{th} percentiles were only 20% higher than the means in literature. The literature models are based on early cycle data so do not take advantage of the available data through life, instead assuming a consistent use protocol.

The robustness to poor data, demonstrated in Figs. 2.14, 2.15 and 2.16, was a consequence of the simplicity of the input features. The model was slow to train at around 10 minutes per model with 3 bins, but the benefit was that the approach was robust to noise and low frequency sampling. These results suggest that the model could be applied where measuring equipment is not as high quality. The results in Fig. 2.15 suggest that performance can be slightly improved with reduced sampling rate from the cycle data, which is a result that has been seen in literature previously [169].

The lack of negative impact on the predictive performance from the quantization of the cycle data was caused by the rounding procedure being a similar procedure to dividing the data into bins. The trial was still included here because high precision observation is unlikely to be always available.

Knee point prediction accuracy was weaker than EoL prediction accuracy. The discrepancy is mostly a consequence of the technique used to calculate the position of the knee point. Small errors in the capacity curve, particularly in the later life gradients, contribute to bigger errors for the intersection point in Fig. 2.9b. Median knee point error of 2.3% still represents a very strong performance for modelling such a poorly understood phenomenon [22, 39].

Cell-specific performance is a significant concern. Percentiles produced consistent and insightful results but there was a sufficiently large spread of results that some cells

were effectively overlooked. For example, of the 157 cells used, only 34 were found to have an RMSE Capacity above and below the 95$^{\text{th}}$ percentile of 2.52% capacity. A further 25 cells were found to not overlap either performance metric for RMSE Capacity. The equivalent numbers for EoL error were 43 and 21 cells respectively.

Another concern was the bias to under-predict the time of end-of-life. While knowledge of this bias could be used to further inform a user of both the accuracy and confidence of a given forecast, it can only be used in cases with many test cells. Knowledge of the bias would be less insightful for users with small numbers of test cases.

The intuitive relationship that better data results in better performance was found in this chapter. However, concern remains that *better* performance could mean better performance for a specific subset of cells and not a general improvement, especially in cases where the median was consistent but 95$^{\text{th}}$ percentiles improved as a function of data quality.

2.7 Conclusions

The chapter presented an automated procedure that performed well for capacity and EoL forecasting. The input features produced were robust to significantly reduced data quality. The selection process was found to have a positive impact on predictive performance. The Gaussian process regression tool successfully mapped to capacity through entire cell lifetimes and produced small knee point errors.

Voltage-based features were consistently found to be the best correlated with degradation and only two were required for accurate forecasting. Time spent in given voltage ranges proved an effective input feature, and was simple to infer information from. However, the information provided by these inputs was lacking diagnostic detail in comparison to those used for capacity estimation in literature.

The Gaussian process regression model was proven to be flexible and effective, especially with an exponential kernel. Compared to the lifetime of these cells, or any real-world cells, the training time was negligible, but remained significant to a user employing a standard desktop computer. The resultant model was a function of the data and hence was difficult to infer from. As an improvement, Chapter 3 presents a different, faster and more transparent approach to modelling health.

This chapter has avoided discussion of the credible intervals that were produced by the GPR model. Credible intervals acknowledge the lack of certainty inherent in data-driven modelling and are one of the benefits of Bayesian approaches [13]. Uncertainty is such a big topic that the final chapter is focussed entirely on it. Chapter 4 investigates the quantifying of cell-to-cell variability and uncertainty, both for empirical and data-driven modelling.

Chapter 3

Piecewise linear regression for battery health modelling

This chapter presents piecewise linear regression (PLR) modelling for lithium-ion battery health prognosis, an approach that aims to be faster and more transparent than the Gaussian process regression (GPR) approach explored in chapter 2.

Section 3.1 introduces the concepts of PLR and why it offers promise. Section 3.2 details the PLR model and procedure used here, followed by a derivation of Bayesian linear regression, the modelling approach used in each sub-model. Sections 3.4 and 3.5 explain how the model was tested and present the results. Finally, the advantages and limitations of the PLR model are discussed in the final two sections, 3.6 and 3.7.

3.1 Introduction

The automated feature selection procedure described in Section 2.2 searched for the strongest linear relationships between ΔQ and the input features. The selection

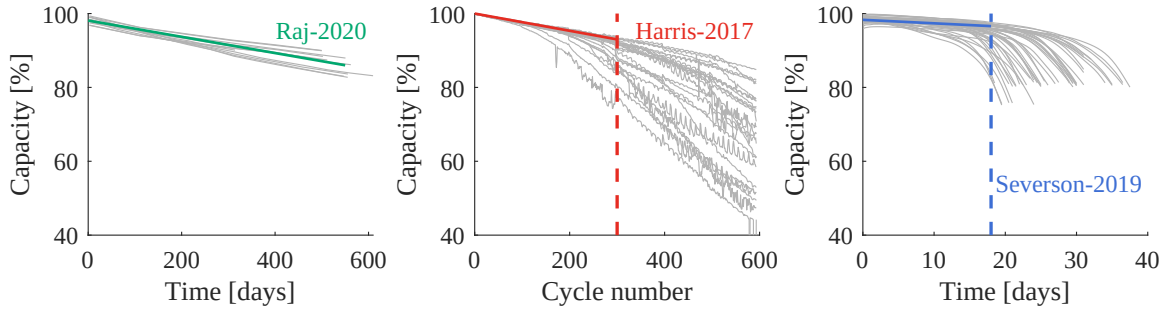


Fig. 3.1 Raj-2020, Harris-2017 and Severson-2019 datasets with example linear fits. Dashed lines indicate the end of consistently linear ageing.

procedure assumes that, loosely linear correlations exist for the Pearson's rank approach to work.

That assumption is supported by existing battery literature. There are known linear relationships between use conditions and degradation, typically found in SoH estimation works. Incremental capacity analysis features, such as peaks and integrals, have been found to have linear relationships with capacity loss [134, 212, 219–223]. Temperature profile properties, such as functions of environmental temperature and the temperature response to charging conditions, have demonstrated linear relationships with capacity [24, 224]. Voltage-based properties have presented linear relationships with capacity [159, 212, 225] alongside more complex observables such as internal stresses [220, 226] and impedance combinations [227].

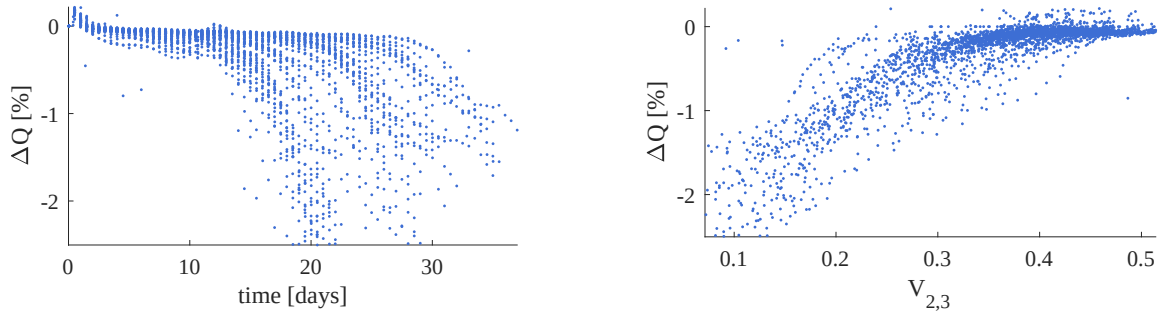
The above linear relationships were found in specific use cases and some were not maintained over full cell lifetimes. Piecewise modelling approaches aim to be more adaptable by splitting complex relationships into sub-sections, and have been used in literature to map a number of relationships. For example, relationships between current and voltage [228], open circuit voltage and state of charge [229], electrochemical

properties and voltage [230], and time varying model parameters [231] have all been modelled in a piecewise fashion. Piecewise linear approaches have been used as part of SoH models too, for instance by splitting lifetime into sections by cycle count to create changing models [113, 232, 233]. Also, a non-linear degradation response to depth of discharge has been approximated with a series of linear models [234].

All PLR battery health models require a decision over how to split the input(s) into distinct regions. Time is the most intuitive input variable to consider and has been used in literature [231, 232, 234] but only the Raj-2020 dataset in Fig. 3.1 appears suited to a split in time. Ageing variability creates a significant challenge in any dataset with changing degradation rates as a function of time, like Harris-2017 and Severson-2019.

There are regions in later life for Harris-2017 and Severson-2019 where cells of the same age were at very different stages of life, and Fig. 3.2a shows how that is accentuated when mapping to ΔQ rather than Q. A piecewise model requires a variable that has a more consistent relationship with degradation.

The strongest variables to calculate how and where to split were assumed to be the ones that correlated best with ΔQ. That assumption also allowed for the feature selection approach of the previous chapter to be retained. The input feature that was selected first, typically $V_{2,3}$ (the proportion of time spent with a voltage between 3.12 V and 3.51 V, see previous chapter), was proposed as the variable to use to calculate how to split the model into sub-regions. Fig. 3.2b suggested an encouragingly tight relationship between $V_{2,3}$ and ΔQ, even with a reasonable amount of noise in later life.



(a) ΔQ as a function of time.

(b) ΔQ as a function of $V_{2,3}$.

Fig. 3.2 Possible variables for splitting a non-linear relationship into piecewise linear sub-regions. (a) The spread of points resulting from using time as the splitting variable. (b) The strongest correlating feature, $V_{2,3}$ demonstrates a far better function for mapping to ΔQ.

For speed, the input features in this chapter were those calculated previously across the 157 selected cells from Severson-2019 and Attia-2020 [22, 107]. The variable bounds were recorded in Table 2.1 in Chapter 2. The automated feature selection step, as previously described, was found to take $\approx 1/250^{\text{th}}$ of a second and thus was not a concern. The rest of this chapter explains how the piecewise model was constructed based on the features previously selected, followed by a thorough validation test, almost always compared with equivalent performance from the GPR model discussed in Chapter 2. The GPR models used either an exponential or squared exponential kernel, denoted as GPR-EXP and GPR-RBF respectively.

3.2 Piecewise model construction

As mentioned in Section 3.1, the best correlated input feature with ΔQ was chosen as the splitting variable. The rest of the input features were selected using the method from Section 2.2, with the five input features retained as standard. Predictive performance

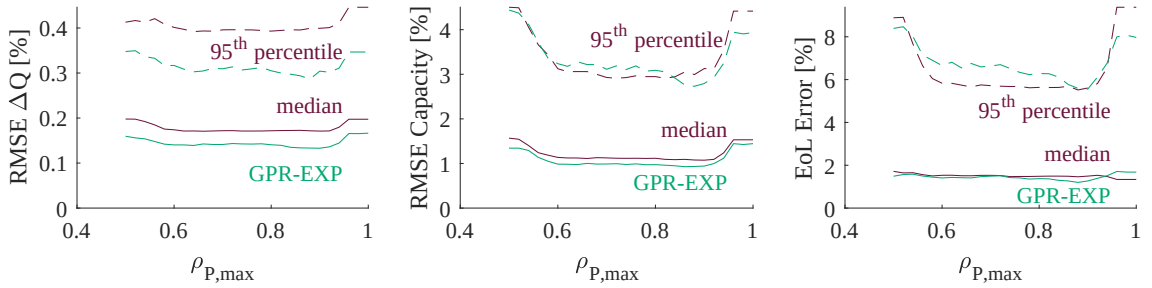


Fig. 3.3 Piecewise-linear (purple) performance as a function of the maximum similarity between input features, $\rho_{P,\max}$, compared with the performance of GPR-EXP (green).

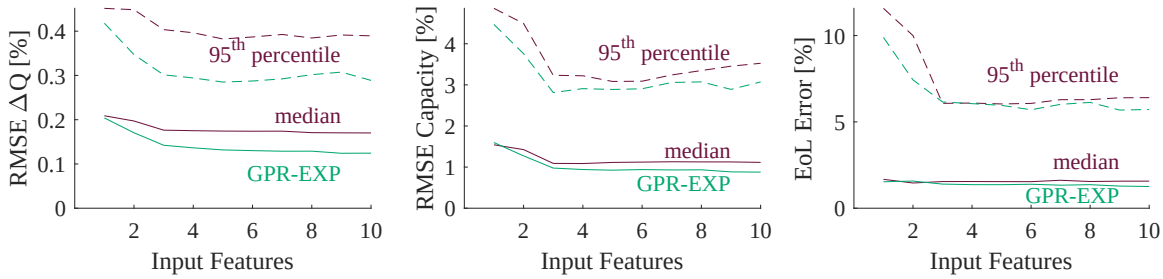


Fig. 3.4 Piecewise-linear performance as a function of input features compared with the performance of GPR-EXP.

was found to be similarly reliant on the maximum shared correlation between input features, $\rho_{P,\max}$, as discussed in Section 2.2. Fig. 3.3 shows that the peak performance of PLR modelling was more consistent than that using GPR-EXP. There was a slight performance advantage to GPR-EXP relative to PLR which is discussed in Section 3.6. Fig. 3.4 replicates the slight predictive performance advantage from GPR-EXP but the difference was small when performance was plotted as a function of the number of input features.

For ease of language, the first selected feature here will be referred to as the splitting feature or splitting variable, labelled x_s. The splitting variable was always $V_{2,3}$, the time spent in the mid-voltage range, in trials here, but that may not be true in general. Splitting was performed using the relationship between the splitting feature and ΔQ

for the training set, shown with a black line in Fig. 3.5a. The aim was to break that functionality into a series of approximately linear sections.

The black line, $f_{\Delta Q}(x_s)$, was a moving average produced using a Gaussian filter. The moving average was evaluated at points x_s^* based on N observations of $\mathbf{x_s}$ and ΔQ. Equation 3.1 shows how the moving average was calculated. The fractional lengthscale β_L needed to be specified by the user. As a default, $\beta_L = 0.1$ was used in all models.

$$f_{\Delta Q}(x_s^*) = \frac{\sum_{k=1}^{N} \Delta Q_k \exp\left(-\frac{(x_s^* - x_{s,k})^2}{\sigma_l(\mathbf{x_s})^2}\right)}{\sum_{k=1}^{N} \exp\left(-\frac{(x_s^* - x_{s,k})^2}{\sigma_l(\mathbf{x_s})^2}\right)} \tag{3.1}$$

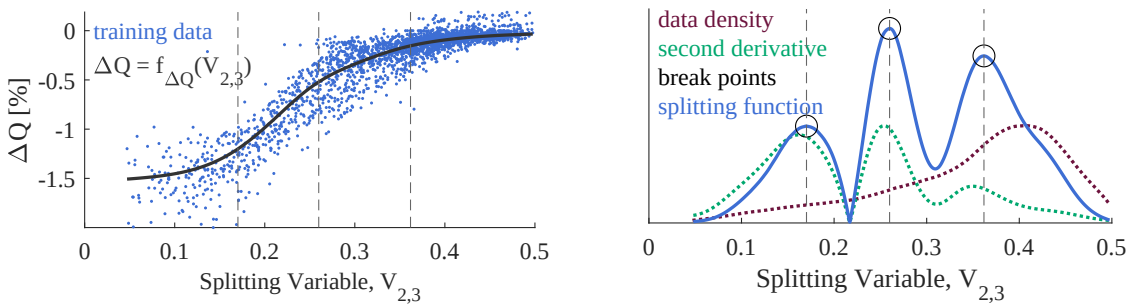
$$\sigma_l(\mathbf{x_s}) = (\max(\mathbf{x_s}) - \min(\mathbf{x_s})) \times \beta_L \tag{3.2}$$

The peaks of the absolute values of the second derivative of $f_{\Delta Q}(x_s)$ would be the ideal points to use, shown in green in Fig. 3.5b. However the smoothed function was noisier at the extreme values of x_s. The final function F_s (blue in Fig. 3.5b) used to split the input features was the absolute value of the second derivative $f_{\Delta Q}(x_s)$ multiplied by a measure of the density of the data, ρ_s, shown in purple in Fig. 3.5b, defined below:

$$F_s(x_s) = \left|\frac{d^2 f_{\Delta Q}}{dx_s^2}\right| \times \rho_s \tag{3.3}$$

$$\rho_s(x_s^*) = \sum_{k=1}^{N} H\left(|x_{s,k} - x_s^*| < \frac{1}{10}(\max(\mathbf{x_s}) - \min(\mathbf{x_s}))\right) \tag{3.4}$$

$$H(x) = \begin{cases} 1 \text{ if } x = \text{True} \\ \\ 0 \text{ if } x = \text{False} \end{cases}$$



(a) (b)

Fig. 3.5 Splitting the inputs using curvature as the principle criterion. (a) ΔQ was mapped as function of the splitting variable to produce a smooth function $f_{\Delta Q}(V_{2,3})$ using a moving average (black line). (b) The second derivative of $f_{\Delta Q}$ is multiplied by the density of the training data points. The peaks of the absolute value of the resultant splitting function were used as the breakpoints.

The density function, $\rho_s(x_s)$, counted the number of data points within a set distance of x_s. Its use in equation 3.3 prioritised changing gradients where there are many data points, i.e. the breakpoints selected earliest should have been the most important.

The approach using F_s was referred to as the *curvature* method because of its dependence on the second derivative. This is distinct from the property of a univariate function known as curvature, which depends on the first and second derivative in a more complex way [235].

A PLR approach with n_m sub-models requires the following steps. First, the $n_m - 1$ highest peaks of F_s are selected as the positions of the breakpoints, then second a linear model is trained on all the input feature data within each subset. For clarity, the individual linear models are referred to as sub-models to distinguish them from the overarching PLR capacity model. Fig. 3.6 demonstrates an example split where the sub-models represents distinct stages of the degradation.

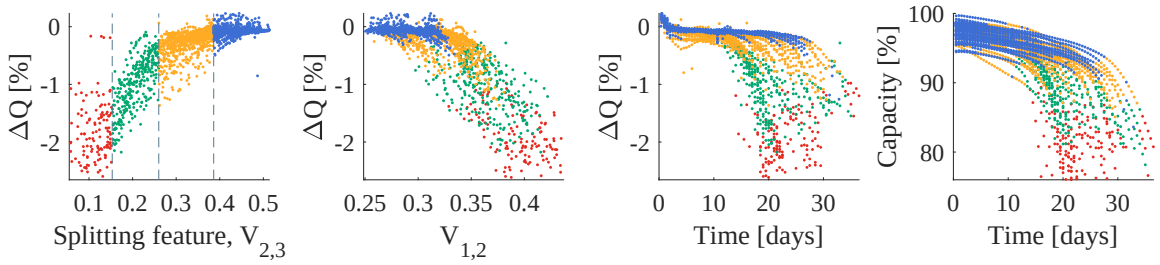


Fig. 3.6 Example split of training data for a PLR model with 4 sub-models. Each colour indicates an individual sub-model.

Other techniques can be used to split the data for a piecewise model. For example, K-means is a clustering algorithm that groups the input data into clusters of similar points [110]. Equally, one could use an optimization algorithm that minimizes a loss function to freely select the optimal number of breakpoints in x_s-space. One trial below investigates the relative performance of these two methods compared with the curvature approach, as well as against a randomly and an evenly spaced selection of breakpoints.

Piecewise models in literature either use a fixed n_m or sequentially add sub-models until a target performance or improvement is met. Early trials using PLR for capacity modelling required a 1% improvement in RMSE ΔQ for each subsequent sub-model size to accept the more complex model and avoid overfitting. But time taken to produce the PLR model was never restrictive because of how fast the linear sub-models trained.

However, later it was decided that instead of searching until a certain condition was met, all possible models were produced for a given training set, up to a maximum of 10 sub-models. The training data in Fig. 3.5a was found to have up to $n_m = 4$ because there are only 3 peaks in F_s in Fig. 3.5b. The performance of each model was measured by the RMSE ΔQ on the training set during model construction. The

n_m	RMSE ΔQ [%]	$\leq 1 + \beta_{\text{improv}}$	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		

Table 3.1 Piecewise model selection by selecting the smallest n_m within β_{improv} of the peak performance.

required model size was the one with the smallest n_m while being within $1 + \beta_{\text{improv}}$ of the optimum performance, where β_{improv} is some fraction of peak performance typically taken as 0.01. There is a study below on the impact of varying β_{improv} on predictive performance.

Table 3.1 demonstrates the model size calculation based on an example training set of 50 cells from Severson-2019 and Attia-2020. The optimum performance was $n_m = 6$, but $n_m = 4$ was sufficiently close (in this case, within a rounding error) that four sub-models were selected.

The final model was formed of the breakpoints and the parameters fitted using Bayesian linear regression (BLR). A derivation is given in Section 3.3. Each sub-model is linear, i.e. each input feature has a single coefficient associated with it.

3.3 Bayesian linear regression

Bayesian linear regression is similar to linear regression, but uses a Bayesian framework for the fitting of parameters. Like GPR, the likelihood is assumed to be Gaussian, shown in equation 3.7. The BLR model is a linear combination of the inputs (equation 3.6). The input and output variables are defined identically to Section 1.5 for consistency. Let the input X be a matrix with N rows of vectors $\mathbf{x}^T$, each of size $1 \times D$. The target variable $\mathbf{y}$ is then a $N \times 1$ vector.

$$y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}\left(0, \sigma_n^2\right) \tag{3.5}$$

$$f(\mathbf{x}) = \sum_{k=1}^{D} x_k w_k = \mathbf{x}^T \mathbf{w} \tag{3.6}$$

$$p(\mathbf{y}|X, \mathbf{w}) = \mathcal{N}\left(X\mathbf{w}, \sigma_n^2\right) \tag{3.7}$$

with $\mathbf{w}$ being a column vector of coefficients. Let the prior over the parameters $\mathbf{w}$ be a zero mean Gaussian with covariance Σ_w. In all models here the parameters were assumed to be independent and identically distributed. As a result, Σ_w in equation 3.8 becomes $\Sigma_w = \sigma_w^2 I$. Based on parameter values in initial trials, the prior variance was set at $\sigma_w^2 = 10^2$ in models unless otherwise specified.

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{0}, \Sigma_w\right) = \mathcal{N}\left(\mathbf{0}, \sigma_w^2 I\right) \tag{3.8}$$

From Bayes' rule, the posterior is proportional to the prior multiplied by the likelihood. The resultant distribution over the parameters can be written as an explicit function

of the prior variance Σ_w, the observation noise σ_n and the training data X and y.

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}, X) \propto & p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w}) \\ = & \mathcal{N}\left(\frac{1}{\sigma_n^2} A^{-1} X^T \mathbf{y}, A^{-1}\right) \end{aligned} \tag{3.9}$$

where $A = \left(\frac{1}{\sigma_n^2} X^T X + \Sigma_w^{-1}\right)$.

The matrix A^{-1} for each sub-model can be easily stored because it is of size $D \times D$, compared to $N \times N$ in the equivalent expression for GPR. The mean estimate of the parameters $\hat{\mathbf{w}}$ can be expressed as,

$$\hat{\mathbf{w}} = \frac{1}{\sigma_n^2} A^{-1} X^T \mathbf{y} \tag{3.10}$$

This chapter only uses the predictive means, so only the coefficients $\hat{\mathbf{w}}$ needed to be stored. The full predictive posterior requires the posterior covariance A^{-1} and is shown in equation 3.11 for test inputs X_*.

$$p(\mathbf{f}_*(\mathbf{X}_*)|X_*, \mathbf{y}, X) = \mathcal{N}\left(X_* \hat{\mathbf{w}}, X_* A^{-1} X_*^T\right) \tag{3.11}$$

The full predictive distributions are briefly used and assessed in Section 4.3.

Control	Value
Training cells	50
Test cells	107
Repeats per test point	20
$\rho_{P,\max}$	0.85
Input features	5
$\max(n_m)$	10
β_{improv}	0.01
σ_w	10
β_L	0.1

Table 3.2 Default controls of the PLR model.

3.4 Modelling and testing

Sub-models included the 5 automatically selected input features and a bias term in the input array, i.e. $D = 6$. The target variable was $y = \Delta Q$. The forecasted capacity profiles were forecasted by summing the predicted changes in capacity, assuming knowledge of initial capacity. End-of-life (EoL) error and RMSE Capacity were the only profile performance metrics used to assess performance.

The main trial was a comparison against GPR-EXP and GPR-RBF (see previous chapter). It used 100 repeats with 50 training cells each time. The remaining 107 cells were used for testing so there were 10,700 test profiles from which to draw percentiles of performance metrics. All defaults for the PLR models are written in Table 3.2.

An equivalent trial was performed on the Sauer-2021 dataset. There were only 48 cells available so 32 training cells were selected. The target variable was $\frac{dQ}{dt}$ because the time between capacity points was inconsistent. Sporadic RPTs were used for the health measurements but there were typically ≈15 data points per cell. Consequently training

datasets were not large, $N \approx 500$, and training was fast. Each training/test split was trialled with GPR-RBF and PLR, and 1,000 repeats were used. The Sauer-2021 dataset was expected to be challenging because there are two changes in degradation behaviour demonstrated, an initial knee point and a later reduced degradation rate, shown in Fig. 1.7 in Chapter 1.

The structure of the model when using Severson-2019 and Attia-2020 was investigated. The number of sub-models and the position of the breakpoints were recorded for 1,000 repeats of the PLR model construction procedure with 50 training cells, taking around 15 minutes to run in total. The selected breakpoints and the number of sub-models were plotted as a histogram and a bar plot respectively.

There were four control parameters of the PLR modelling that required investigation. The model selection parameter β_{improv} was varied between 0 and 1. The maximum allowed number of sub-models, $\max(n_m)$, was tested at values of 1 to 10 sub-models. The prior standard deviation over the parameters σ_w was tested at several values between 10^{-2} and 10^{2} to assess the limits of overly confident and overly flat priors, respectively. Finally, the fractional lengthscale of the Gaussian moving average in equation 3.2, β_L, was varied between 10^{-3} and 10^{0}.

