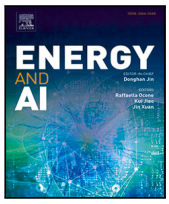
Energy and AI 19 (2025) 100465

Contents lists available at ScienceDirect

Energy and AI

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

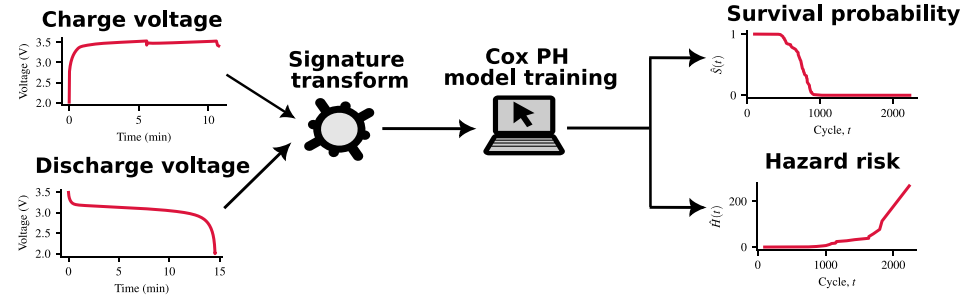


Robust survival model for the prediction of Li-ion battery lifetime reliability and risk functions

Rasheed Ibraheem [a], Timothy I. Cannings [a], Torben Sell [a], Gonçalo dos Reis [a,b,*]

[a] Maxwell Institute for Mathematical Sciences, School of Mathematics, University of Edinburgh, The Kings buildings, Edinburgh, EH9 3JF, Scotland, United Kingdom
[b] Center for Mathematics and Applications (NOVA Math), NOVA School of Science and Technology (NOVA FCT), Caparica, 2829-516, Portugal

GRAPHICAL ABSTRACT



HIGHLIGHTS

- Battery reliability and risk prediction from a single survival model.
- Model validation on data regions capturing protocol-to-protocol and cell-to-cell variability.
- Demonstrated application in battery dynamic warranty adjustment and stock management.
- Rigorous model testing including robustness to low-training sample and data sparsity.
- Documented modeling code publicly available.

ARTICLE INFO

Dataset link: https://data.matr.io/1/

Keywords:
Battery degradation
Cox Proportional Hazards
Path signature methodology
Survival probability function
Cumulative hazard function
Survival analysis
Reliability and risk functions

ABSTRACT

Single-value prediction such as the End of Life and Remaining Useful Life is a common method of estimating the lifetime of Li-ion batteries. Information from such prediction is limited when the entire degradation pattern is needed for practical applications such as dynamic adjustment of battery warranty, improved maintenance scheduling, and battery stock management. In this research, a predictive, semi-parametric survival model called the Cox Proportional Hazards is proposed for the prediction of cell degradation in the form of survival probability (battery reliability) and cumulative hazard (battery risk) functions. Once this model is trained, the two functions can be obtained directly for a new cell without having to predict several cogent points. The model is trained on the first 50 cycles of only the voltage profile from either the charge or discharge data regime, implying that our methodology is data region agnostic. The signature method with both desirable mathematical and machine learning properties was adopted as a feature extraction technique.

* Corresponding author at: Maxwell Institute for Mathematical Sciences, School of Mathematics, University of Edinburgh, The Kings buildings, Edinburgh, EH9 3JF, Scotland, United Kingdom.
E-mail address: g.dosreis@ed.ac.uk (G. dos Reis).

https://doi.org/10.1016/j.egyai.2024.100465
Received 6 September 2024; Received in revised form 16 December 2024; Accepted 19 December 2024
Available online 26 December 2024

The developed models are tested rigorously using application-driven strategies involving model robustness to the number of cycles of data required for model training and prediction, different fractions of training samples, and systematic data sparsity. The codes for modeling and testing are publicly available.

1. Introduction

Common models predicting the lifetime of batteries focus on a single-point prediction. Such models include those that predict the End of Life (EOL) [1,2] and Remaining Useful Life (RUL) [3–5]. These models have limitations regarding the prediction of the entire degradation trajectory which is more informative as battery reliability can be obtained for each cycle count. Several works have addressed these limitations on predicting the full capacity and internal resistance (IR) of Li-ion cells [3,6,7]. While addressing the limitations, these full-curve models rely on the prediction of several cogent points such as the knees of the capacity curves [8] and elbows of IR rise curves [9]. Thus, the prediction of the entire battery degradation reduces to building multiple models to predict these cogent points. This poses challenges in terms of resources (both time and computation infrastructure) needed to train multiple models. In addition, they are not trained to handle incomplete data, i.e., data that fails to meet degradation prediction criteria and is thus removed from modeling. An example of this scenario is when a cell is not cycled to the EOL due to either deliberate dropout, testing failure or error in data measurement. The corresponding data obtained under such a scenario cannot be used for building models that predict the EOL or entire capacity/IR curves, nor can predictions be made on such data.

With all these challenges mentioned, we propose a novel modeling methodology which uses the concept of survival analysis to predict the degradation pattern of Li-ion cells in the form of battery survival probability and hazard rate functions. Survival analysis refers to a collection of statistical methods used for the analysis of data that involves time to occurrence of an event [10]. In this research, we considered the event of a cell reaching the EOL and proposed survival models that model the time to this event. Two powerful and versatile battery degradation predictions are obtained from a single trained survival model: the survival probability and the cumulative hazard functions. The former gives the probability of a cell surviving past a given time while the latter provides the instantaneous rate at which a cell approaches its failure (in this case the EOL), and thus quantifies the risk of battery capacity failure. In terms of incomplete data mentioned earlier, survival analysis is natural for handling incomplete data through incorporating a dichotomy variable which is 0 if the cell does not reach the EOL (and is thus right censored) and 1 otherwise.

After giving a concise introduction to survival analysis, we review some relevant past work on its application to battery lifetime prediction and highlight the challenges therein which we aim to address in this study. Survival analysis was used in [11] to estimate the optimum age at which to replace a medical equipment battery and distinguish between groups of cells with the shortest and longest lifespans. The Kaplan–Meier [12] methodology and Log-Rank Test [13] were used for the estimation of survival probability and hypothesis testing respectively. We emphasize that no predictive models were developed in these works, they instead focus on the estimation of survival curves of groups of cells to be able to use the Log-Rank Test for distribution comparison. In addition, the Kaplan–Meier survival estimate does not take into consideration the cause of battery degradation as no covariates or health indicators are used (only the event time and indicator are needed). In contrast to the work of [11], a data-driven, predictive survival model was developed in [14] using a Random Survival Forest [15] to predict the battery reliability function (i.e., the survival probability function). The data used in the study belongs to the lead–acid starter battery of trucks and were recorded during one of the workshop's maintenance visits. We point out that the data does not contain measurements directly related to battery health indicators and no time series of battery health profiles are available. Thus, their proposed models were developed on covariates specific to a given workshop's maintenance routines. This poses challenges of wide adoption, generalizability and versatility of the developed models. In particular, its adoption in electric vehicle battery prognostics will be limited as popular health indicator data captured in this case are voltage, current, temperature, capacity, and IR (which are not used in their case). On the use of data that directly impact battery degradation for lifetime prediction, an Accelerated Failure Time (AFT) model was developed in [16] to predict the EOL using cycling data including temperature, current, and State of Charge (SOC); the AFT model is a parametric survival model which only predicts the expected time to event, and neither survival nor hazard curves. Here, we emphasize two points. First, the parametric nature of the AFT model poses the challenge of choosing a suitable distribution for the survival time. Second, only a single-point lifetime estimate was considered (the EOL) and hence no information about degradation at every cycle can be accounted for.

Following the challenges and gaps in the literature highlighted above, we present the following contributions of our study to address them:

1. A predictive, semi-parametric survival model called Cox Proportional Hazards (Cox PH) [10] is proposed which does not require the knowledge of the survival time distribution but only a baseline hazard function (which is learned entirely from the data). This model does not only provide a survival probability function once it is trained but also the cumulative hazard function which can be used to quantify battery risk of failure at a point in time.
2. The proposed model was trained and validated on two different cycling data regimes (one at a time): the charge voltage (which captures protocol-to-protocol variabilities) and the constant-current (CC) discharge voltage (which encodes cell-to-cell variabilities). This implies only the voltage profile was considered as the health indicator and the modeling methodology worked seamlessly under the popular battery data regions: charge and discharge. In extracting features from the voltage curve, the signature method [7,17] was adopted. This method is versatile and uniquely captures cogent information for each cell from the corresponding voltage curve. In addition, it has strong mathematical properties (such as uniqueness, universal nonlinearity, and time and translation invariance) that are desirable for a supervised machine learning task.
3. We propose several rigorous and application-driven model tests including robustness to the number of cycles of data required for model training and making inferences, different fractions of training samples, and systematic data sparsity at the training and inference levels. These tests were deployed on the proposed models to guarantee their performance and reliability.

