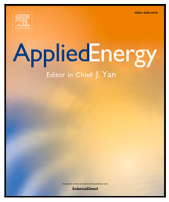
Applied Energy 352 (2023) 121974

Contents lists available at ScienceDirect

Applied Energy

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

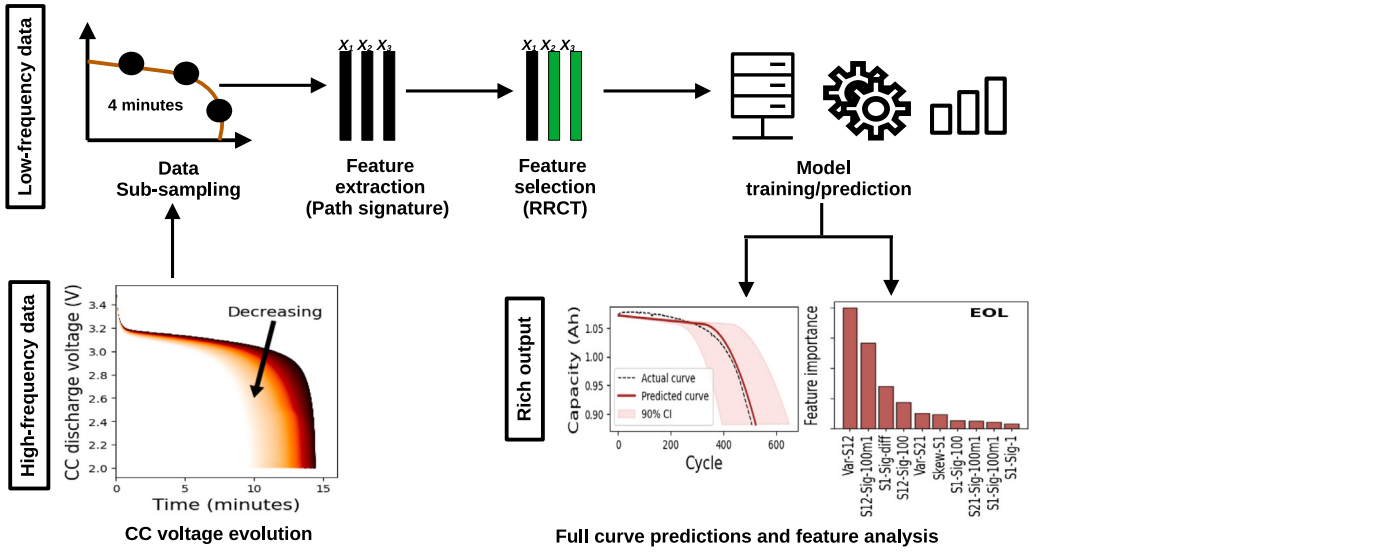


Early prediction of Lithium-ion cell degradation trajectories using signatures of voltage curves up to 4-minute sub-sampling rates

Rasheed Ibraheem [a], Yue Wu [b], Terry Lyons [c], Gonçalo dos Reis [a,d,*]

[a] Maxwell Institute for Mathematical Sciences, School of Mathematics, University of Edinburgh, The Kings buildings, Edinburgh, EH9 3JF, Scotland, UK
[b] Department of Mathematics and Statistics, University of Strathclyde, 26 Richmond St, Glasgow, G1 1XH, Scotland, UK
[c] University of Oxford, Andrew Wiles Building, Woodstock Rd, Oxford, OX2 6GG, UK
[d] Centro de Matematica e Aplicacoes (CMA), Faculdade de Ciencias e Tecnologia, Campus da Caparica, Caparica, 2829-516, Portugal

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Capacity degradation
Path signature methodology
Voltage response under constant current at discharge
Lithium-ion cells
Machine learning
Remaining useful life

ABSTRACT

Feature-based machine learning models for capacity and internal resistance (IR) curve prediction have been researched extensively in literature due to their high accuracy and generalization power. Most such models work within the high frequency of data availability regime, e.g., voltage response recorded every 1–4 s. Outside premium fee cloud monitoring solutions, data may be recorded once every 3, 5 or 10 min. In this low-data regime, there are little to no models available. This literature gap is addressed here via a novel methodology, underpinned by strong mathematical guarantees, called 'path signature'.

This work presents a feature-based predictive model for capacity fade and IR rise curves from only constant-current (CC) discharge voltage corresponding to the first 100 cycles. Included is a comprehensive feature analysis for the model via a relevance, redundancy, and complementarity feature trade-off mechanism. The ability to predict from subsampled 'CC voltage at discharge' data is investigated using different time steps

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

https://doi.org/10.1016/j.apenergy.2023.121974
Received 14 May 2023; Received in revised form 18 July 2023; Accepted 15 September 2023
Available online 28 September 2023

ranging from 4 s to 4 min. It was discovered that voltage measurements taken at the end of every 4 min are enough to generate features for curve prediction with End of Life (EOL) and its corresponding IR values predicted with a mean absolute percentage error (MAPE) of approximately 13.2% and 2.1%, respectively. Our model under higher frequency (4 s) produces an improved accuracy with EOL predicted with an MAPE of 10%. Full implementation code publicly available.

1. Introduction

High energy density and long life are some of the prominent properties of lithium-ion batteries, which make them preferable to other cells from different materials [1,2]. In addition to these desirable characteristics, they are extensively used in powering portable electronic devices, cars, and more interestingly, heavy-duty vehicles which are a newly growing application field. Like any other batteries, lithium-ion batteries' ability to retain charge decreases over time due to factors including charging and discharging processes, storage conditions (such as ambient temperature), and the nature of maintenance routines. As a result, it becomes necessary to study the degradation process and prognostics of lithium-ion batteries for performance and cycle life optimization, safety, maintenance, and replacement cost forecast.

There are various methods that can be applied to the study of lithium-ion battery life prognostics. Among the popular strategies is the use of machine learning techniques for the prediction of capacity fade and internal resistance (IR) curves. This approach draws from feature generation, feature selection, and model building carried out on the battery use data measuring quantities including capacity, IR, voltage, current, and temperature. Models obtained from such an approach have been reported as accurate, robust, and versatile. For instance, machine learning models that address early life prediction of capacity and IR curves have been reported in [3,4], and those which focus on the prediction of certain points on the capacity curve such as End of life (EOL) can be found in [5–13].

One of the important battery health parameters widely predicted in the literature is the EOL (which is here defined as the cycle number at which a cell's capacity drops to 80% of its nominal capacity taking the first cycle as a reference). On this, many methods are based on the rich data sets published in Severson et al. [5] and Attia et al. [6]. Three methods (namely Variance, Discharge, and Full) were proposed in [5] to predict the EOL of cells in the batches of data. The authors extracted features from the first 100 cycles of data including discharge capacity-voltage curve $Q(V)$, voltage, current, temperature, and IR (in which the in-cycle measurement was carried out at approximately 4-s intervals) to build a regularized linear regression model. Their best model achieved a mean absolute percentage error (MAPE) of 7.5% on test data. Using the same measurement frequency and input number of cycles (but only $Q(V)$), a feature-based convolutional neural network (CNN) model was built in [4] to predict the entire capacity curve through an exponential curve parameterization with EOL predicted within an MAPE of approximately 22%. A similar work that used the gradient of $Q(V)$ corresponding to the first 100 cycles is [14] where a feature-based CNN was developed to predict EOL with an average MAPE of 11.7%. Another approach called broad extreme learning machine that used state of health (SOH), $Q(V)$, IR, and charge time corresponding to the first 100 cycles with a 4-s data recording was introduced in [15] to predict the EOL of cells in the considered batches; an average MAPE of 9% was recorded. Similar to this work (using the same number of input cycles but utilized temperature in place of charge time) is [16] where the Gradient Boosting Regression Trees algorithm was used to predict the EOL with an MAPE of 7% on test data. The use of deep learning techniques has also been very successful with the 'explainability' discussion still ongoing — see [17] for a wider discussion and achievements.

All the above-mentioned works rely heavily on high-frequency battery use data measured at frequent intervals such as 1–4 s. However, in real-world scenarios and especially outside frameworks of premium fee cloud monitoring systems for diagnostics and prognostics, data is recorded in wider time gaps (3-10 min) [18]. This poses a loss of accuracy for models sensitive to data sub-sampling (which only work in the rich data regime) and sideline sub-industries that are not able (yet) to tap into cloud BMS systems.

Sub-sampling at a less frequent time is an advantage to any cloud BMS which transmits data more frequently such as 10-30s. Frequent data transmission increases the cost and complexity of the BMS: cloud transmission of high-frequency data requires additional power sources, hardware, and software components. In addition, frequent data transmission can also increase the risk of data loss or corruption, particularly if the BMS is transmitting wirelessly. If there is interference or signal loss, the BMS may not be able to transmit accurate data, which could lead to errors in data processing for model input. Further, sparse data measurements decrease the computational time for data processing for model input and modeling process.

It is reported in [3] that sub-sampling the constant-current (CC) discharge voltage at a time step exceeding one-minute results in a significant loss of model quality. This research focuses on building robust machine learning models that predict the entire capacity and IR degradation curves using battery use data measured at higher and less frequent intervals. In line with [3], a machine learning approach taking as input only the CC voltage response discharge phase is explored — the choice of discharge phase is purely due to constraints on existing data and independent of the modeling.

The data used for this study consists of consistently controlled charge and discharge protocols. However, the discharge profile of each of the measured quantities was taken in CC situations. For many real-world scenarios where batteries are deployed, both the charging and discharging parts might be less helpful in predicting various battery life prognostics. For instance, the charging component of electric vehicles (EVs) use is totally controlled by the nature of the charger (its maximum output energy) and battery management system (BMS) [19]. On the contrary, the discharge component relies on the chosen route, level of traffic, and driver habits, which make it nonmonotonic. Thus, in the case of EVs, it is practical to design prognostics based on the *charging* component. With respect to the situation where the *discharging* component is consistent (such as storage for renewable energy), the charging component depends heavily on the availability of natural sources such as the intensity of the sun in solar panels and wind speed in wind turbines; however, the quantity of derived energy is controlled and monitored [20].

Two technical innovations are considered here. One is a mathematical innovation to deal with the subsampled data, i.e., the novel path signature method for feature generation. The other is a novel feature selection approach called Relevance, Redundancy, and Complementarity Trade-off (RRCT) to select the best features for model building.

Path signature is a rich mathematical structure originating from the field of rough path theory. It is characterized by the ability to extract high-level information from a stream of data using a few summary parameters, which can then be used as features for a machine learning model. From a mathematical point of view, the *signature* provides a graded and faithful description of a curve (up to appropriate reparameterizations) by locally removing the need to look at its fine structure and summarizing it over short intervals. Linear functionals on the signature form a type of algebra that separates points and therefore, they form a basis for continuous functions on compact sets of curves. In short, using this powerful structure, the problem of learning a complex,

highly non-linear function on a dataset of irregular time series can be replaced by a simple, two-step procedure: (1) extract features from the stream by computing its signature and (2) perform regression on the signature features.

The path signature structure has been used successfully in various machine learning tasks such as the analysis of current–voltage response for non-intrusive load monitoring in the identification of electric appliances from a signal [21]; searching for hidden patterns in trading strategies in financial data [22]; sound compression [23], and handwriting character recognition [24] — the evidence of path signatures outperforming classical time-series analysis is clear. Coupled with the success of signatures across various data streams, the motivation to use signatures in this study stems from the nature of the cycling data used and the signature invariance under time reparametrizations [25] (needed to compensate the loss of information from sub-sampling). The time series of discharge voltages (under CC) is characterized by decreasing monotonicity with respect to time and thus taking voltage measurements with monotone but different time values will be insensitive to signatures. In fact, one of the most important features of the signature of a stream is its insensitivity to the choice of sample times provided those samples are taken at a high enough resolution. This study investigates the maximum sample rate at which adequate information can be derived from the signatures based on the data used.

In terms of feature selection, "Relevance, Redundancy, and Complementarity Trade-off" (RRCT) algorithm is a filter-type feature selection tool characterized by robustness and high computation efficiency. The RRCT has been proven to work well with different types of datasets across both regression and classification problems [26] and is more efficient than selection methods using only Pearson correlation. Thus, the combination of path signature and RRCT is natural to efficiently summarize a data stream with a few parameters, and only keeps parameters with the highest impact on the predictions. This choice reduces model complexity and enhances ease of deployment and maintenance.

The rest of the sections of this paper are organized as follows: Section 2 provides information on battery cycling data used in this study, and Section 3 illustrates the various techniques for feature engineering and model building. Experimental results alongside their discussion are presented in Section 4 with concluding remarks in Section 5.

