

Improved Battery Cycle Life Prediction Using a Hybrid Data-Driven Model Incorporating Linear Support Vector Regression and Gaussian

Mohammad Alipour,^{*,[a]} Shiva Sander Tavallaey,^[b, c] Anna M. Andersson,^[b] and Daniel Brandell^[a]

The ability to accurately predict lithium-ion battery life-time already at an early stage of battery usage is critical for ensuring safe operation, accelerating technology development, and enabling battery second-life applications. Many models are unable to effectively predict battery life-time at early cycles due to the complex and nonlinear degrading behavior of lithium-ion batteries. In this study, two hybrid data-driven models, incorporating a traditional linear support vector regression (LSVR) and a Gaussian process regression (GPR), were developed to estimate battery life-time at an early stage, before

more severe capacity fading, utilizing a data set of 124 battery cells with lifetimes ranging from 150 to 2300 cycles. Two type of hybrid models, here denoted as A and B, were proposed. For each of the models, we achieved 1.1% (A) and 1.4% (B) training error, and similarly, 8.3% (A) and 8.2% (B) test error. The two key advantages are that the error percentage is kept below 10% and that very low error values for the training and test sets were observed when utilizing data from only the first 100 cycles. The proposed method thus appears highly promising for predicting battery life during early cycles.

1. Introduction

Lithium-ion (Li-ion) batteries are used in a wide range of applications, from electronic devices to electric vehicles and grid energy storage systems, because of their low cost, long life, and high energy density.^[1,2] These rechargeable batteries lose capacity, energy, and power over time as a result of internal electrochemical processes and external operating conditions. Thus, Li-ion battery aging is generally characterized as an increase in internal resistance and a decrease in capacity, which constitute major problems.^[3,4] Battery aging increases the cost of energy storage systems and may potentially result in serious accidents such as fires and explosions. Therefore, accurate battery cycle life prediction is critical for optimizing the performance of energy storage systems while assuring their safety and reliability.^[5]

Since the emergence of the commercial electric vehicles (EVs), battery life-time has been a focus of research, with different Li-ion batteries being cycled and/or stored in order to identify different degradation mechanisms.^[6] To maintain the

safety and reliability of battery-powered systems, it is generally recommended that batteries should be replaced when they can only store 80% of their initial capacity. Laboratory studies are typically performed to better understand battery aging behavior under various operating conditions, with the resulting data being fed into or used to develop battery cycle life prediction models.^[7] In recent years, a variety of methods for predicting battery lifetime have been presented.^[8–10] Generally, battery life-time prediction methods include model-based, data-driven, and hybrid approaches.^[11–14] Model-based approaches use information of a system's failure mechanisms (e.g., solid electrolyte interface (SEI) growth) to provide a mathematical description of the degradation process, or they build an empirical model (experience-based models) to reproduce the system's declining trajectory.^[15] They normally use different filtering algorithms such as the Kalman filter (KF),^[16] the extended Kalman filter (EKF),^[17] or the particle filter (PF)^[18] to update model parameters recursively by sampling one measurement data point at a time. Hu et al.,^[19] for example, used a dual fractional-order extended Kalman filter (DFOEKF) for co-estimation of state of charge (SOC) and state of health (SOH) for Lithium-ion batteries. Data-driven modeling strategies, on the other hand, use historical data, real-time data, or both to determine the characteristics of the currently observed damage state and estimate future trends.^[12,20–22] Ng et al.^[23] published a list of the recent data-driven models for battery state estimation. Finally, hybrid approaches combine model-based and data-driven methods in order to leverage the strengths of both approaches.^[11,15,24,25]

Data-driven models using statistical and machine learning techniques have gained a lot of interest in battery prognostic applications since they do not necessitate a deep understanding of battery failure and other physical mechanisms. In these models, the battery systems are treated as black box systems to

[a] Dr. M. Alipour, Prof. D. Brandell
Department of Chemistry – Ångström Laboratory,
Uppsala University,
751 21 Uppsala, Sweden
E-mail: mohammad.alipour@kemi.uu.se

[b] Prof. S. S. Tavallaey, Dr. A. M. Andersson
ABB AB Corporate Research, Forskargränd 7,
SE-721 78 Västerås, Sweden

[c] Prof. S. S. Tavallaey
Department of Mechanics, School of Science KTH,
SE-100 44 Stockholm, Sweden

© 2022 The Authors. ChemPhysChem published by Wiley-VCH GmbH.
This is an open access article under the terms of the Creative Commons
Attribution License, which permits use, distribution and reproduction in
any medium, provided the original work is properly cited.

provide a mapping between various input and output variables. An increasing number of articles has been devoted to data-driven algorithms for predicting battery state and life-time in recent years. Che et al.^[26] used a universal deep learning method for prognostic and battery pack state of health estimation. Hu et al.^[27] developed a hybrid approach for lithium-ion battery RUL prediction based on particle filter (PF) and long short-term memory (LSTM) neural network. Liu et al.^[28] employed a Gaussian process regression (GPR) with composite kernels coupling the Arrhenius law and a polynomial equation to capture the electrochemical and empirical knowledge of battery degradation. Nuhic et al.^[29] used the support vector machine (SVM) for the estimation of state of health (SOH) and the remaining useful life (RUL). Ma et al.^[30] used the battery capacity in a specific window (the minimum embedding dimensions of the capacity data) as input features, and created a hybrid neural network that integrated a convolutional neural network and long short-term memory to predict battery life-time. Son et al.^[31] employed a Gaussian process regression using multiphysics features including mechanical and impedance evolutionary responses to estimate the SOH of batteries. Even though these present methods provide satisfactory results in terms of battery life-time prediction, they often require data corresponding to at least 25% aging in order to accurately estimate the target value. Due to the non-linear and complex degradation process of Li-ion batteries, precisely estimating battery life-time at early cycles – where the battery is largely yet to exhibit capacity degradation - is more challenging.

This paper offers two hybrid models combining a linear support vector regression (LSVR) and a Gaussian process regression (GPR) for battery cycle-life prediction using data from only the first 100 cycles in a data set^[32] of 124 cells with lifetimes ranging from 150 to 2300 cycles. The paper is organized as follows: In section 2, a comprehensive mathematical description of the proposed hybrid data-driven model is given. In section 3, the methodologies including the data description, the data preprocessing, the model development, and the model assessment methods are reviewed. Section 4 shows the results of the battery cycle-life prediction and compares them to published data.^[32] The paper is concluded in section 5.

2. Theory

2.1. Regression

Supervised learning can be applied in two different types of problems: regression as well as classification. While the regression approach tries to capture the behavior of the system, the classification tries to group and classify the system behavior in different subsystems.^[33]

Principally, any kind of regression problem could be modeled as

$$y = f(\mathbf{x}) + \varepsilon, \quad (1)$$

where $f(\mathbf{x})$ represents a hidden function of input vector \mathbf{x} and $\varepsilon \sim N(0, \sigma_n^2)$ is an independent and identically distributed Gaussian noise function with zero mean and variance σ_n^2 originating from an observation y .