The final trial compares different splitting techniques with performance presented as a function of the number of training cells. The curvature technique described in section 3.2 was compared against using free selection of breakpoints, K-means, even spacing and random spacing. K-means was only applied across the first two automatically selected input features to reduce the impact of uneven ranges between input features,

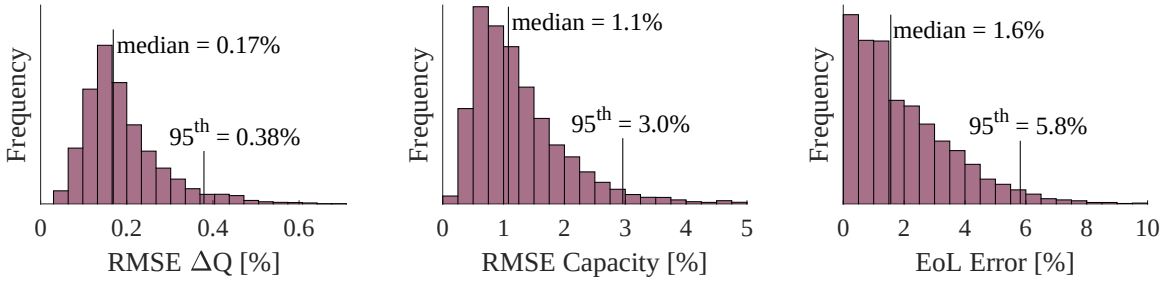


Fig. 3.7 Results of the main trial using piecewise linear regression. All performance metrics were assessed on each test cell and were collated to produce histograms and calculate percentiles of performance. Cell data was from Severson-2019 and Attia-2020.

but there are more sophisticated techniques available[1]. The loss function for free selection of breakpoints was RMSE ΔQ of the training set and Matlab's *fminsearch* was the optimizing tool. A GPR-EXP tool was also used. The mean time taken to produce a model was recorded to assess the speed benefit of PLR.

3.5 Results

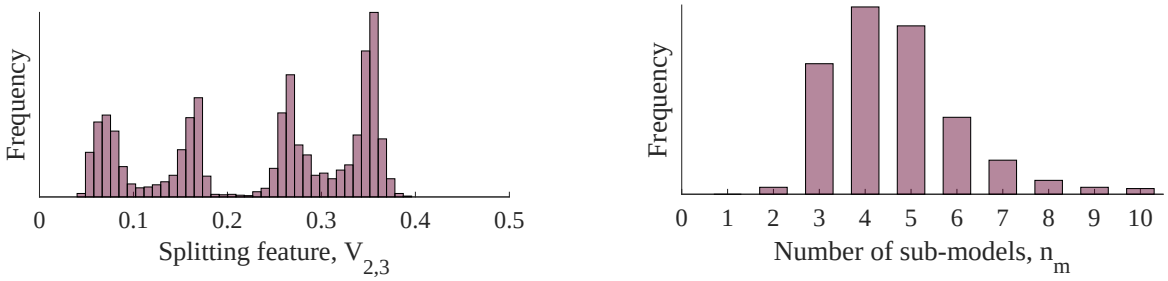
The main trial produced a very tight capacity fit according to the RMSE ΔQ and RMSE Capacity plots shown in Fig. 3.7. The median and 95$^{\text{th}}$ percentile for RMSE ΔQ were 0.17% and 0.38% capacity, and 1.1% and 3.0% capacity for RMSE Capacity and the results for EoL Error were with a median of 1.6%. The predictive performance was comparable to the GPR techniques from Chapter 2.

The 10,700 test profiles allows for a detailed examination of each metric. The best performing regression technique was GPR-EXP because it produced the lowest value in

[1]One method of note would be using the Mahalanobis distance [236]. The first two inputs were chosen as a simple and fast approach that would allow for a comparison with the curvature approach to splitting inputs. Any future use of the Mahalanobis distance would have to handle the complexities involving initialising covariance matrices [237].

Model	RMSE ΔQ [%] median	95^{th}	RMSE Capacity [%] median	95^{th}	EoL Error [%] median	95^{th}
PLR	0.169	0.379	1.08	2.96	1.56	5.82
GPR-EXP	0.133	0.302	0.95	3.00	1.46	6.49
GPR-RBF	0.166	0.412	1.18	4.36	1.76	9.81

Table 3.3 Comparison of full performance of PLR against GPR-EXP and GPR-RBF for each of RMSE ΔQ, RMSE Capacity, and EoL error. Percentiles for each metric were calculated from the 10,700 forecasted profiles for each modelling technique.



(a) Position of breakpoints.

(b) Number of sub-models used.

Fig. 3.8 Results of trial investigating the construction of the model. (a) The positions of all breakpoints plotted on a histogram. (b) The number of sub-models required plotted as a bar chart. Max(n_m) was set at 10.

four of the 6 columns in Table 3.3. PLR appeared to be weaker at the highest levels of performance, especially for RMSE ΔQ, but performed very well at the 95^{th} percentile and was almost equivalent to GPR-EXP for EoL error. RMSE Capacity showed PLR moving from the weakest regression tool at the lowest percentiles to the best at the highest percentiles.

The results in Fig. 3.8 provide insight. The breakpoint positions in $V_{2,3}$ were found in 4 distinct peaks in Fig. 3.8a. Of those, the biggest peak was at $V_{2,3} = 0.356$ above which approximately 60% of the data points were found. The $V_{2,3} \approx 0.356$ peak appeared to represent the end of the linear degradation in Fig. 3.5, akin to a knee-onset

Model	RMSE $\frac{dQ}{dt}$ [%.day^{-1}] median	95$^{\text{th}}$	RMSE Capacity [%] median	95$^{\text{th}}$	EoL Error [%] median	95$^{\text{th}}$
PLR	0.097	0.152	2.65	5.71	2.24	9.15
GPR-RBF	0.060	0.096	1.56	4.70	1.76	4.41

Table 3.4 Large trial with Sauer-2021 comparing PLR and GPR-RBF. 1,000 repeats with 32 training cells were used for this trial.

that has been described in literature [39]. Furthermore, of the 1,000 trials used for Figs. 3.8a and 3.8b, over half of models required 4 or fewer sub-models.

The performance of PLR relative to GPR was weaker on the Sauer-2021 dataset in Table 3.4. Median RMSE-Capacity was 2.7% for PLR and 1.6% for GPR-RBF. There was a smaller difference for EoL error, with 2.2% and 1.8% respectively. The resulting histograms plots are included in Appendix C.4.

Varying β_{improv} and $\max(n_m)$ produced consistently accurate predictions for RMSE Capacity and EoL error until reaching the respective limits of $\beta_{\text{improv}} \approx 0.5$ and $\max(n_m) = 2$. Table 3.1 provides an example model selection and demonstrates that setting $\beta_{\text{improv}} > 0.50$ is likely to be an equivalent condition to $\max(n_m) = 1$.

Changing the prior standard deviation yielded very little variability in predictive performance above $\sigma_w = 10^0$. The 95$^{\text{th}}$ percentile of both RMSE Capacity and EoL error reduced at $\sigma_w \approx 5 \times 10^{-2}$ but the median curves suggested that the performance was weakening because of the increasingly restrictive prior.

The final controllable parameter was β_L. Predictive performance was variable but peaked at $\beta_L \approx 10^{-1}$. The performance of both RMSE Capacity and EoL became

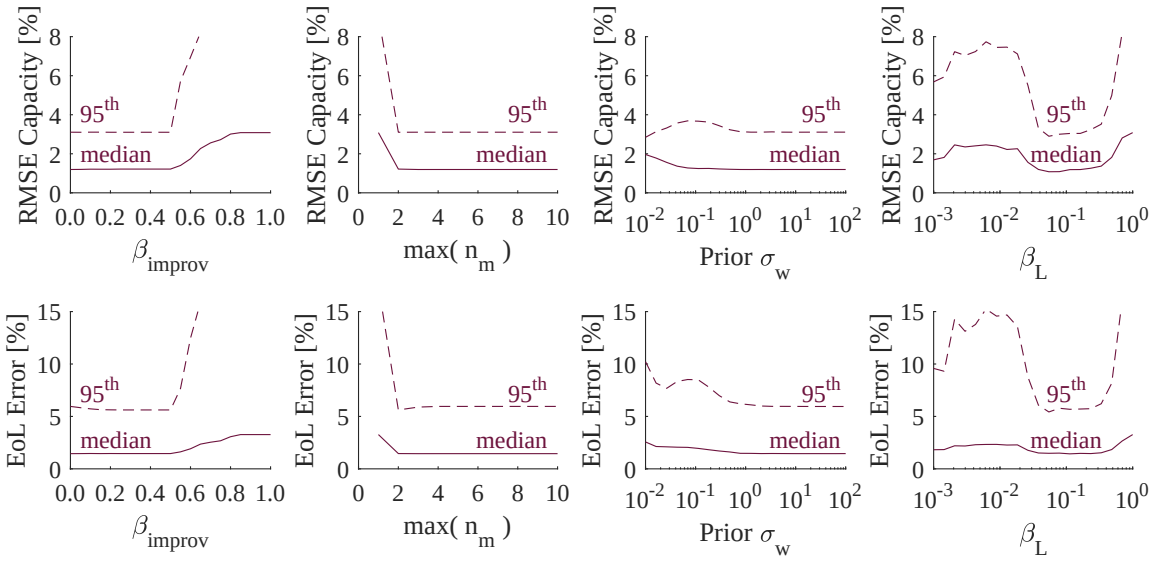


Fig. 3.9 PLR performance as a function of controls β_{improv}, $\max(n_m)$, σ_w, and β_L.

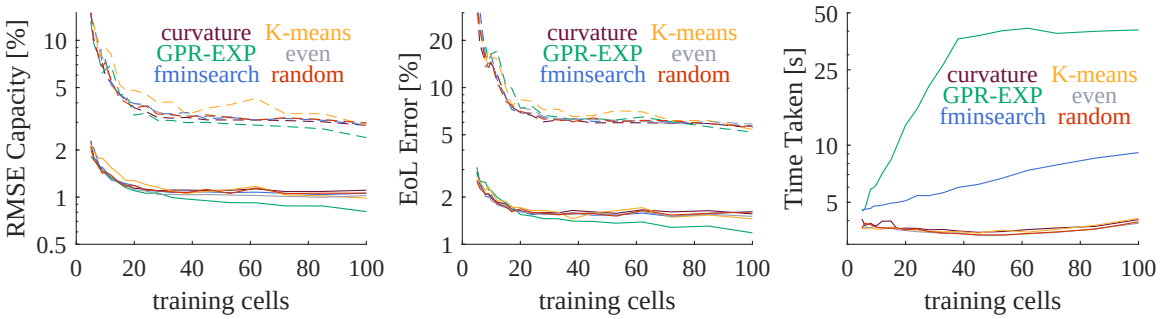


Fig. 3.10 Comparative performance of various splitting mechanisms as a function of the number of input cells. GPR-EXP (green) also included as a comparison.

weaker as β_L became smaller before trending back towards peak performance as β_L approached 10^{-3}.

All methods used in Fig. 3.10 produced rapidly improving RMSE Capacity and EoL error as a function of training cells until 20 training cells were available. From 20 training cells to 100, there was a slow improvement for the PLR approaches whereas GPR-EXP noticeably reduced metrics of RMSE Capacity and the median EoL error.

All of the splitting mechanisms produced very similar results for RMSE Capacity and EoL error in Fig. 3.10. The procedure presented in Section 3.2 did not outperform evenly or randomly spaced breakpoints, but that performance level was attained with fewer sub-models when using the curvature approach. For example, 90% of PLR models using evenly spaced breakpoints required 8 or more sub-models. The model size algorithm presented in Table 3.1 and setting $\max(n_m) = 10$ provided sufficient opportunities for these unintelligent methods to produce a good model of ΔQ which translated into the performance seen in Fig. 3.10. A similar effect caused the improved performance at $\beta_L \approx 10^{-3}$ where the moving average function, $f_{\Delta Q}$, became very noisy, thereby producing many possible breakpoints.

The free choice of breakpoints differentiated from the other PLR techniques for time taken because it scaled linearly with the number of training cells. The other methods required under 5 seconds per model at all training set sizes. The time taken per GPR-EXP model rapidly increased up to 40 seconds above which Matlab's *fitrgp* tool's default limit of 2,000 training points applied.

3.6 Discussion

The performance of PLR was weaker than GPR-EXP and GPR-RBF when measured by RMSE ΔQ, but equivalent or better when looking at the capacity profiles for Severson-2019 and Attia-2020. The linear models appeared less capable of mapping to the smaller variations in ΔQ that were captured by the Gaussian process approaches. However,

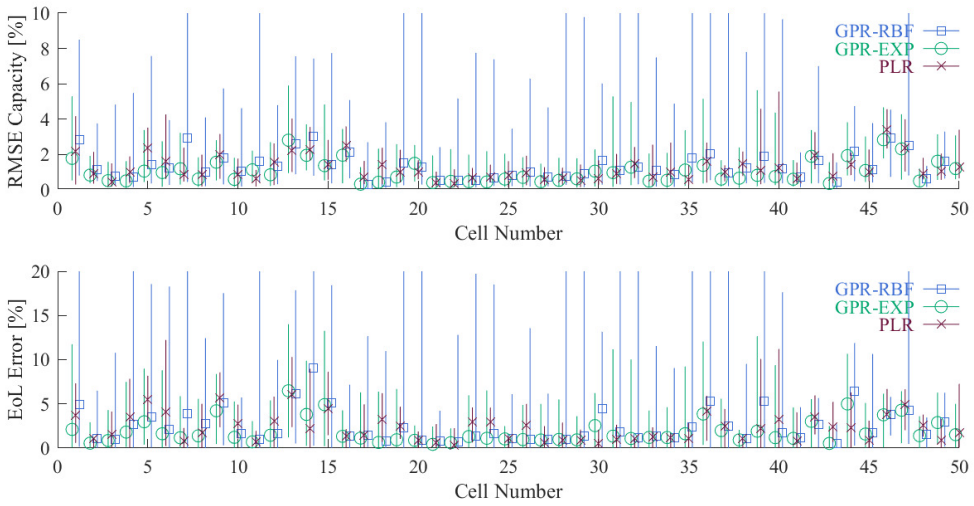


Fig. 3.11 Cell specific performance for test cells 1 to 50. PLR (claret, crosses) performance compared to GPR-EXP (green, circles) and GPR-RBF (blue, squares). Shapes show the median value for each cell, the lines a plotted from the minimum to the maximum of each performance metric for each cell.

that led to more consistent results when averaged over entire profiles, especially at the 95$^{\text{th}}$ percentile of performance in Table 3.3.

Unlike with Gaussian process regression, the PLR models were locally linear in their construction. Linear models appeared more accurate for cells with weaker fits in Fig. 3.11. There remains a concern that the performance of PLR modelling was a result of fitting being more consistent than the GPR equivalents when the fit is poor, rather than being an inherently better model.

The results were broken down according to the cell number in Fig. 3.11, and the median EoL error was found to be smaller for 72 of the 157 cells when using PLR relative to GPR with an exponential kernel. That number rose to 92 cells for the 95$^{\text{th}}$ percentile for each cell[2]. Both of these counts represent approximately half the

[2]The equivalent numbers were 50 and 98 for RMSE Capacity

dataset, suggesting that the two techniques perform similarly for EoL forecasting once cell specific performance is accounted for.

Nonetheless the performance of PLR modelling for capacity forecasting meant that it could be considered an effective replacement for the machine learning approaches here. A version of Fig. 3.11 with data for all test cells is available in appendix C.

The main peak in Fig. 3.8a was found to indicate an onset of the knee point in Severson-2019 and Attia-2020. All sub-models acting in domains of $V_{2,3} < 0.356$ were then providing the flexibility to model the non-linear stage of degradation for these cells. Conditions such as $V_{2,3} \approx 0.356$ and $V_{2,3} \approx 0.267$ were found to be approximate indicators of passing through a knee onset and a knee point respectively. These conditions, as part of the curvature approach to PLR modelling, provided useful markers that could act as a warning signs in a real application. The coefficients in each sub-model show how different inputs contribute to the degradation of a given cell, demonstrating to the transparency of the PLR approach. An example parameter set for a full PLR model is included in Appendix C.

The computational time advantage of PLR versus Gaussian process regression was clear, as can be seen in Fig. 3.10. That difference would have been more significant without the sparse approximation used by Matlab's *fitrgp* tool. Overall, a PLR model with five input features and four sub-models requires 24 coefficients, one noise parameter and three breakpoint values to be stored. This is a significant saving on the approximately 12,500 values of input data, 2,500 target ΔQ values and seven

hyperparameters that would be required to store an equivalent GPR model if using 50 training cells.

The curvature approach for finding breakpoints requires fewer sub-models and there is also value in the ease of understanding. The curvature approach suggested that degradation was limited while the batteries spent > 0.356 of time outside of extreme voltage regions, both of which are known causes of degradation [15, 31]. Rapid degradation occurred when the batteries were at extreme voltages for over two-thirds of the time, but a causal link cannot be proven here.

Using evenly spaced breakpoints performed equally well as a function of the number of training cells but required more sub-models and offered no insight through the breakpoint positions. An equivalent figure to Fig. 3.8 using evenly spaced breakpoints is in Appendix C. Otherwise, any judgement on the relative performance of even spacing versus the curvature method was subjective.

The PLR predictive performance was significantly weaker for the Sauer-2021 dataset because of the second change in degradation rate (see Fig. 1.7). Median EoL error was small which suggests that the predictive performance remains good. The 95^{th} percentiles were less informative because there were only 48 cells. Performance of PLR and GPR-RBF became equivalent when the capacity was truncated, i.e. there was a single behaviour change in the data. On inspection, the problems with Sauer-2021 arose because of the combination of a complex relationship between the splitting variable and $\frac{dQ}{dt}$, and insufficient data in the training set. The result was a smoothed function

that still presented a small amount of noise which influenced the breakpoint selection procedure, similar to the $\beta_L \approx 10^{-2}$ case in Fig. 3.9.

As in Chapter 2, the performance of prediction credible intervals has been ignored here but a PLR model is included in the analysis in Chapter 4. It is of note that including credible intervals in a PLR health model significantly increases the storage requirement because A^{-1} is a $D \times D$ array. The example case with four sub-models would require 24 coefficients and the 144 values in the four arrays to be stored.

3.7 Conclusion

A piecewise linear regression model for lithium-ion battery capacity performed approximately equivalently to Gaussian process models when measured by RMSE Capacity and EoL error for the Severson-2019 and Attia-2020 datasets. The PLR model was robust to significant variations in the controls of the model, typically allowing for variation across an order of magnitude. PLR was found to be flexible to changing degradation rates in the Severson-2019 and Attia-2020 datasets.

The PLR approach trained substantially faster than a GPR model and had a much smaller storage requirement, even if the credible intervals were required. The curvature approach divided the training data into intuitively placed sub-models. The performance was equivalent to other breakpoint finding methods but with fewer models required or less time taken and while providing possible indicators of knee points.

The speed and transparency of the PLR approach are advantages relative to the GPR models. However it was difficult to assign the approach's performance at higher percentiles of RMSE Capacity and EoL error to its being a better fitting tool than the GPR models. The weaker RMSE ΔQ performance suggested that PLR struggled to react to less significant relationships in the data, one benefit of which is a reduced risk of overfitting. The PLR model was also less able to handle multiple significant changes in degradation rate with small quantities of training data. Datasets similar to Sauer-2021 are rare because cells are typically removed from use or experiments before reaching 50% capacity. Nonetheless, further work could investigate the formation and selection of the splitting variable.

The credible intervals of the predictive posterior can be included at a cost to storage efficiency. Further study would look at how to capture uncertainty with a piecewise approach. The noise parameter σ_n in particular could be calculated a number of ways, possibly as a function of the inputs or SoH.

Chapter 4

Capturing degradation uncertainty

The focus of previous chapters was estimating future health of individual battery cells. However, the uncertainty of the forecasts was ignored because of a lack of comprehensive metrics to assess predictive performance. Successfully capturing that uncertainty is an essential component of confident battery health forecasting.

This chapter explores the uncertainty associated with lithium-ion battery degradation forecasts. The introduction discusses the general impacts of noise and uncertainty on data-driven approaches for monitoring lithium-ion battery state of health (SoH). Section 4.2 attempts to quantify intrinsic variability (i.e. caused by manufacturing variability) by observing the consistency of both empirical models and Gaussian process regression (GPR) models as a function of sample size. Multi-level Bayes (MLB) is introduced as a method of estimating parameters of population-level distributions.

Section 4.3 proposes a method to assess the performance of stochastic predictive health models as an improvement on the limited options in literature. Sparse Gaussian process regression is also used to significantly economise the storage requirements of predictive distributions.

4.1 Introduction

Uncertainty in lithium-ion battery degradation forecasting comes from a number of sources [10]. Batteries suffer from intrinsic manufacturing variability that can cause variation in both initial health [17] and subsequent ageing under identical conditions [16, 18]. There is also variability due to differing use protocols or natural forcings [238, 239]. Temperature differences, from environmental conditions and/or inhomogeneities, are sufficiently impactful to merit specific concern [240–242]. Modelling introduces further uncertainty due to measurement noise and prediction uncertainties [138, 243, 244].

The Bayesian methods used in Chapters 2 and 3 include the hyperparameter σ_n which represents the output noise. This is a reflection of the behaviour that cannot be predicted by the model. The noise hyperparameter acts as a catch-all but there remain questions regarding its effectiveness. For instance, the previous models assumed a Gaussian likelihood and all used a single fixed value for σ_n at all stages of ageing, but those assumptions may be weak [245, 246].

The transition models presented in earlier chapters are subject to measurement noise but the impact is challenging to observe. The impact is clearer in SoH estimation tools. To demonstrate this, the input features in Chapter 2 were used in a SoH estimation GPR model with an exponential kernel. Two forms of those input features are compared in Appendix B.2. The first were the same input features as in Chapter 2, i.e. the input features were functions of time spent in regions of use in the current time interval. Here, they are referred to as instantaneous features, and denoted f. The variant of these inputs are the cumulative equivalents, i.e. the proportion of time spent

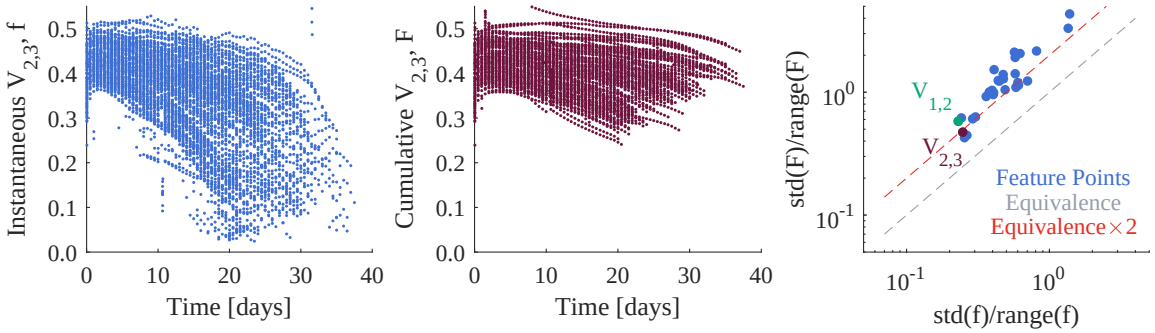


Fig. 4.1 The lack of signal resulting from noisy input features. The right plot shows the relative signal to noise ratios using instantaneous (blue) and cumulative (claret) features. Final plot demonstrated the changing signal to noise ratios of the two feature types.

in regions of use since $t = 0$. These input features are referred to as cumulative features and are denoted F. The cumulative inputs contain the same signal but of a reduced size, whereas the variability between cells is approximately constant, as shown in Fig. 4.1. The size of the noise, taken as the standard deviation of the feature, relative to the signal size, taken as the range of the feature, is approximately double with cumulative input features, F. Consequently, the SoH estimation predictive performance was weaker with the cumulative input features. The median RMSE Capacity when using instantaneous input features was 1.2% capacity, but the value with cumulative input features was 1.7% capacity. The full results are in Appendix B.2.

This chapter introduces two methods to capture uncertainty. First, cell-to-cell variability is quantified by estimating the required sample sizes for consistent model performance. The bulk of Section 4.2 was produced as part of published work in reference [3] with a number of other researchers[1]. The upper bounds on the population-

[1]Author list: P. Dechent, S. Greenbank, F. Hildenbrand, S. Jbabdi, D.U. Sauer and D.A. Howey. PD and I (SG) were joint first authors on this work. FH contributed data-procurement and review, SJ contributed MLB, MCMC code, advice and review, DUS and DH were principle investigators providing advice, guidance and review.

level variance, σ_p^2, were increased here so that the procedure was as consistent as possible for all parameters. That work was extended here by running an equivalent trial using a Gaussian process regression model, but still aiming to find the required sample size.

Section 4.3 focusses on credible intervals, i.e. quantifying the prediction uncertainty that was omitted in Chapters 2 and 3. Section 4.3 presents literature examples of performance metrics for predictive distributions and proposes a new, more comprehensive metric. Further, there is an effort to economise the storage complexity of predictive distributions using sparse Gaussian process regression.

4.2 Quantifying cell-to-cell variability

Quantifying cell-to-cell variability, i.e. the variation in performance of supposedly identical cells, is an important challenge for experimentation, use, and modelling of lithium-ion batteries. Several studies have explicitly demonstrated the presence of variability and its impact on pack performance [247–249], but methods to evaluate the extent of the variability are lacking in battery literature.