2. Data description

The data of this study is the same as that of the companion work [3] and comes from [5,6,27]. They consist of extracted measurements of CC discharge voltage, capacity, and IR which correspond to 158 lithium iron phosphate (LFP)/graphite A123 APR18650M1A cells. These cells are cycled under similar conditions (in terms of ambient temperature and discharging protocol), each with an initial capacity and voltage of 1.1 Ah and 3.3 V respectively.

LFP battery, or equivalently lithium iron phosphate battery, belongs to the family of lithium-ion batteries which uses lithium iron phosphate and graphite carbon as the cathode and anode materials respectively. As described in [5], all cells were cycled at a constant temperature of 30°C and the same discharging protocol, but with different fast-charging policies. Over 80 different charging policies were employed and cells were charged with one of these policies from 0% to 80% state-of-charge (SOC). Each charging policy represents a C-rate[1] applied to the cells at various ranges of SOC levels. For instance, a two-step charging protocol could be as follows: charge a cell at $6C$ from 0% to 50% SOC then apply a C-rate of $4C$ from 50% to 80% SOC. Overall, the charging time for each of the cells is in the range of 9 to 13.3 min, and all considered cells were charged with the same C-rate ($1C$ constant current–constant voltage, CC–CV) from 80% to 100% SOC (with a maximum voltage of 3.6 V). As mentioned earlier, all cells were discharged at the same C-rate of $4C$ CC–CV to a minimum voltage of 2.0 V; further details about the cycling procedure and other properties can be found in [5,6,27]. The generated data is presented in eight batches, namely batches 1 to 8. Each batch has a slightly different cycling procedure with respect to the rest time during charging up to 80% and after discharging (see the method section of [5] for the reported times). The data accompanied by each of the batches are put in three categories namely: descriptors, summary, and cycle. Cell descriptors provide information about the cycling policy, cycle life, barcode, and channel, while the summary includes data on a per-cycle basis comprising cycle number, discharge capacity, charge capacity, internal resistance, maximum temperature, average temperature, minimum temperature, and charge time. Cycle data present information during a given cycle (in-cycle data) and include directly measured and interpolated/derived quantities. The directly measured data consist of time, charge capacity, current, voltage, temperature, and discharge capacity while the derived quantities include a change in discharge capacity with voltage (dQ/dV), linearly interpolated capacity, and linearly interpolated temperature. In line with [3], the same cell naming convention bN_bcN_c is adopted for a cell from batch number N_b and cell number N_c. This work is restricted to cells from batches 1, 2, 3, and 8 because they were all cycled to EOL (taken as the cycle number corresponding to 80% of initial capacity). Since batch 8 does not contain per-cycle measurement of IR, the substitute generated data from [27] is used. In addition, cells that live less than 300 cycles and more than 1200 cycles were excluded from having consistent data across batches; see Fig. 1(b) for the distribution of cycle lives of cells in the training set.

This study's in-cycle extracted measurement of CC discharge voltage forms the basis of feature generation. The average discharging time for CC-controlled situations is approximately 15 min. This time window is similar to that of [28], where the first 30 points on the charging voltage curve collected over 10 min were used to predict the entire voltage curve via a deep neural network (DNN). As described in Fig. 1(a), cells' voltage curve decreases progressively with cycle number, and thus worthwhile to exploit this trend to engineer features that can capture the pattern and be used as predictors for several battery life prognostics. As for the capacity measurements, they were used to extract the knee-onset (k-o) and knee-point (k-p) [8] using the Bacon–Watt models [8,27] in line with the techniques discussed in [29, Section 3]. In the same manner, the IR measurements were used to obtain the elbow-onset (e-o) and elbow-point (e-p) [27]. These points along with EOL (with their corresponding capacity and IR values) were used to describe and predict the full curves in each case; full details are provided in Section 3.

3. Methodology and modeling process

This section presents methods adopted for feature generation, feature selection, and the machine learning modeling. The impact of varying cycle numbers on the predictive power of the generated features is demonstrated, and the details of the technique chosen for entire curve prediction are provided. The key notion of the analysis is the extraction of appropriate features from the time series of discharge voltages using signatures over the observed interval.

3.1. A primer on path signatures

Mathematically, a $\mathbb{R}^d$-valued path X is understood as a continuous mapping from a given real interval $[a, b]$ into a d-dimensional Euclidean space (basically a curve):

$$[a,b] \ni t \mapsto X_t = X(t) = \left\{ X_t^1, X_t^2, X_t^3, \dots, X_t^d \right\}. \quad (1)$$

In simple terms, a path describes a trajectory of a process characterized by its starting and ending points. For instance, the degradation

[1] C-rate depicts the rate of time in which it takes to charge or discharge a cell. It is related to time t in hours by $t = 1/C$-rate.

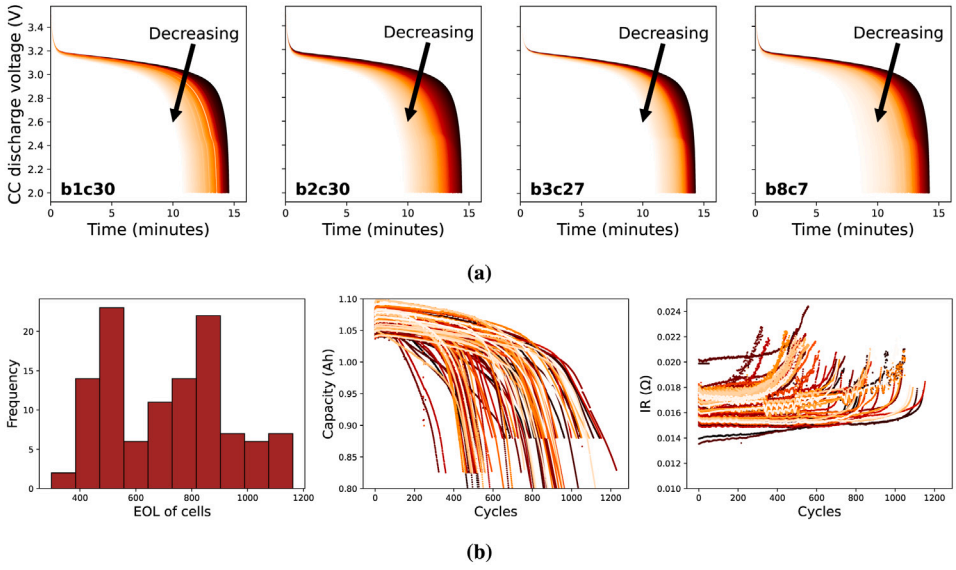


Fig. 1. **(a)** Evolution of the CC voltage curve (with respect to increasing cycle number) at discharge for selected cells in batches 1, 2, 3, and 8. It can be seen that the curves fall as the cells age. **(b)** The EOL histogram, capacity, and IR curves of all the training cells. Most cells live for around 500 and 900 cycles, whereas a few cells have a cycle life of up to 1200. Nominal capacity is seen to range between 1.05–1.10 Ah while IR values range from around 0.013–0.025 Ω.

trajectory of a battery capacity from its nominal value to value at EOL; the in-cycle pair current–voltage; or the stock price within a specified period of time.

From a bird's eye perspective, the signature of X is akin to a basis one selects to express functions in a function space, say, a Fourier basis for continuous functions — critically, the signature is a novel object that is much more expressive than other bases and appropriately encapsulates many of the path's analytical and geometrical properties.

Given the $\mathbb{R}^d$-valued continuous path X of Eq. (1), the kth fold iterated integral of X over $[a,b]$ with respect to its $i_1,\dots,i_k \in \{1,\dots,d\}$ components (possibly repeated) is given by

$$S(X)^{i_i,\dots,i_k}_{a,b} := \int_{a<t_k<b} \cdots \int_{a<t_1<t_2} dX^{i_1}_{t_1} \dots dX^{i_k}_{t_k}. \tag{2}$$

The *path signature* of X over $[a,b]$ is defined as the infinite-dimensional vector collection of all the possible iterated integrals of X. In signature notation, it is written as

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

For more clarity, the signature of X is a sequence of real numbers where each of its terms is an iterated integral defined by Eq. (2) and the superscripts are taken from the set $W = \{(i_1,\dots,i_k) | k \geq 1,\ i_1,\dots,i_k \in \{1,\dots,d\}\}$ called the *set of words* on the *alphabet* $\{1,\dots,d\}$ containing exactly d letters [25]. For instance, given a two-dimensional path $X_t = \{X^1_t, X^2_t\} = \{t+2, t^2+4\}$ with $dX_t = \{dX^1_t, dX^2_t\} = \{dt, 2tdt\}$ and $t \in [0,2]$, some of the terms of the signatures of X_t are

$$S(X)^1_{0,2} = \int_{0<t<2} dX^1_t = \int_0^2 dt, \qquad S(X)^2_{0,2} = \int_{0<t<2} dX^2_t = \int_0^2 2t\,dt,$$

$$S(X)^{1,1}_{0,2} = \int_{0<t_1<t_2<2} dX^1_{t_1}\, dX^1_{t_2} = \int_0^2 \left[\int_0^{t_2} dt_1\right] dt_2,$$

$$S(X)^{1,2}_{0,2} = \int_{0<t_1<t_2<2} dX^1_{t_1}\, dX^2_{t_2} = \int_0^2 \left[\int_0^{t_2} dt_1\right] 2t_2 dt_2,$$

$$S(X)^{2,1}_{0,2} = \int_{0<t_1<t_2<2} dX^2_{t_1}\, dX^1_{t_2} = \int_0^2 \left[\int_0^{t_2} 2t_1 dt_1\right] dt_2,$$

$$S(X)^{2,2}_{0,2} = \int_{0<t_1<t_2<2} dX^2_{t_1}\, dX^2_{t_2} = \int_0^2 \left[\int_0^{t_2} 2t_1 dt_1\right] 2t_2 dt_2,$$

$$S(X)^{1,1,1}_{0,2} = \int_{0<t_1<t_2<t_3<2} dX^1_{t_1}\, dX^1_{t_2}\, dX^1_{t_3} = \int_0^2 \left[\int_0^{t_3} \left[\int_0^{t_2} dt_1\right] dt_2\right] dt_3,$$

$$\vdots$$

Following this pattern, one can obtain all the terms $S(X)^{i_1,\dots,i_k}_{0,2}$ of the signature with given words or multi-index $(i_1,\dots,i_k)$ (where in this case $i_1,\dots,i_k \in \{1,2\}$).

The notion of path signature was originally brought forward in [30] where it was applied to piecewise smooth paths. This idea was extended in [31] to paths characterized by finite length. Signatures have many interesting and useful properties including time invariance under reparameterization, relation with Shuffle product and Chen's identity, time-reversal, and its linear combination to form *Lévy area* of a 2-dimensional path [25].

Path signature does not only work on continuous paths but also can be applied to discrete cases. A discrete path can be a stream of data of any dimension or time series data. Because calculating the signature of a data stream can be computationally intensive, especially for high dimensional data and high signature levels, *iisignature* [32] is a Python package that efficiently handles the calculations.

3.2. Feature generation

In this study, a 2-level signature of the path $X(t) = \{t, V(t)\}$ is considered, where t and $V(t)$ are the in-cycle CC discharge time and discharge voltage respectively, i.e.,

$$S(X)_{t_0,t_f} = \left(1, S(X)^1_{t_0,t_f}, S(X)^2_{t_0,t_f}, S(X)^{1,1}_{t_0,t_f}, S(X)^{1,2}_{t_0,t_f}, S(X)^{2,1}_{t_0,t_f}, S(X)^{2,2}_{t_0,t_f}\right),$$

where t_0, t_f are the initial and final times for the in-cycle CC discharging, respectively. As a remark, other choices for X are possible, e.g., current–voltage-temperature $t \mapsto \{I(t), V(t), T(t)\}$ and combinations thereof. These choices fall outside the scope of this work where the choice $X(t) = \{t, V(t)\}$ during CC is parsimoniously fit for purpose.