The practical applications of the proposed models are clear. One of these is the *dynamic adjustment of warranty*. Battery manufacturers can use the predicted survival probability/hazard functions of battery cells to dynamically adjust warranty terms and conditions. This is made possible because predictions are curves rather than single value such as the EOL. Being able to adjust warrant dynamically has lots of benefits including offering personalized and accurate warranty periods, potentially extending warranties for cells with higher survival probabilities; reduction in the costs associated with warranty claims by anticipating failures and proactively addressing potential issues; improvement of transparency and customer trust and satisfaction because warranty update is backed by a data-driven approach (with battery survival probabilities/hazard rates). Another application is in *data-driven and*

improved maintenance scheduling. Fleet operators and service providers can leverage predicted survival probabilities to optimize maintenance schedules for battery-powered systems and devices. With this, maintenance can be performed proactively, preventing unexpected failures and downtime. Also, efficient allocation of maintenance resources can be achieved by focusing on high-risk batteries and thus unnecessary maintenance on low-risk ones will be reduced. Another benefit lies in *battery stock management*. Companies working with batteries can plan stock more reliably rather than having a large stock of spare batteries, they can order them on a rolling basis accounting for the expected number of dying batteries, for instance, in a month. Data-driven *selection procedure* is another application where a battery user may want to ensure that the battery survives a certain time with high probability. Survival analysis allows a more bespoke selection of batteries, rather than going for a point estimate such as EOL.

The rest of this paper is arranged as follows: Section 2 provides the description of the data used in this study; Section 3 gives an introduction to survival analysis, feature extraction techniques and model building procedures; results and discussion are presented in Section 4; and conclusions were left to Section 5. The mathematical description of the signature method is presented in the Methods Section (after the conclusions of Section 5).

2. Data source and data description

2.1. Data source and cycling procedures

The data used in this research is obtained from the Toyota data [1, 18]. This data contains 373 Lithium Ferrosphosphate (LFP)/graphite A123 APR18650M1A cells grouped into 8 batches (b1-b8). On the high level, each cell has an initial capacity and voltage of 1.1 Ah and 3.3 V respectively. They are cycled using similar conditions of an ambient temperature of 30 °C and the same discharging C-rate of 4C constant current–constant voltage (CC–CV) to a minimum voltage of 2.0 V. On the other hand, each cell is charged from 0%–80% SOC using one of over 80 different charging policies and then charged with the same C-rate (1C CC-CV) from 80% to 100% SOC (reaching a maximum voltage of 3.6 V). As a note, batches 1, 2, 3, and 8 are cycled to EOL while the rest are not cycled to EOL but to a maximum of 120 cycles. From the above descriptions, it can be deduced that the dataset contains two variabilities: *protocol-to-protocol* and *cell-to-cell* variabilities. The protocol-to-protocol differences arise from the charge regime where different charging policies are used to cycle cells resulting in different cycling conditions. As for the cell-to-cell variability, it stems from the discharge regime where cells are discharged identically to observe intrinsic differences among cells. The number of cells in each batch before and after preprocessing is provided in Table 1. Overall, a total of 373 cells were downloaded from the source but 362 (11 cells were dropped after preprocessing) cells were used in this research for all the modeling and experimentation.

2.2. Data analysis: feature and label data

We emphasize the different charging policies that have a direct impact on the charging profiles and introduce more variability in the dataset. With this and the fact that charging profiles in electric vehicles and other battery-powered devices can be controlled during the real-life usage of a battery (unlike discharging profiles which are dependent on the depth of usage, chosen route, level of traffic, and driver habits), the charging voltage profiles were considered as one of the data for feature extraction in this study; see Fig. 1(**a**) for a sample of charging voltage profile for all cells. In addition, the second data regime considered for feature engineering is discharging; see Fig. 1(**b**) for a sample of CC discharging voltage profile for all cells. This is motivated by the fact that some situations might warrant using identical and controlled cycling conditions to study intrinsic or cell-to-cell variabilities (which is the case in the Toyota discharge data regime). Examples of such situations are quality control done by cell manufacturers [19] and in storage for renewable energy where the derived energy is controlled and monitored [6]. Fig. 2 shows the degree of cycle-to-cycle variability. The l_2-norm of the difference between the signatures (see Section 3.4 for the details of signature calculation) of the first cycle voltage curve and each of cycle n ($2 \leq n \leq 50, n \in \mathbb{N}$) was calculated and plotted against n. The norm serves as an indication of 'similarity' between cycle 1 and $n > 1$. This was taken to show the difference of cycle-to-cycle variability in charge and discharge data regimes. It can be seen that variability is more pronounced in the discharge data than in the charge. This is because the discharge data indicate the actual use of a battery (energy is being derived from it), thus, a higher deviation between cycles is expected.

Table 1
The number of cells in each batch. The 'Original' column refers to the actual number of cells in the downloaded Toyota data. After data preprocessing, the 'Processed' column corresponds to the number of cells in each batch. For the Toyota data, 373 cells were originally downloaded, 5 cells were carried into b2 but belonged to b1 and thus removed from b2, and 6 cells from b3 were also removed because of noisy channels, leaving a total of 362.

Dataset	Batches	Number of cells	
		Original	Processed
Toyota	b1	46	46
	b2	48	43
	b3	46	40
	b4	46	46
	b5	46	46
	b6	48	48
	b7	48	48
	b8	45	45
	Total	373	362

Labels for the survival model are EOL and an event indicator (a dichotomy variable which is 1 if a cell is cycled to its EOL, otherwise 0) were obtained from the capacity curves. The EOL is defined as the cycle corresponding to an 80% decrease in capacity relative to the first capacity value. The EOL of all the cells (in each batch) used in this research are presented in Fig. 3. It can be seen that the datasets considered covered a wide range of EOL with a minimum and maximum value of 101 and 2235 cycles, respectively. The figure also shows that batches b4–b7 of the Toyota data were not cycled to EOL and that the last cycle was taken as EOL (thus are censored with an event indicator of 0).

3. Modeling and methodology

3.1. Overview of survival analysis

Survival analysis is a set of statistical techniques that focus on analyzing the time to the occurrence of an event. Time can be measured in different scales or units, for instance, days, months, years, or cycle counts from the start of a procedure until an event of interest occurs. An event is a particular experience that may happen to a subject during their lifetime. This can be the death of an individual, recovery from a disease, or a cell reaching its EOL. In this study, an assumption of one and only one event is adopted. In other words, a subject experiences an event only once in their lifetime. In addition, the *subjects* in our case are battery cells and an *event* is *reaching the EOL*. The EOL is defined as the cycle at which a cell capacity falls to 80% of its initial capacity. The time until EOL is regarded here as *survival time* and EOL is depicted as a *failure*. When a cell is cycled to failure, we say that cell is *uncensored* otherwise *right censored*. Right censoring of a cell may result from losing measurement tracking before reaching EOL or a deliberate dropout.

Following the convention in [10], we denote the random survival time of a cell as $T \geq 0$ (for us this is the number of cycles until EOL)

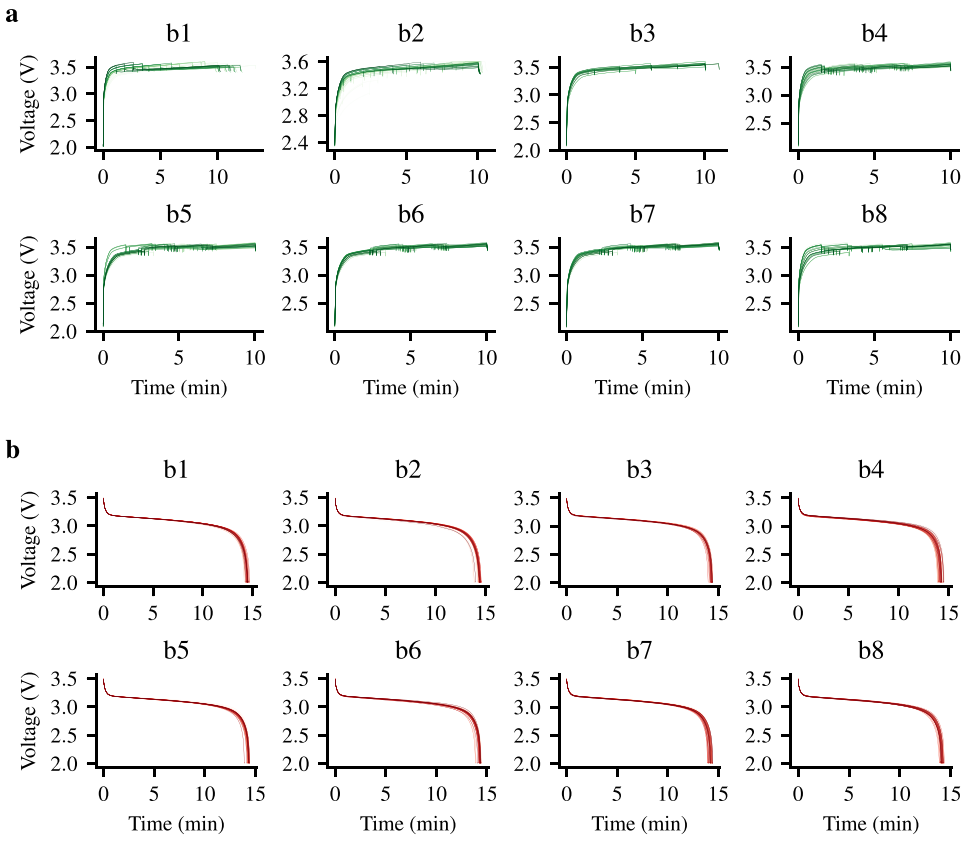


Fig. 1. (**a**) Charge and (**b**) CC discharge voltage curves of the Toyota data for cycle 50. All cells from each batch are considered in each figure block. For the charge regime, it can be seen that variability in the curves exists induced by the different charging policies. As for the discharge regime, cell-to-cell variabilities exist in the curves induced by the intrinsic property of the cells despite the cells being cycled identically.