2.2. Linear Support Vector Regression

For a given training data set D of n observations, $D = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, n\}$, where $\mathbf{x}_i \in R^d$ represents a d -dimensional input feature, y_i represents a scalar target value, and n denotes the number of samples in the training set, Support Vector Regression (SVR) finds a d -dimensional coefficient vector $\mathbf{w} \in R^d$ and intercept coefficient $b \in R$ such that the prediction given by $(\mathbf{w}^T \phi(\mathbf{x}_i) + b)$ is close to target value y_i . Here, the target value is the battery cycle life, and \mathbf{x}_i represents a vector of input features for battery sample i . The Linear SVR, subsequently, solves the following primal problem:^[34]

$$\min_{\mathbf{w}, b} \left(\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \max(0, |y_i - (\mathbf{w}^T \phi(\mathbf{x}_i) + b)| - \varepsilon) \right), \quad (2)$$

where the epsilon-insensitive loss is used, which ignores errors smaller than ε , and $C > 0$ is the regularization term. The dual problem is formulated as:^[35]

$$\min_{\alpha, \alpha^*} \left(\frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \varepsilon e^T (\alpha + \alpha^*) - y^T (\alpha - \alpha^*) \right), \quad (3)$$

subject to $e^T (\alpha, \alpha^*) = 0, 0 < \alpha, \alpha^* < C, i = 1, \dots, n$, where e is a vector of ones, $Q \in R^{n \times n}$ is a matrix with $Q_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Finally, once the optimization problem is solved, the target value is predicted as:

$$\sum_{i \in SV} (\alpha_i - \alpha_i^*) Q_{ij} + b, \quad (4)$$

where only support vectors (SV), i.e. samples that are within the margin, are considered.

2.3. Gaussian Process Regression

Gaussian Process Regression (GPR) is a non-parametric machine learning methodology. Unlike other supervised machine learning algorithms that estimate the probability of parameters of a specific function, the GPR calculates all likely functions that are fitting to the observation data. This approach uses a Bayesian framework to do prediction by collecting prior knowledge and deriving a posterior probability hypothesis.

A GPR is typically defined by two key functions: the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ which are defined as

$$m(\mathbf{x}) = E[f(\mathbf{x})], \quad k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \quad (5)$$

By choosing the mean and covariance functions, one can write the Gaussian process as:^[33]

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (6)$$

Furthermore, by summing the target value and noise distributions, one can simply include independently, identically distributed (i.i.d) Gaussian noise, $\varepsilon \sim N(0, \sigma_n^2)$, to the target value as:

$$y \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I). \quad (7)$$

In supervised learning, locations with comparable observation values \mathbf{x}_i are predicted to have similar response (target) values y_i . In GPR, this similarity is reflected by the covariance function, which determines how responses at one site \mathbf{x}_i are influenced by responses at other sites \mathbf{x}_j , $\mathbf{x}_i \neq \mathbf{x}_j$, $i = 1, 2, \dots, n$. Various kernel functions, with one or several hyper-parameters, can be used to define the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. Thus, the covariance function can be written as $k(\mathbf{x}_i, \mathbf{x}_j | \theta)$. For many conventional kernel functions, kernel variance σ_f and characteristic length scale σ_l are two common hyper-parameters. The characteristic length scales describe how far the input values \mathbf{x}_i can be apart before the response values become uncorrelated. For any collection of input features $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, the GPR defines a jointly Gaussian probability distribution $p(f(\mathbf{x}_1), p(f(\mathbf{x}_2), \dots, p(f(\mathbf{x}_n)))$. Therefore, from the GPR prior, the collection of training points and test points are joint multi-variate Gaussian functions, with zero mean value, distributed as seen in Eq. (8),

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix} \right). \quad (8)$$

Given the number of training samples as n and number of test samples as n^* , $K(X, X^*)$ denotes the $n \times n^*$ matrix of the computed covariances including all pairs of training and test points, and similarly for the other entries $K(X, X)$, $K(X^*, X)$, and $K(X^*, X^*)$. To improve the GPR's performance, the hyper-parameters of the covariance function must be tuned. This can be achieved by maximizing the log marginal likelihood defined as:

$$\log(y|X, \theta) = -\frac{1}{2} \mathbf{y}^T (K + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{n}{2} \log 2\pi, \quad (9)$$

where $-\frac{1}{2} \mathbf{y}^T (K + \sigma_n^2 I)^{-1} \mathbf{y}$ is the data-fit term, $-\frac{1}{2} \log |K + \sigma_n^2 I|$ is the complexity penalty term, and the $-\frac{n}{2} \log 2\pi$ is the normalizing constant term. One can obtain the posterior distribution by limiting the joint prior distribution to the functions that are fitting to observed data points. Subsequently, predictions at test points could be made by computing the conditional distribution as (see e.g.^[33]):

$$p(f^* | X, y, X^*) \sim N(\bar{f}^*, \text{cov}(f^*)), \quad (10)$$

where

$$\bar{f}^* = K(X^*, X) [K(X, X) + \sigma_n^2 I]^{-1} y, \quad (11)$$

$$\text{cov}(f^*) = K(X^*, X^*) - K(X^*, X) [K(X, X) + \sigma_n^2 I]^{-1} K(X, X^*). \quad (12)$$

3. Methodologies

The major purpose of this study is to predict Li-ion battery cycle life at an early stage of battery usage. More specifically, we hypothesize that merging the LSVR and GPR models could yield better results than state-of-the-art methodology,^[32] while still using the same data. Figure 1 depicts the procedure and steps for estimating cycle life, which include data description, data pre-processing, feature selection, and model development, all of which are covered in detail in the following subsections.

3.1. Data Description

Reis et al.^[36] reviewed over 30 datasets associated with Li-ion batteries. The MIT data set^[32] consisting of cycling data for 124 LFP/graphite cells (A 123 systems, model APR18650M1A, 1.1 Ah nominal capacity) was used in this work. All cells were charged using a variety of multi-step fast charging methodologies, then discharged at a constant current. For all cycles, the ambient temperature was fixed to 30 °C. Continuous data including voltage, current, battery temperature, and internal resistance were collected as the battery cells were cycled to end of life (EOL), defined as 80% of their initial capacity. The cycle-life histogram for 124 cell samples ranging from 150 to 2300 cycles is shown in Figure 2.

3.2. Data Pre-Processing

In ML applications, data pre-processing is critical for improving data quality and prediction accuracy. Generally, it includes removing outliers, filling missing values, time-domain synchronization, and normalization.^[37] In this context, some battery samples from noisy channels as well as some batteries that did not reach 80% capacity were removed. Two samples with outliers were noticed in the capacity fade curve for the first 100 cycles. The detected outliers were removed, and the missing data are then filled up using interpolated values. Finally, the whole data set was normalized using the z-score normalization method^[38] as:

$$Z = \frac{x - \mu}{\sigma}, \quad (13)$$

where Z is the standard score, x is the observed value, μ is the sample mean, and σ is the sample standard deviation.