Here, we asked the modelling specific question *how many cells, n_c, are required to fit the parameters of a capacity fade model?* Capacity-time parametric models with up to three parameters were used to reduce the required number of parameters. An early hypothesis was that more complex models would require more cells to capture variability, so it was important to keep the complexity to a minimum. Some of the

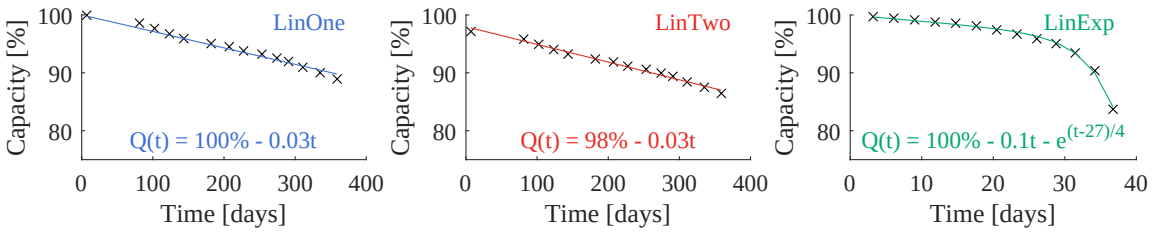


Fig. 4.2 The LinOne (blue), LinTwo (red) and LinExp (green) models used in the cell-to-cell variability study, plotted with example cell data with parameter estimates shown. LinOne and LinTwo plots use a cell from Dechent-2020, the cell in the LinExp plot was from Severson-2019.

datasets available were large by literature standards but the biggest contained only 67 cells.

Three models were used with each of one, two or three parameters, and they are given in the equations below. LinOne and LinTwo were both linear models with a constant degradation rate and either a constant initial capacity or a variable one, respectively, in equations 4.1 and 4.2. The three parameter model, LinExp, assumed linear degradation prior to the knee point, after which the capacity was modelled with an exponential capacity loss. The knee point time is approximated by t_{knee} in equation 4.3. All three models are shown with example parameters in Fig. 4.2.

$$\text{LinOne:} \quad Q(t) = 100\% + c_1 t \tag{4.1}$$

$$\text{LinTwo:} \quad Q(t) = B_2 + c_2 t \tag{4.2}$$

$$\text{LinExp:} \quad Q(t) = c_3 t - \exp\left(-\frac{1}{\tau_D}(t - t_{\text{knee}})\right) \tag{4.3}$$

The datasets used were Sauer-2021 [206], Dechent-2017 [203], Dechent-2020 [204], Severson-2019 [22] and Attia-2020 [107], although not all datasets were used for all

Dataset	LinOne	LinTwo	LinExp	Ref.
Dechent-2017	X	X		[203]
Dechent-2020	X	X		[204]
Sauer-2021			X	[206]
Severson-2019	X	X	X	[22]
Attia-2020	X	X	X	[107]

Table 4.1 The dataset-model combinations used in the multi-level Bayes study.

models, as shown in Table 4.1. The datasets each contained a fixed number of cells but a subset of those cells were used at each test point. From here, these subsets are referred to as sub-samples. Any use of *"sample"* refers to a whole dataset.

Only 67 cells from Severson-2019 were selected based on each having similar use protocols and ageing profiles. Early trials found that including all cells from Severson-2019 produced two distinct distributions for each parameter which over-complicated the early effort to quantify variability presented here. The second, smaller group of cells had lifetimes between 18 and 23 days, and could be included in further studies.

The target of this section was to estimate the sub-sample size, K, required to fit the parameters of a population-level empirical model. Defining *fit* was the biggest challenge here and is the subject of Section 4.2.2.

4.2.1 Multi-level Bayes

Multi-level Bayes (MLB) was used to fit the parameters of the three models at both the individual cell level and population level. The approach is a Bayesian parameter fitting technique that constructs the model in multiple layers. Here, those layers are represented by the sub-sample and the population.

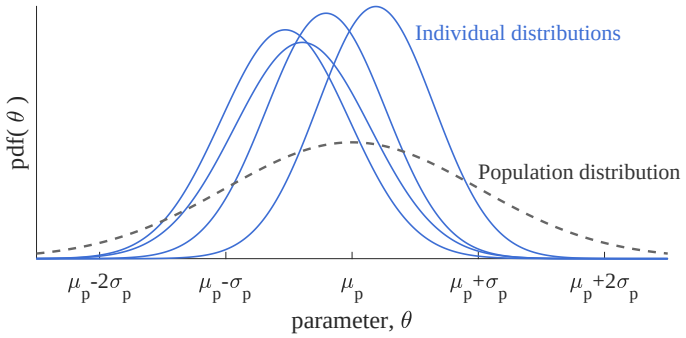


Fig. 4.3 The principle of multi-level Bayes. Individual cells have tighter estimated parameter distributions (blue) whereas the population parameter distribution (grey, dashed) is wider.

The parameters θ were assumed to have come from a population distribution, much as how a batch of lithium-ion cells comes from a much larger population of cells. MLB modelling estimates the parameters of the population-level parameter distributions that would appear if one had fit the model to every cell in that hypothetical population. The population distribution is distinct from parameter distribution for a single cell, as shown in Fig. 4.3.

All parameters were assumed to be drawn from a population normal distribution with associated mean μ_p and variance σ_p^2. The first step was to produce a Bayesian estimate of the parameters for each individual cell, θ_k, in a sub-sample of size K. The parameters in θ_k for each model are shown in equation 4.4.

$$\text{LinOne: } \theta_k = [c_{1,k}], \quad \text{LinTwo: } \theta_k = \begin{bmatrix} B_{2,k} \\ c_{2,k} \end{bmatrix}, \quad \text{LinExp: } \theta_k = \begin{bmatrix} c_{3,k} \\ t_{\text{knee},k} \\ \tau_{D,k} \end{bmatrix} \tag{4.4}$$

Using the Metropolis-Hastings method [250, 251], mean and variance estimates for θ_k were found for each cell with the distributions assumed to be Gaussian:

$$\theta_k \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right) \tag{4.5}$$

The parameter distributions for individual cells, k, are shown in blue in Fig. 4.3.

Population-level prior distributions were formed by taking the mean and variances of the estimates of θ_k and σ_k across the entire sub-sample. Finally, MLB was implemented by using MCMC to estimate the parameters of the population-level posterior probability distributions from the capacity-time data, $\{t, Q\}$, and the sub-sample parameter distributions[2]. We chose the means of the posteriors to give the population parameters $p(\mu_p, \sigma_p|\{\theta_k\}, \{t, Q\})$. Each parameter was then described by a normal distribution:

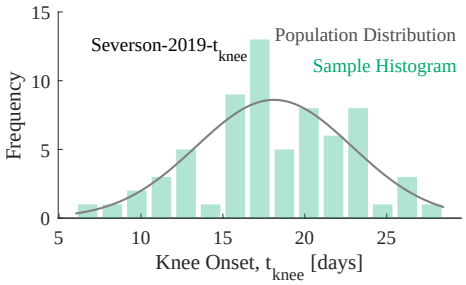
$$\theta_p \sim \mathcal{N}\left(\mu_p, \sigma_p^2\right) \tag{4.6}$$

An example population-level parameter distribution is shown in Fig. 4.4a for Severson-2019-t_{knee}. A fuller derivation and explanation of MLB is in Appendix D.

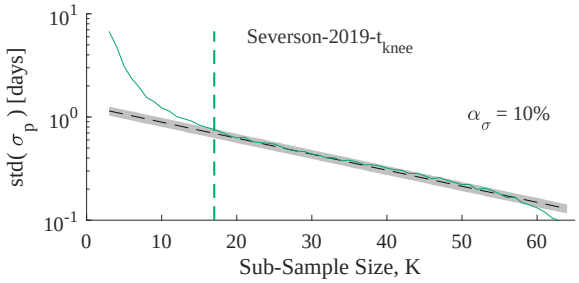
4.2.2 Stable population estimates

The variability across a dataset was observed by the variance of a population parameter estimate, σ_p. As the sub-sample size increases, one expects the variance in the estimate of σ_p, var(σ_p), to decrease if repeatedly using MLB on multiple, randomly selected

[2]All MCMC code was provided by S. Jbabdi, University of Oxford.



(a) Example fitting with MLB.



(b) Method for defining captured variability.

Fig. 4.4 Model output (a) and quantification method (b) used to capture cell-to-cell variability.

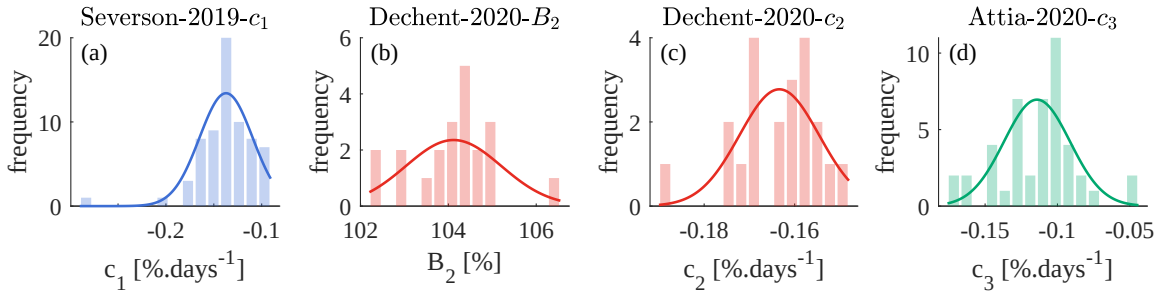


Fig. 4.5 Example population distributions (solid lines) for several example parameter-dataset combinations. Histograms are parameter values for the full sample. Distributions were single estimates from the maximum sub-sample sizes from the main trial.

sub-samples. If the sub-samples were selected from a infinite population, one should see $\text{var}(\sigma_p) \sim \frac{1}{K}$ according to the standard error [252]. However, when drawing from a fixed sample of size N, the variance of the population estimates is limited by the available sample parameter values.

The variability was deemed to have been captured when $\text{var}(\sigma_p)$ began to be influenced by the sample size N, seen on a logarithmic y-axis as a linear reduction in the standard deviation of σ_p as a function of K. Also, standard deviation is preferred to variance because the units are consistent with those of the parameters.

The estimate of the required number of cells, n_c, to stably capture the variance was the smallest sub-sample size where the standard deviation was lower than a linear extrapolation of the mid-sub-sample size region, plus a percentage tolerance α_σ. Here, $\alpha_\sigma = 10\%$ because there was very little noise. Fig. 4.4b demonstrates the technique for the t_{knee} parameter for Severson-2019. A value of n_c was estimated for every dataset-model combination.

Stable parameter estimates are also important for machine learning models. Therefore, an extension is included to estimate n_c for a GPR model, the most common machine learning model in this thesis. Matlab's GPR tool, *fitrgp*, does not return an exact equivalent to σ_p. Consequently, the variability of the hyperparameters as a function of the number of training cells was investigated as an alternative. All 157 cells from the combined Severson-2019 and Attia-2020 datasets were used with a fixed set of inputs, $V_{2,3}$, $V_{1,2}$, time and $|P|_{1,2}$. After 100 repeats were performed at each test point[3], median and interquartile ranges (IQR) of the hyperparameters were returned, along with the RMSE Capacity of the resultant models. Percentiles, not means and variances, were used because the test sample sizes were deemed too small.

4.2.3 Results

The MLB population distribution parameter estimates appeared to be reliable. Fig. 4.5 shows a number of example fits where outliers are successfully avoided (Severson-2019-c_1, 4.5.(a)) and non-Gaussian samples are well approximated (Dechent-2020-B_2,

[3]Only 100 repeats were performed to limit the time taken by this trial.

Parameter	Dechent-2017	Dechent-2020	Sauer-2021	Severson-2019	Attia-2020
c_1	8	7		12	9
B_2	8	8		16	14
c_2	7	7		11	13
c_3			14	19	13
t_{knee}			17	16	12
τ_D			13	13	12

Table 4.2 Estimates of required number of cells, n_c, to fit each parameter-dataset combination.

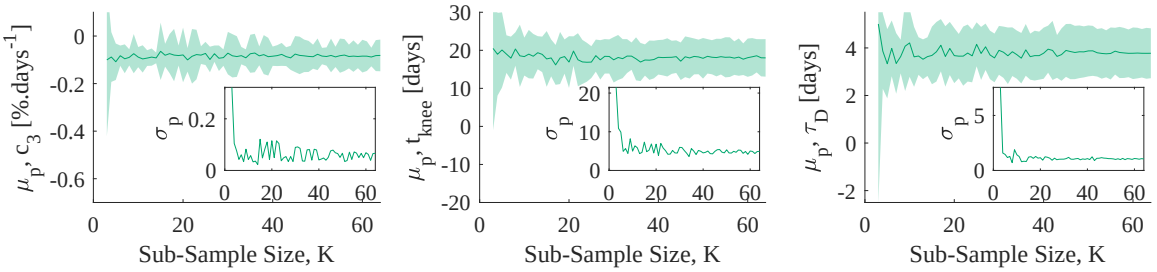


Fig. 4.6 Estimated population distributions for single repetition varying with sub-sample size, K. Figure produced for a cell from Severson-2019 with the LinExp model.

4.5.(b)). The results in Fig. 4.6 demonstrate that MLB responded to increased sub-sample size as expected. The μ_g estimates become more consistent as the sub-sample size grows while the σ_g estimates rapidly reduce in size then plateau.

The sub-sample size versus standard deviation of σ_p curves are very smooth because 1,000 repeats are performed. The results for LinExp were plotted in Fig. 4.7, the equivalent plots for LinOne and LinTwo are in Appendix D. Estimates of n_c vary between 7 and 19 in Table 4.2.

In summary, typically $n_c \approx 12$, with slightly higher n_c estimates being required for the more complex models in Fig. 4.8.

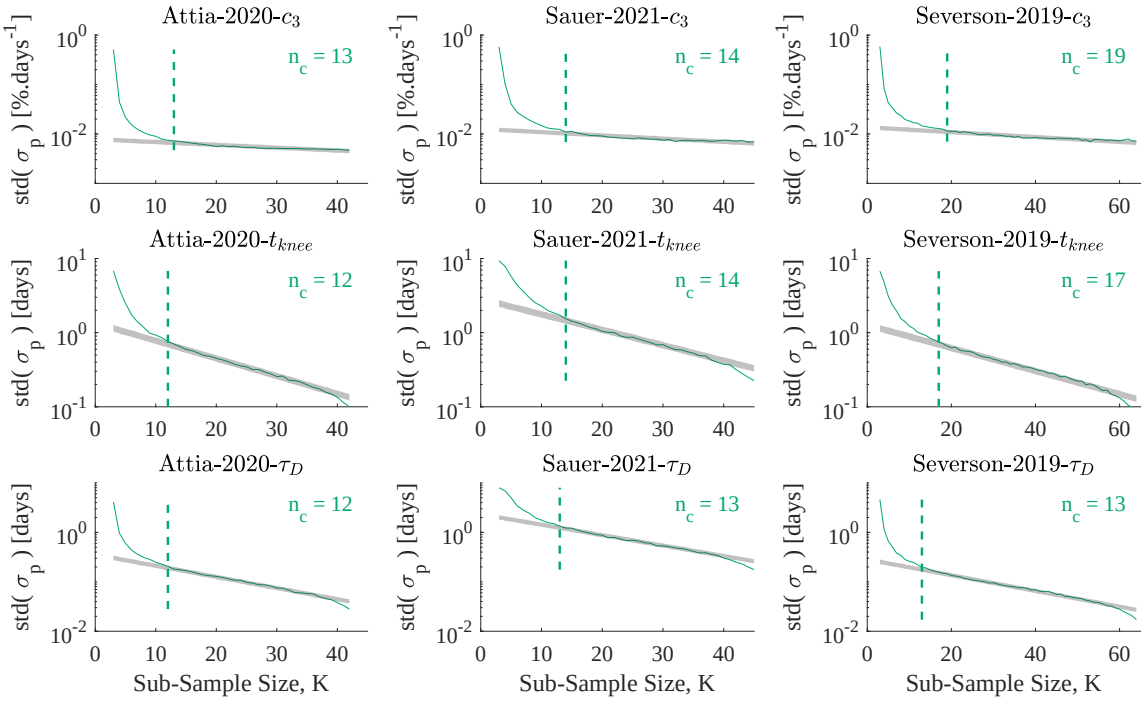


Fig. 4.7 Estimating required sub-sample sizes for the LinExp model.

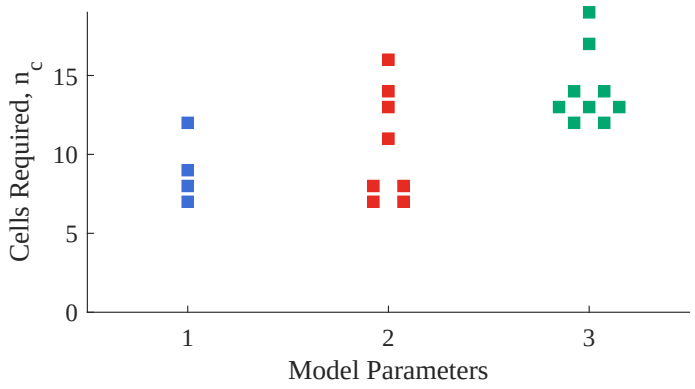


Fig. 4.8 Number of cells required to fit capacity fade models of differing complexity. Offsets included where multiple values of equal n_c were found.

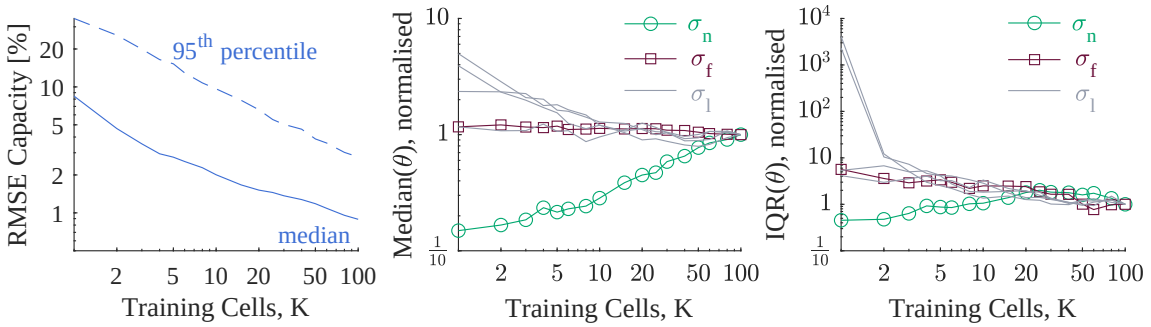


Fig. 4.9 Performance and hyperparameter variability for a GPR-RBF capacity model as a function of the number of training cells. Hyperparameter plots normalized relative to value with $K = 100$.

An equivalent approach for GPR only used 100 repeats resulting in more noise visible in Fig. 4.9. The RMSE Capacity displays a distinct improvement as a function of the increasing number of available training cells. The medians of the lengthscale hyperparameters, σ_l, reduced in size as a function of number of training cells while the noise increased and the magnitude hyperparameter, σ_f, remained consistent. The IQR of the hyperparameters saw a distinct improvement for the lengthscale hyperparameters up to ≈ 4 cells.

4.2.4 Discussion

Multi-level Bayes provides a robust approach for population distribution parameter estimation. The resultant population distributions appeared to match the sample distributions which provides confidence in the technique.

The use of 1,000 repetitions produced a very small amount of noise about the std(σ_p) curves in Fig. 4.7. Such small noise contributed to consistent estimates of n_c, the required number of cells to fit a parameter distribution. Some parameter histograms

were visibly non-Gaussian, such as Dechent-2020-B_2 in Fig. 4.5. In future, attention could be given to model selection based on the shape of the resultant distributions within a sample [3].

Overall, an estimate of $n_c \approx 12$ was drawn from the results but mean n_c was 9, 11 and 14 for the empirical models with 1, 2 and 3 parameters respectively, suggesting an increasing relationship versus model complexity. Further, the number of cells increased for the same datasets with different models. For example, Severson-2019 required successively higher n_c estimates for each more complex model in Table 4.2. However, confidence in that increasing relationship is limited. Attia-2020 did not replicate that result and Dechent-2017 and Dechent-2020 were fit with fewer cells for the two linear models but did not contribute estimates for the LinExp model, effectively contributing to lower n_c for LinOne and LinTwo relative to LinExp.

The estimate of $n_c \approx 12$ was higher than the equivalent result in the published work, $n_c \approx 11$ [3]. The only change was the wider bounds for the population-level variance parameter, σ_p^2. The upper bounds in the published work were significantly above the calculated population variances, but the higher estimate of n_c suggests that there was still an influence from the upper bound on the calculation of required cells.

The goal of Section 4.2 was to answer *how many cells, n_{cells} are required to fit the parameters of a capacity model?* The MLB approach appeared to produce reliable and reasonable n_c values with a small number of simple models. However the technique relies on the impact of repeated sub-sampling from a full dataset to find the apparent relationships seen on logarithmic scales in the plots of Fig. 4.7. Another issue was

that the technique learnt a very small amount from each dataset. For example, the Sauer-2021 dataset contains over 500 MB of battery ageing data [206] but produced only 3 n_c estimates.

The MLB approach to estimating n_c would be challenging to apply to more complex models, such as the electrochemical models mentioned in Section 1.2.1. Future work could look into applying MLB with physics-based, semi-empirical models but particular attention would need to be given to sample distribution parameters and how to limit the impact of sample size on models with large numbers of parameters.

The small trial using GPR produced different results depending on the hyperparameters being investigated. Most hyperparameters were consistently fit with $n_c < 5$, however σ_n needed 50 or more cells before the normalized median was comparable to using 100 cells. Small trials found that σ_f and σ_l can vary over small ranges without significant impact on the outputs, but σ_n is critical for capturing the predictive distribution and hence the uncertainty. These results suggest that capturing uncertainty with GPR models requires n_c to be an order of magnitude higher than that required to capture either the mean estimate of the GPR model or the simple parametric degradation models. RMSE Capacity steadily improved as a function of K, even when the hyperparameters were consistent at $K > 20$, suggesting that the availability of training data is a significant contributor GPR model performance.

4.3 Credible intervals

The previous sections of this chapter have introduced the problem of uncertainty and attempted to quantify the impact of cell-to-cell variability on health modelling. The rest of this chapter is dedicated to investigating the predictive distributions in capacity forecasts, both by quantifying performance and by improving the storage requirements of the GPR models.

Credible intervals specify a region within which an unknown parameter will fall with a certain probability. For GPR the unknown parameters are the target values whereas in Bayesian linear regression (BLR) the unknown parameters are the coefficients. For simplicity, the rest of the chapter always refers to the credible intervals and predictive distributions about capacity forecasts, i.e. the target values of the regression models. Typically, the 2-σ interval is used in literature, meaning the region within 2 standard deviations of the mean estimate [59, 88, 90, 152]. The 2-σ interval on a Gaussian distribution represents a probability of $\approx$95.45%, but a 95% interval is also commonly used [113, 143, 156]. Both BLR and GPR provide probabilistic outputs which here have been assumed Gaussian with some mean function, function variance[4] and noise variance. The derivations of equations 4.7 and 4.8 are in Section 3.3 and 1.5 respectively.

[4]In GPR, the function variance is a function of the covariance function, the training data, and the test data. In BLR, the function variance is a function of the parameter covariance and the test inputs.

These distributions may be used to estimate 95% or 2-σ intervals.

$$\text{BLR:} \quad p(\mathbf{y}_*|X_*, \mathbf{y}, X) = \mathcal{N}\left(\mu_{\text{BLR}}, \sigma^2_{\text{BLR}} + \sigma^2_n\right) \tag{4.7}$$

$$\text{GPR:} \quad p(\mathbf{y}_*|X_*, \mathbf{y}, X) = \mathcal{N}\left(\mu_{\text{GPR}}, \sigma^2_{\text{GPR}} + \sigma^2_n\right) \tag{4.8}$$

Probabilistic outputs are cited as an advantage for Bayesian prognostic methods [13]. Credible intervals have even been used as a weighting method for aggregating predictive models [146]. But there have been limited attempts to assess the honesty of the intervals.

The most common assessment method in battery literature is the calibration score. Typically applied across the 2-σ intervals, the calibration score is the proportion of observations that fall within the given interval. The ideal score for the 95% confidence interval would be 0.95. Calibration scores in literature, and performance later in this chapter, are not accurate to within 0.01 of 0.95 suggesting that differentiating between 95% and 2-σ intervals would be futile for battery health models [26, 138, 149, 152]. The calibration score only assesses the performance of a single point metric of a predictive distribution and so was deemed insufficiently detailed for use here.