To generate the signature $S(X)_{t_0,t_f}$ for feature extraction, the first term of the sequence $S(X)_{t_0,t_f}$ was dropped since it is constant for all the in-cycle voltage curves regardless of their corresponding cycles. The first two levels of the signature of X were considered because of their geometrical intuition [25, Section 1.2.4]. In line with [25] and for brevity, the following conventions (ignoring the constant) for the first two levels of the signature of X were adopted

$$\left(S^1, S^2, S^{1,1}, S^{1,2}, S^{2,1}, S^{2,2}\right). \tag{3}$$

Each $S^{i[,j]}$, $i, j \in \{1,2\}$ in Eq. (3) is related (after simplification) to the components of X as follows:

$$\begin{cases} S^1 = t_f - t_0, \qquad S^2 = V_f - V_0, \\ S^{1,1} = \left(t_f - t_0\right)^2/2, \qquad S^{1,2} = -\int_{t_0}^{t_f} \left(V(t) - V_f\right)\, dt, \\ S^{2,1} = -\int_{t_0}^{t_f} \left(V_0 - V(t)\right)\, dt, \qquad S^{2,2} = \left(V_f - V_0\right)^2/2, \end{cases} \tag{4}$$

where V_0 and V_f are the initial and final CC discharge voltages respectively.

The negative sign in $S^{1,2}$ and $S^{2,1}$ (see Fig. 2) is a result of the orientation of the path X, since it is traversed in a clockwise direction [25]: $V(t)$ decreases with an increase in time t. Each component of the signature has a practical interpretation: S^1 and S^2 are the in-cycle incremental discharge time and voltage under CC respectively; $S^{1,1}$ and $S^{2,2}$ are also proportional to S^1 and S^2 with amplification of the change in the measured quantities (t and $V(t)$); $S^{1,2}$ and $S^{2,1}$ are proportional (current being kept constant) to the in-cycle electrical energy delivered by cells but shifted by V_f and V_0 respectively. The scatter plots of Fig. 3 depict the linear correlations between these signatures and the cycle number: it is revealed that there does exist some correlation between signatures and how each cell ages with a very strong correlation ($\rho < -0.5$ and $\rho > 0.5$) in S^1, $S^{1,1}$, $S^{1,2}$, and $S^{2,1}$. To study the evolution of the CC voltage at discharge using these signatures, cross-cycle features that compare signature components of one cycle to another were built, as also those which capture the distribution of the signatures over the observed cycles. The CC voltage curves at discharge were first cleaned: since the curves are not measured at consistent time intervals, they were interpolated using the SciPy's *interp1d* function [33] and evaluated on a constant time interval with step-size h. The step size h was varied from 4 s to 4 min to generate different sets of sub-samples of the CC voltage at discharge. For each sub-sample, the signature of the corresponding path X defined in Eq. (3) was then calculated using the *iisignature* Python library [32]. Explicitly, suppose $S_i^{(\text{cell})}$ denotes the value of a certain signature component at cycle i for a given cell, features that consider the comparison of a given signature component over n cycles were generated as follows:

$$\begin{cases} \text{Sig}_1 := \text{median}\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, i_{10}\right\}, \\ \text{Sig}_{n/2} := \text{median}\left\{S_i^{(\text{cell})}, i = n/2 - i_{10}, n/2 - i_{10} + 1, \ldots, n/2 + i_{10}\right\}, \\ \text{Sig}_n := \text{median}\left\{S_i^{(\text{cell})}, i = n - i_{10}, n - i_{10} + 1, \ldots, n\right\}, \\ \text{Sig}_{nm1} := \text{Sig}_n - \text{Sig}_1, \\ \text{Sig}_{diff} := \text{Sig}_n - 2\text{Sig}_{n/2} - \text{Sig}_1; \end{cases} \tag{5}$$

where n and i_{10} are the number of input cycles and 10% of n respectively. This procedure resulted in the extraction of 30 features encoded using the convention in Table 1. As for the features measuring the distribution of a fixed signature component over the observed cycles, features for each cell over n cycles were generated using

$$\begin{cases} \min\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, n\right\}, \quad \max\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, n\right\}, \\ \text{mean}\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, n\right\}, \quad \text{var}\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, n\right\}, \\ \text{kurt}\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, n\right\}, \quad \text{skew}\left\{S_i^{(\text{cell})}, i = 1, 2, \ldots, n\right\}; \end{cases} \tag{6}$$

where *var*, *skew*, and *kurt* are the variance, skewness, and kurtosis operations respectively. Under this process, 36 features were generated and were encoded using the convention given in Table 2. Thus, overall, a total of 66 features were extracted and we demonstrate in the next sessions how each group (as well as their combination) was used for model building.

Table 1
Description of the generated features which compare signatures across cycles: the <feature> appendage denotes one of the formulas defined in Eq. (5).

Feature	Description
S1-<feature>	cross-cycle features of the first signature component, S^1
S2-<feature>	cross-cycle features of the second signature component, S^2
S11-<feature>	cross-cycle features of the third signature component, $S^{1,1}$
S12-<feature>	cross-cycle features of the fourth signature component, $S^{1,2}$
S21-<feature>	cross-cycle features of the fifth signature component, $S^{2,1}$
S22-<feature>	cross-cycle features of the sixth signature component, $S^{2,2}$

Table 2
Description of the generated features which capture the distribution of the signatures of the CC discharge voltage (as given in Eq. (6)): the <component> appendage denotes one of the signature components defined in Eq. (3).

Feature	Description
Min-<component>	minimum of a signature component over n cycles
Max-<component>	maximum of a signature component over n cycles
Mean-<component>	mean of a signature component over n cycles
Var-<component>	variance of a signature component over n cycles
Kurt-<component>	kurtosis of a signature component over n cycles
Skew-<component>	skewness of a signature component over n cycles

3.3. RRCT algorithm for feature selection

The RRCT algorithm [26] is a model-agnostic feature selection algorithm that takes as input *features* and *prediction targets* and ranks the input features according to a certain 'Relevance', 'Redundancy' and 'Complementary' Trade-off (RRCT). For the algorithm to be used as a feature selection mechanism, the user must also input the number of features to be kept from the ranked features list.

Ingredients-wise, 'Relevance' reflects the strength (and thus needs to be maximized) of the univariate association between a feature and the target variable. This can be captured using any statistical method which shows the relationship between two variables (for instance, mutual information and correlation coefficient). As for the 'Redundancy', it captures the overlapping or common predicting power of two or more features in the feature set. It thus needs to be minimized to narrow down the dimension of the feature space. Lastly, 'Complementarity' (also known as feature interaction and is the heart of the RRCT), measures the joint predicting power of two or more features since it is possible that individual features might be moderately associated with the target variable but strongly related to it when combined with others.

The three properties are combined in the following equation to rank features in a given feature set:

$$\text{RRCT} := \max_{j \in Q-S} \left[\underbrace{r_{IT}(\boldsymbol{f}_j, \boldsymbol{y})}_{\text{relevance}} - \underbrace{\frac{1}{|S|} \sum_{s \in S} r_{IT}(\boldsymbol{f}_j, \boldsymbol{f}_s)}_{\text{redundancy}} + \underbrace{\text{sign}\left(r_p(\boldsymbol{f}_j, \boldsymbol{y}|S)\right) \cdot \text{sign}\left(r_p(\boldsymbol{f}_j, \boldsymbol{y}|S) - r(\boldsymbol{f}_j, \boldsymbol{y})\right) \cdot r_{p,IT}}_{\text{complementarity}} \right]; \tag{7}$$

where Q is the set of indices of all the features; S is the set of selected feature indices; $\boldsymbol{f}_j$ and $\boldsymbol{y}$ are the feature at index j and target variable respectively; r_{IT} is the non-linearly transformed rank correlation coefficient r_{XY} between two random variables X, Y and is defined by $r_{IT}(X,Y) := -0.5 \cdot \log\left[1 - r_{X,Y}^2\right]$; $r_p(\boldsymbol{f}_j, \boldsymbol{y}|S)$ is the partial correlation coefficient between $\boldsymbol{f}_j$ and $\boldsymbol{y}$ given the existing features in the subset S; $r(\boldsymbol{f}_j, \boldsymbol{y})$ is the Spearman rank correlation between $\boldsymbol{f}_j$ and $\boldsymbol{y}$; $r_{p,IT} = -0.5 \cdot \log\left[1 - r_p^2\right]$ is the transformed computed partial correlation coefficient; sign$(\cdot)$ is the signum function.

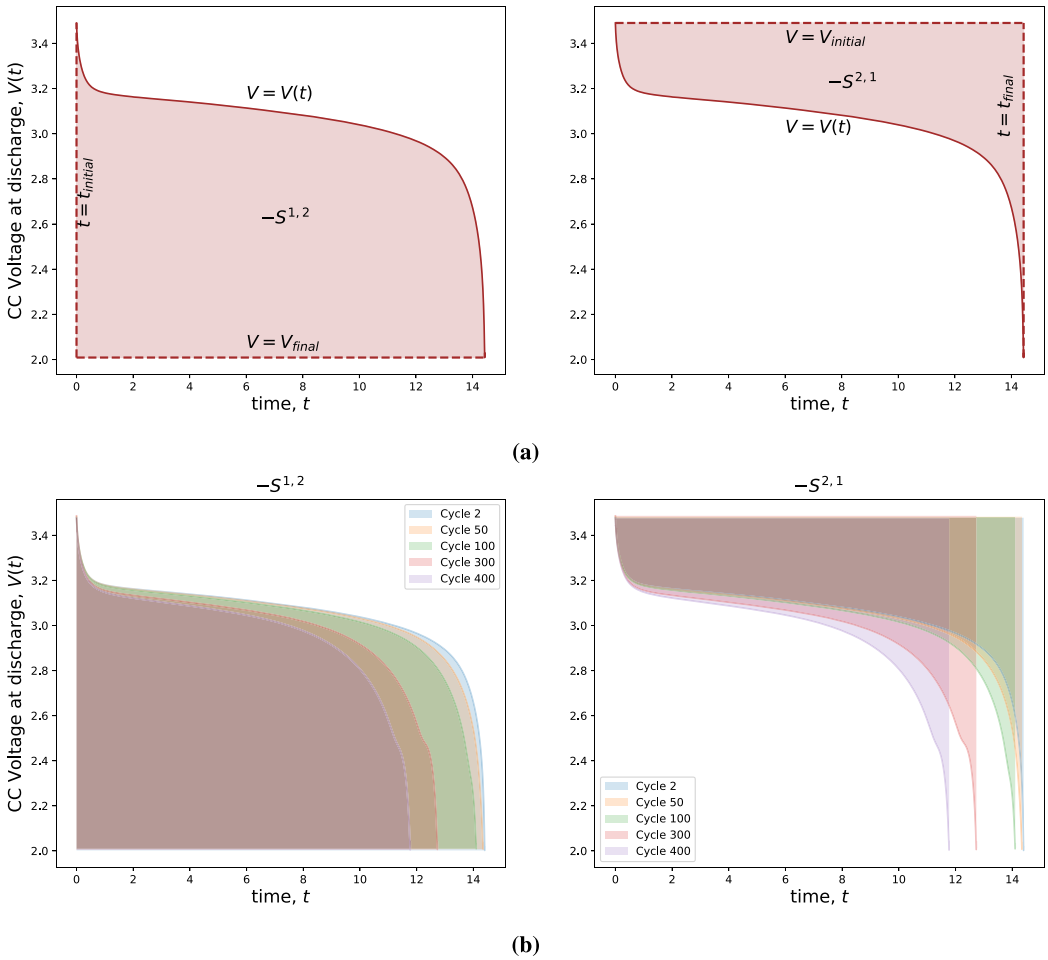


Fig. 2. **(a)** Geometrical interpretation of the first two levels of signature of the path X defined above. The negative signs in $S^{1,2}$ and $S^{2,1}$ are because of the orientation of the path. **(b)** The evolution of the signature terms $S^{1,2}$ and $S^{2,1}$ as the cycle number increases. It can be seen that the areas defined by $-S^{1,2}$ and $-S^{2,1}$ shrink and expand respectively as the input cycle number increases. This is a key point in generating features to capture the evolution.