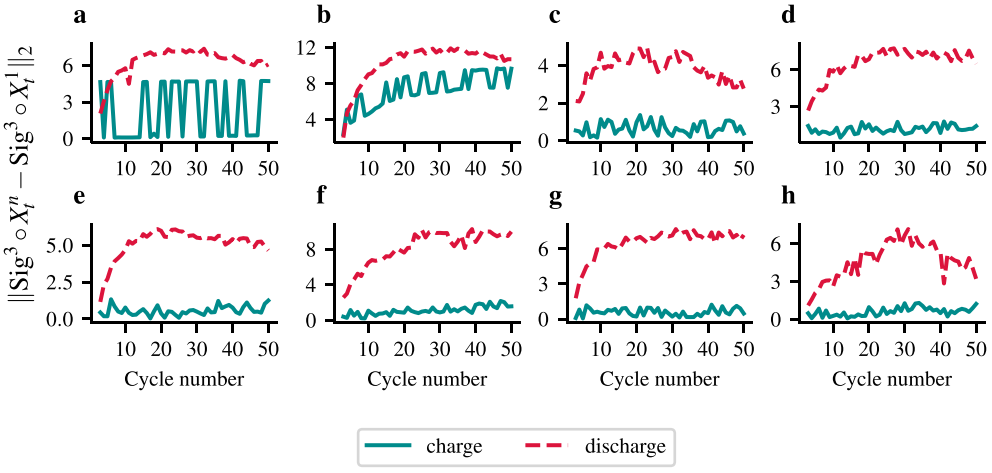


Fig. 2. Cycle-to-cycle variabilities in the charge and discharge curves for a sample cell from each batch: (**a**)-(**h**) corresponding to batches b1-b8 respectively. The l_2-norm of the difference in truncated signatures (calculated up to depth 3) of the voltage curves corresponding to the first cycle and cycle n ($2 \leq n \leq 50, n \in \mathbb{N}$) was used as a quantitative measure of variability. There is more variability in the discharge curves than in the charge. Details about signatures are discussed in Section 3.4 and the Methods section.

and a specific realization of it as t. To specify whether a cell is cycled to failure or right-censored, a binary $(0, 1)$ random variable δ is introduced into the analysis and is defined as

$$\delta = \begin{cases} 0, & \text{if a cell is censored} \\ 1, & \text{if a cell is cycled to failure.} \end{cases} \tag{1}$$

After defining the underlying survival variables, we now introduce and describe two functions which are the quantities of interest in any survival analysis. These are the survival function, $S(t)$, and the hazard function, $h(t)$. The former provides the probability that a cell survives longer than a given time t, and the latter gives the instantaneous rate at which a cell is approaching its EOL at a time t given that the cell has survived up to t. These are defined mathematically as follows (where $\mathbb{P}$ denotes the probability of an event):

$$S(t) = \mathbb{P}(T > t); \quad h(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(t \leq T < t + \Delta t | T \geq t)}{\Delta t}; \tag{2}$$

and they are related via

$$S(t) = e^{-H(t)}; \quad h(t) = -\frac{\frac{dS(t)}{dt}}{S(t)}. \tag{3}$$

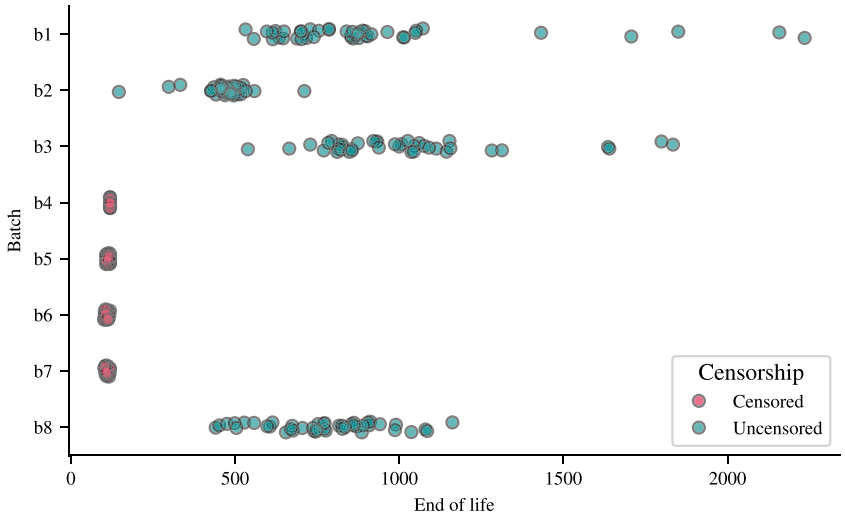


Fig. 3. EOL distribution and censorship summary of cells in each batch. Censored cells are cells that did not witness the event of being cycled to the EOL while uncensored reach the EOL by the end of cycling. All cells from b4 to b7 are censored.

where

$$H(t) := \int_0^t h(u)\,du \tag{4}$$

is known as the cumulative hazard function (usually used to characterize $h(t)$).

The survival function is monotone decreasing, and further, it holds that $S(0) = 1$ (i.e., a cell survives past 0 cycles with probability 1), and $S(\infty) = 0$ (meaning if a cell is cycled or used indefinitely, its probability of survival will eventually collapse to 0). In practice, on the other hand, $S(t)$ is usually estimated using a step function instead of smooth curves (denoted as $\hat{S}(t)$), and this approach is adopted in this study. The hazard function is nonnegative ($h(t) \geq 0$).

3.2. Cox proportional hazards model

The Cox Proportional Hazards (Cox PH) model formula is expressed in terms of the hazard model, which gives the hazard rate at time t for a set of covariates or predictors. Given the covariate for a subject $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$, the Cox PH model is given as

$$h(t, \boldsymbol{x}) = h_0(t) \exp\left[\sum_{i=1}^{p} \beta_i x_i\right], \tag{5}$$

where $h_0(t)$ is the baseline hazard function independent of $\boldsymbol{x}$. We emphasize that $h(t, \boldsymbol{x})$ is the hazard function $h(t)$ of Eq. (2) for a subject with covariate $\boldsymbol{x}$. In this research, we consider the case where $\boldsymbol{x}$ is time-independent.

There are three important properties of the Cox PH model. First, it assumes that the model reduces to the baseline hazard function when all covariates are *nil* (note that this does not necessarily mean zero in all problems as it might not make sense). Second, it is a semiparametric model because $h_0(t)$ is not specified and needs to be estimated. Third, it does not have an intercept term, and hazard functions for two subjects with covariates $\boldsymbol{x}_{s_i}$ and $\boldsymbol{x}_{s_j}$ are proportional at any time t, which motivates the "proportional hazard" assumption highlighted in the model's name. In mathematical terms, using Eq. (5), we have the proportional hazards given by

$$\frac{h(t, \boldsymbol{x}_{s_i})}{h(t, \boldsymbol{x}_{s_j})} = \exp\left[(\boldsymbol{x}_{s_i} - \boldsymbol{x}_{s_j}) \cdot \boldsymbol{\beta}\right], \quad \boldsymbol{\beta} \in \mathbb{R}^p$$

where the right-hand side is independent of time.

Two questions now arise: is it necessary to check the proportional hazards assumption when using Cox PH for predictive regression? How can we best estimate $h_0(t)$ and how can we find $\boldsymbol{\beta}$ via maximum likelihood estimation? To answer the first question, if the goal is to predict the survival function (and thus the hazard function as in the case of this study), then it is not necessary to check for proportional hazards [20]. This is because the goal is to maximize some scoring function (e.g. concordance index), which is independent of how predictions are generated. This is in contrast to a scenario where the survivals of two or more groups are investigated for significant survival differences. To answer the second question, one of the ways of estimating $h_0(t)$ is by using the Breslow estimator [21]. To demonstrate this, suppose that the m times of an event are $0 < t_1 < t_2 < \cdots < t_m$ where $t_i \neq t_j$ for $i \neq j$ and each t_i does not include censored times. Denote s_i as subject with event time t_i. We define the risk set R_i as the set of all subjects s_j such that $t_j \geq t_i$; i.e.,

$$R_i = \left\{s_j : t_j \geq t_i\right\}. \tag{6}$$

Using these ordered event times, the probability that the subject s_i experiences the event at time t_i is the ratio of its hazard at t_i to the sum of the hazards of all subjects who are available to get the event at t_i, that is

$$\mathbb{P}(s_i | t_i) = \frac{h_0(t_i) e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_i}}}{\sum_{s_j \in R_i} h_0(t_i) e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_j}}} = \frac{e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_i}}}{\sum_{s_j \in R_i} e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_j}}},$$

where $\boldsymbol{x}_{s_i}$ are the covariates for subject s_i. It can be seen that this probability is independent of t_i but depends on the order of the occurrences of the events. Breslow's estimator for the baseline hazard and cumulative hazard is given by

$$\hat{h}_0(t_i) = \frac{1}{\sum_{s_j \in R_i} e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_j}}}; \quad \hat{H}_0(t) = \sum_{t_i \leq t} \hat{h}_0(t_i),$$

and the corresponding likelihood function $\mathcal{L}(\boldsymbol{\beta})$ (called the partial likelihood because it only accounts for unique uncensored event times) and log-likelihood $\ell(\boldsymbol{\beta})$ are given by

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^{m} \frac{e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_i}}}{\sum_{s_j \in R_i} e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_j}}}; \quad \ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} \left[\boldsymbol{x}_{s_i} \cdot \boldsymbol{\beta} - \log \sum_{s_j \in R_i} e^{\boldsymbol{\beta} \cdot \boldsymbol{x}_{s_j}}\right]. \tag{7}$$

3.3. Implementation of Cox PH models

Implementing the Cox PH models can be categorized mainly into ordinary and regularized regression, tree-based and ensemble models, and deep learning approaches. In the Python programming language, these three categories have been implemented by popular libraries such

as *lifelines* [20], *scikit-survival* [22] and *pycox* [23,24]. Our implementation choice is that of *scikit-survival* because of its seamless integration with the *scikit-learn* [25] library (and thus allows the use of most of its functionalities such as cross-validation and hyperparameter tuning).