An alternative but similar metric is the β-score [138]. As shown in Fig. 4.10, the β-score is the cumulative probability within a specified margin about the true value [60, 138]. That margin is specified by α in equation 4.9. In plain words, the β-score assesses how close the predictive distribution is to the observed value, measured by the observed value using $\pm\alpha$. By contrast, the calibration score assesses how close the

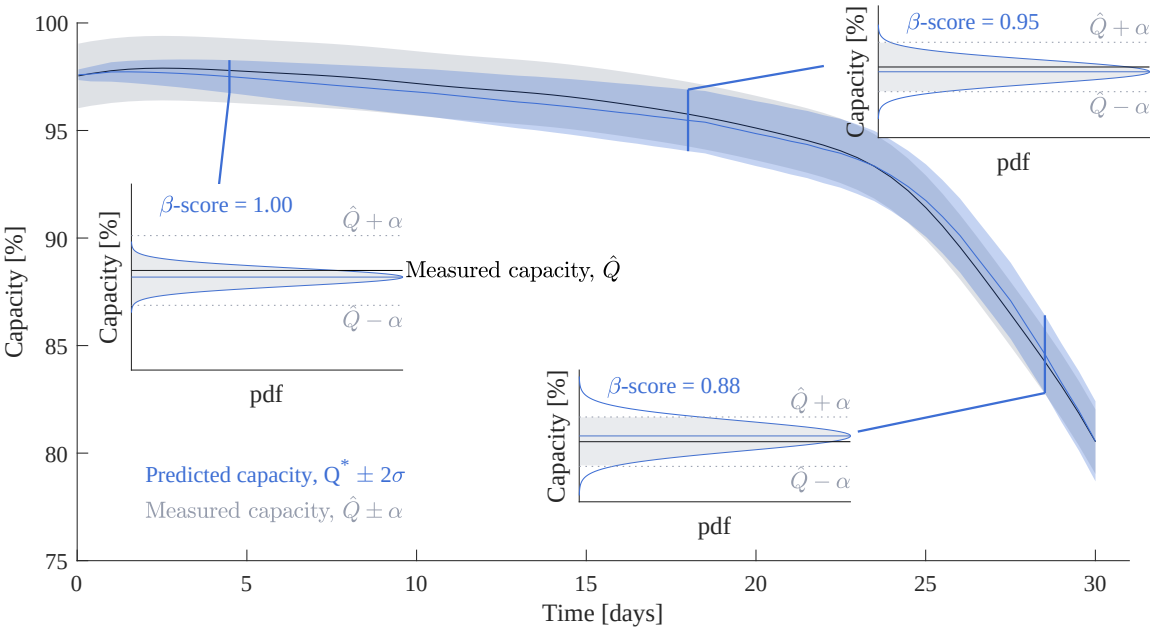


Fig. 4.10 Demonstration of β-score calculation. The output is the cumulative predicted probability within a specified distance, α, of the measured capacity.

observed value is to the prediction, measured by the predictive distribution.

$$\beta\text{-score}(t_i) = \int_{Q(t_i)-\alpha}^{Q(t_i)+\alpha} p(Q)dQ \tag{4.9}$$

The previous work that presented β-scores for battery SoH estimation used $\alpha = 1.5\%$ capacity because that was a limit set by their measurement accuracy [138], but the value of α in other applications is arbitrary. Further, the method works well for SoH estimation because all predictions are composed of single estimates. On the other hand, SoH prognosis requires that variances are summed as subsequent ΔQ forecasts are combined: $\sigma(t_i) = \sqrt{\sum_{k=1}^{i} \sigma(t_k)^2}$. The credible intervals therefore grow through lifetime, effectively changing the possible β-score regardless of predictive performance.

Therefore, a new quantification metric is proposed in Section 4.3.1. Verifying the performance and suitability of this was challenging without access to extremely large datasets or known ground-truths. Alternatively, a test was performed with a predictable relationship between independent variable and predictive performance and the aim was to reproduce that relationship using the new metric.

4.3.1 Quantifying uncertainty performance: RMSE-Freq

The proposed metric for predictive distributions is based on Brier scores [253] and is similar to the optimization strategy in reference [138]. The metric targets the specific application of battery health where a user cares mainly about whether a cell is above or below a given predicted SoH.

Any given predictive distribution in battery capacity can be converted into a cumulative distribution function (cdf) with $\text{cdf}(Q = -\infty) = 0$ and $\text{cdf}(Q = +\infty) = 1$. Fig. 4.11.(a) shows three example probability density functions (pdf), and the cumulative probability of the measured capacity is calculated in Fig. 4.11.(b). That cdf is used to produce a set of binary outputs depending on whether or not the forecasted frequency is above or below the observation, as in Figs. 4.11.(c) and 4.11.(d). Simplistically, in a well-calibrated model, the observed capacity should appear below a forecasted frequency of 0.30 in 30% of predicted capacity values and similarly for all values of forecasted frequency.

If a model is well-calibrated, it should present a very small deviation from the dashed grey line in Fig. 4.11.(e). Consequently, the measure of performance for a

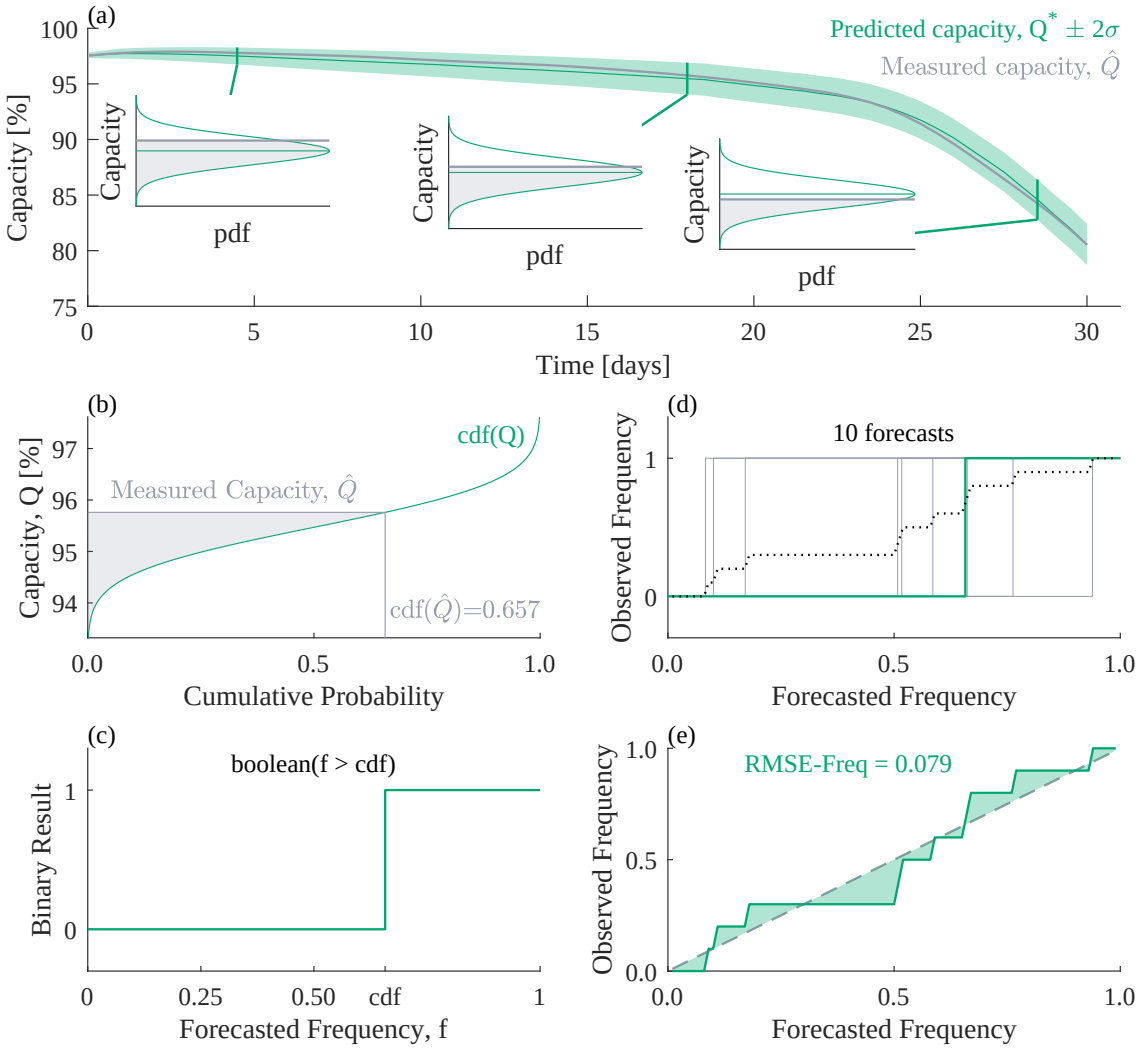


Fig. 4.11 Demonstration of RMSE-Freq calculation. The output is the root mean square distance between the forecasted frequency of capacity values and the observed frequencies, as calculated based on the cumulative probability of the predictive distributions. (a) The forecasted capacity predictive distributions are compared with the measured capacity. (b) The measured capacity at each test point is used to calculate the predicted probability of the capacity being below the measurement. (c) The cumulative predictive forecast is converted into a series of binary scores with a 1 for every forecasted frequency above cdf($\hat{Q}$). (d) Multiple binary forecasts are combined and averaged to produce a mean average observed frequency (dotted line). (e) The root mean square difference between the mean average observed frequency and the ideal forecast (dashed grey line) is taken as the RMSE-Freq.

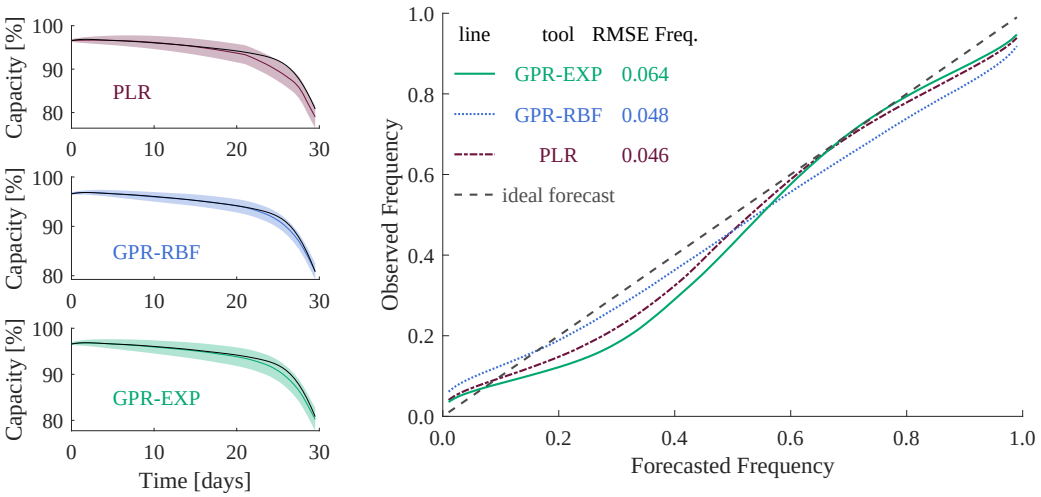


Fig. 4.12 Brier-style plot for RMSE-Freq of previously tested models, PLR, GPR-EXP and GPR-RBF.

predictive distribution will be RMSE-Freq, the root mean square error between the perfectly calibrated model and the actual performance.

The performance metric RMSE-Freq is therefore a measure of how accurately forecasted probability distributions resemble the observed SoH, measured according to the battery-specific condition of being above or below a target SoH value.

Results using the models in Chapters 2 and 3 are given in Fig. 4.12. Despite being the best performing model in previous chapters, GPR with an exponential kernel (GPR-EXP) is the weakest of the three tested when measured with RMSE-Freq. These results were consistent - the trial in Fig. 4.12 used 100 repetitions with 50 training cells in all cases. There was a consistent bias to overpredict the frequency at which capacity was below the predictions, most likely resulting from the bias in the mean estimate, as shown in all three profiles in Fig. 4.12.

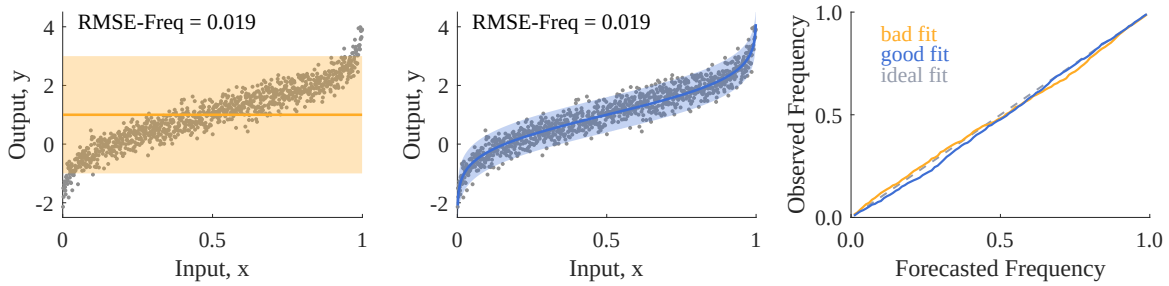


Fig. 4.13 Example failure of RMSE-Freq on generated data. Two very different fits were shown to score equally with RMSE-Freq.

RMSE-Freq does appear to be an effective measure but it could fail to distinguish between a distribution which fits an unintelligent model to a full data array and one that is fitting a more detailed model. Fig. 4.13 demonstrates a good example of that failure on carefully generated random data. This error can be avoided by using RMSE-Freq in conjunction with the RMSE of the target variable, which is RMSE Capacity in this thesis.

Testing results given below in Section 4.3.3 include EoL error and RMSE Capacity as extra performance metrics. The β-score is presented as a comparison. These tests also assess the predictive performance of Sparse Gaussian process regression (SparseGPR) as a function of the number of pseudo-inputs used. SparseGPR is the improvement to GPR described in Section 1.5.1 but a new implementation for battery health forecasting proposed here is described below.

4.3.2 Sparse Gaussian processes

Gaussian processes have been found to produce flexible and accurate capacity health forecasts with well-calibrated predictive distributions, as shown in Fig. 4.12. But storing the full model is expensive because all training points must be retained.

Sparse Gaussian process regression, also known as sparse pseudo-input Gaussian process regression, is an approximation of GPR [188, 191, 195]. It involves selecting a replacement training set of pseudo-inputs of reduced size, $M \ll N$ rows, which can approximate the full training set. The resultant regression model then scales as $\mathcal{O}(M^2N)$, representing a significant computational saving compared to $\mathcal{O}(N^3)$ [188, 195]. A brief derivation of SparseGPR is included in Section 1.5.1 and references [188] and [191] are recommended for further reading.

There are a number of works investigating different methods of optimising the location of the pseudo-inputs [195, 191, 190]. That optimisation is computationally intensive because there are $MD + D + 2$ parameters to fit [193]. Applied to this work, the additional complexity was not found to significantly improve SoH forecasting in initial trials. As a faster alternative approach, K-means clustering was used to produce the pseudo-inputs prior to hyperparameter tuning, based on finding M focal points among the full training set. Fixing these M points as the pseudo inputs speeds up the process and avoids the difficult judgement of how to place a prior over the pseudo-input points. One can use points randomly selected from the training set [193], but K-means

was preferred here. The time input was normalized such that $t = 1$ was equivalent to 40 days[5] in order to prevent the time variable dominating the K-means calculation.

Applying SparseGPR

The SparseGPR approach not only reduces computational complexity but also offers an important opportunity to significantly reduce storage requirements. The proposal was to replace the training inputs with the pseudo-inputs, Z, in the predictive model with pseudo-targets, $\mathbf{f_Z}$, calculated as the mean estimates of the predictive posterior in equation 1.20. That switch could reduce storage requirements from 1,000's of rows down to 10's. The full training set is still required to calculate the pseudo-targets with K-means and fit the hyperparameters but the final predictive posterior only depends on the pseudo-inputs and pseudo-targets:

$$p(\mathbf{y}_*|X_*, Z, \mathbf{f_z}) = \mathcal{N}\left(K_{*Z}(K_Z + \sigma_n^2 I)^{-1}\mathbf{f_z}, K_* - K_{*Z}(K_Z + \sigma_n^2 I)^{-1}K_{*Z}^T + \sigma_n^2 I\right) \quad (4.10)$$

The rest of the SparseGPR model used identical processes to the models in Chapters 2 and 3. Input features were calculated based on Table 2.1 and automated feature selection using the methods previously discussed was used to select 5 input features.

SparseGPR using pseudo-inputs was expected to produce a distinct relationship between predictive performance and M, the number of pseudo-inputs. Higher values of M were assumed likely to produce smaller RMSE-Freq values. That relationship

[5]As described in Section 2.1, the maximum lifetime of cells selected was 40 days. This scaling placed all time variables between 0 and 1, approximately the same range as the other input features.

was verified as a function of RMSE Capacity and EoL error in Section 4.3.3, as well as with RMSE-Freq.

Two trials were performed using the sparse GPR implementation from GPy [194]. Both trials varied M between 3 and 100. There were 100 repeats at each test point, each with 50 randomly selected training cells and 107 test cells. RMSE Capacity, EoL Error, RMSE-Freq and the β-score were recorded for all 10,700 test profiles.

The first trial exclusively used the pseudo-inputs Z as proposed in equation 4.10 but still compared against the performance from a standard GPR-RBF model. The second trial compared the predictive performance between using SparseGPR with an RBF kernel as in literature, i.e. storing and using the full training data set X for the predictive posterior (denoted RBF(X)), against the pseudo-input method, i.e. only storing the pseudo-inputs Z and using equation 4.10 (denoted RBF(Z)), proposed here. The aim of this second trial was to establish the predictive performance cost of approximating the training set using M pseudo-input points.

4.3.3 Results

As expected, small numbers of input points resulted in poor predictive performance of lithium-ion battery capacity. RMSE Capacity and EoL error were both significantly worse below $M = 20$, see Fig. 4.14. SparseGPR-EXP was the worst performing model by all metrics at almost all values of M. SparseGPR-RBF appeared approximately equivalent to GPR-RBF at $M \approx 50$.

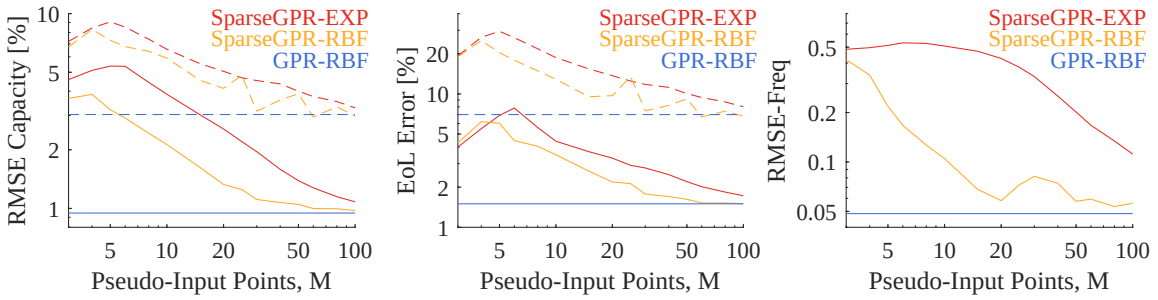


Fig. 4.14 Median (solid) and 95$^{\text{th}}$ percentile (dashed) performance of regression tools as a function of number of pseudo-inputs. Compares SparseGPR using an EXP kernel (red) and a RBF kernel (yellow) against a GPR model with a RBF kernel (blue).

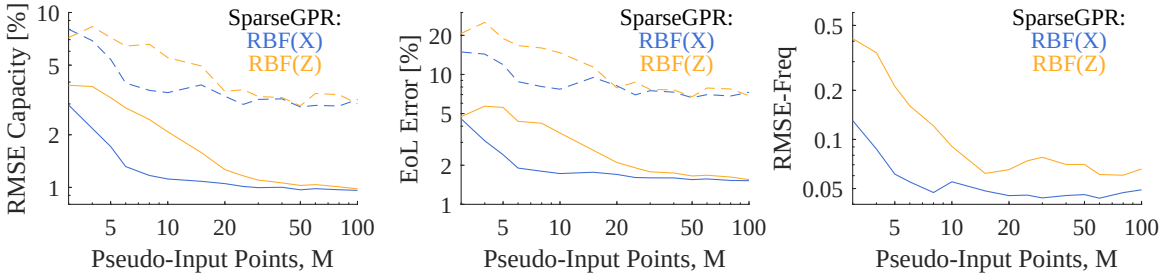


Fig. 4.15 Comparison of performance between using the pseudo-inputs (RBF(Z), yellow lines) and the training set (RBF(X), blue lines) in SparseGPR predictive posterior with an RBF kernel. Median and 95$^{\text{th}}$ percentiles in solid and dashed lines respectively.

The proposed prediction uncertainty accuracy metric, RMSE-Freq, displayed the expected relationship between M and predictive performance. That relationship was clearer for SparseGPR-RBF relative to SparseGPR-EXP for which RMSE-Freq was consistently poor below $M = 20$. SparseGPR-RBF never reaches the performance of the standard GPR-RBF in Fig. 4.14 when measured by RMSE-Freq.

In the second trial, the pseudo-input approach produced worse capacity models than the standard SparseGPR approach from literature at all values of M. However performance was approximately equivalent for $M \geq 40$, aside from the consistent difference in RMSE-Freq in Fig. 4.15.

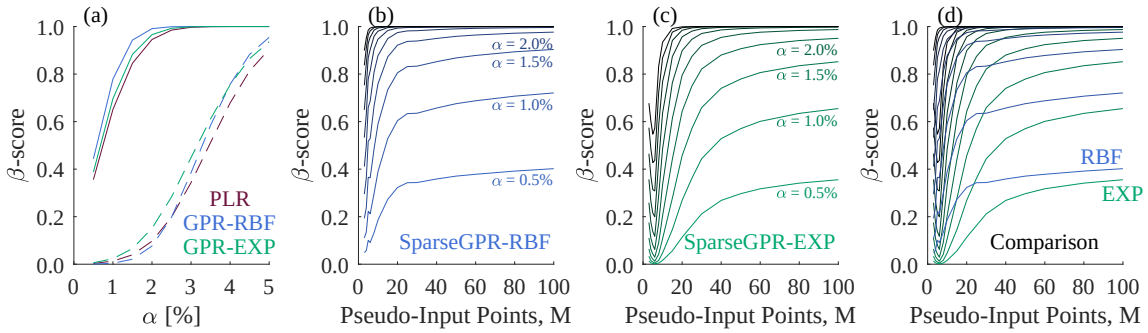


Fig. 4.16 Performance of the GPR methods measured by β-scores. Continuous lines are medians, dashed lines are 95$^{\text{th}}$ percentiles.

The β-score results in Fig. 4.16 show that SparseGPR-RBF produced tighter fits than SparseGPR-EXP for all values of α. Piecewise linear regression (PLR) performed worst as a function of α in Fig. 4.16.a), but all three techniques were approximately equivalent.

4.3.4 Discussion

The pseudo-input approach was weak for $M < 30$ because there was insufficient data to produce an accurate health model. The model appeared accurate at $M \approx 50$, representing a significant improvement on the storage requirements compared to a standard GPR model. Storing 50 pseudo-inputs and pseudo-outputs would reduce storage requirements by $\approx 98\%$ without a significant impact on RMSE Capacity or EoL error.

Fig. 4.15 demonstrates that using only pseudo-inputs produced larger RMSE-Freq values, even at higher values of M where RMSE Capacity and EoL error were equivalent whether the training data was included or not. The consistent difference in RMSE-Freq between using pseudo-inputs and using the full SparseGPR model suggests that the

pseudo-inputs still lacked sufficient detail to capture the full variability among the test cells.

Presenting β-score results in a comprehensive manner is challenging. Fig. 4.16.(a) shows median and 95^{th} percentiles as a function of α whereas Figs. 4.16.(b) and 4.16.(c) map how the β-score evolves as a function of both M and α. These plots were uninformative until compared in Fig. 4.16.(d). The results suggested that the predictive distributions were more accurate when calculated with an RBF kernel. Much like RMSE-Freq, β-scores were difficult to interpret without also knowing the profile-focussed predictive performance using RMSE Capacity. For example, Figs. 4.14 and 4.16 show that the β-scores were good for SparseGPR-RBF at $M \approx 20$ but RMSE Capacity was poor until $M \approx 50$.

The RBF kernel was found to consistently outperform the EXP kernel when using pseudo-inputs. Exponential kernels consistently returned lengthscale hyperparameters approximately 4 times the size of those with RBF kernels on the same data. Figs. 4.17.(a) and 4.17.(b) demonstrated the relative impact on the pseudo-input approach by marking the position at which the covariance between inputs had halved in size. The scale for the EXP kernel is such that the model struggled to discern between pseudo-input training points with small numbers of input data points.

The pseudo-targets in Fig. 4.17.c) were spread over a far smaller range with the EXP kernel and the weighted targets, calculated as in equation 4.11, did not correlate as well as with the RBF kernel. Despite being the best performing model in previous chapters, the exponential kernel was concluded to be a poor one when using a pseudo-input

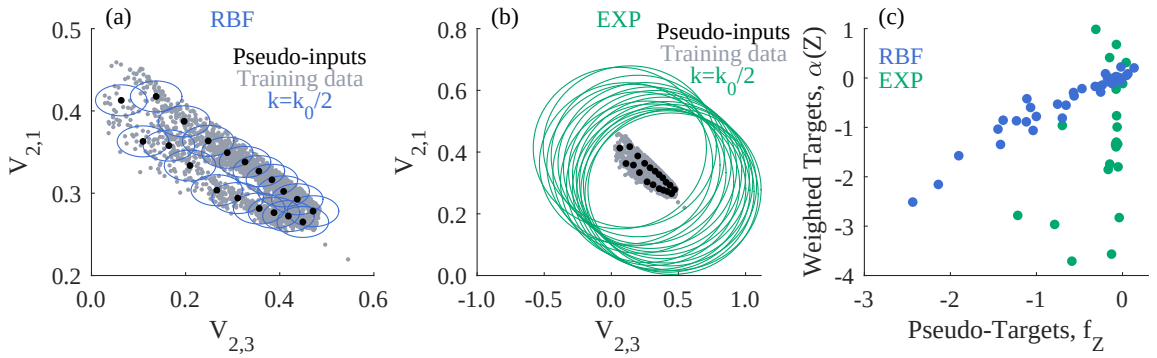


Fig. 4.17 Depiction of the difference between the RBF and EXP kernel functions when used for the pseudo-input approach for SparseGPR. For (a) and (b), the training data (grey dots) was used to calculate the pseudo-inputs (black dots) using K-means with K = 20. The point of covariance being half the maximum (coloured circles) is shown for each kernel choice, centered on each pseudo-input. (c) The weighted targets are shown as calculated by the RBF kernel (blue dots) and the EXP kernel (green dots).

technique.

$$\alpha(X) = \left(K_Z + \sigma_n^2 I\right)^{-1} \mathbf{f_Z} \implies f_*(\mathbf{x}_*) = \sum_{i=1}^{M} \alpha(\mathbf{z}_i) k(\mathbf{x}_*, \mathbf{z_i}) \tag{4.11}$$

This analysis was led by the RMSE-Freq results in Figs. 4.14 and 4.15. The metric is capable of distinguishing between distinct but similar modelling approaches and provides insight into predictive performance. Profile-focussed metrics, such as RMSE Capacity, are still required for reliable measurements and the metric was only meaningful if comparisons are available. Another issue with RMSE-Freq is the reliance on the shape of the posterior distribution, especially for the tails of the distributions. The tails represent the most interesting part of the distribution to a user interested in safety, so future metrics could add weights to RMSE-Freq such that lower capacity values are prioritised.