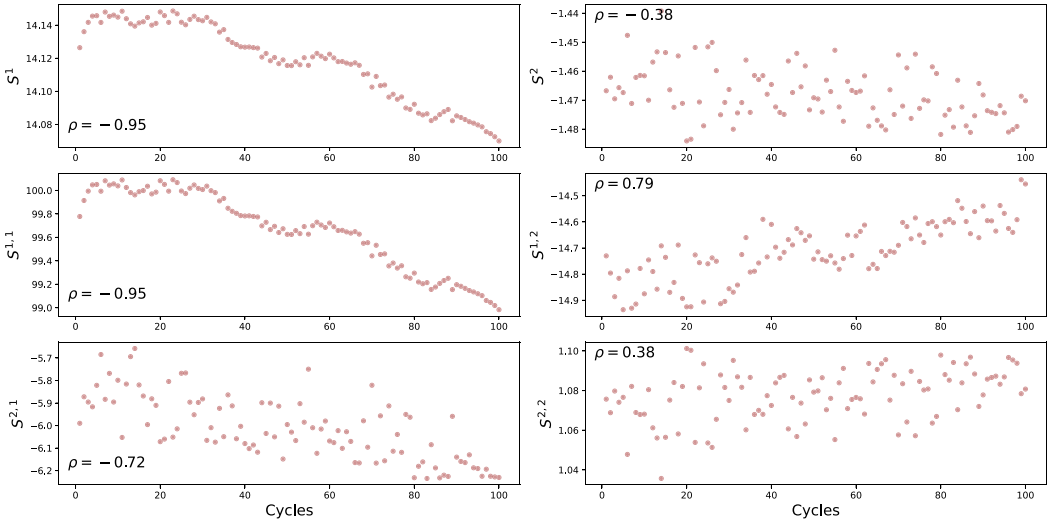


Fig. 3. Scatter plots (and the corresponding Pearson correlation coefficient ρ) of signature components and the first 100 cycles of a random cell b1c30. Each plot shows that there are some linear correlations between signatures and cycle numbers.

Looking in more detail into the complementarity term of Eq. (7), the term sign $\left(r_p(\boldsymbol{f}_j, \boldsymbol{y}|S) - r(\boldsymbol{f}_j, \boldsymbol{y})\right)$ determines whether the conditional relevance $r_p(\boldsymbol{f}_j, \boldsymbol{y}|S)$ is greater than $r(\boldsymbol{f}_j, \boldsymbol{y})$. This implies that including $\boldsymbol{f}_j$ has additional explanatory relevance given the features in S. Furthermore, the term sign $\left(r_p(\boldsymbol{f}_j, \boldsymbol{y}|S)\right)$ forces the overall complementarity contribution to be positive when $r(\boldsymbol{f}_j, \boldsymbol{y}) < 0$, $r_p(\boldsymbol{f}_j, \boldsymbol{y}|S) < 0$, and sign $\left(r_p(\boldsymbol{f}_j, \boldsymbol{y}|S) - r(\boldsymbol{f}_j, \boldsymbol{y})\right) < 0$.

Table 3
Description of the machine learning models considered in this study.

Model name	What are predicted
cycle model	k-o, k-p, e-o, e-p, EOL
capacity-IR model	Qatk-o, Qatk-p, IRate-o, IRate-p, IRatEOL

Concretely, the RRCT algorithm uses Eq. (7) in an incremental way to rank features (in decreasing order) based on relevance, redundancy, and complementarity through the following main three steps: (i) select the first feature $\boldsymbol{f}_j$ using $\max_{j\in Q}\left[r_{IT}(\boldsymbol{f}_j, \boldsymbol{y})\right]$ and place it in the initially empty subset S of Q, i.e., $j \rightarrow S$; (ii) apply Eq. (7) repeatedly to select the next feature from $Q - S$ and put it in S, i.e., $S \cup j \rightarrow S$; (iii) the features are now ranked based on the three criteria and the top k features $\{\boldsymbol{f}_j\}_{j=1}^{k}$, $j \in S$ can be kept from the original feature matrix.

These steps have been originally implemented in MATLAB by the author of [26], and its Python implementation (as used in this study) is available on the PyPI platform under the name *rrct* (see Code Availability section below).

3.4. Machine learning

Features, targets, and a model to capture nonlinearity

The feature set was described in Section 3.2. In terms of prediction targets, the aim is to predict the degradation trajectory of the capacity and internal resistance curve from early life. This is done by predicting a small amount of specific, cogent points of each curve and then re-constructing the full curve [29, Fig. 1]. For each Capacity and IR curve (Fig. 1(b)), and using the Bacon–Watt model [8,27] (an overview of identification methods appears in [34]), knees and elbows coordinates of the curves are obtained. Concretely, the prediction targets are: the cycle-points for knee-onset (k-o), knee-point (k-p), elbow-onset (e-o), elbow-point (e-p), and (capacity) EOL, plus the capacities at k-o and k-p (Qatk-o and Qatk-p, respectively) as well as IR values at e-p, e-o and EOL (IRate-o, IRate-p and IRatEOL, respectively).

The bar chart of Fig. 4 shows the linear Pearson correlation coefficient ρ between all the generated features and two of the key targets for prediction: EOL and its corresponding IR value (IRatEOL). It can be seen that only 10 and 24 out of the total extracted features have a stronger linear correlation with each of the chosen targets, respectively, which implies that the majority have a more complex relationship with the targets (than linear).

Nonlinear modeling, its design and details of model implementation.

In line with [3], the modeling choice taken here is to consider a non-linear, tree-based, and ensemble model called Extreme Gradient Boosting (XGBoost) [35]. In this study, the modeling strategy involved building two different models named *cycle model* and *capacity-IR model*. The former was designed to jointly predict cycle number-related targets (namely knee-onset (k-o), knee-point (k-p), elbow-onset (e-o), elbow-point (e-p), and EOL) while the latter was built to jointly predict the capacities at k-o and k-p (Qatk-o and Qatk-p) as well as IR values at e-p, e-o and EOL (IRate-o, IRate-p, and IRatEOL); Table 3 summarizes.

The XGBoost implementation used is that of scikit-learn [36] and details about its theory can be found in the Methods section (also [35]). Each of the models of Table 3 was trained using a $70-30\%$ train-test splitting strategy (i.e., 70% of the 158 cells were used for training and 30% were used for testing) using the features generated and the above-mentioned targets. Since the XGBoost is designed to predict a single target, the scikit-learn *MultiOutputRegressor* class [36] was used to wrap the model to predict multiple targets.

As for the hyperparameter tuning, both trial and error and grid search approaches were employed. Trial and error was considered to identify suitable values of the hyperparameters for the model. Grid search approach, accomplished by the scikit-learn *GridSearchCV* class [36], was considered to generate other possible values around those obtained in the first technique (a parameter space), which were then optimized using squared error scoring function to obtain the best set. In particular, it was observed that the learning rate, number of estimators, l_1-regularization, minimum sample split, and the maximum depth of grown trees were the most important to tune (see Table 4 for the optimized values of these parameters); details about the meanings of these parameters can be found in [35]. Model evaluation was carried out on the test data via model metrics including mean absolute error (MAE), mean absolute percentage (MAPE), and root mean squared error (RMSE); see Eq. (9) in the Methods section for their definitions.

Table 4
Parameter specification for the two trained XGBoost models: cycle and capacity-IR models. In places where "default" is specified, it means that the scikit-learn's default parameters were used.

Parameters	Values	
	Cycle model	Capacity-ir model
n_estimators	100	500
learning_rate	default	0.1
reg_alpha	0.1	default
max_depth	2	6
min_samples_split	3	default

3.5. Longitudinal data exploration

To determine a base value for the number of cycles of battery cycling data needed for our study, the modeling strategies discussed in Section 3.4 were deployed on all the 66 features extracted from the in-cycle CC discharge voltage measured at approximately 4 s intervals. A 3-fold cross-validation was carried out on the training set corresponding to the data obtained under the first n cycles, $n = 1, 2, \ldots, 100$. Errors were further categorized into two groups (one related to points on the capacity curve (k-o, k-p, and EOL) and the other related to IR curve (e-o and e-p)) to see the impact of varying cycle number inputs on the model; see Fig. 5 for the summary of results. In general, errors decrease as the input cycle progresses. This is different from the previous paper [3] where increasing the cycle number threshold (in the given range) does not significantly improve the model. In addition, it is a selling point with respect to [3] because as more data is observed in the defined range of n, the corresponding model becomes more accurate. Following this result and the fact that the gap between input cycles has to be bigger for the signature to uniquely define the CC discharge voltage of different cycles (see Fig. 2), 100 cycles were chosen for all the modeling activities. This is in line with other papers predicting EOL and capacity fade curves of batteries [4,5,7]. For comparison purposes only, errors on test data were presented for models using data generated under 50 cycles.

3.6. Data sub-sampling and feature selection

Upon establishing the number of input cycles for the models, the focus turns to the effect of CC discharge voltage sub-sampling and feature selection on model accuracy. The former was accomplished by considering different values of time steps, h (in minutes, as described in Section 3.2) between 0.05 and 4 min with a constant increase of 0.05. Under each h, features were generated accordingly and fed to the machine learning algorithm. Feature selection was performed for a given h by considering a different threshold of the percentage of features to retain after the process. In particular, keeping $p\%$ was considered for modeling (where $p \in \{10, 20, 30, \ldots, 90\}$), and the model performance metrics were recorded accordingly.

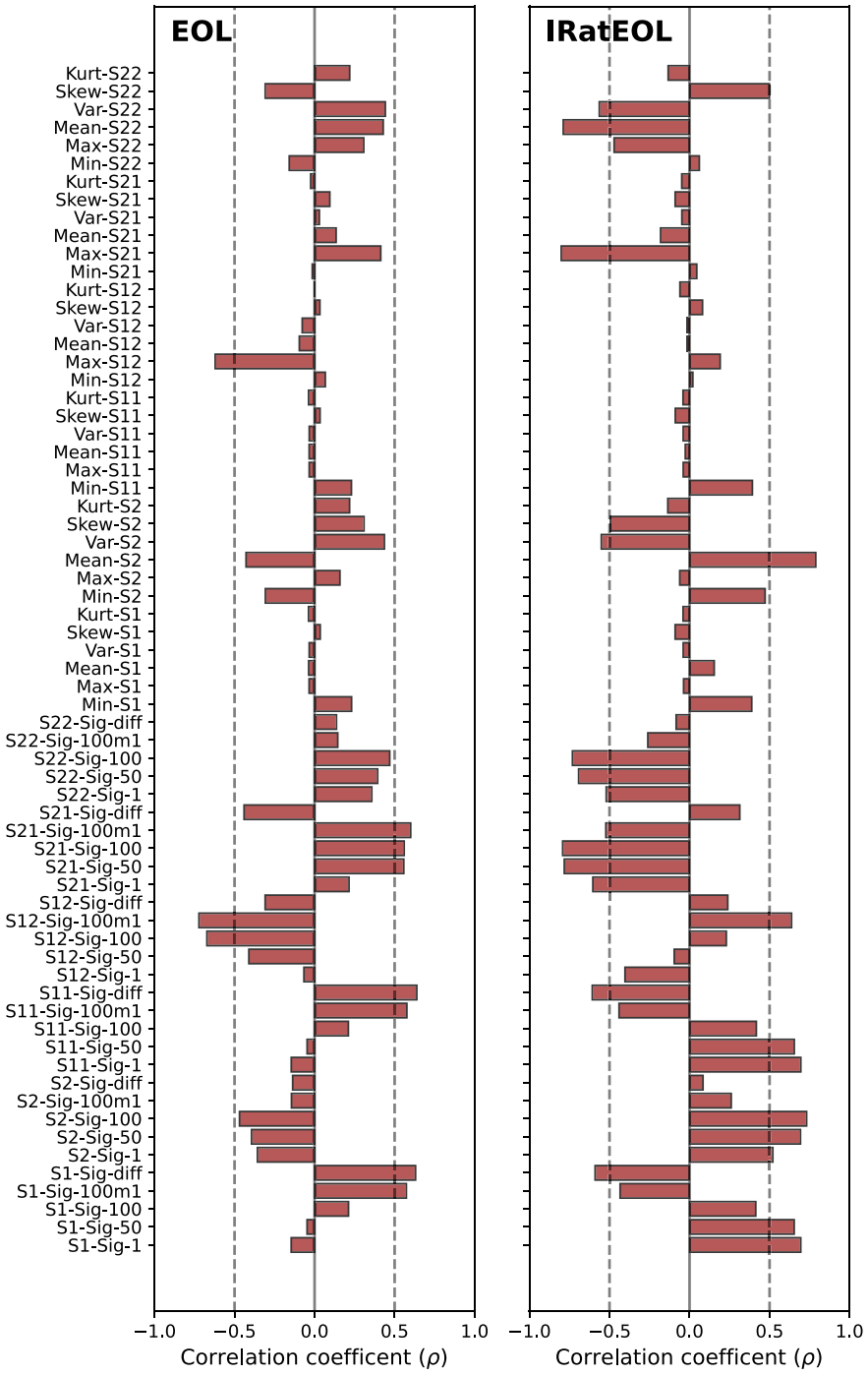


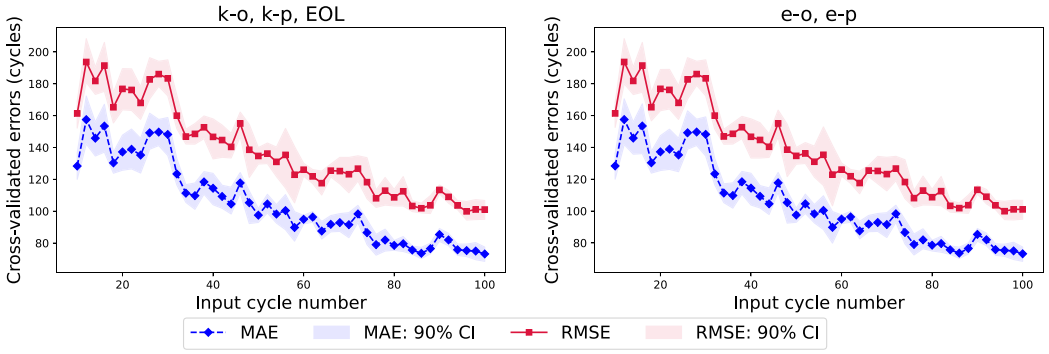
Fig. 4. Pearson correlation coefficients between the features generated from Tables 1 and 2 and EOL as well as IRatEOL.