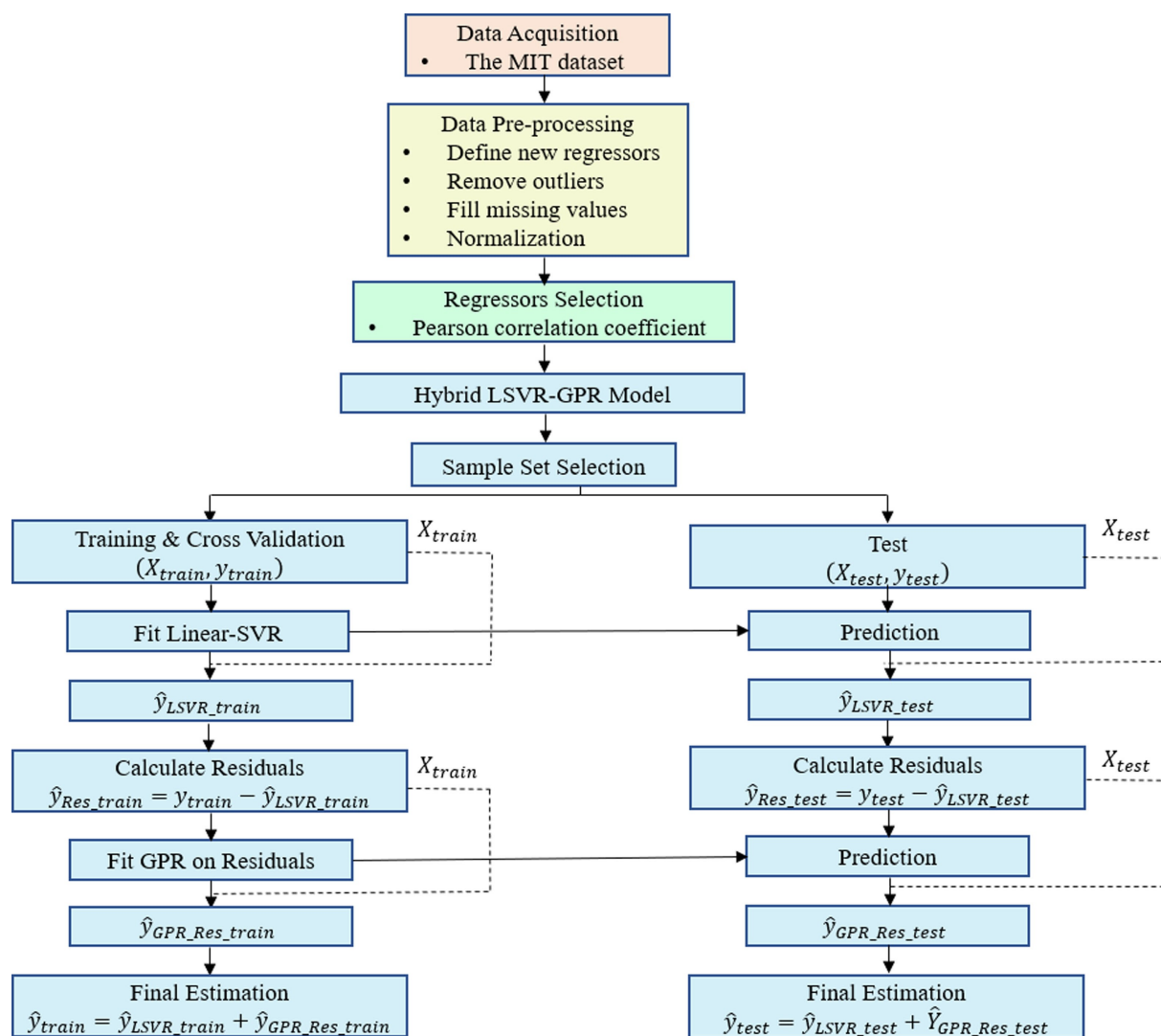


Figure 1. Cycle life estimation procedure.

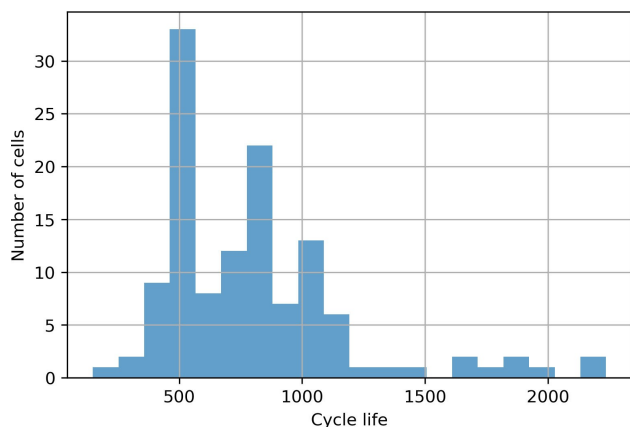


Figure 2. Cycle-life histogram for 124 battery samples in the MIT dataset.

3.3. Feature Selection

Normally, machine learning applications contain plenty of input features in the dataset. While some of these features might have good predictive strength, the presence of non-informative features can add uncertainty to the predictions. Therefore, when it comes to creating a machine learning model, feature selection is crucial to minimize the number of input variables, to lower the computational cost of modeling, and to increase the model's performance. The two fundamental types of feature selection approaches are supervised and unsupervised procedures. The distinction is whether or not the features are chosen based on the target variable. Unsupervised feature selection strategies, such as those that remove redundant variables using correlation, disregard the target variable. Approaches that use the target variable, such as methods that eliminate irrelevant variables, are supervised feature-selection techniques. In this section, an unsupervised method was used to remove redundant features. Features with high correlation have approx-

imately the same influence on the observed output. Therefore, when two features have a high correlation, one of them might be dropped without losing relevant information for predicting the output of interest.

Before eliminating redundant features, additional features were added to the available ones developed by Severson et al.^[32] All features with their respective definition are listed in Table 1. Below is a description of how the features are derived:^[32]

$$\begin{aligned}\Delta Q(V) &= Q_{100}(V) - Q_{10}(V), \Delta Q(V) \in R^p, \\ \Delta T(V) &= T_{100}(V) - T_{10}(V), \Delta T(V) \in R^p,\end{aligned}\quad (14)$$

$$\overline{\Delta Q}(V) = \frac{1}{p} \sum_{i=1}^p \Delta Q(V), \quad (15)$$

$$\mathbf{b}^* = \arg \min_b \frac{1}{m} \|\mathbf{q} - \mathbf{N}\mathbf{b}\|_2^2, \quad (16)$$

where m is the number of cycles in the prediction, $\mathbf{q} \in R^m$ is a vector of discharge capacities as a function of the cycle number, $\mathbf{N} \in R^{m \times 2}$ is a matrix with the first column containing cycle numbers and the second column containing a vector of ones, and $b \in R^2$ is a coefficient vector.

Table 1. List of input features.