Different algorithms were explored in *scikit-survival* including ordinary Cox regression, regularized Cox regression, random survival forest [15], extratrees, and gradient boosting [26,27]. Of all these, the gradient boosting algorithm was able to capture the feature-label relation most as quantified by the metrics considered. The gradient boosting algorithm is similar to how random forest works but differs in how built trees are combined. While random forest averages the predictions made by fitted trees, gradient boosting builds greedily and sequentially (additively) such that a new tree (called base or weak learner) is added by fitting it on the residuals of the previous tree. Thus, the overall model f (which takes the role $\boldsymbol{x}\cdot\boldsymbol{\beta}$) is given as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_T} w_i g_i(\boldsymbol{x};\theta_i); \tag{8}$$

where N_T is the total number of base learners, w_i are weights, and g_i is a function that represents each tree structure parameterized by θ_i. The partial log-likelihood function in Eq. (7) is modified by replacing $\boldsymbol{x}\cdot\boldsymbol{\beta}$ with f and introducing the binary random variable in Eq. (1) to allow the following optimization problem on the uncensored unique event times:

$$\arg\max_{f} \sum_{i=1}^{m} \delta_i \left[f(\boldsymbol{x}_{s_i}) - \log \sum_{s_j \in R_i} e^{f(\boldsymbol{x}_{s_j})} \right].$$

Three metrics were considered for evaluating the chosen model: the concordance index (C-index), the cumulative dynamic receiver operating characteristic curve (cumulative dynamic AUC), and the integrated Brier score. The C-index measures the rank correlation between the predicted risk scores and observed event times. To define it, we need to define concordance. Given two subjects s_i and s_j, they are said to be comparable if their observed event times t_i, t_j satisfies $t_j > t_i$ and $\delta_i = 1$; i.e., the subject with lower observed time experienced the event. Comparable subjects s_i, s_j are said to be concordant if their predicted risks $\hat{r}_i, \hat{r}_j$ satisfies $\hat{r}_i > \hat{r}_j$ and $t_j > t_i$; i.e. the subject with lower risk has a larger observed survival time (and vice versa). Thus, the C-index is the ratio of the number of concordant pairs to the number of comparable pairs:

$$\text{C-index} = \frac{\#\text{ concordant pairs}}{\#\text{ comparable pairs}}. \tag{9}$$

In situations of equal predicted risk scores for a comparable pair, the number of concordant pairs is incremented by 0.5 instead of 1. To define the cumulative dynamic AUC, it is necessary to introduce cumulative and dynamic subjects. The former are all subjects that experience an event before or at time t (all subjects s_i such that $t_i \le t$) and the later are subjects whose survival times satisfy $t_i > t$. Following these definitions, the cumulative dynamic AUC measures the effectiveness of a fitted survival model to distinguish cumulative from dynamic subjects:

$$\text{AUC}(t) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}(t_j > t)\mathbb{1}(t_i \le t)\omega_i \mathbb{1}\left(\hat{r}(\boldsymbol{x}_{s_j}) \le \hat{r}(\boldsymbol{x}_{s_i})\right)}{\left[\sum_{i=1}^{n} \mathbb{1}(t_i > t)\right]\left[\sum_{i=1}^{n} \mathbb{1}(t_i \le t)\omega_i\right]}$$

Here, $\mathbb{1}$ is an indicator function, $\omega = 1/\hat{G}(t)$ is the inverse probability of censoring weights (IPCW) estimated from the Kaplan–Meier estimate of the censoring survival function (i.e Kaplan–Meier formula applied to censored times in which censored times are treated as failure times and vice versa), $\hat{G}(t)$, defined by

$$\hat{G}(t) = \prod_{\{i|t_i<t\}} 1 - \frac{n_i}{|R_i|}, \tag{10}$$

where n_i is the number of subjects that fail at time t_i and R_i is the risk set defined in Eq. (6). The function $\hat{r}(\boldsymbol{x})$ is the model's predicted risk score for a subject with covariate $\boldsymbol{x}$. For an ordinary Cox regression, it is estimated by $\hat{r}(\boldsymbol{x}) = \boldsymbol{x}\cdot\boldsymbol{\beta}$ while for a gradient boosting ensemble model (as in the case of this study), it is estimated by $\hat{r}(\boldsymbol{x}) = f(\boldsymbol{x})$ where f is as defined in Eq. (8). To summarize this metric, its mean value over a time range $t \in [a, b]$ is considered:

$$\overline{\text{AUC}}(a,b) = \frac{1}{\hat{S}(a) - \hat{S}(b)} \int_a^b \text{AUC}(t)\, d\hat{S}(t) \tag{11}$$

where $\hat{S}(t)$ is the estimated survival function. As for the time-dependent Brier score, it measures the mean squared error of survival states at time t:

$$\text{BS}(t) = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbb{1}(t_i \le t \wedge \delta_i = 1)\frac{\left(0 - \hat{\pi}(t|\boldsymbol{x}_{s_i})\right)^2}{\hat{G}(t_i)} + \mathbb{1}(t_i > t)\frac{\left(1 - \hat{\pi}(t|\boldsymbol{x}_{s_i})\right)^2}{\hat{G}(t)}\right]$$

where $\hat{\pi}(t|\boldsymbol{x})$ is the model's predicted probability of remaining event-free up to time t for a given covariate (i.e., $\hat{S}(t)$ for a subject or sample with covariate $\boldsymbol{x}$); and $\hat{G}(t)$ is the Kaplan–Meier estimate of the censoring survival function given by Eq. (10). The corresponding integrated Brier score calculated in the time interval $t \in [a, b]$ is given by

$$\text{IBS}(a,b) = \int_a^b \text{BS}(t)\, dw(t); \quad w(t) = t/b. \tag{12}$$

The *scikit-survival* implementation of all these metrics was considered and further information can be found in the library documentation. To the best of our knowledge, there are no standardized benchmarks for the values of each of the metrics in the literature and thresholds are treated case-by-case based on the nature of the problem under consideration. Moreover, the literature reviewed in this study (see [15,22,24]) has reported ≥ 0.65 for C-index/cumulative dynamic AUC; and ≤ 0.25 for integrated Brier score.

3.4. Data modeling

In this study, two groups of features were used as covariates for the Cox PH models. The first group was extracted from the charge voltage curves and the second from the discharge voltage curves. The two groups were engineered from the first n cycles of the corresponding profile, where the integer $n \in [2, 100]$ is chosen via cross-validation. The choice of this cycle number interval was motivated by the notion of using the minimum cycle data possible for modeling. Information from these curves was extracted via path signatures. This extraction involves creating a two-dimensional path $X_t = \{t, V(t)\}$ where t (in minutes) and $V(t)$ are the time and voltage values respectively. Signatures up to depth k were then extracted from this path using the *iisignature* Python library [28]; see the Methods section for the mathematical description and properties of the signature transform. The signature depth, $k \in \{2, 3, 4\}$ was tuned to balance between model accuracy and computational complexity. The higher the value of k, the more information is derived from the voltage–time profile, however, the higher the feature space dimension and thus the more complex the resulting model. In particular,

$$\text{Sig}^k \circ X_t^i := \left(S^1, S^2, S^{1,1}, S^{1,2}, S^{2,1}, \ldots, S^{i_1,\ldots,i_k}\right)$$

are the truncated signatures of depth k of a time–voltage path of cycle i corresponding to a given cell. The matrix of signatures of the time–voltage path was then calculated for the first n cycles resulting in

$$\text{Sig}^k \circ X_t^{1:n} := \begin{bmatrix} S_1^1 & S_1^2 & S_1^{1,1} & S_1^{1,2} & S_1^{2,1} & \ldots & S_1^{i_1,\ldots,i_k} \\ S_2^1 & S_2^2 & S_2^{1,1} & S_2^{1,2} & S_2^{2,1} & \ldots & S_2^{i_1,\ldots,i_k} \\ S_3^1 & S_3^2 & S_3^{1,1} & S_3^{1,2} & S_3^{2,1} & \ldots & S_3^{i_1,\ldots,i_k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_n^1 & S_n^2 & S_n^{1,1} & S_n^{1,2} & S_n^{2,1} & \ldots & S_n^{i_1,\ldots,i_k} \end{bmatrix} \in \mathbb{R}^{n\times 2(2^k-1)}. \tag{13}$$

Table 2
Description of the proposed survival models considered in this study. Predicted survival probability is denoted as $\hat{S}(t)$ and cumulative hazard function as $\hat{H}(t)$, which are the estimates of $S(t)$ (given in Eq. (3)) and $H(t)$ (given in Eq. (4)) respectively.