3.7. Entire curve prediction

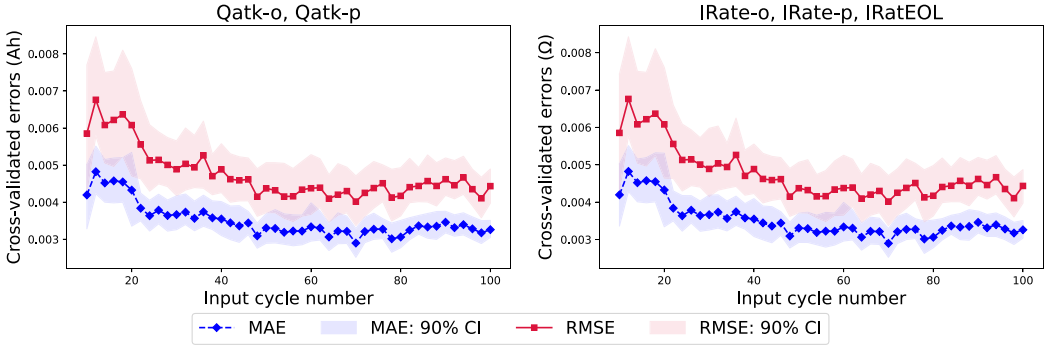
The predictions from the machine learning model described in Section 3.4 are the key points on the capacity and IR curves. Following the method developed in [3,17,27,29], these points (knees, elbows, EOL, and their corresponding capacity and IR values — see [34] for an in-depth review of "Knees") together with the initial values of the cycle number, capacity, and IR were fed to a modified quadratic spline to predict the entire curves. In other words, the spline fits a straight line between the initial point and knee/elbow onset, a quadratic polynomial between the knee/elbow onset and knee/elbow point, and a quadratic polynomial from the knee/elbow point to the EOL. Further details about the spline can be found in [3].

4. Results and discussion

In this section, the results of the various experiments performed in this study are presented. In particular, metrics on the models developed under the high and low-frequency data measurement and feature selection are reported. A visualization of the full prediction of the capacity fade and IR rise curves corresponding to randomly selected cells in the test data is also given.



(a)



(b)

Fig. 5. **(a)** Average cross-validated errors on the training set corresponding to the *cycle model* trained on the combination of features from Tables 1 and 2. Errors were further grouped into two categories (one related to points on the capacity curve (k-o, k-p, and EOL) and the other related to IR curve (e-o and e-p)) to see the impact of varying cycle number inputs on the model. It can be observed that, on average, the model captures the evolution of signatures of the CC voltage at discharge more at higher values of input cycle number. **(b)** The same investigation but on *capacity-IR model* is shown by averaging the cross-validated errors on capacity at k-o and k-p, and IR at e-o, e-p, and EOL predictions. In general, the error decreases significantly in the first 40 cycles after which it approximately remains the same for the rest of the cycle number inputs.

4.1. Model results under high-frequency data

Model metrics. Table 5 presents the performance metrics (and their 95% confidence intervals) of the cycle model. The model was trained on features in Table 1 as well as its combination with those in Table 2 which were generated under high-frequency data (4 s). Generally, errors (train and test) are lower under the combination of both tables with EOL predicted on test data to the accuracy of 65 and 91 cycles of MAE and RMSE respectively. A similar trend is also observed in Table 6 for the predictions made under the capacity-IR model where IRatEOL was predicted with MAE and RMSE of approximately 4.06×10^{-4} and $6.32 \times 10^{-4}\ \Omega$ respectively. This highlights the non-negligible role of the distributional features of Table 2. In fact, the signature's distribution captures summary information about cells' loss in capacity: for instance, the distribution of S^2 and $S^{1,2}$ components give information about the incremental change in voltage and energy respectively; see Eq. (4).

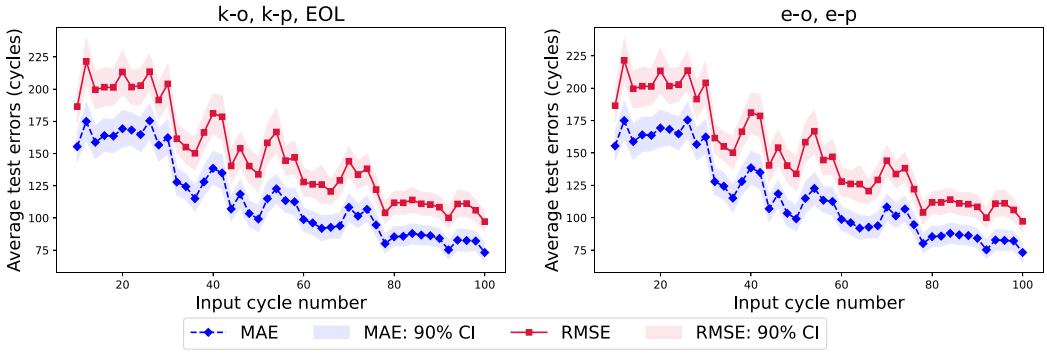
The comparative analysis (using the MAPE) of the models obtained under cycles 50 and 100 for the groups of generated features are presented in Table 7. Longitudinal data under 100 cycles yielded fewer test errors than that of 50 cycles. In fact, Fig. 6 illustrates that test errors decrease as the input cycles increase. This also agrees with what was reported in Section 3.5 where cross-validation on the training data was considered. It is remarked, for this result, that the models still have a competitive accuracy (see [4,5]) under 50 cycles with an MAPE of 15.8%.

Parity plots. In Fig. 7, parity plots of model predictions are provided (with an embedded histogram of residuals) to investigate how the predicted data points are close to the true parity line. Overall, predicted points in both train and test data are close to the true parity line. The embedded histograms highlight that most off-predictions are concentrated in the bin symmetrically (i.e., no skew) centered around zero, showing that prediction errors are minimized on both training and test cells.

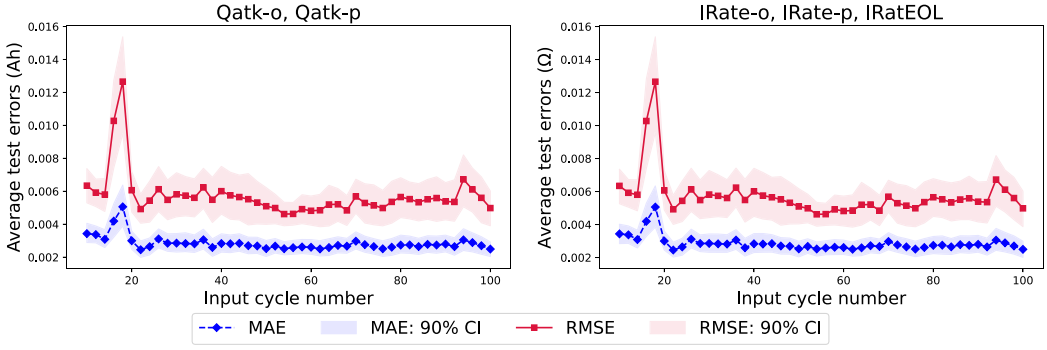
Table 5
Performance metrics together with 95% confidence interval of the *cycle model* for the prediction of k-o, k-p, e-o, e-p, and EOL. Values for the metrics using features from Table 1 only and their combination with those in Table 2 generated under 4-s data are provided.

Table		MAE (cycles)		RMSE (cycles)	
		Train	Test	Train	Test
1	k-o	21 ± 3.4	64 ± 16.5	28 ± 4.4	86 ± 18.5
	k-p	23 ± 3.4	66 ± 17.1	29 ± 4.7	89 ± 18.8
	e-o	28 ± 4.4	87 ± 19.6	35 ± 5.0	110 ± 21.1
	e-p	27 ± 4.3	78 ± 18.0	35 ± 5.0	101 ± 18.5
	EOL	30 ± 4.7	74 ± 19.7	39 ± 5.9	101 ± 23.1
1 & 2	k-o	13 ± 2.4	68 ± 15.7	19 ± 3.3	87 ± 16.8
	k-p	16 ± 2.9	66 ± 17.2	23 ± 3.9	89 ± 20.0
	e-o	16 ± 2.8	86 ± 19.2	22 ± 3.8	110 ± 22.7
	e-p	18 ± 3.1	76 ± 22.1	25 ± 3.9	108 ± 26.4
	EOL	19 ± 3.2	65 ± 17.8	26 ± 4.2	91 ± 24.3

Feature importance. To account for the role of each of the generated features in building the XGBoost models, feature importance corresponding to the first 10 most important predictors for each target are presented in Fig. 8. Feature importance values were calculated using the XGBoost impurity-based feature importance analysis embedded in the algorithm [35] and were scaled to a range between 0 and 1 for ease of interpretation. In general, the most important features were spread across those presented in Tables 1 and 2 which show the predicting power of both methods employed for feature extraction. In particular,



(a)



(b)

Fig. 6. **(a)** Average test errors corresponding to the *cycle model* built on the combination of features from Tables 1 and 2 generated under 4-s data. The errors are split into two parts (one related to points on the capacity curve (k-o, k-p, and EOL) and the other related to IR curve (e-o and e-p)) to further see the impact of varying cycle number inputs on the model. It is shown that, roughly, the error decreases with an increase in the input number of cycles. **(b)** The effect on input cycles on *capacity-IR model* is shown by averaging the test errors on capacity at k-o and k-p, and IR at e-o, e-p, and EOL predictions. In general, errors do not change significantly over the cycle numbers considered.

Table 6
Performance metrics together with 95% confidence interval of the *capacity-IR model* for the prediction of Qatk-o, Qatk-p, IRate-o, IRate-p, and IRatEOL. Values for the metrics using features in Table 1 only and its combination with those in Table 2 generated under 4-s data are presented.

Table		MAE		RMSE	
		Train ($\times 10^{-4}$)	Test ($\times 10^{-3}$)	Train ($\times 10^{-4}$)	Test ($\times 10^{-3}$)
1	Qatk-o	9.29 ± 1.40	5.353 ± 1.54	12.02 ± 2.04	7.566 ± 2.36
	Qatk-p	9.28 ± 1.31	5.652 ± 1.68	11.60 ± 1.80	8.103 ± 2.88
	IRate-o	0.18 ± 0.02	0.498 ± 0.12	0.24 ± 0.04	0.640 ± 0.15
	IRate-p	0.17 ± 0.03	0.493 ± 0.12	0.22 ± 0.03	0.637 ± 0.16
	IRatEOL	0.19 ± 0.03	0.490 ± 0.13	0.24 ± 0.03	0.660 ± 0.12
1 & 2	Qatk-o	7.65 ± 1.23	5.637 ± 1.52	10.19 ± 2.02	7.689 ± 2.50
	Qatk-p	7.81 ± 1.16	5.634 ± 1.93	9.91 ± 1.73	8.722 ± 3.77
	IRate-o	0.15 ± 0.02	0.416 ± 0.11	0.18 ± 0.02	0.554 ± 0.16
	IRate-p	0.14 ± 0.02	0.395 ± 0.11	0.18 ± 0.03	0.546 ± 0.17
	IRatEOL	0.17 ± 0.03	0.406 ± 0.14	0.21 ± 0.02	0.632 ± 0.22

Table 7
A comparative analysis of mean absolute percentage error (MAPE): performance metrics of the *cycle* and *capacity-IR* models for the prediction of EOL and IRatEOL respectively. Metric values for using two different cycle number inputs (n) under features from Table 1 only as well as its combination with those from Table 2 generated under 4-s data are shown.