	Features description	Symbol	Equation
ΔQ_{100} 10(V) features	Minimum	x_min	$\log(\min(\Delta Q(V)))$
	Mean	x_mean	$\log(\overline{\Delta Q}(V))$
	Variance	x_var	$\log\left(\frac{1}{p-1} \sum_{i=1}^p (\Delta Q(V) - \overline{\Delta Q}(V))^2\right)$
	Skewness	x_skew	$\log\left[\frac{\frac{1}{p} \sum_{i=1}^p (\Delta Q(V) - \overline{\Delta Q}(V))^3}{\left(\sqrt{\frac{1}{p} \sum_{i=1}^p (\Delta Q(V) - \overline{\Delta Q}(V))^2}\right)^3}\right]$
	Kurtosis	x_kurt	$\log\left[\frac{\frac{1}{p} \sum_{i=1}^p (\Delta Q(V) - \overline{\Delta Q}(V))^4}{\left(\sqrt{\frac{1}{p} \sum_{i=1}^p (\Delta Q(V) - \overline{\Delta Q}(V))^2}\right)^4}\right]$
Discharge capacity fade curve features	Slope of the linear fit to the capacity fade curve, cycles 2 to 100	x_slopeDC	the first value in the vector \mathbf{b} where $d=99$
	Intercept of the linear fit to capacity fade curve, cycles 2 to 100	x_constDC	the second value in the vector \mathbf{b} where $d=99$
	Slope of the linear fit to the capacity fade curve, cycles 91 to 100	x_slope90	the first value in the vector \mathbf{b} where $d=10$
	Intercept of the linear fit to capacity fade curve, cycles 91 to 100	x_const90	the second value in the vector \mathbf{b} where $d=10$
	Discharge capacity, cycle 2	x_QD2	$Q(n=2)$
	Difference between max discharge capacity and cycle 2	x_Qdiff	$\max_n Q(n) - Q(n=2)$
	Discharge capacity, cycle 100	x_QD100	$Q(n=100)$
Other features	Average charge time, first 5 cycles	x_chargetime	$\frac{1}{5} \sum_{i=1}^5 \text{Charge Time}_i$
	Maximum temperature, cycles 2 to 100	x_maxT	$\max_n T(n)$
	Minimum temperature, cycles 2 to 100	x_minT	$\min_n T(n)$
	Integral of temperature over time, cycles 2 to 100	x_tempint	$\int_{t_2}^{t_{100}} T(t) dt$
	Internal resistance, cycle 2	x_IR2	$\text{IR}(n=2)$
	Minimum internal resistance, cycles 2 to 100	x_IRmin	$\min_n \text{IR}(n)$
	Internal resistance, difference between cycle 100 and 2	x_IRdiff	$\text{IR}(n=100) - \text{IR}(n=2)$
Features added by this work	Variance of $\Delta T(V)$	x_varT	$\log\left(\frac{1}{p-1} \sum_{i=1}^p (\Delta T(V) - \overline{\Delta T}(V))^2\right)$
	Mean of dVdQ curve at cycle 100	x_mean dVdQ	$\log\left(\frac{dVdQ_{100}}{dQdT_{100}}\right)$
	Mean of dVdT curve at cycle 100	x_mean dVdT	$\log\left(\frac{dQdV_{100}}{dQdT_{100}}\right)$
	Mean of dQdV curve at cycle 100	x_mean dQdV	$\log\left(\frac{dQdV_{100}}{dQdT_{100}}\right)$
	Coulombic efficiency at cycle 2	x_CE2	$\frac{QD}{QC}(n=2)$
	Coulombic efficiency at cycle 100	x_CE100	$\frac{QD}{QC}(n=100)$
	Variance of Coulombic efficiency cycle 2 to 100	x_CEvar	$\log\left(\frac{1}{p-1} \sum_{i=1}^p (\text{CE}_{2-100} - \overline{\text{CE}_{2-100}})^2\right)$

Figure 3 shows the correlation heat-map including all features. To remove redundant input variables, columns with correlation greater than 0.9 were dropped. As a result, six features of twenty-six were removed.

3.4. Model Development

In this section, a comprehensive data-driven model was employed to predict battery cycle life before more severe capacity degradation phenomenon occurs. To this end, two hybrid models combining a LSVR and a GPR model were developed. While the LSVR model was used to forecast battery cycle life, the GPR model was used to model the cycle life residual, which is defined as the difference between the real cycle life and the LSVR model's predicted cycle life. Severson et al.^[32] utilized a linear model, and used the lasso and elastic net techniques for regularization to avoid over-fitting. They used four-fold cross-validation and Monte Carlo sampling for tuning hyper-parameters. Because recreating the same results would be difficult, the LSVR model, which employs the linear kernel, is used in this study. The GPR model was tested in the form of two different models: model A and model B. As illustrated in Figure 1, the final predictions were obtained by adding the LSVR model's predicted cycle life and the GPR model's predicted cycle life residual. The final models are therefore called hybrid model A and hybrid model B. It is worth noting that this design is theoretically equal to setting the LSVR model as a mean function of the GPR model.

In section [FS], an unsupervised feature selection strategy was used to remove redundant features. In this section, the filter feature selection method was used to select the most

relevant features. The filter-based feature selection method is a supervised method which uses statistical techniques to assess the relevance of features and target variable outside of the predictive models.^[39] The absolute valued Pearson correlation coefficient, as the most commonly used ranking criterion in the filter methods, was employed to select the most relevant features correlated to the target values. It determines the linear relationship between the feature x and the target y , as:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (17)$$

where x_i and y_i denote the i -th sample of feature x and the target y , and \bar{x} and \bar{y} are the independent and dependent sample means, respectively. Figure 4 shows the listed computed Pearson coefficients between the remaining features and the cycle life value. A threshold of 0.5 was utilized to filter the relevant features to be used as an input variables in the LSVR model, leading to the final choice of x_var , x_mean_dVdQ , x_minT , x_mean_dQdV , as well as x_IR2 .

Learning the parameters of a prediction function and testing it on the same data set is a fundamental error that can result in over-fitting. In machine learning applications, the common practice is to divide the entire data into three sets of data, i.e. training, cross-validation and testing, 60:20:20. It is well-known that the basic idea of cross-validation is to split the training set into two disjoint sets, one which is actually used for training, and the other, the validation set, which is used to monitor the performance of the trained model. The answer to