4.4 Conclusions

Two initial efforts at capturing the uncertainty within lithium-ion battery degradation models were proposed. The first was an attempt to determine the number of cells required to fit the parameters of an empirical model. The method established a consistent estimate of around 12 cells for datasets of significantly differing sizes. The multi-level Bayes approach produced consistent population distributions for the parameters in the empirical models. The required sub-sample size estimation was reliant on the use of a limited dataset and and remained an arbitrary condition. Future works could search for methods that do not require limited datasets. Another avenue would be to find a much larger dataset that can be modelled with a single empirical equation. Alternatively, finding appropriate models with increased complexity would be instructive.

The second half of Chapter 4 focussed on quantifying the performance of probabilistic predictive distributions. The proposed metric, RMSE-Freq, provided insightful comparisons and produced the expected relationship between the number of pseudo-inputs and predictive performance. The pseudo-input approach provided almost equivalent predictive performance to a full GPR model above 50 pseudo-inputs. The performance of kernel functions was distinguishable using RMSE-Freq and the β-score, but the β-score required the extensive plots in Fig. 4.16 in order to be insightful.

The pseudo-input approach found that the RBF kernel produced more accurate forecasts than the EXP kernel for a sparse Gaussian process regression tool, contrary to previous chapters. SparseGPR is complex but the proposed predictive model is

relatively simple. The output mean and variances are single calculations and the storage requirement is reduced by around 98% for the case here. Consequently, SparseGPR-RBF competes with the piecewise linear model in Chapter 3 for storage requirements and predictive capability, but only if uncertainties are required.

Chapter 5

Conclusions

5.1 Contributions and conclusions

5.1.1 Input feature generation

The feature generation procedure proposed in Chapter 2 produced a large number of potential input features. As predictors of change in battery capacity ΔQ, the input features were found to be robust to increased noise and reduced data collection frequency. The input features were calculated as proportions of time spent in regions of each use variable so that the values could be understood and, hopefully, be predictable in real world applications. The thresholds were found to be extremely consistent and would easily scale to other datasets.

The feature selection process was found to be fast and produced accurate capacity fade models when used with GPR, PLR and SparseGPR. The use of feature discrimination condition, $\rho_{P,\max}$, i.e. the maximum correlation coefficient between input features, had a positive impact on performance and ensured a diverse training dataset. The

strongest correlation with the target variable ΔQ was for $V_{2,3}$, the proportion of time spent between $V \approx 3.12$ V and $V \approx 3.51$ V, for the Severson-2019 and Attia-2020 datasets.

The feature generation and selection procedures were a small part of the work presented in this thesis, but the vast majority of subsequent work made use of the performance and flexibility provided by these two steps.

Gaussian process regression methods are adaptable regression tools that were able to accurately map complex patterns in degradation in Chapters 2 and 4. The models using the exponential kernel were found to be the most accurate predictors of capacity profiles but were demonstrably weaker when using pseudo-inputs for SparseGPR. Compared to battery literature, the models produced accurate capacity profiles values and represented a significant improvement for end-of-life forecasting. Knee point prediction was achieved using the predicted capacity profiles but with slightly lower accuracy relative to end-of-life forecasting performance.

5.1.2 Faster, more transparent modelling

Piecewise linear regression was found to be fast and interpretable. The approach was not as accurate as GPR when mapping the smaller, more detailed variations in the ageing datasets, but the end-of-life prediction performance was comparable. The resultant capacity models were simpler and could be stored at a considerably reduced cost relative to GPR. The curvature approach to splitting the input data was found to be insightful, fast and to produce accurate predictions.

5.1.3 Quantifying uncertainty

The proposed method for quantifying cell-to-cell variability returned the expected result that more complex models (i.e. models having more parameters) would require data from more cells, but only a limited supply of datasets was used. The number of cells, n_c, required to fit the parameter distributions for empirical capacity-time models was found to be ≈ 12 which is higher than the result in previously published research [3]. The difference was attributed to the higher upper bound on the population-level variance parameter used here. Multi-level Bayes was an effective modelling technique and the parameter distributions were found to be representative.

The proposed performance metric for quantifying the accuracy of predictive distributions was RMSE-Freq, which is a measure of the frequency at which measured capacity values are below predicted capacity values. This was adapted from a known optimisation strategy to apply to lithium-ion battery SoH prognosis. RMSE-Freq demonstrated with the GPR models that the squared exponential kernel produced more accurate probabilistic forecasts than the exponential kernel for SparseGPR, and also that the predictive distributions from the piecewise linear model was comparable to both by this metric.

The SparseGPR approach with pseudo-inputs was found to reduce storage requirements for a GPR model by 98%, for a small cost in decreased RMSE-Freq accuracy. The reduction in storage requirements suggests that the pseudo-input approach could be used to apply a flexible machine learning model in the real world at low cost.

5.2 Limitations and future work

5.2.1 Feature engineering

All of the work in this thesis assumed knowledge of intended battery use. This assumption is weak in more general applications. The prediction of future battery use is beyond the scope of this thesis but attempting to forecast the input features in Chapter 2 would be an interesting avenue of study.

Future research could investigate whether more detailed input features would improve performance and add flexibility. For example, including semi-empirical models would increase the number of potential input features and improve the physical justification behind the automated approach. Specific attention could then be given to choosing $\rho_{P,\max}$, i.e. what value is required to ensure that similar models are avoided. Alternatively, there are simple input features available that were omitted here, such as time spent at very small or zero current, that could be effective predictors of degradation.

A remaining problem with all these methods is the reliance on supervised learning because battery health measurements are not always available. Estimating capacity or internal resistance is beyond the scope of this work, but the possibility of using variables such as $V_{2,3}$ as indirect health measures could be researched. While $V_{2,3}$ was found to produce estimates of knee onset in Chapter 3, other simple features could provide equally informative details.

The splitting mechanism presented in Chapter 3 aimed to use the feature with the least noise across the dataset. For this, methods such as principle component analysis (PCA) could be investigated in future to split the input, with the rest of the procedure remaining identical. Using PCA to produce input features was considered as part of the algorithms in both Chapters 2 and 3 but was rejected because there was an insufficient performance improvement to justify the increased complexity and there was a risk of producing overly complex inputs. Nonetheless, PCA might produce inputs features with reduced noise that could be used as the splitting variable at the cost of reduced interpretability. Splitting variables with reduced noise could also reduce the impact of datasets with few capacity measurements, such as Sauer-2021.

5.2.2 Health diagnosis and prognosis

All of the models here assumed adequate access to battery test data. However, the checking of robustness of the approach in Chapter 2 had limited ability to replicate the challenges of real world battery use. Extrapolation beyond available data is a further challenge. The models here could be weak if a test cell is used in a significantly different manner to the training set.

Future work should investigate how forecasting battery SoH can be combined with battery control (for example, of charging protocols). A prediction has value in itself, but there is far more value if the trained model can be incorporated into a control structure. Simpler models, such as PLR with $\beta_{\text{improv}} = 0.2$ would probably be of most

use, but attempting to introduce the pseudo-input approach into a control algorithm could be beneficial.

5.2.3 Capturing uncertainty

The predictive performance of using pseudo-inputs with SparseGPR suggested that $M > 50$ was sufficient for accurate and trustworthy capacity prognosis, where M is the number of pseudo-inputs. The selection of pseudo-input points using K-means could be significantly improved. In its current form, the approach requires M to be sufficiently big to capture the rarer behaviours in the training dataset. However that inevitably led to many pseudo-inputs being in the densely populated areas, therefore some pseudo-inputs may have been superfluous.

All methods presented here would benefit from access to larger and more varied datasets. But the methods focussed on quantifying or capturing uncertainty would benefit most. Equally, investigating transfer learning methods to extrapolate the regression models from one dataset to another would be valuable.

Forecasting battery usage was alluded to at various points in this work and remains an open question. Ultimately, accuracy of the models of the future ageing of a given cell or battery pack will be a function of the accuracy of the future use profile. Comparing the relative impacts of uncertainty of use versus uncertainty of degradation would be an extremely interesting avenue of study.

All models in this thesis predicted capacity profiles, and so the uncertainty was in units of capacity. Real-world interventions, whether for cost or safety, require actions

which must be planned and implemented. Consequently, forecasting capacity profiles that can reliably be converted into estimates of remaining useful life (RUL) would be extremely valuable. Verifying the performance of predictive distributions for RUL is extremely difficult because cells are typically removed from testing once they have met some end-of-life criterion. There are many cases where forecasted profiles must extrapolate based on no usage data. Here, that was done with linear regression where required, so this is a clear area for improvement.

References

[1] Samuel Greenbank and David Howey. Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life. *IEEE Transactions on Industrial Informatics*, pages 1–1, 8 2021.

[2] Samuel Greenbank and David A. Howey. Piecewise-linear modelling with feature selection for Li-ion battery end of life prognosis. 4 2021.

[3] Philipp Dechent, Samuel Greenbank, Felix Hildenbrand, Saad Jbabdi, Dirk Uwe Sauer, and David A. Howey. Estimation of Li-ion degradation test sample sizes required to understand cell-to-cell variability. *Batteries & Supercaps*, 7 2021.

[4] Aashutosh Mistry, Ankit Verma, Shashank Sripad, Rebecca E. Ciez, Valentin Sulzer, Ferran Brosa Planella, Robert Timms, Yumin Zhang, Rachel Kurchin, Philipp Dechent, Weihan Li, Samuel Greenbank, Zeeshan Ahmad, Dilip Krishnamurthy, Jr. Alexis Fenton, Kevin Tenny, Prehit Patel, Daniel Juarez Robles, Paul Gasper, Andrew Colclasure, Artem Baskin, Corinne Scown, Venkat Subramanian, Edwin Khoo, Srikanth Allu, David Howey, Steven DeCaluwe, Scott Roberts, and Venkatasubramanian Viswanathan. A minimal information set to enable verifiable theoretical battery research. *ACS Energy Letters*, 6:3831–3835, 2021.

[5] Peter M. Attia, Alexander Bills, Ferran Brosa Planella, Philipp Dechent, Goncalo dos Reis, Matthieu Dubarry, Paul Gasper, Richard Gilchrist, Samuel Greenbank, David Howey, Ouyang Liu, Edwin Khoo, Yuliya Preger, Abhishek Soni, Shashank Sripad, Anna G. Stefanopoulou, and Valentin Sulzer. Knees in lithium-ion battery aging trajectories. 2021.

[6] O. Ramstrom. The Nobel Prize in Chemistry 2019: Advanced Science Background, 2019.

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

[8] Pankaj Arora, Marc Doyle, and Ralph E White. Mathematical modeling of the lithium deposition overcharge reaction in lithium-ion batteries using carbon-based negative electrodes. *Journal of The Electrochemical Society*, 146:3543–3553, 1999.

[9] S Park, A Savvides, and M B Srivastava. Battery capacity measurement and analysis using lithium coin cell battery. In *Proceedings of the International Symposium on Low Power Electronics and Design, Digest of Technical Papers*, pages 382–387, 2001.

[10] Seyed Mohammad Rezvanizaniani, Zongchang Liu, Yan Chen, and Jay Lee. Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle safety and mobility. *Journal of Power Sources*, 256:110–124, 2014.

[11] Yinjiao Xing, Qiang Miao, K. L. Tsui, and Michael Pecht. Prognostics and health monitoring for lithium-ion battery. In *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics, ISI 2011*, pages 242–247, 2011.

[12] Kai Liu, Yayuan Liu, Dingchang Lin, Allen Pei, and Yi Cui. Materials for lithium-ion battery safety. *Science Advances*, 4, 2018.

[13] Yi Li, Kailong Liu, Aoife M. Foley, Alana Zülke, Maitane Berecibar, Elise Nanini-Maury, Joeri Van Mierlo, and Harry E. Hoster. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. *Renewable and Sustainable Energy Reviews*, 113, 2019.

[14] Gregory L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs - Part 1. Background. *Journal of Power Sources*, 134:252–261, 8 2004.

[15] Jorn M. Reniers, Grietus Mulder, and David A. Howey. Review and performance comparison of mechanical-chemical degradation models for lithium-ion batteries. *Journal of The Electrochemical Society*, 166:A3189–A3200, 2019.

[16] Thorsten Baumhöfer, Manuel Brühl, Susanne Rothgang, and Dirk Uwe Sauer. Production caused variation in capacity aging trend and correlation to initial cell performance. *Journal of Power Sources*, 247:332–338, 2014.

[17] Markus Schindler, Johannes Sturm, Sebastian Ludwig, Julius Schmitt, and Andreas Jossen. Evolution of initial cell-to-cell variations during a three-year production cycle. *eTransportation*, page 100102, 1 2021.

[18] Stephen J. Harris, David J. Harris, and Chen Li. Failure statistics for commercial lithium ion batteries: A study of 24 pouch cells. *Journal of Power Sources*, 342:589–597, 2017.

[19] Xianke Lin, Jonghyun Park, Lin Liu, Yoonkoo Lee, A. M. Sastry, and Wei Lu. A comprehensive capacity fade model and analysis for Li-ion batteries. *Journal of The Electrochemical Society*, 160:A1701–A1710, 2013.

[20] R Spotnitz. Simulation of capacity fade in lithium-ion batteries. *Journal of Power Sources*, 113:72–80, 2003.

[21] Qi Zhang and Ralph E. White. Capacity fade analysis of a lithium ion cell. *Journal of Power Sources*, 179:793–798, 5 2008.

[22] 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, and R. D. Braatz. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4:383–391, 2019.

[23] Xiao Guang Yang, Yongjun Leng, Guangsheng Zhang, Shanhai Ge, and Chao Yang Wang. Modeling of lithium plating induced aging of lithium-ion batteries: Transition from linear to nonlinear aging. *Journal of Power Sources*, 360:28–40, 2017.

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

[25] Tobias Bach, Simon F. Schuster, Elena Fleder, Jana Müller, Martin Brand, Andreas Jossen, and Gerhard Sextl. Nonlinear aging characteristics of lithium-ion cells under different operational conditions. *Journal of Energy Storage*, 5:44–53, 2016.

[26] 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 A: Storage operation. *Journal of Energy Storage*, 30:101409, 8 2020.

[27] Peter Keil, Simon F. Schuster, Jörn Wilhelm, Julian Travi, Andreas Hauser, Ralph C. Karl, and Andreas Jossen. Calendar aging of lithium-ion batteries. *Journal of The Electrochemical Society*, 163:A1872–A1880, 2016.

[28] I. Bloom, B. W. Cole, J. J. Sohn, S. A. Jones, E. G. Polzin, V. S. Battaglia, G. L. Henriksen, C. Motloch, R. Richardson, T. Unkelhaeuser, D. Ingersoll, and H. L. Case. An accelerated calendar and cycle life study of Li-ion cells. *Journal of Power Sources*, 101:238–247, 2001.

[29] M Broussely, S Herreyre, P Biensan, P Kasztejna, K Nechev, and R J Staniewicz. Aging mechanism in Li-ion cells and calendar life predictions. *Journal of Power Sources*, 97:13–21, 2001.

[30] Eduardo Redondo-Iglesias, Pascal Venet, and Serge Pelissier. Eyring acceleration model for predicting calendar ageing of lithium-ion batteries. *Journal of Energy Storage*, 13:176–183, 2017.

[31] M. Schimpe, M. E. von Kuepach, M. Naumann, H. C. Hesse, K. Smith, and A. Jossen. Comprehensive modeling of temperature-dependent degradation mechanisms in lithium iron phosphate batteries. *Journal of The Electrochemical Society*, 165:A181–A193, 2018.

[32] Paul Gasper, Kevin Gering, Eric Dufek, and Kandler Smith. Challenging practices of algebraic battery life models through statistical validation and model identification via machine-learning. *Journal of the Electrochemical Society*, 168:020502, 2 2021.

[33] Brian Bole, Chetan Kulkarni, and Matthew Daigle. Randomized battery usage data set, 2014.

[34] Trishna Raj, Andrew A Wang, Charles W Monroe, and David A Howey. Investigation of path-dependent degradation in lithium-ion batteries. *Batteries & Supercaps*, 3:1377–1385, 2020.

[35] Thomas Waldmann, Björn Ingo Hogg, and Margret Wohlfahrt-Mehrens. Li plating as unwanted side reaction in commercial Li-ion cells – a review. *Journal of Power Sources*, 384:107–124, 4 2018.

[36] Saurabh Saxena, Myeongsu Kang, Yinjiao Xing, and Michael Pecht. Anomaly detection during lithium-ion battery qualification testing. In *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*. Institute of Electrical and Electronics Engineers Inc., 8 2018.

[37] Simon F. Schuster, Tobias C. Bach, Elena Fleder, Jana Müller, Martin J. Brand, Henning Lorrmann, Andreas Jossen, and Gerhard Sextl. Nonlinear aging of cylindrical lithium-ion cells linked to heterogeneous compression. *Journal of Energy Storage*, 5:212–223, 2 2016.

[38] Selcuk Atalay, Muhammad Sheikh, Alessandro Mariani, Yu Merla, Ed Bower, and W. Dhammika Widanage. Theory of battery ageing in a lithium-ion battery: Capacity fade, nonlinear ageing and lifetime prediction. *Journal of Power Sources*, 478:229026, 12 2020.

[39] Paula Fermín-Cueto, Euan McTurk, Michael Allerhand, Encarni Medina-Lopez, Miguel F. Anjos, Joel Sylvester, and Gonçalo dos Reis. Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells. *Energy and AI*, page 100006, 4 2020.

[40] Gregory L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs - Part 3. State and parameter estimation. *Journal of Power Sources*, 134:252–261, 2004.

[41] Weihan Li, Jiawei Zhang, Florian Ringbeck, Dominik Jöst, Lei Zhang, Zhongbao Wei, and Dirk Uwe Sauer. Physics-informed neural networks for electrode-level state estimation in lithium-ion batteries. *Journal of Power Sources*, 506:230034, 9 2021.

[42] Weihan Li, Neil Sengupta, Philipp Dechent, David Howey, Anuradha Annaswamy, and Dirk Uwe Sauer. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. *Journal of Power Sources*, 482:228863, 1 2021.

[43] Sheng Shen, Mohammadkazem Sadoughi, Xiangyi Chen, Mingyi Hong, and Chao Hu. A deep learning method for online capacity estimation of lithium-ion batteries. *Journal of Energy Storage*, 25, 2019.

[44] Piyush Tagade, Krishnan S. Hariharan, Sanoop Ramachandran, Ashish Khandelwal, Arunava Naha, Subramanya Mayya Kolake, and Seong Ho Han. Deep Gaussian process regression for lithium-ion battery health prognosis and degradation mode diagnosis. *Journal of Power Sources*, 445, 2020.

[45] Xiaosong Hu, Yunhong Che, Xianke Lin, and Simona Onori. Battery health prediction using fusion-based feature selection and machine learning. *IEEE Transactions on Transportation Electrification*, pages 1–1, 8 2020.

[46] M. Berecibar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, and P. Van Den Bossche. Critical review of state of health estimation methods of Li-ion batteries for real applications. *Renewable and Sustainable Energy Reviews*, 56:572–587, 2016.

[47] Yujie Wang, Jiaqiang Tian, Zhendong Sun, Li Wang, Ruilong Xu, Mince Li, and Zonghai Chen. A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. *Renewable and Sustainable Energy Reviews*, 131, 10 2020.

[48] Ming Feng Ge, Yiben Liu, Xingxing Jiang, and Jie Liu. A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. *Measurement: Journal of the International Measurement Confederation*, 174, 4 2021.

[49] Akash Basia, Zineb Simeu-Abazi, Eric Gascard, and Peggy Zwolinski. Review on state of health estimation methodologies for lithium-ion batteries in the context of circular economy. *CIRP Journal of Manufacturing Science and Technology*, 32:517–528, 1 2021.

[50] Huixing Meng and Yan Fu Li. A review on prognostics and health management methods of lithium-ion batteries. *Renewable and Sustainable Energy Reviews*, 116, 12 2019.

[51] Datong Liu, Jingyue Pang, Jianbao Zhou, Yu Peng, and Michael Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectronics Reliability*, 53:832–839, 2013.

[52] Guangzhong Dong, Zonghai Chen, Jingwen Wei, and Qiang Ling. Battery health prognosis using brownian motion modeling and particle filtering. *IEEE Transactions on Industrial Electronics*, 65:8646–8655, 11 2018.

[53] Jianbo Yu. State of health prediction of lithium-ion batteries: Multiscale logic regression and Gaussian process regression ensemble. *Reliability Engineering and System Safety*, 174:82–95, 6 2018.

[54] Roozbeh Razavi-Far, Shiladitya Chakrabarti, and Mehrdad Saif. Multi-step-ahead prediction techniques for lithium-ion batteries condition prognosis. In *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, pages 4675–4680. Institute of Electrical and Electronics Engineers Inc., 2 2017.

[55] Xiaosong Hu, Le Xu, Xianke Lin, and Michael Pecht. Battery lifetime prognostics. *Joule*, 4:310–346, 2 2020.

[56] C. Su and H. J. Chen. A review on prognostics approaches for remaining useful life of lithium-ion battery. In *IOP Conference Series: Earth and Environmental Science*, volume 93. Institute of Physics Publishing, 11 2017.

[57] Lifeng Wu, Xiaohui Fu, and Yong Guan. Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies. *Applied Sciences*, 6:166, 5 2016.

[58] Junxia Li, Miao Zhang, Hui Zheng, and Jing Jie. Battery remaining useful life prediction using improved mutated particle filter. *Energy Storage*, 12 2020.

[59] Robert R. Richardson, Michael A. Osborne, and David A. Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017.

[60] Jie Liu, Abhinav Saxena, Kai Goebel, Bhaskar Saha, and Wilson 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.

[61] Kai Goebel, Bhaskar Saha, Abhinav Saxena, Jose R Celaya, and Jon P Christophersen. Prognostics in battery health management. *IEEE Instrumentation & Measurement Magazine*, 11:33–40, 2008.

[62] Valentin Sulzer, Peyman Mohtat, Suhak Lee, Jason B. Siegel, and Anna G. Stefanopoulou. Promise and challenges of a data-driven approach for battery lifetime prognostics. 10 2020.

[63] Ye Wen, Rich Wolski, and Chandra Krintz. Online prediction of battery lifetime for embedded and mobile devices. *Third Power-Aware Computer Systems*, pages 57–72, 2004.

[64] Calum Strange and Gonçalo dos Reis. Prediction of future capacity and internal resistance of Li-ion cells from one cycle of input data. *Energy and AI*, 5:100097, 9 2021.

[65] Weiping Diao, Saurabh Saxena, Bongtae Han, and Michael Pecht. Algorithm to determine the knee point on capacity fade curves of lithium-ion cells. *Energies*, 12, 7 2019.

[66] Calum Strange, Shawn Li, Richard Gilchrist, and Gonçalo Dos Reis. Elbows of internal resistance rise curves in Li-ion cells. *Energies*, 14, 2 2021.

[67] Marcia L. Baptista, Elsa M.P. Henriques, and Kai Goebel. More effective prognostics with elbow point detection and deep learning. *Mechanical Systems and Signal Processing*, 146, 1 2020.

[68] T. R. Ashwin, A. Barai, K. Uddin, L. Somerville, A. McGordon, and J. Marco. Prediction of battery storage ageing and solid electrolyte interphase property estimation using an electrochemical model. *Journal of Power Sources*, 385:141–147, 2018.

[69] Christian Kupper, Björn Weißhar, Sascha Rißmann, and Wolfgang G. Bessler. End-of-life prediction of a lithium-ion battery cell based on mechanistic aging models of the graphite electrode. *Journal of The Electrochemical Society*, 165:A3468–A3480, 2018.

[70] C. Delacourt and M. Safari. Life simulation of a graphite/LiFePO 4 cell under cycling and storage. *Journal of The Electrochemical Society*, 159:A1283–A1291, 2012.

[71] Rutooj Deshpande, Mark Verbrugge, Yang-Tse Cheng, John Wang, and Ping Liu. Battery cycle life prediction with coupled chemical degradation and fatigue mechanics. *Journal of The Electrochemical Society*, 159:A1730–A1738, 2012.

[72] Justin Purewal, John Wang, Jason Graetz, Souren Soukiazian, Harshad Tataria, and Mark W. Verbrugge. Degradation of lithium ion batteries employing graphite negatives and nickel-cobalt-manganese oxide + spinel manganese oxide positives: Part 2, chemical-mechanical degradation model. *Journal of Power Sources*, 272:1154–1161, 12 2014.

[73] Hao Ge, Tetsuya Aoki, Nobuhisa Ikeda, Sohei Suga, Takuma Isobe, Zhe Li, Yuichiro Tabuchi, and Jianbo Zhang. Investigating lithium plating in lithium-ion batteries at low temperatures using electrochemical model with NMR assisted parameterization. *Journal of The Electrochemical Society*, 164:A1050–A1060, 2017.