		MAPE (%)			
		Table 1		Tables 1 and 2	
		Train	Test	Train	Test
$n = 50$	EOL	5.6	15.5	4.2	15.8
	IRatEOL	0.1	3.0	0.09	2.9
$n = 100$	EOL	4.1	11.7	2.7	10.0
	IRatEOL	0.1	2.6	0.09	2.1

for EOL prediction, the features that measure the variance of $S^{1,2}$ (Var-S12) and compare its values at cycles 1 and 100 (S12-Sig-100m1) were discovered to be on top of the list. This can be linked to the rich geometrical interpretation of this signature component (see Fig. 2): it is the negative time integral of the CC discharge voltage whose values are shifted by the final voltage. This is directly linked and proportional to the energy discharged from the cells as the output current is kept constant.

RRCT feature selection. The bar charts of Fig. 9 show the results of the RRCT algorithm (see code availability section) applied to all the features generated to predict the EOL and IRatEOL. On the training set, errors decreased progressively as we increased the proportion of retained features in both cases. On the other hand, test errors did not change significantly as the feature percentages increased. This shows the robustness of the signature method and the quality of the

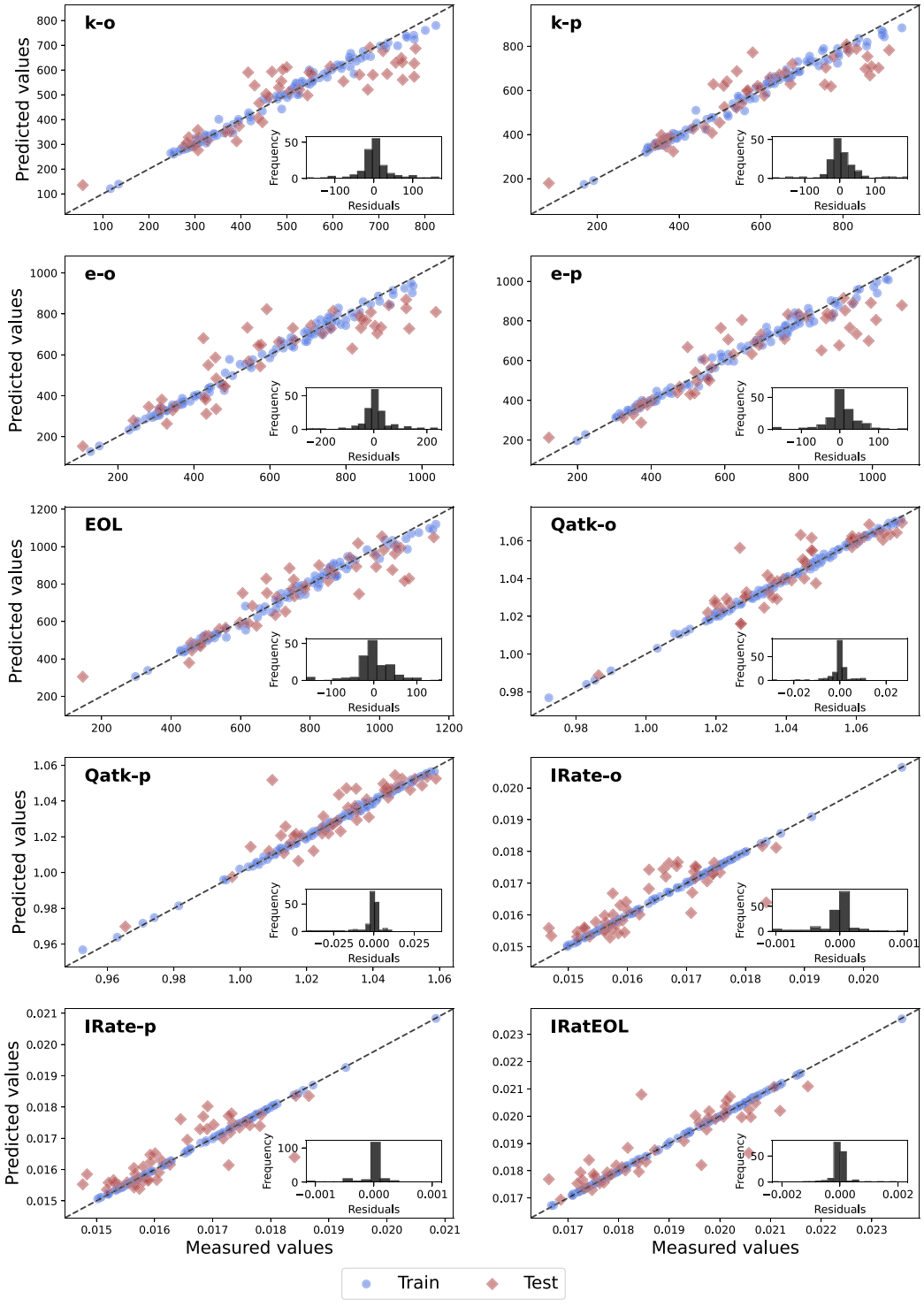


Fig. 7. Parity plots showing the comparison of the predicted values to the measured values (obtained under *cycle model* and *capacity-IR model* built from the combination of features from Tables 1 and 2 generated under high-frequency data (4 s)). Embedded in each of the plots is a histogram showing the distribution of both training and test residuals.

features selected by the RRCT. In addition, the results prove useful for dimensionality reduction, model simplicity, ease of model deployment, and reduced computation time.

Entire curve prediction. Upon obtaining knees, elbows, EOL, and their respective capacity and IR values, the method discussed in Section 3.7 was employed to predict the entire curves for randomly selected cells in the test data. The results are presented with the 90% confidence intervals in Fig. 10. The confidence intervals were obtained via the method discussed in [3, Section 4.2]. It can be seen that the models were able to predict both the capacity and IR curves as the predicted curves were very close to the measured ones. In addition, the visualization shows the robustness of the models to the noise in the IR data: the features

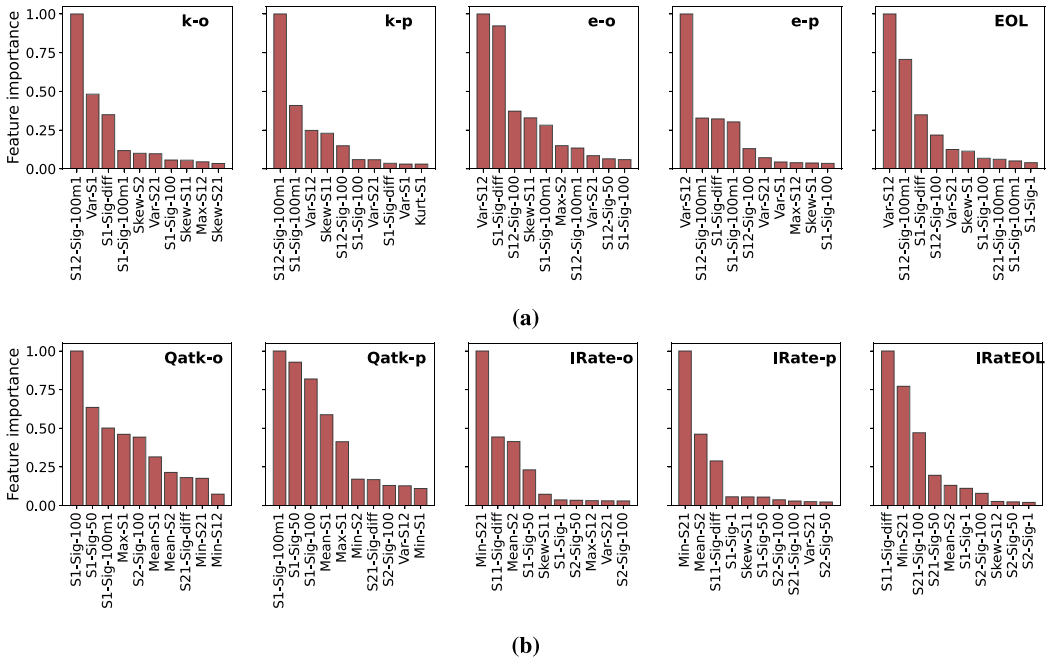


Fig. 8. Feature importance of the generated features calculated by the XGBoost algorithm. The first 10 features with the highest importance in predicting each of the targets in **(a)** *cycle model*, **(a)** *capacity-IR model* built on the combination of features in Tables 1 and 2 generated under high-frequency data (4 s) are displayed.

Table 8
Performance metrics for the prediction of EOL and IRatEOL using all the features in both Tables 1 and 2 generated under CC discharge voltage recorded at every 4 min. MAE and RMSE are in cycles for the case of EOL; in Ohms (Ω) for IRatEOL. All MAPE are in percentages.

	MAE		MAPE		RMSE	
	Train	Test	Train	Test	Train	Test
EOL	22	85	3.0	13.2	30	111
IRatEOL	1.7×10^{-5}	3.94×10^{-4}	0.09	2.1	2.30×10^{-5}	5.91×10^{-4}

generated through the signature method mitigated the sensitivity of the XGBoost model to the noise and thus was able to predict the IR curves more accurately.

4.2. Model results under low-frequency data: CC discharge voltage sub-sampling

This section reports the models' findings under a low-frequency data regime. On cross-validation, the 3-fold cross-validation results obtained under training data with different sub-sampling time steps for the case of EOL prediction are displayed in Fig. 11. Roughly, the errors increase as time steps rise. It is observed that the model still maintains its quality up to a time step of 2 min with an MAE and RMSE of about 90 and 120 cycles respectively. In addition, the 90% confidence intervals at time steps from 3.5 min upward were contained in those just before it. This is an indication that the voltage measurement frequency can be extended to up to 4 min without compromising model accuracy.

Model performance metrics. As highlighted in the above paragraph, model building was restricted to CC discharge voltage sub-sampled every 4 min. For brevity, the performance metrics for the EOL and IRatEOL predictions were only reported in Table 8. It was observed that voltage measurement at the end of every 4 min generated a model which predicts EOL with MAE, MAPE, and RMSE of 85 cycles, 13.2%, and 111 cycles respectively. Noise in the IR data has a minimal effect on the model and MAPE on IRatEOL prediction was observed to be 2.1%.

Full curve prediction. In Fig. 12, the models obtained under 4-min sub-sampling were used for full curve prediction. Each curve predicted for a random cell in the test data shows a good approximation of the actual measurement.

RRCT feature selection and sub-sampling. The use of RRCT feature selection tool was extended to the features extracted under 4-min CC discharge voltage sub-sampling. Only the top 10% of features (see the bar chart of Fig. 13(a)) ranked by the algorithm were retained. Interestingly, EOL and IRatEOL were predicted to have an accuracy very close to that of models using all the features; see Table 9 for the summary of the metrics. This two-level model simplification (4-min sub-sampling and feature selection) does not only applicable in real-life where data are recorded at a low frequency but also proposes an effective and economical choice for model deployment and maintenance.

The heat map of Fig. 13(b) shows the similarity of the first ten features selected by the RRCT algorithm under different voltage sub-sampling rates. Each score was calculated between two sub-sampling time steps by finding the ratio of the number of common features to the total number of features selected; thus higher ratio means greater similarity. This experiment was motivated by checking the robustness of the algorithm under changing voltage measurement frequency. It was observed that, in general, there are some similarities between features generated under any given two time frequencies. In addition, the strength of similarity grows as measurements are taken with close time steps. This is an indication that similar information is derived from the data even when the voltage reading is taken at different time frequencies.