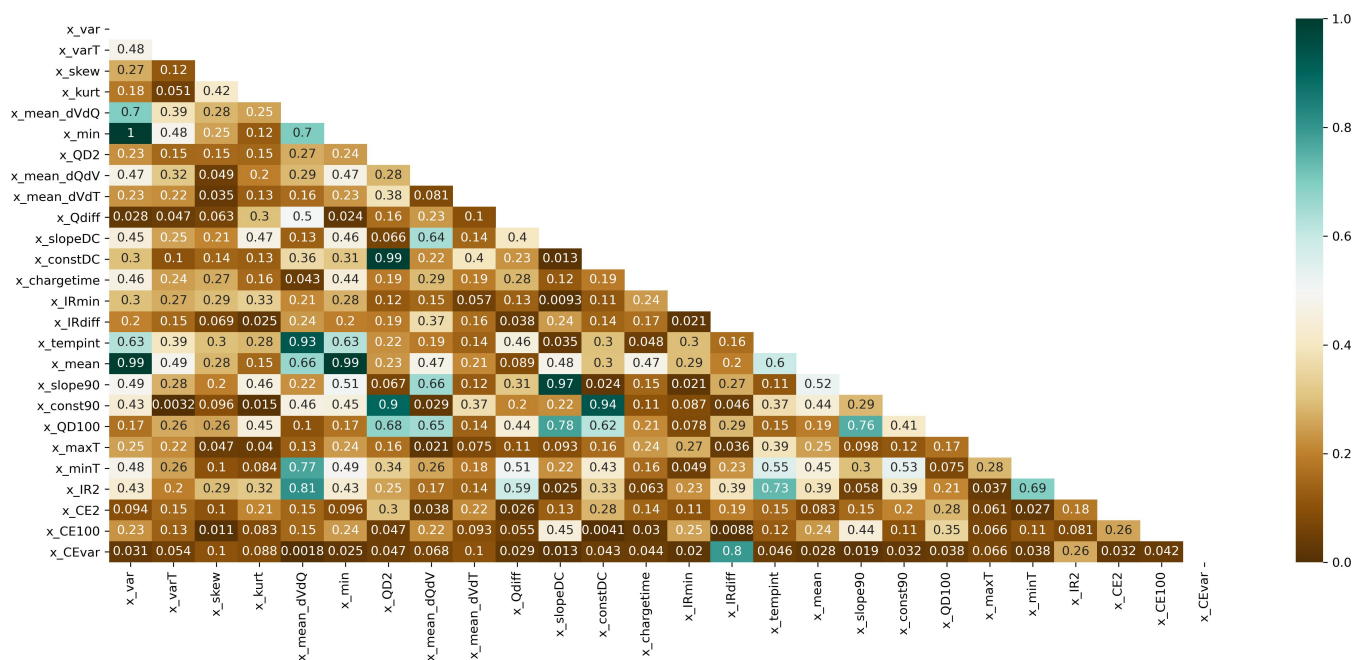


Figure 3. Triangle Correlation Heatmap for the dataset.

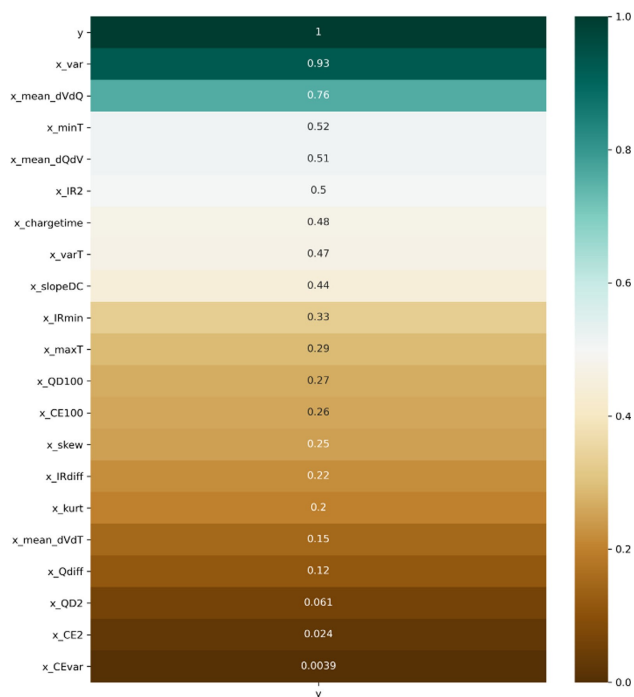


Figure 4. Pearson correlation coefficients between individual regressors and battery cycle lives.

the question on what the optimal number of the chosen folds would be, is more based on experimental rather than theoretical studies. One approach would be to choose the so-called leave-one-out cross-validation (LOO-CV), i.e. an extreme case of k -fold cross-validation obtained for $k=n$, the number of training cases. While this approach can be computational heavy, but the typical values for k are often in the range 3 to 10. In this work, the 80/20 training/test split on the data-set was used. Furthermore, the training set was split in to 5 smaller sub-sets, meaning that the 5-fold cross-validation was performed. Figure 5 depicts the procedure for k -fold cross-validation, in which a model is trained using $k-1$ of folds as training data and the resulting model is validated on the remaining data. After fitting the model using the training data and thereafter cross-validating it, the model was evaluated using the test set. We

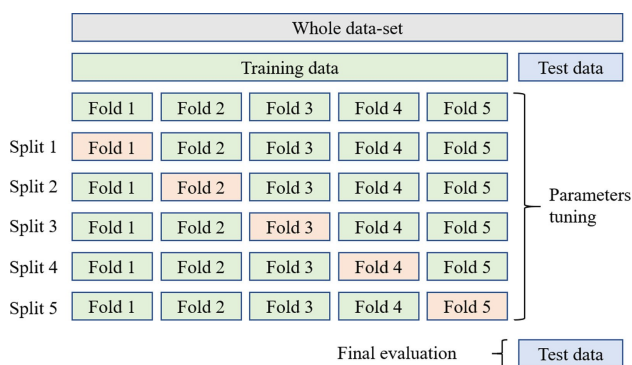


Figure 5. 5-fold cross validation procedure.

evaluated various cross-validation with different k -folds ($k = 1, 2, \dots, 5$), with the results showing that our choice of 5-fold cross-validation had the lowest error.

3.5. Model A

It is worth noting that the covariance function must be carefully chosen or built since it determines the GPR's functionality. As discussed earlier, the covariance function determines how responses at one site x_i are influenced by responses at other sites x_j , $x_i \neq x_j$, $i = 1, 2, \dots, n$. In model A, firstly, relevant features with the cycle life residual were filtered using the Pearson correlation coefficient. The Pearson coefficients vary from 0.0079 to 0.43, as shown in Figure 6. As a result, a 0.25 threshold was set to filter the relevant features, and five features were chosen to be used in Model A. Then, five different isotropic kernel functions, i.e. with the same length scale hyperparameter, see section [GPR], for each feature, were used in the GPR model. The isotropic squared exponential (radial basis function- RBF) kernel function is one of the most common used covariance functions, and defined as:

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j | \theta) = \sigma_f^2 \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma_l^2}\right), \quad (18)$$

where σ_l is the characteristic length scale, and σ_f is the signal standard deviation. The isotropic Matern 3/2 kernel is defined by:

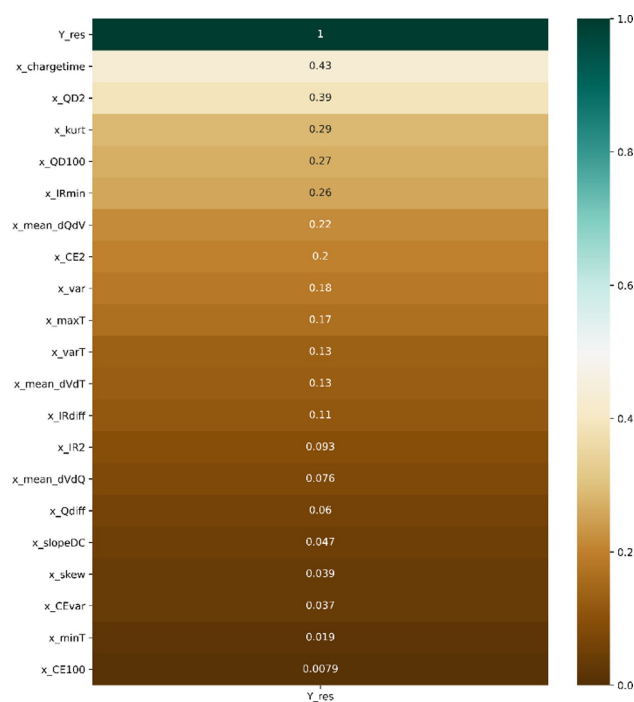


Figure 6. Pearson correlation coefficients between individual regressors and battery cycle life residual.