Model name	Model input	What is predicted
Charge model	*Charge voltage* profiles of the first n cycles were used to extract features in Eq. (14) for each cell.	$\hat{S}(t)$ and $\hat{H}(t)$
Discharge model	*CC discharge voltage* profiles of the first n cycles were used to extract features in Eq. (14) for each cell.	$\hat{S}(t)$ and $\hat{H}(t)$

Then, a path $P^c = \{i, (\mathrm{Sig}^k \circ X_t^{1:n})_c\}$ consisting of cycles $i \in [1, n]$ as the first component and column c of the matrix $\mathrm{Sig}^k \circ X_t^{1:n}$ as the second component was formed. Finally, the truncated signatures of P^c were calculated up to depth k for each column c, and the resulting vectors were stacked horizontally forming a single vector for each cell. Thus for a cell, its corresponding stacked row feature vector is given by

$$f_{\text{cell}} := \left(\mathrm{Sig}^k \circ P^1, \mathrm{Sig}^k \circ P^2, \mathrm{Sig}^k \circ P^3, \ldots, \mathrm{Sig}^k \circ P^{2(2^k-1)}\right) \in \mathbb{R}^{4(2^k-1)^2}, \tag{14}$$

where each of $\mathrm{Sig}^k \circ P^c$ has exactly $2(2^k - 1)$ elements.

The target for survival analysis is a structured array in which each element is the pair (T, δ) where T and δ are the EOL and the binary random variable defined in Eq. (1) respectively. Thus, the target vector $\boldsymbol{y}$ of all 362 cells considered in this research is given by

$$\boldsymbol{y} := \begin{bmatrix} (T_1, \delta_1) \\ (T_2, \delta_2) \\ (T_3, \delta_3) \\ \vdots \\ (T_{362}, \delta_{362}) \end{bmatrix}.$$

3.5. Model training

As described in Section 3.3, the model of choice is the gradient boosting implementation of the Cox PH model. Two models were proposed based on whether they use features extracted from charge or discharge data regime: the *charge*, and *discharge* models; see Table 2 for the summary of each model description. The charge model was trained using the features extracted from the charge voltage profiles of the first n cycles for each cell. On the other hand, the discharge model was built on top of features engineered from the CC discharge voltage.

The rationale behind these proposed models is as follows. When there is data that captures the protocol-to-protocol or cycling conditions variabilities, the charge model can be built to predict the survival and hazard functions of cells. On the other hand, when the available data captures the cell-to-cell variability (a situation where all cells are cycled identically to capture the intrinsic properties of cells), then the discharge model will be preferred. Thus, these two modeling options were proposed to handle the two well-known cycling options and to show their robustness (to different data regimes) and versatility.

For the two models, the 80%–20% train–test split strategy was adopted, and hyperparameter tuning was achieved via a grid search using the C-index as the scoring metric. The train–test splitting was done in a way that each batch maintains the split ratio in the train and test set to ensure unbiased representation. Each model was trained and evaluated across 100 train–test splits (each split was randomly generated). This ensures that accurate and representative performance metrics are obtained for each model. Performance metrics on the test set were estimated using the three metrics defined in Eqs. (9), (11) and (12).

Table 3
Performance metrics of the charge and discharge models on the test set. The mean ($\bar{x}$) and standard deviation (σ) of the metrics were calculated across 100 train–test splits. Both the cumulative dynamic AUC and integrated Brier score were calculated for $t \in [500, 1000]$. For both C-index and cumulative dynamic AUC (both in the range $[0, 1]$), the higher they are the better the model while the reverse is the case for the integrated Brier score.

Model	C-index		Cumulative dynamic AUC		Integrated Brier score	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
Charge	0.7944	0.0281	0.9015	0.0269	0.1122	0.0163
Discharge	0.8098	0.0302	0.9149	0.0248	0.1046	0.0152

4. Results and discussion

4.1. Choosing the signature depth and number of cycles of data for modeling

Central to this study is the choice of the number of cycles of data required to achieve accurate models. To accomplish this, 50 cycles worth of data was first fixed for both the charge and discharge models to choose the right value of signature depth k. As discussed in Section 3.4, $k \in \{2, 3, 4\}$ was considered and the cross-validated results (using the C-index) on the train data are presented in Fig. 4(**a**). The C-index in both charge and discharge models significantly increases as k rises from 2 to 3, however, the difference at $k = 3$ and $k = 4$ is less significant. While this trend may not apply consistently for $k > 4$, $k = 3$ was chosen for signature depth in extracting features from charge and discharge data regimes.

As for choosing the number of cycles n of data for modeling, integer values $n \in [2, 100]$ were considered. Features under each cycle were fed to the charge and discharge models for cross-validation on the training data. The results of this experiment are presented in Fig. 4(**b**). It was observed that each model's performance increases between cycles 2 and 50, but approximately remain the same over the rest of the cycles. In this regard, the optimal cycle number was chosen to be 50 throughout the rest of the experiments performed in this study.

Further in this experiment, it was observed that the charge model appears less sensitive to changes in the selected signature depth and cycle threshold than the discharge model. This is because inter-cycle variability in voltage curves is more significant in the discharge data than in the charge data. As a result, the discharge model is expected to be more sensitive to both changes in signature depth and cycle number; this finding also agrees with the results of Section 4.4.

4.2. Model performance

The performance metrics of the charge and discharge models calculated on the test set are given in Table 3. In both models, the performance across the 100 train–test splits is consistent and this is evident in the low standard deviation (≈0.03 for all models and metrics). In addition, both models perform similarly on the test set and this implies that the modeling methodology employed in this research is robust to different data regimes used for feature extraction. Thus, if either cell-to-cell or protocol-to-protocol variabilities need to be captured, the proposed models can be used for the early prediction of battery survival probability and hazard rate curves.

Figs. 5 and 6 show the predicted survival and cumulative hazard functions for the test cells in each batch. For brevity, only results obtained under the discharge model were presented. It can be seen that each batch has different survival patterns which suggests there is a significant variability in degradation across batches. This trend is also repeated within a batch. The variability in survival can be linked to differences in cells' intrinsic properties as survival is taken as a function of features extracted from the identically discharged voltage curves.

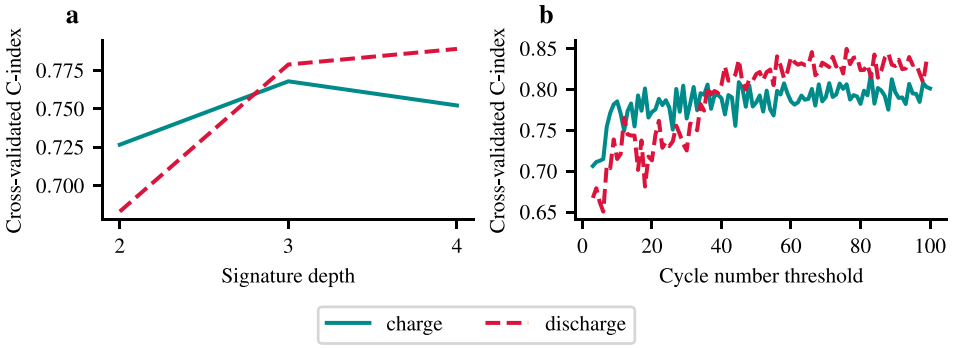


Fig. 4. Choosing optimal (**a**) signature depth for feature extraction and (**b**) number of cycles of data for modeling. For brevity, 50 cycles were fixed to observe the effect of signature depth on model performance. Signature depth $k = 3$ was chosen as optimal depth and was used to extract features under different cycle number thresholds to show the effect of varying cycle numbers on model performance. Lastly, 50 cycles worth of data was chosen as the optimal threshold for all other experiments in this study.

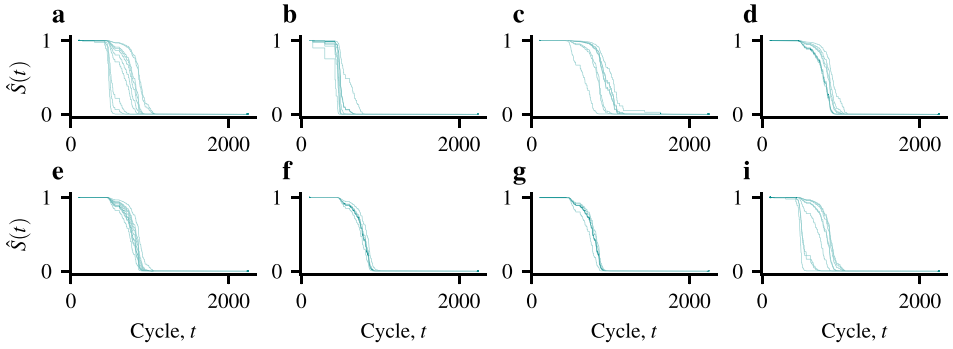


Fig. 5. Predicted survival probabilities for the test data using the discharge model. Survival functions for all cells in each batch ((**a**)-(**i**) correspond to batches b1-b8 respectively) are plotted in a single figure block.