[74] John Cannarella and Craig B. Arnold. The effects of defects on localized plating in lithium-ion batteries. *Journal of The Electrochemical Society*, 162:A1365–A1373, 2015.

[75] Seyed Saeed Madani, Erik Schaltz, and Søren Knudsen Kær. A review of different electric equivalent circuit models and parameter identification methods of lithium-ion batteries. *ECS Transactions*, 87:23–37, 11 2018.

[76] S. Nejad, D. T. Gladwin, and D. A. Stone. A systematic review of lumped-parameter equivalent circuit models for real-time estimation of lithium-ion battery states. *Journal of Power Sources*, 316:183–196, 6 2016.

[77] Gregory L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs - Part 2. Modeling and identification. *Journal of Power Sources*, 134:262–276, 8 2004.

[78] Bor Yann Liaw, Ganesan Nagasubramanian, Rudolph G. Jungst, and Daniel H. Doughty. Modeling of lithium ion cells - a simple equivalent-circuit model approach. In *Solid State Ionics*, volume 175, pages 835–839, 11 2004.

[79] Matthieu Dubarry and Bor Yann Liaw. Development of a universal modeling tool for rechargeable lithium batteries. *Journal of Power Sources*, 174:856–860, 12 2007.

[80] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth, and D. U. Sauer. Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. ii: Modelling. *Journal of Power Sources*, 196:5349–5356, 6 2011.

[81] Yang Li, Mahinda Vilathgamuwa, Troy Farrell, San Shing Choi, Ngoc Tham Tran, and Joseph Teague. A physics-based distributed-parameter equivalent circuit model for lithium-ion batteries. *Electrochimica Acta*, 299:451–469, 2019.

[82] Simo Särkka. *Bayesian Filtering and Smoothing*. Cambirdge University Press, 3 edition, 2013.

[83] P. Ramadass, Bala Haran, Parthasarathy M. Gomadam, Ralph White, and Branko N. Popov. Development of first principles capacity fade model for Li-ion cells. *Journal of The Electrochemical Society*, 151:A196, 2004.

[84] Madeleine Ecker, Jochen B. Gerschler, Jan Vogel, Stefan Käbitz, Friedrich Hust, Philipp Dechent, and Dirk Uwe Sauer. Development of a lifetime prediction model for lithium-ion batteries based on extended accelerated aging test data. *Journal of Power Sources*, 215:248–257, 2012.

[85] Peter M. Attia, William C. Chueh, and Stephen J. Harris. Revisiting the $t^{0.5}$ dependence of sei growth. *Journal of The Electrochemical Society*, 167:090535, 5 2020.

[86] Julius Schmitt, Arpit Maheshwari, Michael Heck, Stephan Lux, and Matthias Vetter. Impedance change and capacity fade of lithium nickel manganese cobalt oxide-based batteries during calendar aging. *Journal of Power Sources*, 353:183–194, 2017.

[87] Romain Mathieu, Issam Baghdadi, Olivier Briat, Philippe Gyan, and Jean Michel J.-M. Vinassa. D-optimal design of experiments applied to lithium battery for ageing model calibration. *Energy*, 141:2108–2119, 2017.

[88] Chao Hu, Hui Ye, Gaurav Jain, and Craig Schmidt. Remaining useful life assessment of lithium-ion batteries in implantable medical devices. *Journal of Power Sources*, 375:118–130, 1 2018.

[89] Yongquan Sun, Xueling Hao, Michael Pecht, and Yapeng Zhou. Remaining useful life prediction for lithium-ion batteries based on an integrated health indicator. *Microelectronics Reliability*, 88-90:1189–1194, 9 2018.

[90] Xiujuan Zheng and Huajing Fang. An integrated unscented Kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction. *Reliability Engineering and System Safety*, 144:74–82, 12 2015.

[91] Qiang Miao, Lei Xie, Hengjuan Cui, Wei Liang, and Michael Pecht. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*, 53:805–810, 2013.

[92] Chao Hu, Gaurav Jain, Prabhakar Tamirisa, and Tom Gorka. Method for estimating capacity and predicting remaining useful life of lithium-ion battery. In *2014 International Conference on Prognostics and Health Management, PHM 2014*. Institute of Electrical and Electronics Engineers Inc., 2 2015.

[93] Heng Zhang, Qiang Miao, Xin Zhang, and Zhiwen Liu. An improved unscented particle filter approach for lithium-ion battery remaining useful life prediction. *Microelectronics Reliability*, 81:288–298, 2 2018.

[94] Bhaskar Saha, Kai Goebel, and Jon Christophersen. Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, 31:293–308, 6 2009.

[95] Murilo Osorio Camargos, Iury Bessa, Luiz A. Q. Cordovil Junior, Pedro Henrique Silva Coutinho, Daniel Furtado Leite, and Reinaldo Martinez Palhares. Evolving fuzzy system applied to battery charge capacity prediction for fault prognostics, 2021.

[96] Nicolas Watrin, Benjamin Blunier, and Abdellatif Miraoui. Review of adaptive systems for lithium batteries state-of-charge and state-of-health estimation. In *2012 IEEE Transportation Electrification Conference and Expo*, pages 1–6, 2012.

[97] P. Singh and D. Reisner. Fuzzy logic-based state-of-health determination of lead acid batteries. In *24th Annual International Telecommunications Energy Conference*, pages 583–590, 2002.

[98] Plamen Angelov and Xiaowei Zhou. Evolving fuzzy systems from data streams in real-time. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 29–35, 2006.

[99] Nachaat Khayat and Nabil Karami. Adaptive techniques used for lifetime estimation of lithium-ion batteries. In *2016 Third International Conference on Electrical, Electronics, Computer Engineering and their Applications*, pages 98–103, 2016.

[100] M. Lucu, E. Martinez-Laserna, I. Gandiaga, and H. Camblong. A critical review on self-adaptive Li-ion battery ageing models. *Journal of Power Sources*, 109:138–159, 2018.

[101] Yapeng Zhou and Miaohua Huang. Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and arima model. *Microelectronics Reliability*, 65:265–273, 10 2016.

[102] Shuaiqi Shen, Song Ci, Kuan Zhang, and Xiaohui Liang. Lifecycle prediction of second use electric vehicle batteries based on arima model. In *2019 IEEE Globecom Workshops*, pages 1–6, 2019.

[103] Xiaodong Xu, Chuanqiang Yu, Shengjin Tang, Xiaoyan Sun, Xiaosheng Si, and Lifeng Wu. State-of-health estimation for lithium-ion batteries based on wiener process with modeling the relaxation effect. *IEEE Access*, 7:105186–105201, 2019.

[104] Shengjin Tang, Chuanqiang Yu, Xue Wang, Xiaosong Guo, and Xiaosheng Si. Remaining useful life prediction of lithium-ion batteries based on the wiener process with measurement error. *Energies*, 7(2):520–547, 2014.

[105] Zeyu Wu, Zili Wang, Cheng Qian, Bo Sun, Yi Ren, Qiang Feng, and Dezhen Yang. Online prognostication of remaining useful life for random discharge lithium-ion batteries using a gamma process model. In *2019 20th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems*, pages 1–6, 2019.

[106] Yu-Chang Lin and Kuan-Jung Chung. Lifetime prognosis of lithium-ion batteries through novel accelerated degradation measurements and a combined gamma process and Monte Carlo method. *Applied Sciences*, 9(3), 2019.

[107] Peter M. Attia, Aditya Grover, Norman Jin, Kristen A. Severson, Todor M. Markov, Yang-Hung Liao, Michael H. Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, Patrick K. Herring, Muratahan Aykol, Stephen J. Harris, Richard D. Braatz, Stefano Ermon, and William C. Chueh. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature*, 578:397–402, 2 2020.

[108] Wei He, Nicholas Williard, Michael Osterman, and Michael Pecht. Prognostics of lithium-ion batteries based on dempster-shafer theory and the bayesian monte carlo method. *Journal of Power Sources*, 196:10314–10321, 12 2011.

[109] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning*. Massachusetts Institute of Technology Press, 2006.

[110] C M Bishop. *Pattern Recognition and Machine Learning*. Springer Science, 8 edition, 2007.

[111] Kevin P Murphy. *Machine Learning: a probabilistic perspective*. MIT press, 2012.

[112] Zhongwei Deng, Xiaosong Hu, Xianke Lin, Le Xu, Yunhong Che, and Lin Hu. General discharge voltage information enabled health evaluation for lithium-ion batteries. *IEEE/ASME Transactions on Mechatronics*, pages 1–1, 11 2020.

[113] Rui Xiong, Yongzhi Zhang, Ju Wang, Hongwen He, Simin Peng, and Michael Pecht. Lithium-ion battery health prognosis based on a real battery management system used in electric vehicles. *IEEE Transactions on Vehicular Technology*, 68:4110–4121, 5 2019.

[114] Li Yang, Lingling Zhao, Xiaohong Su, and Shuai Wang. A lithium-ion battery RUL prognosis method using temperature changing rate. In *2016 IEEE International Conference on Prognostics and Health Management, ICPHM 2016*. Institute of Electrical and Electronics Engineers Inc., 8 2016.

[115] Pucheng Pei, Qibin Zhou, Lei Wu, Ziyao Wu, Jianfeng Hua, and Huimin Fan. Capacity estimation for lithium-ion battery using experimental feature interval approach. *Energy*, 203, 7 2020.

[116] Erik Schaltz, Daniel-Ioan Stroe, Kjeld Norregaard, Bjarne Johnsen, and Andreas Christensen. Partial charging method for lithium-ion battery state-of-health estimation. In *International Conference on Ecological Vehicles and Renewable Energies*, 2019.

[117] Zeyu Ma, Ruixin Yang, and Zhenpo Wang. A novel data-model fusion state-of-health estimation approach for lithium-ion batteries. *Applied Energy*, 237:836–847, 3 2019.

[118] Soren Byg Vilsen, Soren Knudsen Kaer, and Daniel Ioan Stroe. Log-linear model for predicting the lithium-ion battery age based on resistance extraction from dynamic aging profiles. *IEEE Transactions on Industry Applications*, 56:6937–6948, 11 2020.

[119] Jinsong Yu, Jie Yang, Yao Wu, Diyin Tang, and Jing Dai. Online state-of-health prediction of lithium-ion batteries with limited labeled data. *International Journal of Energy Research*, 44:11345–11360, 11 2020.

[120] Vladimir Vapnik, Steven E Golowich, and Alex Smola. Support vector method for function approximation, regression estimation, and signal processing. 1997.

[121] Michael E. Tipping. The relevance vector machine. In *NIPS*, 1999.

[122] Christopher M Bishop and Michael E Tipping. Variational relevance vector machines. In *Uncertainty in Artificial Intelligence Proceedings*, 2000.

[123] Adnan Nuhic, Tarik Terzimehic, Thomas Soczka-Guth, Michael Buchholz, and Klaus Dietmayer. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of Power Sources*, 239:680–688, 2013.

[124] Achmad Widodo, Min Chan Shim, Wahyu Caesarendra, and Bo Suk Yang. Intelligent prognostics for battery health monitoring based on sample entropy. *Expert Systems with Applications*, 38:11763–11769, 2011.

[125] Yuchen Song, Datong Liu, and Yu Peng. Data-driven on-line health assessment for lithium-ion battery with uncertainty presentation. In *2018 IEEE International Conference on Prognostics and Health Management*. Institute of Electrical and Electronics Engineers Inc., 8 2018.

[126] Xiaoli Qin, Qi Zhao, Hongbo Zhao, Wenquan Feng, and Xiumei Guan. Prognostics of remaining useful life for lithium-ion batteries based on a feature vector selection and relevance vector machine approach. In *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, pages 1–6. Institute of Electrical and Electronics Engineers Inc., 7 2017.

[127] Jianbao Zhou, Datong Liu, Yu Peng, and Xiyuan Peng. An optimized relevance vector machine with incremental learning strategy for lithium-ion battery remaining useful life estimation. In *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, pages 561–565, 2013.

[128] Yang Chang and Huajing Fang. A hybrid prognostic method for system degradation based on particle filter and relevance vector machine. *Reliability Engineering and System Safety*, 186:51–63, 6 2019.

[129] Yuchen Song, Datong Liu, Yandong Hou, Jinxiang Yu, and Yu Peng. Satellite lithium-ion battery remaining useful life estimation with an iterative updated RVM fused with the KF algorithm. *Chinese Journal of Aeronautics*, 31:31–40, 1 2018.

[130] Sheng Shen, Venkat Nemani, Jinqiang Liu, Chao Hu, and Zhaoyu Wang. A hybrid machine learning model for battery cycle life prediction with early cycle data. In *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2020.

[131] Hailin Feng and Dandan Song. A health indicator extraction based on surface temperature for lithium-ion batteries remaining useful life prediction. *Journal of Energy Storage*, 34:102118, 2 2021.

[132] Meru A. Patil, Piyush Tagade, Krishnan S. Hariharan, Subramanya M. Kolake, Taewon Song, Taejung Yeo, and Seokgwang Doo. A novel multistage support vector machine based approach for li ion battery remaining useful life estimation. *Applied Energy*, 159:285–297, 12 2015.

[133] Jingcai Du, Weige Zhang, Caiping Zhang, and Xingzhen Zhou. Battery remaining useful life prediction under coupling stress based on support vector regression. In *Energy Procedia*, volume 152, pages 538–543. Elsevier Ltd, 2018.

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

[135] Alexander Lamprecht, Moritz Riesterer, and Sebastian Steinhorst. Random forest regression of charge balancing data: A state of health estimation method for electric vehicle batteries. In *International Conference on Omni-layer Intelligent Systems*, 2020.

[136] Kodjo S.R. Mawonou, Akram Eddahech, Didier Dumur, Dominique Beauvois, and Emmanuel Godoy. State-of-health estimators coupled to a random forest approach for lithium-ion battery aging factor ranking. *Journal of Power Sources*, 484, 2021.

[137] Roozbeh Razavi-Far, Maryam Farajzadeh-Zanjani, Shiladitya Chakrabarti, and Mehrdad Saif. Data-driven prognostic techniques for estimation of the remaining useful life of lithium-ion batteries. In *2016 IEEE International Conference on Prognostics and Health Management, ICPHM 2016*. Institute of Electrical and Electronics Engineers Inc., 8 2016.

[138] Darius Roman, Saurabh Saxena, Valentin Robu, Michael Pecht, and David Flynn. Machine learning pipeline for battery state-of-health estimation. *Nature Machine Intelligence*, 4 2021.

[139] Luca Martino and Jesse Read. A joint introduction to Gaussian processes and relevance vector machines with connections to kalman filtering and other Kernel smoothers, 9 2020.

[140] Carl Edward Rasmussen and Joaquin Quiñonero-Candela. Healing the relevance vector machine through augmentation. In *International Conference on Machine Learning*, 2005.

[141] Jianshe Feng, Xiaodong Jia, Haoshu Cai, Feng Zhu, Xiang Li, and Jay Lee. Cross trajectory Gaussian process regression model for battery health prediction. *Journal of Modern Power Systems and Clean Energy*, 09 2020.

[142] Datong Liu, Jingyue Pang, Jianbao Zhou, and Yu Peng. Data-driven prognostics for lithium-ion battery based on Gaussian process regression. In *Proceedings of IEEE 2012 Prognostics and System Health Management Conference*, 2012.

[143] Duo Yang, Xu Zhang, Rui Pan, Yujie Wang, and Zonghai Chen. A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *Journal of Power Sources*, 384:387–395, 4 2018.

[144] Yi Jun He, Jia Ni Shen, Ji Fu Shen, and Zi Feng Ma. State of health estimation of lithium-ion batteries: A multiscale Gaussian process regression modeling approach. *AIChE Journal*, 61:1589–1600, 5 2015.

[145] Xiaoyu Li, Changgui Yuan, Xiaohui Li, and Zhenpo Wang. State of health estimation for Li-ion battery using incremental capacity analysis and Gaussian process regression. *Energy*, 190, 1 2020.

[146] Xueying Zheng and Xiaogang Deng. State-of-health prediction for lithium-ion batteries with multiple Gaussian process regression model. *IEEE Access*, 7:150383–150394, 2019.

[147] Antti Aitio and David Howey. Predicting battery end of life from solar off-grid system field data using machine learning. 2021.

[148] Kailong Liu, Yi Li, Xiaosong Hu, Mattin Lucu, and Widanalage Dhammika Widanage. Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, 16:3767–3777, 6 2020.

[149] 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*, 30:101410, 8 2020.

[150] Mattin Lucu, Markel Azkue, Haritza Camblong, and Egoitz Martinez-Laserna. Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data: Holistic validation with EV driving profiles. In *IEEE Energy Conversion Congress and Exposition*, pages 5600–5607. Institute of Electrical and Electronics Engineers (IEEE), 10 2020.

[151] Kailong Liu, Yunlong Shang, Quan Ouyang, and Widanalage Dhammika Widanage. A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery. *IEEE Transactions on Industrial Electronics*, 68:3170–3180, 4 2021.

[152] Robert R. Richardson, Michael A. Osborne, and David A. Howey. Battery health prediction under generalized conditions using a Gaussian process transition model. *Journal of Energy Storage*, 23:320–328, 2019.

[153] Lingling Li, Pengchong Wang, Kuei Hsiang Chao, Yatong Zhou, and Yang Xie. Remaining useful life prediction for lithium-ion batteries based on Gaussian processes mixture. *PLoS ONE*, 11, 9 2016.

[154] Jian Liu and Ziqiang Chen. Remaining useful life prediction of lithium-ion batteries based on health indicator and Gaussian process regression model. *IEEE Access*, 7:39474–39484, 2019.

[155] Lim Sze Li Harry, Pham Luu Trung Duong, and Nagarajan Raghavan. Exploration of multi-output Gaussian process regression for residual storage life prediction in lithium ion battery. In *Proceedings - 2020 Prognostics and Health Management Conference, PHM-Besancon 2020*, pages 263–269. Institute of Electrical and Electronics Engineers Inc., 5 2020.

[156] Yapeng Zhou, Miaohua Huang, Yupu Chen, and Ye Tao. A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. *Journal of Power Sources*, 321:1–10, 7 2016.

[157] Meng Li, Mohammadkazem Sadoughi, Sheng Shen, and Chao Hu. Remaining useful life prediction of lithium-ion batteries using multi-model Gaussian process. In *IEEE International Conference on Prognostics and Health Management*, pages 1–6, June 2019.

[158] Sergios Theodoridis. Chapter 13 - Bayesian learning: Approximate inference and nonparametric models. *Machine Learning*, pages 639 – 706, 2015.

[159] Bin Gou, Yan Xu, and Xue Feng. An ensemble learning-based data-driven method for online state-of-health estimation of lithium-ion batteries. *IEEE Transactions on Transportation Electrification*, pages 1–1, 10 2020.

[160] F. Cadini, C. Sbarufatti, F. Cancelliere, and M. Giglio. State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters. *Applied Energy*, 235:661–672, 2019.

[161] Ji Wu, Chenbin Zhang, and Zonghai Chen. An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. *Applied Energy*, 173:134–140, 2016.

[162] Shuzhi Zhang, Baoyu Zhai, Xu Guo, Kaike Wang, Nian Peng, and Xiongwen Zhang. Synchronous estimation of state of health and remaining useful lifetime for lithium-ion battery using the incremental capacity and artificial neural networks. *Journal of Energy Storage*, 26, 12 2019.

[163] Wei Liu and Yan Xu. Data-driven online health estimation of Li-ion batteries using a novel energy-based health indicator. *IEEE Transactions on Energy Conversion*, 35:1715–1718, 9 2020.

[164] Yitao Wu, Qiao Xue, Jiangwei Shen, Zhenzhen Lei, Zheng Chen, and Yonggang Liu. State of health estimation for lithium-ion batteries based on healthy features and long short-term memory. *IEEE Access*, 8:28533–28547, 2020.

[165] Yongzhi Zhang, Rui Xiong, Hongwen He, and Michael G. Pecht. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, 67:5695–5705, 7 2018.

[166] Joonki Hong, Dongheon Lee, Eui Rim Jeong, and Yung Yi. Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning. *Applied Energy*, 278, 11 2020.

[167] Guijun Ma, Yong Zhang, C. Cheng, Beitong Zhou, Pengchao Hu, and Ye Yuan. Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network. *Applied Energy*, 253:113626, 2019.

[168] Lyu Li, Yu Peng, Yuchen Song, and Datong Liu. Lithium-ion battery remaining useful life prognostics using data-driven deep learning algorithm. In *Prognostics and System Health Management Conference*, pages 1094–1100. Institute of Electrical and Electronics Engineers Inc., 1 2018.

[169] Kirandeep Kaur, Akhil Garg, Xujian Cui, Surinder Singh, and Bijaya Ketan Panigrahi. Deep learning networks for capacity estimation for monitoring SOH of Li-ion batteries for electric vehicles. *International Journal of Energy Research*, 2020.

[170] Shuangqi Li, Hongwen He, and Jianwei Li. Big data driven lithium-ion battery modeling method based on sdae-elm algorithm and data pre-processing technology. *Applied Energy*, 242:1259–1273, 5 2019.

[171] Mingye Zhu, Quan Ouyang, Yong Wan, and Zhisheng Wang. Remaining Useful Life Prediction of Lithium-Ion Batteries: A Hybrid Approach of Grey-Markov Chain Model and Improved Gaussian Process; Remaining Useful Life Prediction of Lithium-Ion Batteries: A Hybrid Approach of Grey-Markov Chain Model and Improved Gaussian Process. pages 2168–6777, 2021.

[172] M. F.H. Rani, Z. M. Razlan, A. B. Shahriman, Z. Ibrahim, and W. K. Wan. Comparative study of surface temperature of lithium-ion polymer cells at different discharging rates by infrared thermography and thermocouple. *International Journal of Heat and Mass Transfer*, 153, 6 2020.

[173] Zhonghua Yun and Wenhu Qin. Remaining useful life estimation of lithium-ion batteries based on optimal time series health indicator. *IEEE Access*, 8:55447–55461, 2020.

[174] Zengkai Wang, Shengkui Zeng, Jianbin Guo, and Taichun Qin. State of health estimation of lithium-ion batteries based on the constant voltage charging curve. *Energy*, 167:661–669, 1 2019.

[175] Xiaoyu Li, Zhenpo Wang, Lei Zhang, Changfu Zou, and David D. Dorrell. State-of-health estimation for Li-ion batteries by combing the incremental capacity analysis method with grey relational analysis. *Journal of Power Sources*, 2019.

[176] Xiaoyu Li, Zhenpo Wang, and Jinying Yan. Prognostic health condition for lithium battery using the partial incremental capacity and Gaussian process regression. *Journal of Power Sources*, 421:56–67, 2019.

[177] S. Khaleghi, Y. Firouz, J. Van Mierlo, and P. Van den Bossche. Developing a real-time data-driven battery health diagnosis method, using time and frequency domain condition indicators. *Applied Energy*, 255, 12 2019.

[178] Peter M Attia, Kristen A Severson, and Jeremy D Witmer. Statistical learning for accurate and interpretable battery lifetime prediction. *Journal of the Electrochemical Society*, 168:090547, 2021.

[179] C. Huang and L. Wang. Gaussian process regression-based modelling of lithium-ion battery temperature-dependent open-circuit-voltage. *Electronics Letters*, 53:1214–1216, 8 2017.

[180] Kailong Liu, Xiaosong Hu, Zhongbao Wei, Yi Li, and Yan Jiang. Modified Gaussian process regression models for cyclic capacity prediction of lithium-ion batteries. *IEEE Transactions on Transportation Electrification*, 5:1225–1236, 12 2019.

[181] Michael L. Stein. *Interpolation of Spatial Data*. Springer Series in Statistics. Springer, New York, NY, 1999.

[182] David J C Mackay. Bayesian non-linear modeling for the prediciton competition. *Maximum Entropy and Bayesian Methods*, pages 221–234, 1996.

[183] Yuan Qi, Thomas P Minka, Rosalind W Picard, and Zoubin Ghahramani. Predictive automatic relevance determination by expectation propagation. 2004.

[184] Jun Zhao, Long Chen, Witold Pedrycz, and Wei Wang. Variational inference-based automatic relevance determination kernel for embedded feature selection of noisy industrial data. *IEEE Transactions on Industrial Electronics*, 66:416–428, 1 2019.

[185] Kailong Liu, Zhongbao Wei, Zhile Yang, and Kang Li. Mass load prediction for lithium-ion battery electrode clean production: A machine learning approach. *Journal of Cleaner Production*, 289, 3 2021.

[186] Yunwei Zhang, Qiaochu Tang, Yao Zhang, Jiabin Wang, Ulrich Stimming, and Alpha A. Lee. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. *Nature Communications*, 11, 12 2020.

[187] Mikhail Belyaev, Evgeny Burnaev, and Yermek Kapushev. Exact inference for Gaussian process regression in case of big data with the cartesian product structure. 3 2014.

[188] Felix Leibfried, Vincent Dutordoir, ST John, and Nicolas Durrande. A tutorial on sparse gaussian processes and variational inference, 12 2020.

[189] Edward Snelson, Zoubin Ghahramani, and Carl Rasmussen. Warped Gaussian processes. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004.

[190] Michalis K Titsias. Variational learning of inducing variables in sparse Gaussian processes. 2009.

[191] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18, 2006.

[192] Michalis Titsias. Variational model selection for sparse Gaussian process regression. 2009.

[193] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[194] GPy. GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy, since 2012.

[195] Matthias Bauer, Mark Van Der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Neural Information Processing Systems*, 2016.