$$k_{\text{Matern}3/2}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \left(1 + \frac{\sqrt{3}}{\sigma_l} r\right) \exp\left(-\frac{\sqrt{3}}{\sigma_l} r\right), \quad (19)$$

where $r = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}$. Similarly, the Matern 5/2, rational quadratic (RatQuad), exponential (Exp), and linear kernel functions are defined as following equations:

$$k_{\text{Matern}5/2}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \left(1 + \frac{\sqrt{5}}{\sigma_l} r + \frac{5r^2}{3\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5}}{\sigma_l} r\right), \quad (20)$$

$$k_{\text{RatQuad}}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \left(1 + \frac{r^2}{2\alpha\sigma_l^2}\right), \quad (21)$$

$$k_{\text{Exp}}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \left(\frac{r}{\sigma_l}\right). \quad (22)$$

3.6. Model B

In model B, in contrast to model A, the entire input features were used where kernels with different length scales σ_l were used for each feature. Here, \mathbf{x}_{il} denotes a single feature l of sample i and \mathbf{x}_{jl} denotes a single feature l of sample j where $\mathbf{x}_i \neq \mathbf{x}_j$, $i = 1, 2, \dots, n$, and $l = 1, 2, \dots, d$. The automatic relevance determination (ARD) structure was implemented to extract the highly relevant input features for cycle life estimation. In principle, through using model B, irrelevant features might be limited by setting large length scales for them, resulting in a reduced and descriptive dataset.^[40]

Five alternative ARD-kernels were investigated to assess the GPR performance with model B, just as they were with model A. The ARD Squared Exponential kernel is defined as:

$$k_{\text{SE-ARD}}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{l=1}^d \frac{(\mathbf{x}_{il} - \mathbf{x}_{jl})^2}{\sigma_l^2}\right). \quad (23)$$

The ARD Matern 3/2 is defined as:

$$k_{\text{Matern}3/2_ARD}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 (1 + \sqrt{3}r) \exp(-\sqrt{3}r), \quad (24)$$

where $r = \sqrt{\sum_{l=1}^d \frac{(\mathbf{x}_{il} - \mathbf{x}_{jl})^2}{\sigma_l^2}}$. Similarly, the ARD Matern 5/2, ARD rational quadratic (RatQuad), ARD exponential (Exp), and ARD linear kernel functions are defined in the following equations:

$$k_{\text{Matern}5/2_ARD}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5r^2}{3}\right) \exp(-\sqrt{5}r), \quad (25)$$

$$k_{\text{RatQuad_ARD}}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 \left(1 + \frac{1}{2\alpha} \sum_{l=1}^d \frac{(\mathbf{x}_{il} - \mathbf{x}_{jl})^2}{\sigma_l^2}\right)^{-\alpha}, \quad (26)$$

$$k_{\text{Exp}}(\mathbf{x}_i, \mathbf{x}_j|\theta) = \sigma_f^2 (-r). \quad (27)$$

3.7. Model Assessment

To evaluate the outcomes, the predicted cycle lives should be compared to the actual cycle lives from the observation data. In this sense, this work employs two distinct metrics, root mean square error (RMSE) and mean percent error (% err), similar to those employed by Severson et al.^[32] The metrics are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (28)$$

$$\% \text{err} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \times 100. \quad (29)$$

4. Results and Discussion

Section 3.4 covered the design of the developed hybrid data-driven models. The major point of interest in this study has been to improve the accuracy of the predicted remaining useful life for the studied batteries. Different statistical and data-driven-models were examined as described in chapter 3. The GPR model was used to forecast the cycle life residuals after subtracting the predicted cycle life from the observed cycle life values using the LSVR model. The hybrid models were developed in two forms: hybrid model A and hybrid model B. The key differences between them are the method of input feature selection and the type of kernels used in the covariance matrix for each case.

Figure 7 shows the cycle life residual data distribution across all battery samples. The goal here is to use the GPR model to estimate the cycle life residual for each of the samples. To this end, a GPR model with alternative kernel functions was examined, as described in section 3.4. Although the squared exponential (SE) kernel function is powerful for machine learning applications, one drawback could be the smoothness

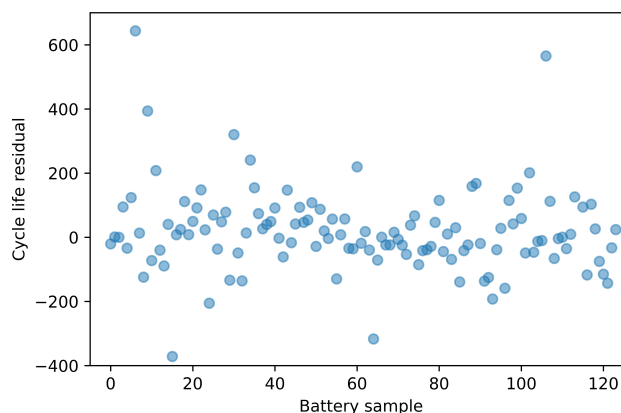


Figure 7. Distribution of cycle life residual data for all the battery samples.

of the predicted model which can exclude specific behaviors in the studied data. Here, the Matern class of covariance with or without ARD (Automatic Relevance Determination) can be of use. This class of kernel functions use Bessel functions and additional positive hyperparameters. The scaling parameter is chosen so that for an infinitely large scale factor, the kernel will converge to the ordinary SE covariance function. Thus, there is a trade-off between the smoothness and required hardness when choosing the right value for the scaling parameter. Low values (e.g. $\frac{1}{2}$) would be too rough, whereas high values (e.g. $\frac{7}{2}$) would be too smooth. The results provided in Table 2 clearly indicate this fact.

Table 2 lists the prediction accuracy of hybrid model A using the RBF, Matern 3/2, Matern 5/2, rational quadratic, and exponential kernels. Despite the fact that the exponential kernel had the highest RMSE for the training set among all the kernels, it was chosen to represent model A since it had the lowest RMSE and %err for the test set. The hybrid model A has the advantage of keeping the %err for both the training and test sets below 10%, despite the kernel function used in the GPR model.

Similarly, Table 3 lists the performance of hybrid model B with five different ARD kernels. Using all ARD kernels in the GPR model, hybrid model B, like hybrid model A, is capable of keeping the %err below 10%. Among these, the model using the exponential kernel has the best performance, with RMSE of 16.6 and 152, and %err of 1.4 and 8.2 for the training and test set, respectively.