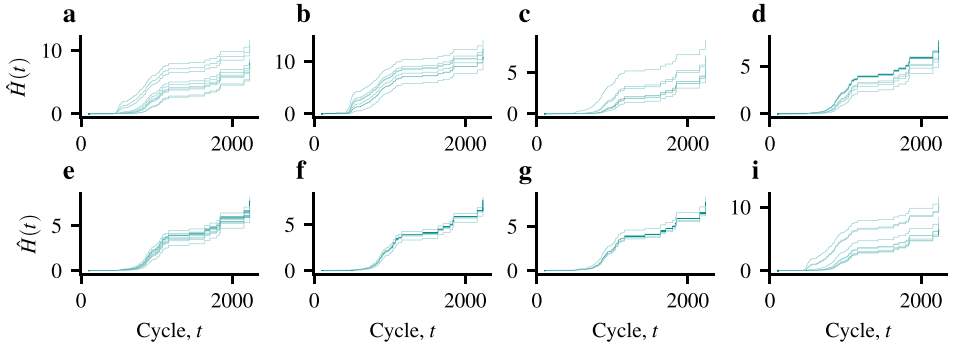


Fig. 6. Predicted cumulative hazards for the test set using the discharge model. Each cumulative hazard is transformed using $\log(1 + \hat{H}(t))$ to see differences within a batch. Cumulative hazards for all cells in each batch ((**a**)-(**i**) correspond to batches b1-b8 respectively) are plotted in a single figure block.

4.3. Effect of number of training samples on model performance

This section focuses on the effect of varying the number of training samples on the models' performances. This experiment is motivated by two practical reasons: (i) a practical model should improve in accuracy as the number of training samples increases; (ii) we want to investigate how the proposed models perform under small training samples. For this experiment, we fixed a train–test split (80%–20%), and then sampled (without replacement) a given fraction of cells from the whole training set. These sampled cells were then used to train the charge and discharge models. The resulting models were evaluated on the fixed test set. We considered sample fractions in the set $\{0.2, 0.3, \ldots, 1.0\}$. Each sample fraction was repeated 100 times and the corresponding metrics were averaged. The results of this investigation are presented in Fig. 7. As expected, both models' accuracy improves (C-index and cumulative dynamic AUC increases while the integrated Brier score decreases accordingly) as the sample fraction increases. The proposed models demonstrate robust performance in our experiments even with limited training data. Specifically, under scenarios with low sample sizes, the model consistently achieves high accuracy (C-index, cumulative dynamic AUC, and integrated Brier score of approximately 0.72, 0.84, and 0.18 for both models) compared to the highest sample fraction of 1.0 (C-index, cumulative dynamic AUC, and integrated Brier score of approximately 0.82, 0.93, and 0.10 for both models), indicating its effectiveness in data-scarce environments. This performance can be attributed to the quality of features (signatures) extracted from the charge and discharge regimes.

4.4. Low-cycle prediction experiment

The feature extraction technique introduced in Section 3.4 allows for the use of different numbers of cycles of data when training models

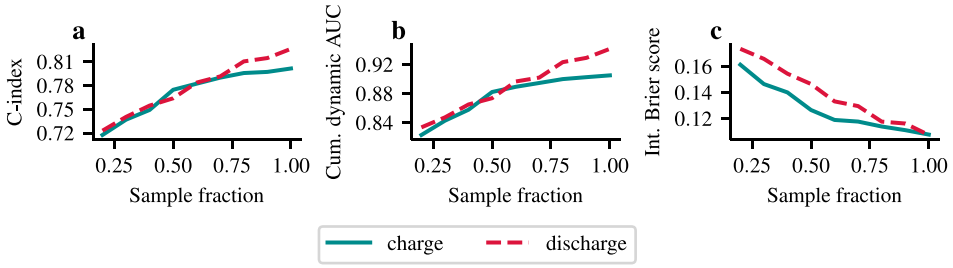


Fig. 7. Results of the training sample increment experiment. The charge and discharge models were trained on varying sample fractions from the whole training set and then validated on a fixed test set. The averaged metric value is presented at each sample fraction for (**a**) C-index, (**b**) cumulative dynamic AUC, and (**c**) integrated Brier score.

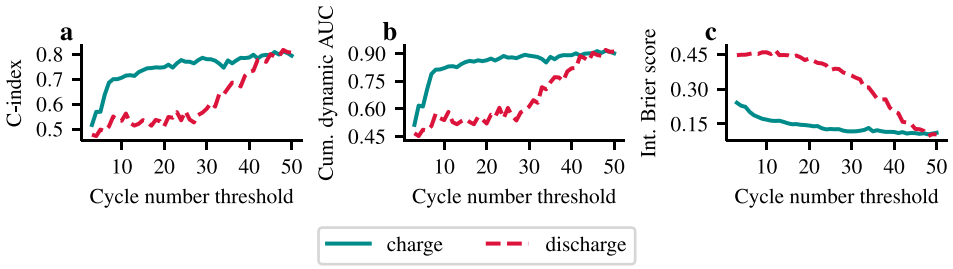


Fig. 8. Results of the low-cycle prediction experiments. The charge and discharge models were trained on features extracted from 50 cycles of voltage profiles but validated on features obtained under different cycle number thresholds. The charge model was observed to be more robust under low-cycle data situations at inference time. The averaged metric value is presented at each cycle number threshold for (**a**) C-index, (**b**) cumulative dynamic AUC, and (**c**) integrated Brier score.

and when using trained models for inferences. This is because the final feature vector of Eq. (14) will always have a fixed length (for a given signature depth) regardless of the number of rows of the matrix in Eq. (13). However, as the number of rows of the matrix increases (i.e. increasing the number of cycles of data used), more information is expected to be derived from the voltage of a given data region. In this section, we investigate the performance of the proposed models when they are trained on the first 50 cycles worth of data but making predictions with a lower number of cycles of data. This experiment is applicable in a low-data situation at inference time. Each model was trained on training data with 50 cycles worth of voltage profiles and then evaluated on test data with an increasing number of cycles (integer cycle numbers $n \in [3, 50]$ were considered). This procedure was repeated 100 times for each testing cycle using a different random train–test split at each repetition. The resulting performance metrics were averaged over the considered repetitions.

The results of this experiment are presented in Fig. 8. For the charge model, model performance improves steadily in the first 10 cycles but becomes less significant over the rest of the cycle numbers. On the other hand, the discharge model's performance rises slightly in the first 10 cycles, remains approximately the same between cycles 10 and 30, and eventually increases from cycle 20 until it plateaus at the 40-cycle mark. These results suggest that the charge model shows a significant level of robustness in a low-data prediction situation with C-index and cumulative dynamic AUC climbing over 0.75 in $10 \leq n \leq 30$. In contrast, when using low-cycle numbers ($n \leq 30$) in the discharge regime, the C-index and cumulative dynamic AUC values are approximately 0.5, indicating a limited predictive accuracy in these low-cycle scenarios. The reason behind the difference in robustness is not far-fetched. Cycle-to-cycle variability in voltage curves is more pronounced in the discharge data region than the charge counterpart, thus the resulting model is expected to be more sensitive to the cycle number drift at inference time.

4.5. Effect of voltage curve sparsity on model performance

In this section, we investigate the impact of data sparsity on the predictive performance of the proposed models. By data sparsity, we mean systematically considering only voltage values recorded at every index step $i \in \{1, 2, 3, \ldots, 10\}$. For instance if $V = [2.0, 2.5, 3.0, 3.5, 4.0]$ and $i = 2$, the corresponding sparse $V_{\text{sparse}} = [2.0, 3.0, 4.0]$. Note that the case $i = 1$ gives the original voltage curve and this was considered for benchmark purposes. To obtain more representative results, each case of i was repeated 100 times using a different random train–test split at each repetition. We considered two cases of sparsity: (i) sparsity at the training data level, where sparsity was introduced to voltage curves used for training models, and (ii) sparsity at the test data level, where sparsity was considered in the voltage curves used for model testing.

The results of this investigation are presented in Fig. 9. In both sparsity cases, the discharge model was robust to every resulting sparse voltage curve and its performance was relatively the same compared to the original voltage curves (when $i = 1$). In contrast to the charge model, model performance decreases as the degree of voltage curve sparsity increases, suggesting that the charge model is less robust. The difference in sparsity is most likely due to the induced variation in the voltage curves. Because the charge model used the charge voltage values and variations in the charge data region are induced by the different charging protocols, it will be more sensitive to the distribution of recorded measurements and value omission. In addition, within a given charge voltage curve, at least 3 different charging policies were employed, and thus more variations are introduced into voltage value distributions. However, the discharge model utilized data from the discharge data region, where variabilities are only induced by the intrinsic differences of the cells as the cells were discharged identically. As a result, the corresponding voltage distribution geometry will be conserved even in the presence of systematic sparseness. Another factor that is likely to contribute to the significant difference in sparsity robustness is the duration of each data region considered. The CC discharge regime takes approximately 15 min for the maximum cycle 50 while that of the charge takes between 9 to 13 min (see Fig. 1). Given that data are recorded at the same rate in both regions, more points are expected to be concentrated at the CC discharge than at the charge region, thus resulting in a different degree of sparsity at the same index step.