[196] Gozde Ozcan, Milutin Pajovic, Zafer Sahinoglu, Yebin Wang, Philip V. Orlik, and Toshihiro Wada. Online battery state-of-charge estimation based on sparse Gaussian process regression. In *IEEE Power and Energy Society General Meeting*, volume 2016-November. IEEE Computer Society, 11 2016.

[197] MathWorks: Statistics and Machine Learning Toolbox. fitrgp: Fit a Gaussian process regression (gpr) model, 2021.

[198] Brian Bole, Chetan S Kulkarni, and Matthew Daigle. Adaptation of an electrochemistry-based Li-ion battery model to account for deterioration observed under randomized use. In *Annual Conference of the Prognositcs and Health Management Society*, 2014.

[199] Google scholar reference search, 2021. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Adaptation+of+an+Electrochemistry-based+Li-Ion+Battery+Model+to+Account+for+Deterioration+Observed+Under+Randomized+Use&btnG=, last accessed 03/08/2021.

[200] Brian Bole, Chetan Kulkarni, and Matthew Daigle. Adaptation of an electrochemistry-based Li-ion battery model to account for deterioration observed under randomized use, 2021. https://www.researchgate.net/publication/266386006_Adaptation_of_an_Electrochemistry-based_Li-Ion_Battery_Model_to_Account_for_Deterioration_Observed_Under_Randomized_Use, last accessed 03/08/2021.

[201] Trishna Raj and David Howey, 2020. https://ora.ox.ac.uk/objects/uuid:de62b5d2-6154-426d-bcbb-30253ddb7d1e, last accessed 03/08/2021.

[202] Philipp Dechent, 2021. Personal communication.

[203] Philipp Dechent, Susanne Lehner, and Dirk Uwe Sauer, 2021. Time series data on cycle tests on Samsung inr18650-1511.

[204] Philipp Dechent, Lisa Willenberg, Marcel Eckert, and Dirk Uwe Sauer, 2021. Time series data on cycle tests on Samsung inr18650-35e.

[205] Weihan Li, Neil Sengupta, Philipp Dechent, David Howey, Anuradha Annaswamy, and Dirk Uwe Sauer. One-shot battery degradation trajectory prediction with deep learning. *Journal of Power Sources*, 2021.

[206] Dirk Uwe Sauer, 2021. https://publications.rwth-aachen.de/record/818642, last accessed 03/08/2021.

[207] Zhongwei Deng, Xiaosong Hu, Xianke Lin, Yunhong Che, Le Xu, and Wenchao Guo. Data-driven state of charge estimation for lithium-ion battery packs based on Gaussian process regression. *Energy*, 205, 8 2020.

[208] Lin Chen, Zhiqiang Lü, Weilong Lin, Junzi Li, and Haihong Pan. A new state-of-health estimation method for lithium-ion batteries through the intrinsic relationship between Ohmic internal resistance and capacity. *Measurement: Journal of the International Measurement Confederation*, 116:586–595, 2 2018.

[209] Yapeng Zhou, Miaohua Huang, Yupu Chen, and Ye Tao. A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. *Journal of Power Sources*, 321:1–10, 7 2016.

[210] Lin Chen, Huimin Wang, Bohao Liu, Yijue Wang, Yunhui Ding, and Haihong Pan. Battery state-of-health estimation based on a metabolic extreme learning machine combining degradation state model and error compensation. *Energy*, 215, 1 2021.

[211] Tingting Xu, Zhen Peng, and Lifeng Wu. A novel data-driven method for predicting the circulating capacity of lithium-ion battery under random variable current. *Energy*, 218, 3 2021.

[212] Yuanyuan Li, Daniel-Ioan Stroe, Yuhua Cheng, Hanmin Sheng, Xin Sui, and Remus Teodorescu. On the feature selection for battery state of health estimation based on charging–discharging profiles. *Journal of Energy Storage*, 33:102122, 1 2021.

[213] Philippe Leray and Patrick Gallinari. Feature selection with neural networks. *Behaviormetrika*, 26:145–166, 1999.

[214] Aileen Bahl, B. Hellack, Mihaela Balas, Anca Dinischiotu, Martin Wiemann, Joep Brinkmann, Andreas Luch, Bernhard Y. Renard, and Andrea Haase. Recursive feature elimination in random forest classification supports nanomaterial grouping. *NanoImpact*, 15, 3 2019.

[215] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27:659–678, 5 2017.

[216] Berit Floor Lund and Bjarne A. Foss. Parameter ranking by orthogonalization-applied to nonlinear mechanistic models. *Automatica*, 44:278–281, 1 2008.

[217] Daniel-Ioan Stroe, Maciej Swierczynski, Søren Knudsen Kær, and Remus Teodorescu. Degradation behavior of lithium-ion batteries during calendar ageing - the case of the internal resistance increase. In *IEEE Transactions on Industry Applications*, volume 54, pages 517–525. Institute of Electrical and Electronics Engineers Inc., 1 2018.

[218] Florian Ringbeck, Christiane Rahe, Georg Fuchs, and Dirk Uwe Sauer. Identification of lithium plating in lithium-ion batteries by electrical and optical methods. *Journal of The Electrochemical Society*, 167:090536, 5 2020.

[219] Nassim A. Samad, Youngki Kim, Jason B. Siegel, and Anna G. Stefanopoulou. Battery capacity fading estimation using a force-based incremental capacity analysis. *Journal of The Electrochemical Society*, 163:A1584–A1594, 5 2016.

[220] Alexander Bartlett, James Marcicki, Kevin Rhodes, and Giorgio Rizzoni. State of health estimation in composite electrode lithium-ion cells. *Journal of Power Sources*, 284:642–649, 2015.

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

[222] Xue Li, Jiuchun Jiang, Le Yi Wang, Dafen Chen, Yanru Zhang, and Caiping Zhang. A capacity model based on charging process for state of health estimation of lithium ion batteries. *Applied Energy*, 177:537–543, 9 2016.

[223] C. Pastor-Fernández, W.D. Widanage, G.H. Chouchelamane, and J. Marco. A soh diagnosis and prognosis method to identify and quantify degradation modes in Li-ion batteries using the IC/DV technique. In *6th Hybrid and Electric Vehicles Conference (HEVC 2016)*, pages 1–6, 2016.

[224] Yao Wu and Andreas Jossen. Entropy-induced temperature variation as a new indicator for state of health estimation of lithium-ion cells. *Electrochimica Acta*, 276:370–376, 6 2018.

[225] Lisa K. Willenberg, Philipp Dechent, Georg Fuchs, Dirk Uwe Sauer, and Egbert Figgemeier. High-precision monitoring of volume change of commercial lithium-ion batteries by using strain gauges. *Sustainability (Switzerland)*, 12, 1 2020.

[226] John Cannarella and Craig B. Arnold. State of health and charge measurements in lithium-ion batteries using mechanical stress. *Journal of Power Sources*, 269:7–14, 12 2014.

[227] B Saha, S Poll, K Goebel, and J Christophersen. An integrated approach to battery health monitoring using bayesian regression and state estimation. In *2007 IEEE Autotestcon*, pages 646–653, 9 2007.

[228] Christian Fleischer, Wladislaw Waag, Hans Martin Heyn, and Dirk Uwe 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. *Journal of Power Sources*, 260:276–291, 8 2014.

[229] Yonghua Li, R. Dyche Anderson, Jing Song, Anthony M. Phillips, and Xu Wang. A nonlinear adaptive observer approach for state of charge estimation of lithium-ion batteries. In *Proceedings of the American Control Conference*, pages 370–375, 2011.

[230] Peyman Mohtat, Suhak Lee, Jason B. Siegel, and Anna G. Stefanopoulou. Towards better estimability of electrode-specific state of health: Decoding the cell expansion. *Journal of Power Sources*, 427:101–111, 2019.

[231] Zheng Xin Zhang, Xiao Sheng Si, Chang Hua Hu, and Michael G. Pecht. A prognostic model for stochastic degrading systems with state recovery: Application to Li-ion batteries. *IEEE Transactions on Reliability*, 66:1293–1308, 12 2017.

[232] Dong Wang, Jin Zhen Kong, Yang Zhao, and Kwok Leung Tsui. Piecewise model based intelligent prognostics for state of health prediction of rechargeable batteries with capacity regeneration phenomena. *Measurement: Journal of the International Measurement Confederation*, 147, 12 2019.

[233] Karkulali Pugalenthi, Hyunseok Park, and Nagarajan Raghavan. Piecewise model-based online prognosis of lithium-ion batteries using particle filters. *IEEE Access*, 8:153508–153516, 2020.

[234] Bolun Xu, Jinye Zhao, Tongxin Zheng, Eugene Litvinov, and Daniel S. Kirschen. Factoring the cycle aging cost of batteries participating in electricity markets. *IEEE Transactions on Power Systems*, 33:2248–2259, 3 2018.

[235] Ville Satopää, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *International Conference on Distributed Computing Systems*, pages 166–171, 2011.

[236] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Science of India*. National Institute of Science of India, 1936.

[237] Igor Melnykov and Volodymyr Melnykov. On k-means algorithm with the use of mahalanobis distances. *Statistics Probability Letters*, 84:88–95, 2014.

[238] Georg Angenendt, Sebastian Zurmühlen, Ramin Mir-Montazerri, Dirk Magnor, and Dirk Uwe Sauer. Enhancing battery lifetime in PV battery home storage system using forecast based operating stategies. In *International Renewable Energy Storage Conference*, pages 80–88, 2016.

[239] Maciej Swierczynski, Daniel Ioan Stroe, Ana Irina Stan, and Remus Teodorescu. Lifetime and economic analyses of lithium-ion batteries for balancing wind power forecast error. *International Journal of Energy Research*, 39:760–770, 5 2015.

[240] Katharina Rumpf, Maik Naumann, and Andreas Jossen. Experimental investigation of parametric cell-to-cell variation and correlation based on 1100 commercial lithium-ion cells. *Journal of Energy Storage*, 14:224–243, 2017.

[241] Xinhua Liu, Weilong Ai, Max Naylor Marlow, Yatish Patel, and Billy Wu. The effect of cell-to-cell variations and thermal gradients on the performance and degradation of lithium-ion battery packs. *Applied Energy*, 248:489–499, 2019.

[242] Sebastian Paul, Christian Diegelmann, Herbert Kabza, and Werner Tillmetz. Analysis of ageing inhomogeneities in lithium-ion battery systems. *Journal of Power Sources*, 239:642–650, 2013.

[243] Zhongbao Wei, Difan Zhao, Hongwen He, Wanke Cao, and Guangzhong Dong. A noise-tolerant model parameterization method for lithium-ion battery management system. *Applied Energy*, 268, 6 2020.

[244] Seongjun Lee, Jonghoon Kim, Jaemoon Lee, and B. H. Cho. State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge. *Journal of Power Sources*, 185:1367–1373, 12 2008.

[245] Paul Goldberg, Christopher Williams, and Christopher Bishop. Regression with input-dependent noise: A Gaussian process treatment. *Advances in Neural Information Processing Systems*, 10, 02 1998.

[246] Dong Wang, Fangfang Yang, Yang Zhao, and Kwok Leung Tsui. Prognostics of lithium-ion batteries based on state space modeling with heterogeneous noise variances. *Microelectronics Reliability*, 75:1–8, 8 2017.

[247] Simone Orcioni, Adriana Ricci, Luca Buccolini, Cristiano Scavongelli, and Massimo Conti. Effects of variability of the characteristics of single cell on the performance of a lithium-ion battery pack. In *Workshop on Intelligent Solutions in Embedded Systems*, pages 15–21. Institute of Electrical and Electronics Engineers Inc., 2017.

[248] Michael Baumann, Leo Wildfeuer, Stephan Rohr, and Markus Lienkamp. Parameter variations within Li-ion battery packs – theoretical investigations and experimental quantification. *Journal of Energy Storage*, 18:295–307, 2018.

[249] Susanne Rothgang, Thorsten Baumhöfer, and Dirk Uwe Sauer. Diversion of aging of battery cells in automotive systems. In *2014 IEEE Vehicle Power and Propulsion Conference, VPPC 2014*. Institute of Electrical and Electronics Engineers Inc., 2014.

[250] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953.

[251] W K Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[252] Douglas J. Altman and J. Martin Bland. Standard deviations and standard errors. *British Medical Journal*, 331:903, 2005.

[253] Glenn Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

Appendix A

The models behind the figures and tables

This appendix contains information on all models used to produce predictive plots. Figures not produced from a model are not included. RPTP stands for repeats per test point.

Description	**Value**
Tool	Gaussian process regression
Kernel	Matérn-3/2 with ARD
Dataset	NASA-2014 [33]
Test cell	2
Training cells	1, 7, 8
Input variables	(left) time [days], charge throughput [Ah] (middle) $t_{I>3\mathrm{A}}$ [days], $t_{T>20\,^{\circ}\mathrm{C}}$ [days] (right) time [days], charge throughput [Ah], $t_{I>3\mathrm{A}}$ [days], $t_{T>20\,^{\circ}\mathrm{C}}$ [days]
Target variable	$\frac{dQ}{dt}$ [%.days^{-1}]

Table A.1 Model details for Fig. 1.2

Description	Value
Tool	Gaussian process regression
Kernel	radial basis function with ARD
Dataset	NASA-2014 [33]
Test cell	2
Training cells	(left) 1, 7, 8
	(middle) 17, 18, 19
	(right) 1, 7, 8, 17, 18, 19
Input variables	calendar time [days]
	cumulative charge throughput [Ah]
Target variable	capacity [%]

Table A.2 Model details for Fig. 1.3

Description	Value
Kernel functions	exponential (yellow)
	Matérn-3/2 (purple)
	Matérn-5/2 (green)
	radial basis function (blue)
σ_l	1
σ_f	1
Tool	Matlab's *mvnrnd*, zero mean

Table A.3 Model details for Fig. 1.4

Description	Value
Tool	Gaussian process regression
Kernel	(left) radial basis function with ARD
	(right) radial basis function without ARD
Dataset	NASA-2014 [33]
Test cell	2
Training cells	1, 7, 8
Input variables	normalised calendar time
	normalised cumulative charge throughput
	(blue) noise $\times\frac{1}{15}$
	(yellow) noise $\times 15$
Target variable	capacity [%]

Table A.4 Model details for Fig. 1.5

Description	**Value**
Tool	Gaussian process regression
Kernel	exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	100 randomly selected cells
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	number of bins created by thresholds
RPTP	10

Table A.5 Model details for Fig. 2.3

Number of bins	Percentiles used
1	1.0, 99.0
2	1.0, 50.0, 99.0
3	1.0, 33.7, 66.3 99.0
4	1.0, 25.5, 50.0, 74.5, 99.0
5	1.0, 20.6, 40.2, 59.8, 79.4, 99.0
6	1.0, 17.3, 33.7, 50.0, 66.3, 82.7, 99.0
7	1.0, 15.0, 29.0, 43.0, 57.0, 71.0, 85.0, 99.0
8	1.0, 13.3, 25.5, 37.8, 50.0, 62.3, 74.5, 86.8, 99.0
9	1.0, 11.9, 22.8, 33.7, 44.6, 55.4, 66.3, 77.2, 88.1, 99.0
10	1.0, 10.8, 20.6, 30.4, 40.2, 50.0, 59.8, 69.6, 79.4, 89.2, 99.0
Limit	For speed, no features covering $< 20\%$ were generated i.e. at least two bins are combined above 4 bins

Table A.6 Percentiles used in Fig. 2.3

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
	10 for time performance plot
Input variables	automated feature generation (Section 2.1) over full dataset
	automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	RPT frequency
RPTP	20

Table A.7 Model details for Fig. 2.4

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
Input variables	automated feature generation (Section 2.1) over full dataset
	automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	maximum shared correlation, $\rho_{P,\max}$
RPTP	20

Table A.8 Model details for Fig. 2.6

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	number of input features
RPTP	20

Table A.9 Model details for Fig. 2.7

Description	Value
Tool	Gaussian process regression
Kernel	(respectively, by row) radial basis function with ARD Matérn-5/2 with ARD Matérn-3/2 with ARD exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
RPTP	20 for each kernel function
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Target variable	ΔQ [%]

Table A.10 Model details for Table 2.3

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD (green, where applicable)
	RBF with ARD (blue, only in Fig. 2.13)
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	100 randomly selected cells
Input variables	automated feature generation (Section 2.1)
	automated feature selection (Section 2.2)
Target variable	ΔQ [%]
RPTP	100 (where applicable)

Table A.11 Model details for Figs. 2.10, 2.12, 2.13 & 2.17

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
Input variables	automated feature generation (Section 2.1) over full dataset
	automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	noise added to raw data, σ_R
RPTP	20

Table A.12 Model details for Fig. 2.14

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
Input variables	automated feature generation (Section 2.1) over full dataset
	automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	frequency of raw data selection
RPTP	20

Table A.13 Model details for Fig. 2.15

Description	Value
Tool	Gaussian process regression
Kernel	Exponential with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Test cells	All cells not selected in training set
Training cells	50 randomly selected cells
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Target variable	ΔQ [%]
Control variable	number of available raw data values per variable
RPTP	20

Table A.14 Model details for Fig. 2.16

Description	Value
Dataset	Severson-2019 [22] and Attia-2020 [107]
Training cells	50 randomly selected cells
Testing cells	All cells not selected in training set
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Target variable	ΔQ [%]
β_L	$\frac{1}{10}$
σ_w	10
$\rho_{P,\max}$	0.85
β_{improv}	0.01
Splitting technique	curvature
Maximum models	10
Input features	5
RPTP	20

Table A.15 Default controls for piecewise linear regression models, used throughout unless specified.

Description	Value
Tool	Piecewise linear regression (purple) GPR-EXP (green)
Control variable	$\rho_{P,\max}$
Defaults	Table A.15

Table A.16 Model details for Fig. 3.3

Description	Value
Tool	Piecewise linear regression (purple) GPR-EXP (green)
Control variable	number of input features
Defaults	Table A.15

Table A.17 Model details for Fig. 3.4

Description	Value
Tool	piecewise linear regression (purple) GPR-EXP (green) GPR-RBF (blue)
RPTP	100
Defaults	Table A.15

Table A.18 Model details for Figs. 3.7 and 3.11, and Table 3.3.

Description	Value
Tool	piecewise linear regression (purple)
Defaults	Table A.15
RPTP	1,000

Table A.19 Model details for Fig. 3.8

Description	Value
Dataset	Sauer-2021 [206]
Tool	piecewise linear regression (purple) GPR-RBF (blue)
Defaults	Table A.15
RPTP	20

Table A.20 Model details for Fig. **??**

Description	Value
Dataset	Severson-2019 [22] and Attia-2020 [107]
Tool	piecewise linear regression
Defaults	Table A.15
Control variables	β_{improv} max(n_m) σ_w β_L
RPTP	20

Table A.21 Model details for Fig. 3.9

Description	Value
Tool	PLR, split by curvature (purple) PLR, split by *fminsearch* (blue) PLR, split by K-means (yellow) PLR, split evenly (grey) PLR, split randomly (red) GPR-EXP (green)
Control variables	number of input features
Defaults	Table A.15

Table A.22 Model details for Fig. 3.10

Splitting mechanism	Description
Curvature (purple)	As per Section 3.2
fminsearch (blue)	Use Matlab's *fminsearch* to optimise position of breakpoints with RMSE ΔQ as the loss function
K-means (yellow)	Use K-means across the first two input features.
Even (grey)	Evenly spaced breakpoints
Random (red)	Random placing of the breakpoints

Table A.23 Description of alternative splitting mechanisms used in Fig. 3.10

Description	**Value**
Datasets	Dechent-2017 [203] Dechent-2020 [204] Sauer-2021 [206] Severson-2019 [22] Attia-2020 [107]
Training cells	3 to full sample minus 3
Tool	Multi-level Bayes
Likelihood	Gaussian
μ_p prior	$\mathrm{N}(0, 10^4)$
σ_p^2 prior	$\mathrm{U}(0, \infty)$
Bounds	(lower, upper)
c_1	mean: (-10, 10), variance: (0, 100^2)
c_2	mean: (-10, 10), variance: (0, 100^2)
B_2	mean: (90, 110), variance: (0, 100^2)
c_3	mean: (-10, 10), variance: (0, 100^2)
t_{knee}	mean: (0, 100), variance: (0, 100^2)
τ_D	mean: (-10, 10), variance: (0, 100^2)
Points per cell	15
RPTP	1,000

Table A.24 Model details for Figs. 4.4, 4.5, 4.6, 4.7 & 4.8, and Table 4.2

Description	**Value**
Tool	Gaussian process regression
Kernel	radial basis function with ARD
Dataset	Severson-2019 [22] and Attia-2020 [107]
Training cells	1 to 100 randomly selected cells
Testing cells	57 randomly selected cells
Input variables	$V_{2,3}$, $V_{1,2}$, time [days], $\lvert P\rvert_{1,2}$
Target variable	ΔQ [%]
RPTP	100

Table A.25 Model details for Fig. 4.9

Description	Value
Tool	piecewise linear regression (purple) GPR-EXP (green) GPR-RBF (blue)
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Training cells	50 randomly selected cells
Testing cells	All cells not selected in training set
Target variable	ΔQ [%]
PLR Defaults	Table A.15
RPTP	100

Table A.26 Model details for Fig. 4.12

Description	Value
Tool	SparseGPR-RBF (yellow) SparseGPR-EXP (red) GPR-RBF (blue)
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Training cells	50 randomly selected cells
Testing cells	All cells not selected in training set
Target variable	ΔQ [%]
Control Variable	Pseudo-input points, K
RPTP	100

Table A.27 Model details for Figs. 4.14 & 4.16

Description	Value
Tool	SparseGPR-RBF(Z) (yellow) SparseGPR-RBF(X) (blue)
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Training cells	50 randomly selected cells
Testing cells	All cells not selected in training set
Target variable	ΔQ [%]
Control Variable	Pseudo-input points, K
RPTP	100

Table A.28 Model details for Fig. 4.15

Description	**Value**
Tool	Gaussian process regression
Kernel	exponential with ARD (green) radial basis function with ARD (blue)
Dataset	Severson-2019 [22] and Attia-2020 [107]
Training cells	50
Input variables	automated feature generation (Section 2.1) over full dataset automated feature selection (Section 2.2)
Target variable	ΔQ [%]
K-means, K	20
RPTP	100

Table A.29 Model details for Fig. 4.17

Appendix B

Extra trials - Chapter 2

This appendix contains the trials referenced but not presented in Chapter 2.

B.1 Prognosis on full dataset

The main trial in Chapter 2 used datasets Severson-2019 [22] and Attia-2020 [107] but only cells with lifetimes between 15 and 40 days. The results in this section are for the same procedure as the main trial in Chapter 2, but running on all available cells in the two datasets. That amounted to 175 cells, an increase from the 157 used in Chapter 2.

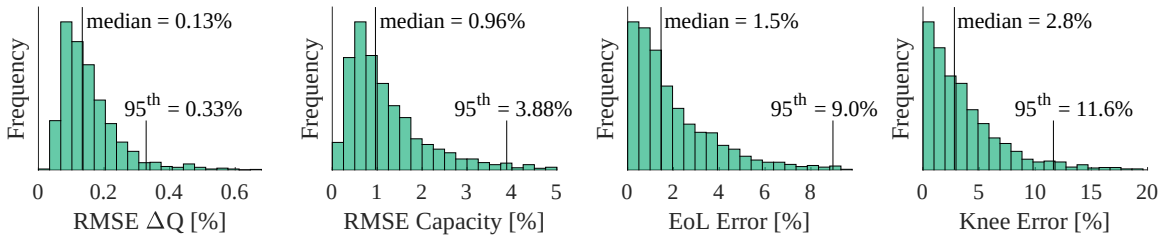


Fig. B.1 Predictive performance of automated approach in Chapter 2 on all cells in Severson-2019 and Attia-2020.

B.2 State of Health Estimation Trial

The title of this thesis includes *"diagnosis"*. This section looks explicitly at state of health (SoH) estimation using the input features and modelling techniques presented in this thesis.

Two methods of SoH estimation were tested. First by using the input features produced in Chapter 2 and mapping directly to capacity Q. The second model was constructed similarly to the first model, but the input features were the proportion of time spent in each region between time $t = 0$ and time $t = t$. This method was referred to as using cumulative features and labelled $Q(F)$ whereas the first is using instantaneous features and labelled $Q(f)$. Lower case f symbolised instantaneous features while F indicated cumulative versions. All input features were calculated using the thresholds in Table 2.1.

Both approaches were compared to the techniques from Chapter 2, labelled $\Delta Q(f)$ here because instantaneous input features are used to map to ΔQ. The regression tool selected for all models was Gaussian process regression with an exponential kernel (GPR-EXP). Only RMSE Capacity was used as a performance metric. Each technique was performed with 100 repeats with 50 randomly selected training cells and 107 test cells.

The estimation results from using cumulative features are poor with a median of 1.7% capacity. The 95^{th} percentiles were similar between using cumulative and instantaneous features, but there is a loosely applied upper limit to the the RMSE Capacity set by the range of the test target data.

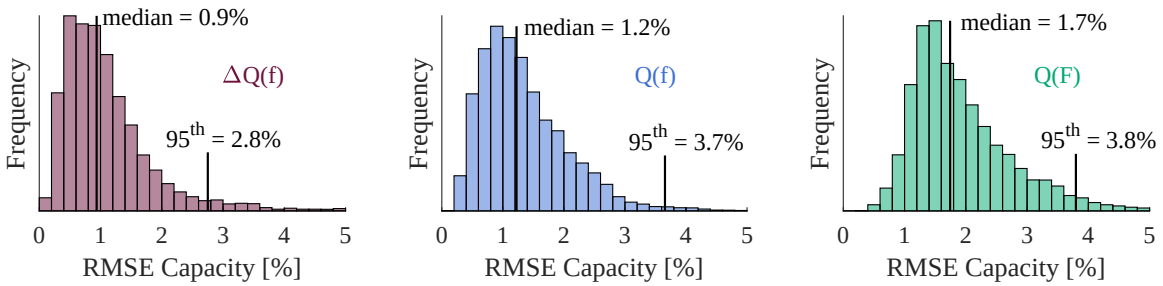


Fig. B.2 RMSE Capacity of the two SoH estimation methods compared against SoH prognosis method from Chapter 2. All models used GPR-EXP for the Severson-2019 and Attia-2020 datasets.