The final form of the hybrid models A and B is accepted as those with the exponential kernel in the GPR model. The predicted versus real cycle lifes for the LSVR, hybrid model A, and hybrid model B are depicted in Figure 8, with the blue points representing training samples and the red points representing test points. The more linear the distribution is, the higher the prediction performance. The hybrid models are clearly more linearly distributed, implying that the predicted cycle lifes are closer to the real values.

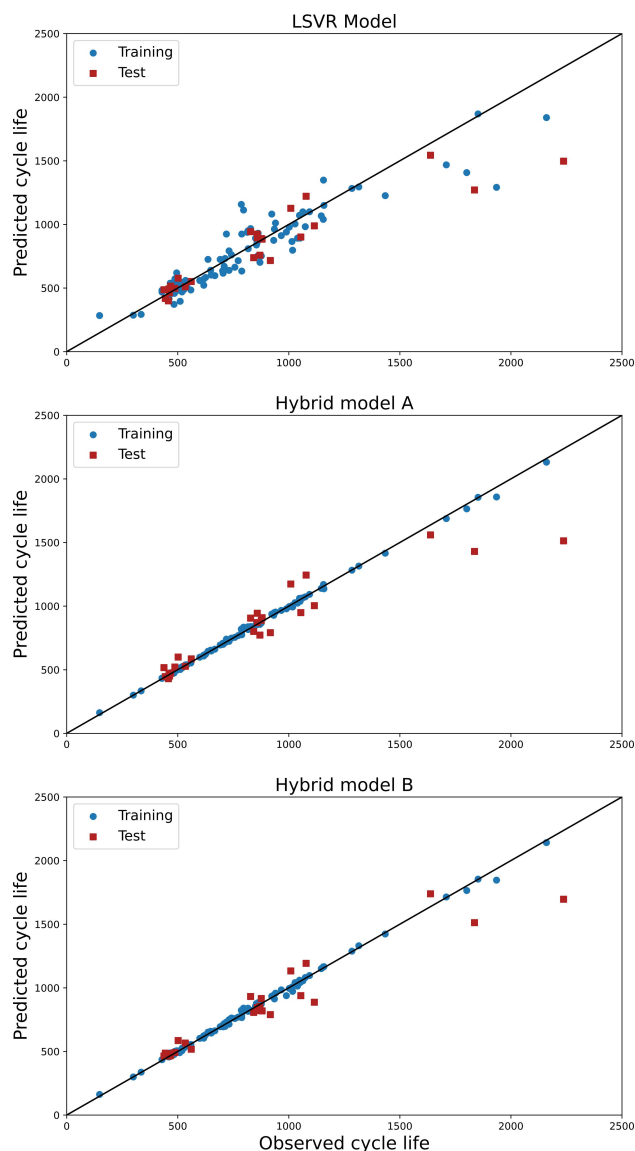


Figure 8. Predicted cycle life versus the real cycle life for the LSVR model, the hybrid model A, and the hybrid model B.

Hybrid model A	RMSE		Mean Percent Error (%)	
	Training	Test	Training	Test
RBF	12.8	180.8	1.1	9.3
Matern 32	13.2	177.2	1.1	8.6
Matern 52	13.0	178.5	1.1	8.9
RatQuad	13.5	179.2	1.1	8.6
Exponential	13.8	177	1.1	8.3

Hybrid model B	RMSE		Mean Percent Error (%)	
	Training	Test	Training	Test
RBF_ARD	21.4	173.2	2.0	9.2
Matern 32_ARD	19.2	176.5	1.8	9.7
Matern 52_ARD	20.2	176.9	1.9	9.7
RatQuad_ARD	20.2	175.8	1.9	9.6
Exponential_ARD	16.6	152	1.4	8.2

Performance prediction of the LSVR model, hybrid model A, and hybrid model B was thereafter evaluated. The models were tested using five different kernels, and the best results were chosen and compared with Severson et al.^[32] Two metrics, the RMSE and %err, were used to evaluate the prediction performance of the models.

Table 4 benchmarks the current work with the linear model developed by Severson et al.^[32] who developed three separate models: the “Variance”, the “Discharge”, and the “Full” model, based on the feature types selected from different subgroups, and predicted and classified cells by cycle life. They reported their results in two ways (including and excluding an outlier sample that reached the end of life before cycle 100) for two sets of test: test 1 and test 2. They obtained high error values for the entire training, test 1, and test 2 sets using the “Variance” model, with RMSE values greater than 100 and %err

Table 4. Benchmarking the models.

Benchmark	Model	RMSE		Mean Percent Error (%)			
		Training	Test 1	Test 2	Training	Test 1	Test 2
Severson et al. ^[32]	Variance	103	138 (138)	196	14.1	14.7 (13.2)	11.4
	Discharge	76	91 (86)	173	9.8	13 (10.1)	8.6
	Full	51	118 (100)	214	5.6	14.1 (7.5)	10.7
	Model	Training	Test		Training	Test	
This work (excluding added features)	LSVR	136.4	191.2	–	12.2	12.6	–
	Hybrid model A	15.9	167.9	–	1.3	10.2	–
	Hybrid model B	17.7	167.6	–	1.4	10.2	–
	Model	Training	Test		Training	Test	
This work (including added features)	LSVR	128.4	204.8	–	10.9	10.6	–
	Hybrid model A	13.8	177.0	–	1.1	8.3	–
	Hybrid model B	16.6	152.0	–	1.4	8.2	–
	Model	Training	Test		Training	Test	

values greater than 10. They had slightly better results using the “Discharge” model with %err of 8.6 for test 2 and RMSE values less than 100 for training and test 1. Using their “Full” model, they got 118 for RMSE and 14.1 for %err for test 1 including the outlier. They achieved high prediction accuracy after excluding that sample, with an RMSE of 100 and a %err of 7.5 for test 1. However for test 2, even excluding the outlier, the “Full” model failed to keep %err below 10.

In this study, results were reported, as shown in Table 4, both with and without the added input features. Without the added input features, the LSVR model shows comparable % err values both for training (12.2%) and test (12.6%) set. However, when comparing the LSVR model to the hybrid models A and B, the latter perform better, especially on training data. With the new input features added to this study, hybrid model A outperforms all other models in terms of the RMSE (13.8) and % err (1.1%) for the training set, while hybrid model B, with the RMSE and %err of 15.2 and 8.2, showed the best performance for the test data. Both models offer two key advantages over the other models: the first is that they keep the %err below 10% for both the training and test sets, and the second is that the metrics of the training and test sets are not drastically different.

All of the computations were done on a personal computer (Intel(R) Core(TM) i9-10885H CPU @ 2.40 GHz). It's worth mentioning that loading the data takes the longest time. The LSVR model takes 0.29 seconds, while the hybrid models A and B with exponential kernels take 8 and 11 seconds to run, respectively.