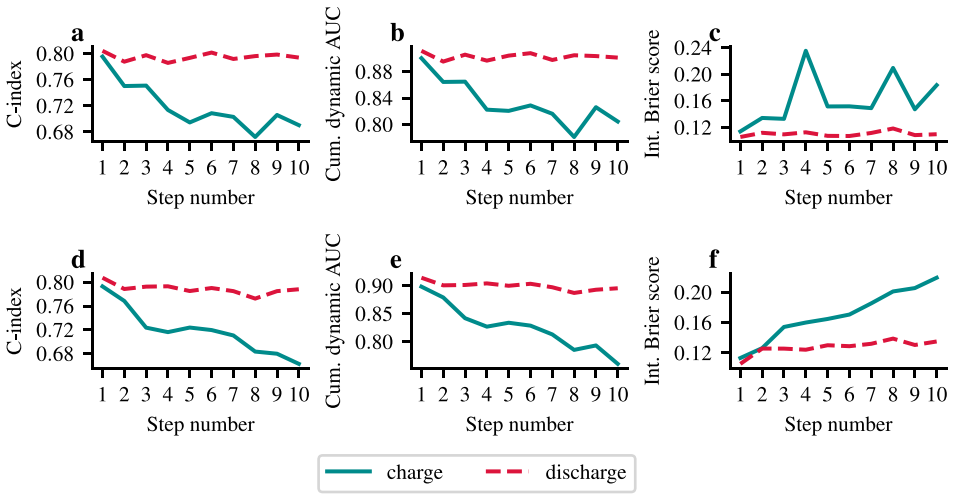


Fig. 9. Effect of voltage curve sparsity on the proposed models' performance. Sparsity was introduced at training (**a-c**) and test (**d-f**) data levels. It was observed that the discharge model was more robust to the systematic sparsity introduced at both levels.

5. Conclusion

A study has been carried out using different battery data regimes to develop battery survival models. We have proposed battery survival models which work seamlessly with features extracted from either the charge or discharge regime. This is outstanding as the models cater for both protocol-to-protocol differences (using different cycling conditions) and cell-to-cell variabilities (when cells are cycled identically). The models are meant to be used for making predictions at the early life of batteries taking only the first 50 cycles of the voltage profiles. In addition, the models are robust to small training samples and decent/accurate predictions can be achieved even at low training sample rates. When more training data are available, the proposed models' performances have been shown to improve steadily. All models were trained on an HP EliteBook 850 G7 Notebook PC with 32 GiB RAM and 1.0 TB Disk capacity. Each model training takes approximately 20.78 s (wall-clock time) to run for the dataset considered in this study.

We conducted additional real-life application-driven experiments: (i) robustness to low-cycle data at prediction time, and (ii) robustness to data sparsity at both training and inference time. Both proposed models show a significant degree of robustness when given less than 50 cycles worth of data at prediction time. In this case, the charge model showed a more significant robustness. On data sparsity, we carried out a systematic sub-sampling of voltage values at different degrees both at training and test data levels. The discharge model accuracy maintains its predictive performance at all sparsity degrees and levels while the charge model performance decays as the sparsity rate rises. This is an indication that the discharge model is preferable for a sparsity-aware application.

The predicted cells' survival and cumulative hazard functions are great tools for battery manufacturers and fleet operators. They can be used to adjust the battery warranty and schedule data-informed maintenance. In terms of future research, other known data regimes such as the Electrochemical Impedance Spectroscopy (EIS) and Reference Performance Test (RPT) data could be investigated for potential data for feature extraction to further show the versatility and robustness of the proposed methods.

Methods

Introduction to signature method

The signature method is a set of feature extraction strategies for multivariate time series data. It takes a continuous, piecewise smooth path representing the multivariate time series and transforms it into a sequence of real numbers that encode the area, segmental, and geometric information of such path over a real interval. The truncated sequence (of fixed length) of the transform can then be used as features for any machine learning task such as supervised learning. Its application to machine learning problems has been studied extensively. This includes, among others, sound compression [29], handwriting character recognition [30], detecting hand-to-face touch with supervised contrastive learning [31], and early detection of sepsis in patients [32]. More recently, its application in the prediction of battery lifetime was introduced in [7] where signatures of a two-dimensional path consisting of the time and the discharge voltage of the CC were used as features for machine learning models that predict the full capacity and IR curves. In light of the success above of the signature method, we adopted it in this research with an additional layer to capture signature changes as a function of cycle number; see Section 3.4 for how the signature method was deployed in this research.

After giving high-level information about the signature, we delve into its mathematical build-up and structures. We first define a path, iterated integrals and finally present a formal definition of path signature.

Definition 5.1 (*Path*)**.** A d-dimensional real-valued path X_t is a continuous mapping from a real interval $[a,b] \ni t$ into a d-dimensional real space:

$$X_t = \{X_t^1, X_t^2, X_t^3, \dots, X_t^d\} : [a,b] \mapsto \mathbb{R}^d. \tag{15}$$

Definition 5.2 (*Iterated Integral [17]*)**.** Suppose X_t is a real d-dimensional, continuous piecewise smooth path defined on $[a,b]$ as given in Eq. (15). Then the kth level iterated integral of X_t is given by

$$S(X_t)_{a,t}^{i_1,\dots,i_k} := \int_{a<t_k<t} \cdots \int_{a<t_1<t_2} dX_{t_1}^{i_1} \dots dX_{t_k}^{i_k}, \tag{16}$$

where $i_1, \dots, i_k \in \{1, \dots, d\}$.

Definition 5.3 (*Signature [17]*)**.** The path signature of X_t over $[a,b]$ is defined as an infinite sequence of all possible calculated iterated integrals defined in Eq. (16):

$$\text{Sig}\circ X_t := \left(1, S(X_t)_{a,b}^1, S(X_t)_{a,b}^2, S(X_t)_{a,b}^{1,1}, S(X_t)_{a,b}^{1,2}, \dots, S(X_t)_{a,b}^{i_1,\dots,i_k}, \dots\right).$$

If the sequence is truncated at depth k (as in the case of feature extraction for machine learning task), then we obtain a truncated

signature of X_t (of fixed length) given by

$$\mathrm{Sig}^k \circ X_t := \left(1, S(X_t)^1_{a,b}, S(X_t)^2_{a,b}, S(X_t)^{1,1}_{a,b}, S(X_t)^{1,2}_{a,b}, \dots, S(X_t)^{i_1,\dots,i_k}_{a,b}\right).$$

Usually, the first value of the sequence $\mathrm{Sig}^k \circ X_t$ is omitted during feature extraction as this will be constant for all samples. An example of the signature of a two-dimensional continuous piecewise path $X_t = \{f(t), g(t)\}$ calculated up to depth $k = 2$ is

$$\begin{aligned}\mathrm{Sig}^2 \circ X_t = \Bigg(1, \int_a^b f' \, dt, \int_a^b g' \, dt, \int_a^b \left[\int_a^{t_2} f'(t_1)\, dt_1\right] f'(t_2) dt_2, \\ \int_a^b \left[\int_a^{t_2} f'(t_1)\, dt_1\right] g'(t_2)\, dt_2, \\ \int_a^b \left[\int_a^{t_2} g'(t_1)\, dt_1\right] f'(t_2)\, dt_2, \int_a^b \left[\int_a^{t_2} g'(t_1) dt_1\right] g'(t_2) dt_2\Bigg).\end{aligned}$$

Properties of signature transform

The path signature has some properties that make it natural to extract information from a time series data that can be carved into a path. These properties also enable signatures to be used for various machine-learning tasks. We state and discuss three of these properties in this section which most relate to feature extraction and machine learning; see [17,33] for the details of other properties. The first property focuses on the uniqueness of the signature of a path. This essentially implies that there is no loss of information from a path when its signature is calculated as its encoding.

Proposition 5.1 (*Uniqueness of Path Signature [17,34]*). *Suppose X_t is a real d-dimensional, continuous piecewise smooth path defined on $[a, b]$ as given in Eq. (15). Then Sig$\circ X_t$ uniquely characterizes X_t up to translation.*

Next, a function of the path X_t can be approximated by a linear function of its signature. This is called the universal nonlinearity property of signature [33,35]. This property is applicable in a supervised learning task. For a machine learning problem of predicting a label y by learning a function f of the path X_t (i.e., $f(X_t) = y$), the universal nonlinearity implies that we can instead learn a linear function L of the signature of X_t without losing information about f. This is powerful as f might be highly nonlinear and X_t might also be highly dimensional with a large number of data points.