To further investigate the robustness of the proposed models, both cycle and capacity-IR models were trained on the high-frequency data (CC discharge voltage measured at 4 s) and then used for prediction using data from low-frequency data sub-sampled at 0.5-4 min. The MAE of this experiment is presented in Fig. 14. It was observed that, generally, errors grow slowly (about 10 cycles, 10^{-3} Ah, 10^{-4} Ω per

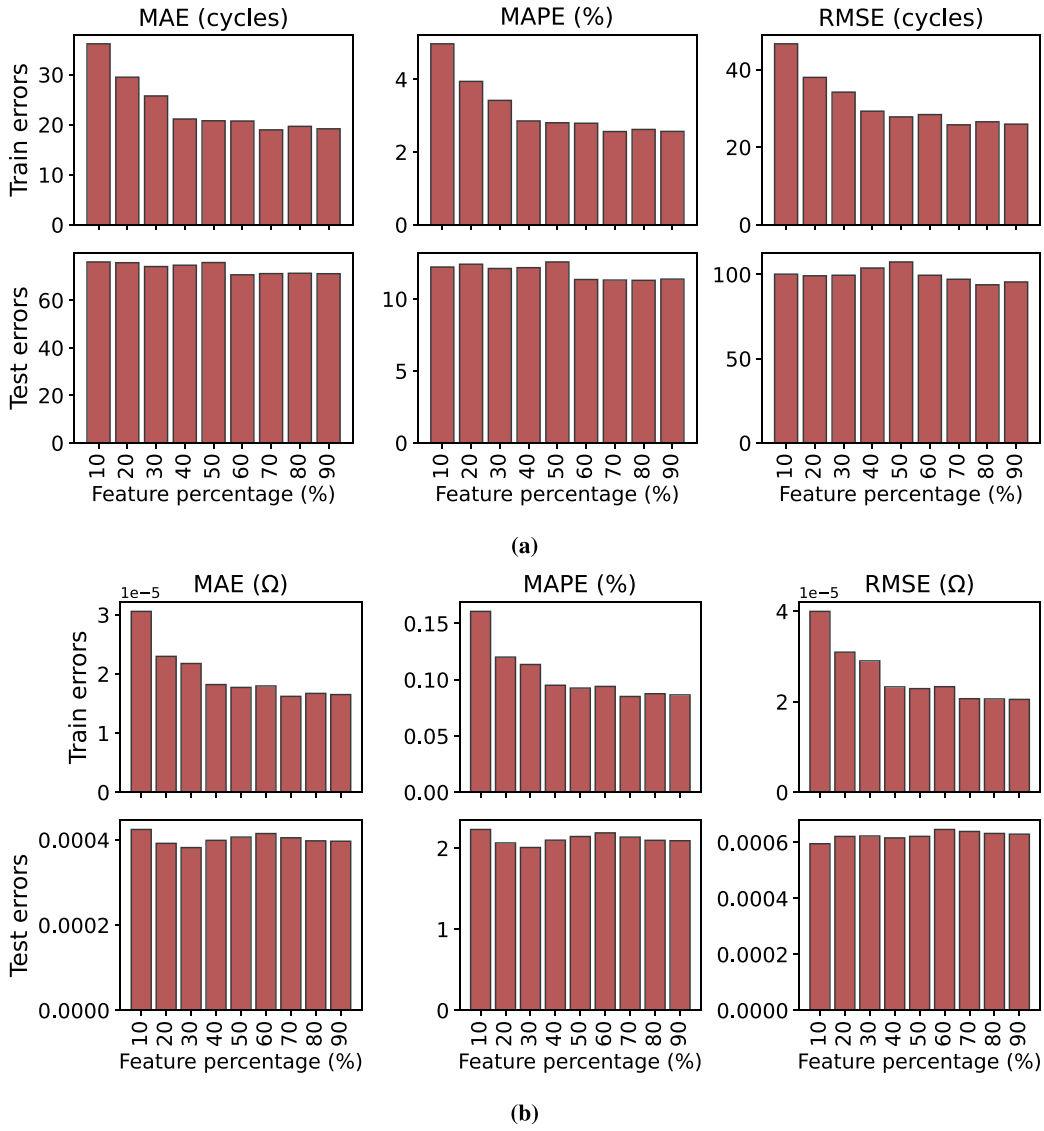


Fig. 9. Feature selection using the RRCT algorithm– this is independent of the feature importance of the XGBoost algorithm. Feature selection with different percentages of the combination of features from Tables 1 and 2 generated under 4-s data are provided. The selected features were used for EOL and IRatEOL predictions. Performance metrics are provided for EOL and IRatEOL in **(a)** and **(b)** respectively.

Table 9
Performance metrics for the prediction of EOL and IRatEOL using only the best 10% (selected by the RRCT algorithm) of features in both Tables 1 and 2 generated under CC discharge voltage recorded at every 4 min. MAE and RMSE are in cycles for the case of EOL; in Ohms (Ω) for IRatEOL. All MAPE are in percentages.

	MAE		MAPE		RMSE	
	Train	Test	Train	Test	Train	Test
EOL	40	85	5.3	13.4	53	111
IRatEOL	2.90×10^{-5}	3.63×10^{-4}	0.15	1.9	3.90×10^{-5}	5.19×10^{-4}

0.5 min for cycle-like, capacity-like, and IR-like targets respectively), but remain relatively the same from 1.5 min onward. *This implies that the signature-based models trained when there is high availability of data can be saved and reused to make predictions for data taken at wider time gaps.* Again, this proves beneficial in terms of reduction in the cost of data collection. It also facilitates online prediction as data need not be recorded at the initial fixed frequency before sending for prediction.

4.3. Comparison with past literature

Here, the models built using the 66 features generated with CC voltage at discharge measured at 4-s and 4-min frequencies are compared with similar methods in the literature. In order to ensure a just and accurate comparison, only papers that adopted a feature-based approach and trained models utilizing the first 100 cycles of the battery

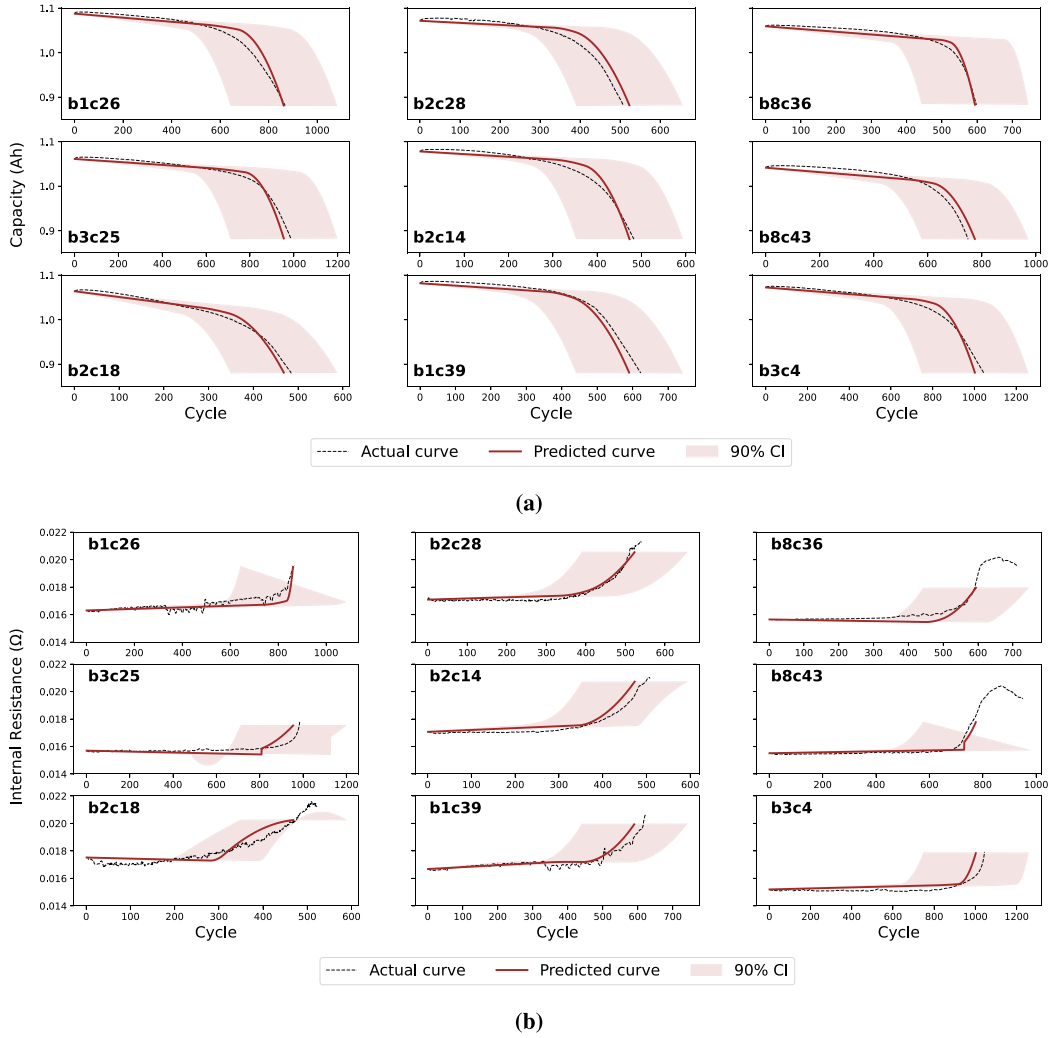


Fig. 10. Full prediction of **(a)** capacity fade and **(b)** IR curves for randomly selected cells from batches 1, 2, 3, and 8 under models built using the combination of features from Tables 1 and 2 generated under high-frequency data (4 s).

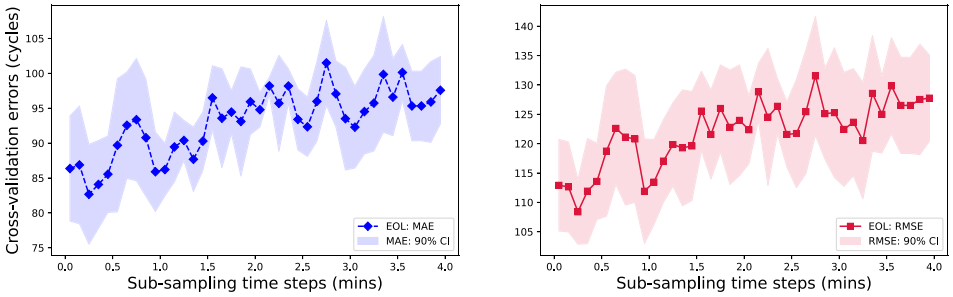


Fig. 11. Cross-validation results (together with 90% confidence interval) using different time steps for CC discharge voltage sub-sampling — a case of EOL prediction using a combination of features in Tables 1 and 2.

cycling data (with the exception of [3], which used 50 cycles) employed in this study were considered for citations. For brevity, the MAPE on EOL prediction is compared. The summary of the comparative analysis is presented in Table 10. The key focus and contribution are the data regime used for feature generation, the frequency of data measurement, and the amount of data needed for making predictions.

An extreme gradient boosting regression model was built in this study utilizing only in-cycle CC discharge voltage curves which makes it stand out from all reviewed models except those of [3]. This work produces a 4-min model that is robust to sub-sampling rate with only a 13% error increase (with respect to the 4-s model) in the case of a model trained on batches 1, 2, and 3 (where it is a 63% increase in [3]).

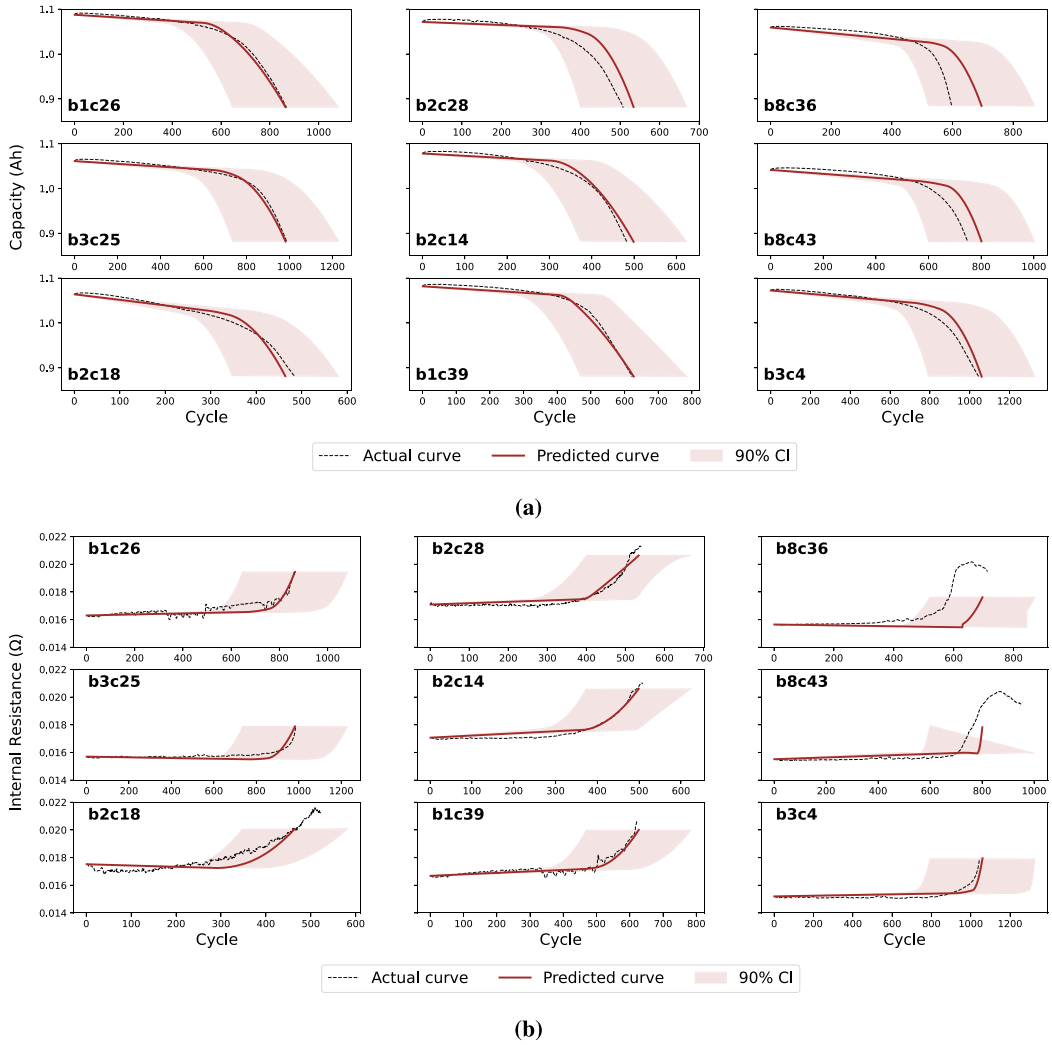


Fig. 12. Full prediction of **(a)** capacity fade and **(b)** IR curves for randomly selected cells from batches 1, 2, 3, and 8 under models built using the combination of features from Tables 1 and 2 and CC discharge voltage measured at the end of every 4 min.