The signal-to-noise ratio was weaker when using cumulative features. As shown in Fig. 4.1, there is insufficient variation in cumulative features to capture the varying capacity values. That result would be replicated in any environment with input features averaged over significant time periods, and sudden and catastrophic changes in degradation rate.

The prognosis approach appeared to have distinctly better performance than both diagnosis methods. The prognosis method assumed knowledge of initial health before extrapolating the capacity through the life of the cell. Effectively, that assumption removed the variability in initial capacity, artificially improving results.

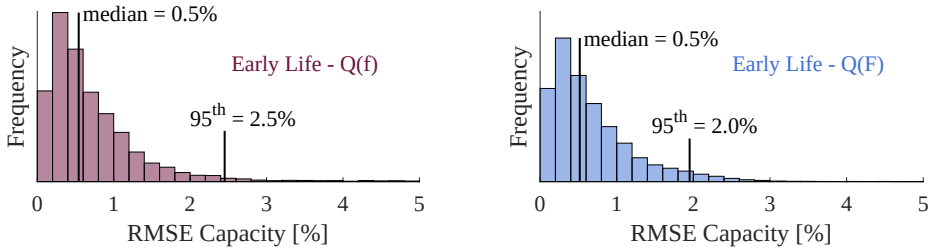


Fig. B.3 RMSE Capacity of the two proposed methods for SoH estimation in early life.

The SoH estimation was accurate when training and testing on the first few data points for each cell. Fig. B.3 used the first 5 data points for each training and test cell

and the results were a median of 0.5% capacity for both cumulative and instantaneous features.

It is possible to combine the median performances of the ΔQ method and the early life capacity estimation to approximate performance. The result would be a model that extrapolates a full health profile without any prior knowledge of health. As estimate of performance can be calculated as per equation B.1 and should result in a smaller RMSE Capacity than the above SoH estimation methods.

$$\begin{aligned}\text{RMSE Capacity(full prediction)} =& \sqrt{\text{RMSE Capacity(early life)}^2 + \text{RMSE Capacity}(\Delta Q)^2} \\ =& \sqrt{0.53\%^2 + 0.94\%^2} \approx 1.1\% \end{aligned} \tag{B.1}$$

B.3 Automated feature selection trial

10,000 repeats of the automated feature selection procedure described in Section 2.2 were performed to assess speed and investigate the outputs. The whole trial took around 7 minutes, 43 seconds of which were the selection process. The input features were selected from a random selection of 50 of the cells in Severson-2019 and Attia-2020, features calculated using the standard variable thresholds in Table 2.1. 10 features were selected each time for detail, as opposed to only 5 input features selected in most models in this thesis.

Below is a full depiction of Table 2.4, but with absolute frequencies not percentages quoted.

	Selection number									
Feature	1	2	3	4	5	6	7	8	9	10
$V_{2,3}$	10000									
$V_{1,2}$		7067	2							
time		2933	7067							
$\|P\|_{1,2}$			2852	6935	38	1				
$\|I\|_{1,2}$			73	1015	1099	6				
$I_{3,4}$			3	459	977	185	14	1		
$V_{1,4}$			2	80	1395	3551	1895	1396	949	450
$I_{1,3}$			1	1323	4119	1172	113	3	1	
$\|P\|_{2,4}$				188	1413	1041	456	47		
$\|I\|_{2,3}$					473	2126	3483	2338	1122	372
$\|P\|_{3,4}$					275	1095	2009	2586	2185	1097
$P_{3,4}$					207	783	1716	2363	1751	726
$\|P\|_{2,3}$					3	26	118	459	1695	2914
$\|I\|_{2,4}$					1	5	1			
$\Delta V_{1,4}$						6	147	502	1349	2687
$P_{1,3}$						1	27	191	459	274
$P_{2,4}$						1	5	12	45	111
$\Delta\|P\|_{3,4}$						1	3	25	80	283
$\|P\|_{1,3}$							10	67	320	788
$\Delta V_{2,3}$							3	4	13	33
$\Delta V_{1,3}$								4	25	203
$P_{1,4}$								1	1	5
$\Delta\|P\|_{1,2}$								1	1	0
$T_{2,3}$									3	38
$\Delta P_{3,4}$									1	4
$\|P\|_{1,4}$										12
$\Delta P_{1,3}$										2
$\|I\|_{1,4}$										1

Table B.1 Frequency of selection for the most common features. Trial contained 10,000 repeats and the maximum shared correlation was set at $\rho_{P,\max} = 0.85$.

Appendix C

Extra trials - Chapter 3

C.1 Example model parameters

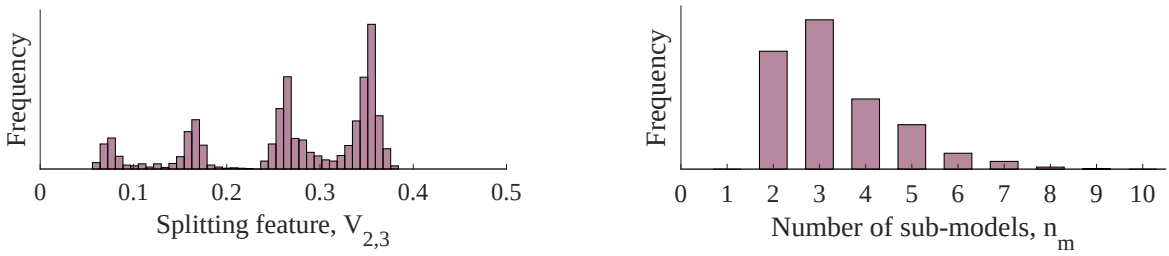
The below table contains all of the parameters for a piecewise linear regression (PLR) model. This model was formed using 50 randomly selected training cells from Severson-2019 and Attia-2020, and was used in the main train in Chapter 3. There were 4 sub-models, and a single value of σ_n was used for all sub-models.

Sub-model	bias	$V_{2,3}$	time	$\lvert P\rvert_{1,2}$	$I_{1,3}$	$\lvert P\rvert_{3,4}$	x_s range	σ_n
1	-9.21	9.03	0.068	2.85	5.04	-0.37	$V_{2,3} < 0.14$	
2	-3.59	7.48	-0.022	1.42	1.18	0.43	$0.14 \leq V_{2,3} < 0.34$	0.221
3	-0.22	-0.13	-0.012	0.62	-0.60	0.21	$0.34 \leq V_{2,3}$	

Table C.1 All parameters for a PLR model with 3 sub-models. The values were taken from a model in the main trial in Chapter 3. The bias term is the constant in the linear model.

C.2 Model structure investigation

Chapter 3 includes an investigation into the structure of the PLR model, but only a single model is assessed. An equivalent plot to Fig. 3.8 was produced for two extra values of the required improvement β_{improv} in the curvature method for finding suitable breakpoints. The same was done for using evenly spaced breakpoints.



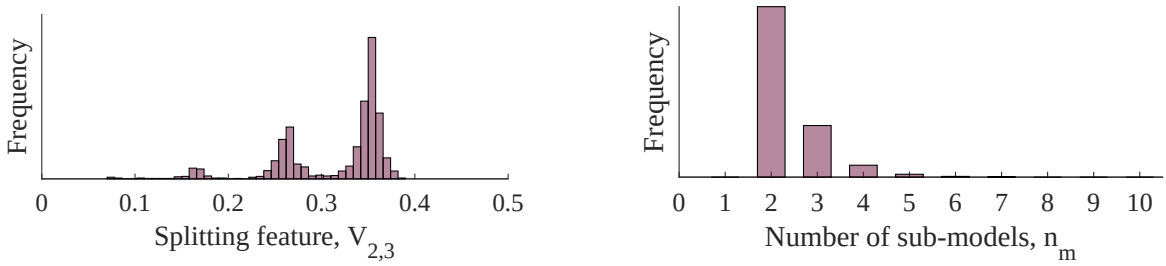
(a) Position of breakpoints.

(b) Number of sub-models used.

Fig. C.1 Results of trial investigating the construction of the PLR model with $\beta_{\text{improv}} = 0.10$.

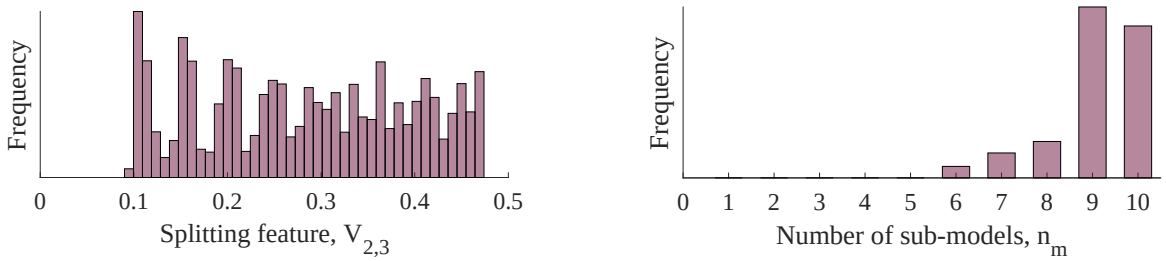
The two extra trials into the value of β_{improv} used the same set up as for Fig. 3.8, but with $\beta_{\text{improv}} = 0.10$ and $\beta_{\text{improv}} = 0.20$. Fewer sub-models were required in both cases, with that restriction being particularly acute for $\beta_{\text{improv}} = 0.20$ in Fig. C.2b. The results in Fig. 3.9 suggest that predictive performance was unaffected for $\beta_{\text{improv}} < 0.50$, but model complexity is reduced above $\beta_{\text{improv}} \geq 0.10$ values.

The model for Fig. C.3a used $\beta_{\text{improv}} = 0.01$ but used evenly spaced breakpoints. Models using evenly spaced breakpoints performed equivalently to the curvature method when measured by RMSE Capacity, EoL error and when computational time was compared. Fig. C.3a demonstrates the lack of information provided by evenly spaced breakpoints. Fig. C.3b shows the increased number of sub-models required to match the performance of the curvature approach.



(a) Position of breakpoints. (b) Number of sub-models used.

Fig. C.2 Results of trial investigating the construction of the PLR model with $\beta_{\text{improv}} = 0.20$.



(a) Position of breakpoints. (b) Number of sub-models used.

Fig. C.3 Results of trial investigating the construction of a PLR model using evenly spaced breakpoints and $\beta_{\text{improv}} = 0.01$.

C.3 Full cell-specific performance plot

Fig. 3.11 demonstrates the cell-specific predictive performance of PLR and two Gaussian process regression models, GPR-RBF and GPR-EXP. For clarity, only a subset of the test cells are included in Fig. 3.11. Fig. C.4 contains the same results but for all cells.

C.4 Health prognosis - Sauer-2021

The PLR approach was applied to the Sauer-2021 dataset [206] and the predictive performance is compared to that of GPR with a squared exponential kernel in Fig. **??**.

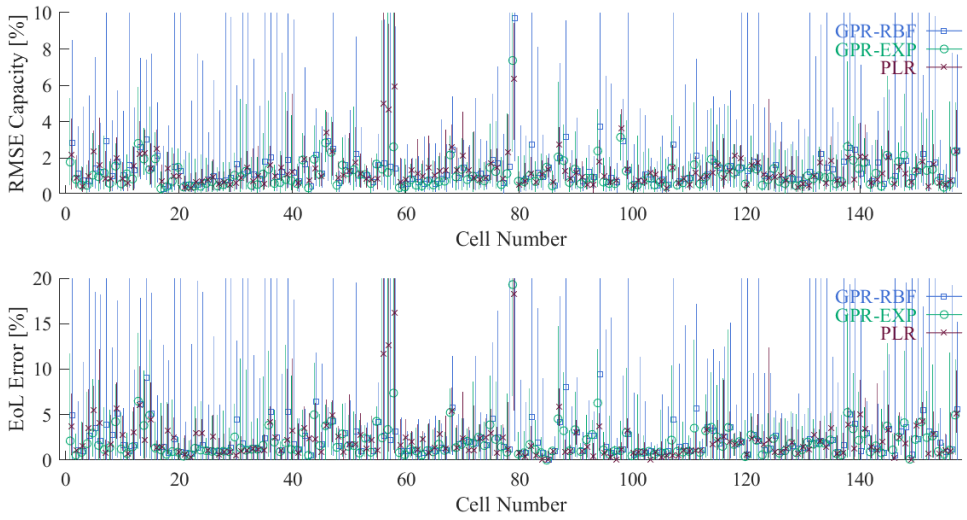


Fig. C.4 Cell specific performance of PLR (claret, crosses) relative to GPR-EXP (green, circles) and GPR-RBF (blue, squares) for all cells. Shapes show the median value for each cell, the lines a plotted from the minimum to the maximum of each performance metric for each cell.

The input features were generated using all available cells because there were only 48 available. The thresholds are in Table C.2.

Percentile	Current [A]	Voltage [V]	Temperature [°C]	Power [W]	Abs. Current [A]	Abs. Power [W]
1st	-4.10	3.10	24.0	-14.6	0.00	0.0
33rd	-0.24	3.51	27.5	-0.8	0.19	0.7
67th	0.18	3.90	28.5	0.7	0.98	3.6
99th	4.10	4.10	31.9	15.8	4.10	15.8

Table C.2 Feature thresholds calculated from the Sauer-2021 dataset

Feature selection was performed with $\rho_{P,\max} = 0.85$. There were 5 input features used and end of life was set at 60% but RMSE Capacity was measured over the full life down to around 30% capacity. The results from a Gaussian process regression model with a squared exponential kernel are in Fig. C.5.

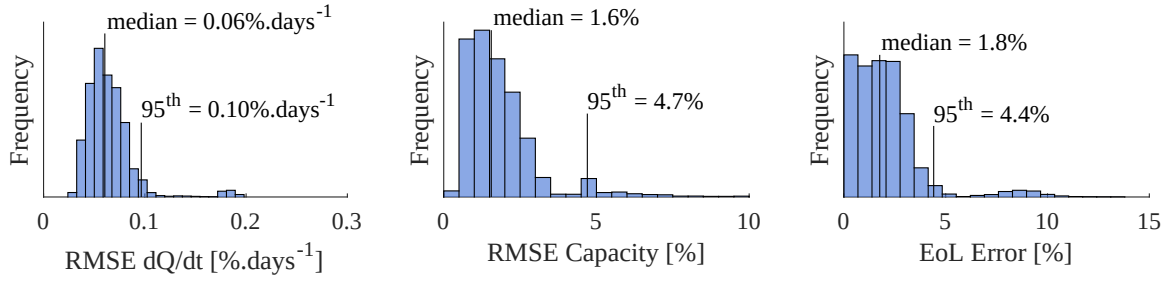


Fig. C.5 GPR-RBF predictive performance for Sauer-2021

The piecewise linear model was controlled as per the defaults in Chapter 3: $\beta_{\text{improv}} = 0.01$, $\beta_L = 0.1$, $\max(n_m) = 10$, $\sigma_w = 10$. The results are in Fig. C.6 and are compared with GPR-RBF in Table 3.4.

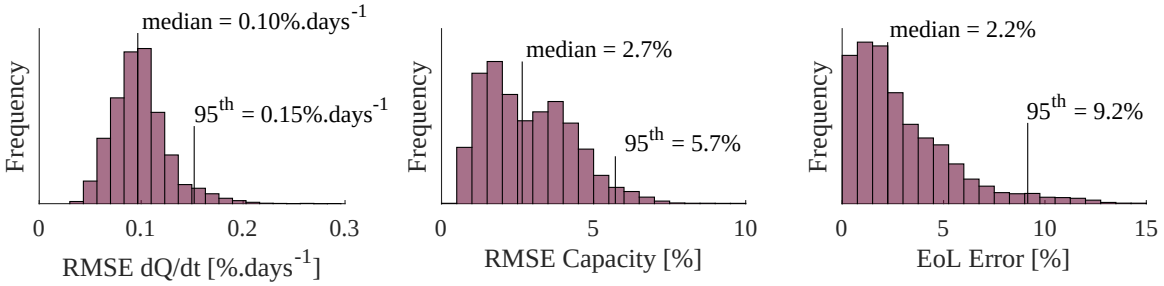


Fig. C.6 PLR performance with Sauer-2021

The piecewise model appeared to struggle with the double change in degradation behaviour in Sauer-2021. A second trial was performed with the training data truncated above 50% of nominal capacity to avoid the late life reduction in degradation rate.

The RMSE Capacity was significantly improved in this case. The PLR approach appeared to be insufficiently flexible to manage the two behavioural changes. The selection step was most vulnerable to the complex behaviour. A measure of noise about some arbitrary smoothed function could perform better to select the first feature. The other features could still be selected using correlation-based procedures.

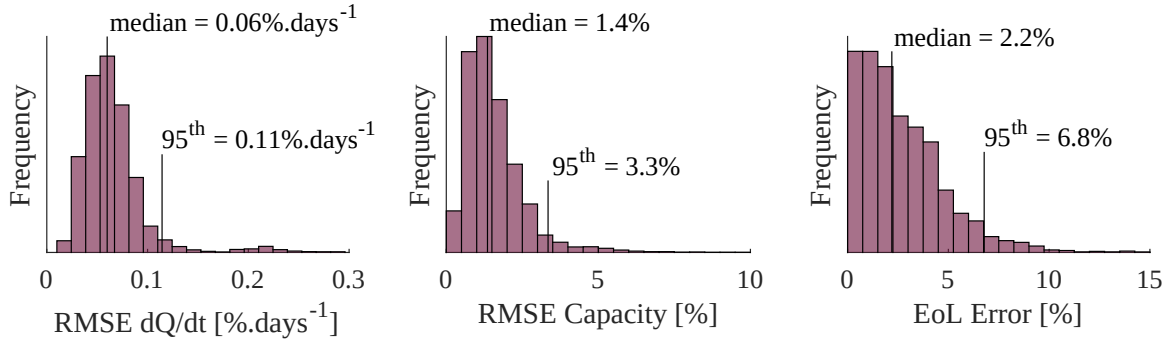


Fig. C.7 PLR performance with Sauer-2021 truncated above 50%

Appendix D

Quantifying cell-to-cell variability

This appendix contains the extra details required for the attempt to quantify cell-to-cell variability in Chapter 4. There is a derivation of multi-level Bayes (MLB) followed by the results for LinOne and LinTwo that were excluded from the main text.

D.1 Derivation of multi-level Bayes

The following derivation is adapted from the published work in reference [3]. The mathematics and code behind the use of MLB were provided by S. Jbabdi of the University of Oxford. The explanation in the published work was largely written and edited by S. Jbabdi.

When using MLB, the parameters of a model for each individual cell are assumed to be drawn from a population distribution. That distribution is unique for each dataset and always assumed to be Gaussian, i.e. each dataset has individual but unknown population mean and variance for each parameter.

The target variable for all models in Chapter 4 was capacity. For the following derivation, let $\mathbf{y_k}$ denote the target variable of cell k over a number of measurements, i.e. $\mathbf{y_k}$ is a vector. The model parameters for cell k are denoted θ_k. For example, the LinExp model is specified by $\theta_k = (c_k, t_{f,k}, \tau_k)^T$ and θ_k for LinOne and LinTwo were of length one and two respectively. The observations were assumed to be a sum of the model and Gaussian measurement noise.

$$\mathbf{y_k}(t) = f(\theta_k, t) + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, \sigma_{n,k}^2) \tag{D.1}$$

where f is one of LinOne, LinTwo or LinExp, the three capacity-time models used in Section 4.2.

As previously mentioned, each set of parameters for an individual cell model were assumed to be drawn from a Gaussian population distribution,

$$\theta_k \sim \mathcal{N}(\mu_p, \sigma_p^2) \tag{D.2}$$

where μ_p is the population mean and σ_p^2 is the population variance. The population means were assumed to have wide Gaussian priors, $p(\mu_g) = \mathcal{N}(0, 10^4)$, and the priors over the population variances were uniform distributions. The priors for the noise parameters were assumed to be Jeffrey's prior, $p(\sigma_{n,k}^2) \sim \frac{1}{\sigma_{n,k}^2}$ and these were integrated out analytically.

The multi-level Bayes model is built in two stages. The first level fit the parameters for individual cells, θ_k, by using Markov chain Monte Carlo (MCMC) to create samples

from $p(\mathbf{y_k}|\theta_k) = \int p(\mathbf{y_k}|\theta_k, \sigma_{n,k})p(\sigma_{n,k})d\sigma_{n,k}$. Based on these samples, the means μ_k and variances σ_k^2 were estimated. These estimates were used in the second level of inference, the one that produced the population distributions. The posterior of the population distribution can be expressed as,

$$p(\mu_p, \sigma_p|\{\mathbf{y_k}\}) = \int \cdots \int p(\mathbf{y_k}|\theta_k)p(\theta_k|\mu_p, \sigma_p)p(\mu_p, \sigma_p)d\theta_1 ... d\theta_K \tag{D.3}$$

The above integral can be evaluated analytically when all distributions are Gaussian,

$$p(\mu_p, \sigma_p|\{\mathbf{y_k}\}) \propto \prod_{k=1}^{K} \mathcal{N}(\mu_k, \sigma_k^2 + \sigma_p^2)p(\mu_p, \sigma_p) \tag{D.4}$$

where K is the sub-sample size.

The population posterior in equation D.4 shows that the population mean is a weighted average of the individual cell parameters. The weights are a combination of the individual, first-level variances, σ_k^2, and the population variance σ_p^2. The population distribution is then sampled using MCMC to estimate the summary statistics, μ_p and σ_p^2.

D.2 Results

The plots below are the results for LinOne (Fig. D.1) and LinTwo (Fig. D.2). The plots are the equivalents of those in Fig. 4.7 for the LinExp model. The y-axes are the standard deviations of 1,000 estimates of σ_p and the x-axes are the sub-sample sizes,

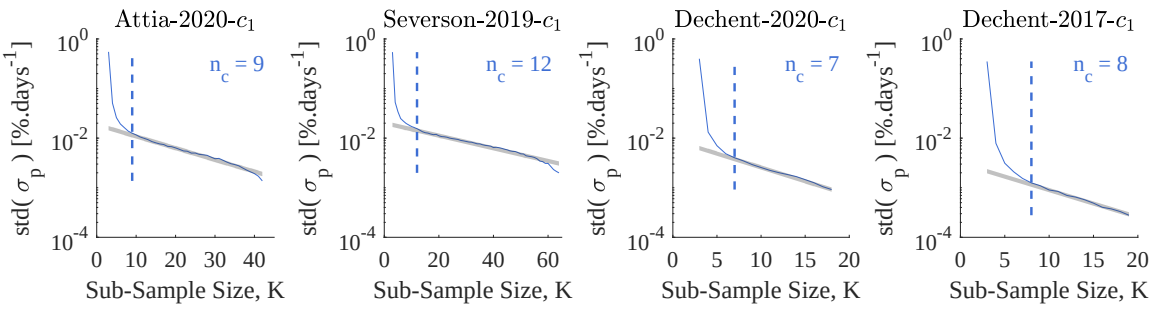


Fig. D.1 Estimating required sub-sample sizes for the LinOne model.

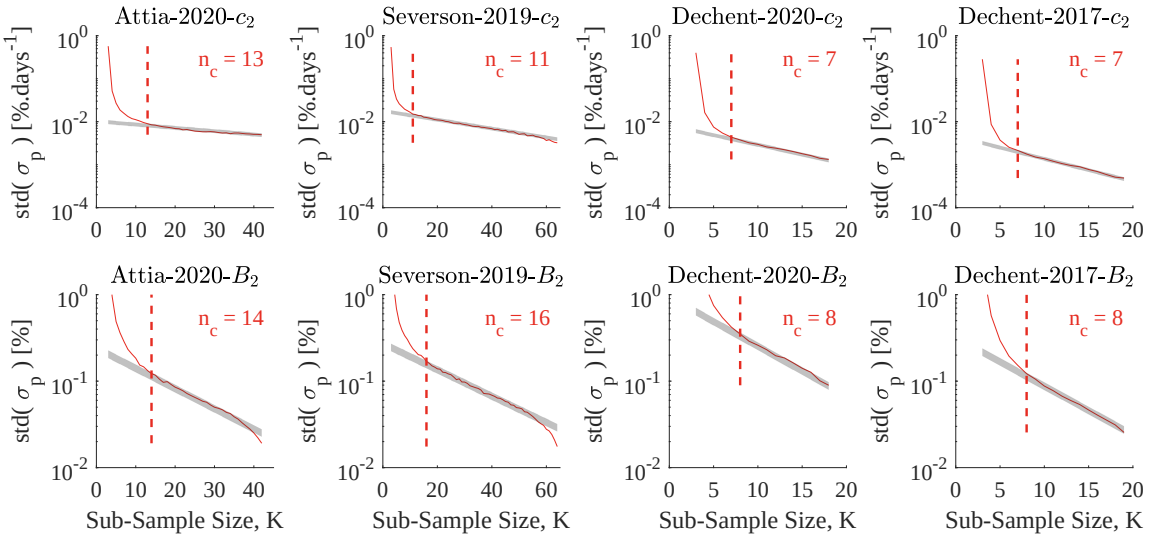


Fig. D.2 Estimating required sub-sample sizes for the LinTwo model.

K. Each plot shows the estimate of required sample size, n_c, for that dataset-model combination.