Proposition 5.2 (*Universal Nonlinearity [33,35]*). *Suppose X_t is a real d-dimensional, continuous piecewise smooth path defined on $[a, b]$ as given in Eq. (15). Let f be a real-valued continuous function defined on all X_t and let C be a compact set of all X_t. If the translation invariance is removed for all $X_t \in C$, then there exists a linear function L such that for all $X_t \in C$,*

$$\left|f(X_t) - L(Sig\circ X_t)\right| < \epsilon, \quad \epsilon > 0.$$

Lastly, the signature is invariant to time reparameterizations and translation. This property is a direct consequence of the signature being a sequence of iterated path integrals. Invariance to time reparametrization is important when encoding information about the order in which a given measured data profile arrives at a point without caring about when it precisely arrives at such a point. Thus, if one desires features that are both invariant to time parameterization and translation of the underlying path, then signature transform is a natural choice.

Proposition 5.3 (*Time Reparameterization Invariance [17,36]*). *Suppose X_t is a real d-dimensional, continuous piecewise smooth path defined on $[a, b]$ as given in Eq. (15). Let $\psi : [a, b] \mapsto [a, b]$ be a reparametrization (a continuously differentiable, non-decreasing, and subjective function). Then Sig$\circ X_t$ = Sig$\circ X_{\psi(t)}$.*

CRediT authorship contribution statement

Rasheed Ibraheem: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Timothy I. Cannings:** Writing – review & editing, Writing – original draft, Validation. **Torben Sell:** Writing – review & editing, Writing – original draft, Validation. **Gonçalo dos Reis:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Code availability

The codes for all modeling and experiments performed in this research can be found in this well-documented GitHub repository: https://github.com/Rasheed19/battery-survival.

Funding

This project was funded by an industry-academia collaborative grant EPSRC EP/R511687/1 awarded by *EPSRC & University of Edinburgh*, United Kingdom program *Impact Acceleration Account (IAA)*.

R. Ibraheem is a Ph.D. student in EPSRC's MAC-MIGS Centre for Doctoral Training. MAC-MIGS is supported by the UK's Engineering and Physical Science Research Council (grant number EP/S023291/1).

G. dos Reis acknowledges partial support from the FCT – Fundação para a Ciência e a Tecnologia, Portugal, I.P., under the scope of the projects UIDB/00297/2020 (https://doi.org/10.54499/UIDB/00297/2020) and UIDP/00297/2020 (https://doi.org/10.54499/UIDP/00297/2020) (Center for Mathematics and Applications, NOVA Math). G. dos Reis acknowledges support from the Faraday Institution, United Kingdom (grant number FIRG049).

Additional information

This work has no supplementary information file.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used for all the experiments and modeling in this research is from [1,18] and can be downloaded from https://data.matr.io/1/.

References

[1] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. Nature Energy 2019;4(5):383–91. http://dx.doi.org/10.1038/s41560-019-0356-8.

[2] Paulson NH, Kubal J, Ward L, Saxena S, Lu W, Babinec SJ. Feature engineering for machine learning enabled early prediction of battery lifetime. J Power Sources 2022;527:231127. http://dx.doi.org/10.1016/j.jpowsour.2022.231127.

[3] Strange C, dos Reis G. Prediction of future capacity and internal resistance of li-ion cells from one cycle of input data. Energy AI 2021;5:100097. http://dx.doi.org/10.1016/j.egyai.2021.100097.

[4] Strange C, Ibraheem R, dos Reis G. Online lifetime prediction for lithium-ion batteries with cycle-by-cycle updates, variance reduction, and model ensembling. Energies 2023;16(7):3273. http://dx.doi.org/10.3390/en16073273.

[5] Hong J, Lee D, Jeong E-R, Yi Y. Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning. Appl Energy 2020;278:115646. http://dx.doi.org/10.1016/j.apenergy.2020.115646.

[6] Ibraheem R, Strange C, dos Reis G. Capacity and internal resistance of lithium-ion batteries: Full degradation curve prediction from voltage response at constant current at discharge. J Power Sources 2023;556:232477. http://dx.doi.org/10.1016/j.jpowsour.2022.232477.

[7] Ibraheem R, Wu Y, Lyons T, dos Reis G. Early prediction of lithium-ion cell degradation trajectories using signatures of voltage curves up to 4-minute sub-sampling rates. Appl Energy 2023;352:121974. http://dx.doi.org/10.1016/j.apenergy.2023.121974.

[8] Attia PM, Bills A, Planella FB, Dechent P, Dos Reis G, Dubarry M, et al. Knees in lithium-ion battery aging trajectories. J Electrochem Soc 2022;169(6):060517.
[9] Strange C, Li S, Gilchrist R, dos Reis G. Elbows of internal resistance rise curves in li-ion cells. Energies 2021;14(4):1206. http://dx.doi.org/10.3390/en14041206.
[10] David GK, Mitchel K. Survival analysis: a self-learning text. Spinger; 2012.
[11] Kuhajda D. Using survival analysis to evaluate medical equipment battery life. Biomed Instrum Technol 2016;50(3):184–9. http://dx.doi.org/10.2345/0899-8205-50.3.184.
[12] Kaplan EL, Meier P. Nonparametric estimation from incomplete observations. J Amer Statist Assoc 1958;53(282):457–81. http://dx.doi.org/10.1080/01621459.1958.10501452, http://eudml.org/doc/39555.
[13] Machin D, Cheung YB, Parmar M. Survival analysis: a practical approach. John Wiley & Sons; 2006.
[14] Voronov S, Frisk E, Krysander M. Data-driven battery lifetime prediction and confidence estimation for heavy-duty trucks. IEEE Trans Reliab 2018;67(2):623–39. http://dx.doi.org/10.1109/TR.2018.2803798.
[15] Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. Ann Appl Stat 2008;2(3):841–60. http://dx.doi.org/10.1214/08-AOAS169.
[16] Santhira Sekeran M, Živadinović M, Spiliopoulou M. Transferability of a battery cell end-of-life prediction model using survival analysis. Energies 2022;15(8). http://dx.doi.org/10.3390/en15082930, http://eudml.org/doc/39555.
[17] Chevyrev I, Kormilitzin A. A primer on the signature method in machine learning. 2016, http://dx.doi.org/10.48550/ARXIV.1603.03788, http://eudml.org/doc/39555.
[18] Attia PM, Grover A, Jin N, Severson KA, Markov TM, Liao Y-H, et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. Nature 2020;578:397–402.
[19] Geslin A, van Vlijmen B, Cui X, Bhargava A, Asinger PA, Braatz RD, et al. Selecting the appropriate features in battery lifetime predictions. Joule 2023;7(9):1956–65. http://dx.doi.org/10.1016/j.joule.2023.07.021, http://eudml.org/doc/39555.
[20] Davidson-Pilon C. lifelines: survival analysis in python. J Open Source Softw 2019;4(40):1317. http://dx.doi.org/10.21105/joss.01317.
[21] Breslow N. Covariance analysis of censored survival data. Biometrics 1974;89–99.
[22] Pölsterl S. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. J Mach Learn Res 2020;21(212):1–6, http://eudml.org/doc/39555.
[23] Kvamme H, Borgan Ø, Scheel I. Time-to-event prediction with neural networks and cox regression. J Mach Learn Res 2019;20(129):1–30, http://eudml.org/doc/39555.
[24] Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC Med Res Methodol 2018;18:1–12.
[25] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. J Mach Learn Res 2012. http://dx.doi.org/10.48550/ARXIV.1201.0490, http://eudml.org/doc/39555.
[26] Friedman JH. Greedy function approximation: A gradient boosting machine. Ann Statist 2001;29(5). http://dx.doi.org/10.1214/aos/1013203451.
[27] Friedman JH. Stochastic gradient boosting. Comput Statist Data Anal 2002;38(4):367–78. http://dx.doi.org/10.1016/S0167-9473(01)00065-2, nonlinear Methods and Data Mining, http://eudml.org/doc/39555.
[28] Reizenstein JF, Graham B. Algorithm 1004: The iisignature library: Efficient calculation of iterated-integral signatures and log signatures. ACM Trans Math Software 2020;46(1):1–21. http://dx.doi.org/10.1145/3371237.
[29] Lyons TJ, Sidorova N. Sound compression: a rough path approach. In: Proceedings of the 4th International Symposium on Information and Communication Technologies. Trinity College Dublin; 2005, p. 223–8.
[30] Li C, Zhang X, Jin L. Lpsnet: a novel log path signature feature based hand gesture recognition framework. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017, p. 631–9.
[31] Ibrahim MR, Lyons T. Facetouch: Detecting hand-to-face touch with supervised contrastive learning to assist in tracing infectious diseases. Plos one 2024;19(6):e0288670.
[32] Cohen SN, Foster J, Foster P, Lou H, Lyons T, Morley S, et al. Subtle variation in sepsis-iii definitions markedly influences predictive performance within and across methods. Sci Rep 2024;14(1):1920.
[33] Kidger P, Bonnier P, Perez Arribas I, Salvi C, Lyons T. Deep signature transforms. Adv Neural Inf Process Syst 2019;32.
[34] Hambly B, Lyons T. Uniqueness for the signature of a path of bounded variation and the reduced path group. Ann of Math 2010;171(1):109–67. http://dx.doi.org/10.4007/annals.2010.171.109.
[35] Arribas IP. Derivatives pricing using signature payoffs. 2018, arXiv preprint arXiv:1809.09466.
[36] Lyons TJ. Differential equations driven by rough signals. Rev Mat Iberoam 1998;14(2):215–310, http://eudml.org/doc/39555.