5. Conclusion and Future Work

Battery lifetime prediction at an early stage of cycling is critical for safe operation, considering the rapid technology development, and need for accurate state of health (SOH) monitoring in EV applications. Most data-driven models described in literature need data relating to at least 25% of the aging process in order to properly predict battery lifetime. In this paper, a hybrid data-driven model combining the LSVR and GPR is proposed to

effectively predict battery cycle life using data from only the first 100 cycles.

Although the presented approach has shown the inherent potential of using data-driven approaches for describing and predicting the complex physical processes such as estimation of the Li-ion battery cycle life, the data greediness of these methods still calls for need of further research in the field. A smart combination of a physical reduced order model (ROM) with less parameters to be identified together with real as well as synthetic data would be one option track for future work.

Acknowledgment

This work has been supported by the Swedish Energy Agency through the Battery Fund program (project number 50100-1). MA and DB also acknowledge STandUP for Energy. We thank Mikael Unge for fruitful discussions.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords: battery cycle life · cycle life prediction · data-driven modeling · linear support vector regression · Gaussian process regression

- [1] M. Armand, P. Axmann, D. Bresser, M. Copley, K. Edström, C. Ekberg, D. Guyomard, B. Lestriez, P. Novák, M. Petrankova, W. Porcher, S. Trabesinger, M. W. Mehrens, H. Zhang, *J. Power Sources* **2020**, 479, 228708.
 [2] B. Dunn, H. Kamath, J.-M. Tarascon, *Science* **2011**, 334, 928–935.

- [3] Y. Li, K. Liu, A. M. Foley, A. Zülke, M. Berecibar, E. Nanini-Maury, J. Van Mierlo, H. E. Hoster, *Renewable Sustainable Energy Rev.* **2019**, *113*, 109254.
- [4] R. Xiong, Y. Pan, W. Shen, H. Li, F. Sun, *Renewable Sustainable Energy Rev.* **2020**, *131*, 110048.
- [5] Z. Fei, F. Yang, K.-L. Tsui, L. Li, Z. Zhang, *Energy* **2021**, *225*, 120205.
- [6] J. Vetter, P. Novák, M. R. Wagner, C. Veit, K.-C. Möller, J. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, A. Hammouche, *J. Power Sources* **2005**, *147*, 269–281.
- [7] M. S. Hosen, J. Jaguemont, J. Van Mierlo, M. Berecibar, *iScience* **2021**, *24*, 102060.
- [8] E. Wood, M. Alexander, T. H. Bradley, *J. Power Sources* **2011**, *196*, 5147–5154.
- [9] J. Neubauer, A. Pesaran, *J. Power Sources* **2011**, *196*, 10351–10358.
- [10] J. Y. Che, Z. Deng, X. Lin, L. Hu, X. Hu, *IEEE Trans. Veh. Technol.* **2021**, *70*, 1269–1277.
- [11] X. Hu, L. Xu, X. Lin, M. Pecht, *Joule* **2020**, *4*, 310–346.
- [12] D. An, N. H. Kim, J.-H. Choi, *Reliab. Eng. Syst. Saf.* **2015**, *133*, 223–236.
- [13] S. M. Rezvanzaniani, Z. Liu, Y. Chen, J. Lee, *J. Power Sources* **2014**, *256*, 110–124.
- [14] U. Krewer, F. Röder, E. Harinath, R. D. Braatz, B. Bedürftig, R. Findeisen, *J. Electrochem. Soc.* **2018**, *165*, A3656.
- [15] L. Liao, F. Köttig, *IEEE Trans. Reliab.* **2014**, *63*, 191–207.
- [16] Z. Wang, D. T. Gladwin, M. J. Smith, S. Haass, *Appl. Energy* **2021**, *294*, 117022.
- [17] W. He, N. Williard, M. Osterman, M. Pecht, *Prognostics of lithium-ion batteries using extended Kalman filtering*, IMAPS advanced technology workshop on high reliability microelectronics for military applications **2011**, pp. 1–4.
- [18] M. F. Samadi, S. M. Alavi, M. Saif, *An electrochemical model-based particle filter approach for Lithium-ion battery estimation*, 2012 IEEE 51st IEEE conference on decision and control (CDC) **2012**, 3074–3079.
- [19] X. Hu, H. Yuan, C. Zho, Z. Li, L. Zhang, *IEEE Trans. Veh. Technol.* **2018**, *67*, 10319–10329.
- [20] L. Lu, X. Han, J. Li, J. Hua, M. Ouyang, *J. Power Sources* **2013**, *226*, 272–288.
- [21] M. U. Cuma, T. Koroglu, *Renewable Sustainable Energy Rev.* **2015**, *42*, 517–531.
- [22] J. Zhang, J. Lee, *J. Power Sources* **2011**, *196*, 6007–6014.
- [23] M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, Z. W. Seh, *Nat. Mach. Intell.* **2020**, *2*, 161–170.
- [24] Y. Xing, Q. Miao, K.-L. Tsui, M. Pecht, *Prognostics and health monitoring for lithium-ion battery*, Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics **2011**, 242–247.
- [25] J. Xu, L. Xu, *J. Syst. Eng.* **2011**, *22*, 428–436.
- [26] Y. Che, Z. Deng, P. Li, X. Tang, K. Khosravinia, X. Lin, X. Hu, *Energy* **2022**, *238*, 121857.
- [27] X. Hu, X. Yang, F. Feng, K. Liu, X. Lin, *J. Dyn. Syst. Meas. Control.* **2021**, *143*, 061001.
- [28] K. Liu, X. Hu, Z. Wei, Y. Li, Y. Jiang, *IEEE Trans. Transp. Electrification* **2019**, *5*, 1225–1236.
- [29] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, K. Dietmayer, *J. Power Sources* **2013**, *239*, 680–688.
- [30] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, Y. Yuan, *Appl. Energy* **2019**, *253*, 113626.
- [31] S. Son, S. Jeong, E. Kwak, K. Oh, *Energy* **2022**, *238*, 121712.
- [32] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, M. Z. Bazant, S. J. Harris, W. C. Chueh, R. D. Braatz, *Nat. Energy* **2019**, *4*, 383–391.
- [33] S. E. Rasmussen, C. K. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, ISBN 026218253X **2006**.
- [34] C.-H. Ho, C.-J. Lin, *J. Mach. Learn. Res.* **2012**, *13*, 3323–3348.
- [35] Q. Klopfenstein, S. Vaiter, *Mach. Learn.* **2021**, *110*, 1939–1974.
- [36] G. Dos Reis, C. Strange, M. Yadav, S. Li, *Energy and AI* **2021**, *5*, 100081.
- [37] G. Z. Deng, X. Hu, X. Lin, Y. Che, L. Xu, W. Guo, *Energy* **2020**, *205*, 118000.
- [38] G. S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas, *IAENG Int J Comput Sci* **2006**, *1*, 111–117.
- [39] M. Kuhn, K. Johnson, *Applied predictive modeling*, Springer **2013**, 26.
- [40] R. M. Neal, *Bayesian learning for neural networks*, Springer Science & Business Media **2012**, 118.

Manuscript received: November 19, 2021

Revised manuscript received: January 22, 2022

Accepted manuscript online: January 24, 2022

Version of record online: March 1, 2022