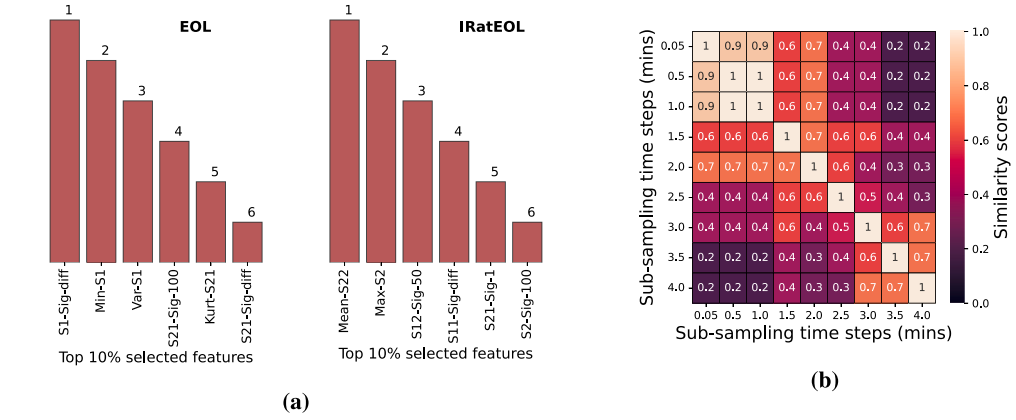


Fig. 13. **(a)** Bar chats of the ranks of top 10% features obtained by applying RRCT algorithm to the combination of features in Tables 1 and 2 which were generated using CC discharge voltage sub-sampled at every 4 min. **(b)** Heat map of similarity scores showing how the first ten selected features (by the RRCT) are similar across different sub-sampling time steps.

5. Conclusion

The deployment of the signature method for predicting capacity and internal resistance degradation curves of lithium-ion batteries from early data and under a low data regime has been addressed.

Empirically, a sample rate of one sample every 4 min returns adequate information for prediction (using 6 features – RRCT selected). This is a strong reduction over previous approaches encoding this information and a significant gain over existing literature that could sample only up to one minute. The power of the signature methodology

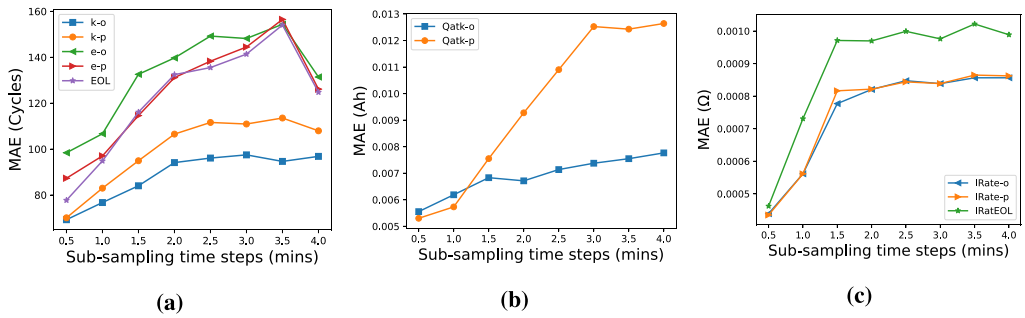


Fig. 14. Mean absolute errors in the prediction of **(a)** knees, elbows and EOL, **(b)** capacity at knees, and (c) IR at elbows and EOL using the model trained under 4-s data which was applied to features obtained from 0.5 to 4-min data sub-sampling.

Table 10
Comparison of the performance of our 4-s and 4-min data models in predicting EOL. For comparison purposes, feature-based models built on the cycling data belonging to batches 1, 2, and 3 from [5] were considered. In places where numbers are marked with an asterisk (*), it means the metric is obtained from a model built including batch 8 [6]. Meaning of symbols: measured voltage at time t (V), discharge capacity (Q), current (I), temperature (T), IR (IR), discharge capacity-voltage curve ($Q(V)$), state of health (SOH) and charge time (ct). In places of FaD (Full at discharge), it means the method makes use of the entire information at discharge; whereas CCaD (CC at discharge) depicts the data is derived only from constant-current conditions at discharge. The MAPE in parenthesis in some methods refers to the errors obtained on secondary test data.

Papers	Data used from [5]	Data regime	Sampling frequency	MAPE (%)
This work	V	CCaD	4 min	14.3, 13.2*
	V	CCaD	4 s	12.6, 10.0*
[3]	V	CCaD	4 min	19.5, 19.7*
	V	CCaD	4 s	12.0, 12.6*
[4]	$Q(V)$	FaD	4 s	~22.0
Variance [5]	$Q(V)$	FaD	4 s	15.0 (11.0)
Discharge [5]	$Q(V), V, I$	FaD	4 s	10.1 (8.6)
Full [5]	$Q(V), V, I, T, IR$	FaD	4 s	7.5 (10.7)
[15]	SOH, $Q(V)$, IR, ct	FaD	4 s	9.0
[14]	$Q(V)$	FaD	4 s	11.7
[16]	SOH, $Q(V), V, T$	FaD	4 s	7.0

for this type of application was confirmed by two industry-relevant stress tests: by (i) building an accurate predictive model from data sampled directly at every 4 min, and (ii) training an accurate model in the high-data regime and then predict using the data sampled at every 4 min. The latter showcases model robustness towards loss of input data.

A comparative analysis showed a substantial dissimilarity of the model's top relevant features set between the high- and low-data regimes. This hints at a separation of predictive factors between data regimes, and thus the low-data regime still requires further study. The manuscript's model outperforms existing low-data regime models and is moderately competitive within the existing high-date regime literature (where comparison is possible).

The proposed models were built for robustness and interpretability. On *robustness*, we were able to establish that the developed models estimate both capacity and IR curves with high accuracy even when there is noisiness in the input data.

With respect to model *interpretability*, only the second level of the signature for feature extraction was used. The reviewed literature has established this level to have an interesting geometrical and physical interpretation. Moreover, the importance of each generated feature through the XGBoost feature importance functionality is accounted for.

To support future research, the modeling code (in CC BY 4.0) is available to anyone wanting to replicate or develop the findings. In terms of open avenues for further exploration, the signature method can be applied as a feature generation mechanism to the more infrequent Reference Performance Test (RPT) data to predict battery degradation. Also open is the application of the method to *varying currents* under charging/discharging conditions — the simplest version is the question would be lifetime predictions for fast charging of electric vehicles whose batteries are charged in (mostly) multi-stage constant current charging conditions. Lastly, the models of this manuscript were developed using the Severson–Attia–Strange data and took as input the discharge voltage data at CC. It is left for future research to carry out the analysis using *charge voltage at CC* as the input data instead of the discharge one.

Methods

An introduction to extreme gradient boosting (XGBoost)

XGBoost is a supervised learning algorithm where a set of features are used for predicting a set of targets. In contrast to many tree-based ensemble models, which average predictions made on individual trees, XGBoost builds regression trees additively where a candidate tree is added only if it improves the value of a chosen objective function.

Following [35], given a training set $\{\boldsymbol{X}, \boldsymbol{y}\}$ where $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_m)$, $\boldsymbol{x}_i \in \mathbb{R}^d$ is the matrix of features, $\boldsymbol{y} = (y_1, y_2, \dots, y_m) \in \mathbb{R}^m$ is the target vector, d is the dimension of each of the features, and m is the number of samples, XGBoost makes use of the following to predict output:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), \quad f_k \in \mathbb{F}, \quad i = 1, 2, \dots, m,$$

where $\mathbb{F}$ denotes the space of all classification and regression trees (CARTs) and K is the number of trees. Here, every f_k corresponds to an independent tree structure as well as leaf weights or scores $\boldsymbol{w}$. Contrary to decision trees, each regression tree has a continuous score on each of the associated leaves. For a given sample in the data, the decision rules in the trees are used to classify it into the leaves and evaluate the final prediction by simply adding up the scores $\boldsymbol{w}$ in the corresponding leaves. To learn the set of functions f_k used in the model, the following

regularized objective function is minimized:

$$L = \sum_{i=1}^{m} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{8}$$

where $\Omega(f)$ is a function defined by $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\boldsymbol{w}\|^2$ which put a penalty on the complexity of the model, l depicts the differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i, and T is the number of leaves in each tree. It is worth noticing that the regularization terms, λ and γ, are added for the sole purpose of smoothening the final learned weights to curb over-fitting. In fact, when the regularization parameters are set to zero, the regularized objective function in Eq. (8) boils down to the ordinary gradient tree boosting [36]. Due to the fact that the tree ensemble model described above in Eq. (8) uses functions as parameters, it cannot be optimized through the regular optimization techniques in Euclidean space. To circumvent this barrier, the model is trained in an additive way: if $\hat{y}_i^{(t)}$ is the prediction on the ith instance at the tth iteration, it will be needed to include f_t in order to minimize the objective function given by

$$L^{(t)} = \sum_{i=1}^{m} l(y_i, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{x}_i)) + \Omega(f_t).$$

In other words, f_t, which most improves the model objective function is greedily added at each iteration. XGBoost is available in the scikit-learn library and further mathematical details can be found in [35].

Machine learning performance metrics

In this study, we employed *mean absolute error* (MAE), *mean absolute percentage error* (MAPE), and *root mean squared error* (RMSE) for model performance measure, and they are defined below:

$$\begin{cases} \text{MAE}(y, \hat{y}) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} |y_i - \hat{y}_i| \\ \text{MAPE}(y, \hat{y}) = \dfrac{100\%}{n}\displaystyle\sum_{i=1}^{n} \dfrac{|y_i - \hat{y}_i|}{y_i} \\ \text{RMSE}(y, \hat{y}) = \sqrt{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \end{cases} \tag{9}$$

where $y_i, \hat{y}_i$ are the actual and predicted values for sample i respectively, and n is the number of samples.

CRediT authorship contribution statement

Rasheed Ibraheem: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation. **Yue Wu:** Writing – original draft, Methodology, Conceptualization. **Terry Lyons:** Supervision, Methodology. **Gonçalo dos Reis:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

The data has already been made available.

Funding

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

R. Ibraheem was supported by the EPSRC Centre for Doctoral Training in Mathematical Modelling, Analysis and Computation (MAC-MIGS) funded by the UK Engineering and Physical Sciences Research Council (grant EP/S023291/1), Heriot-Watt University and the University of Edinburgh.

G. dos Reis acknowledges partial support from the *Fundação para a Ciência e a Tecnologia* (Portuguese Foundation for Science and Technology) through the project UIDB/00297/2020 and UIDP/00297/2020 (Center for Mathematics and Applications, CMA/FCT/UNL). G. dos Reis acknowledges support the Faraday Institution [grant number FIRG049].

T. Lyons acknowledges funding from Alan Turing Institute through Engineering and Physical Sciences Research Council (EPSRC) grant EP/N510129/1 and funding from EPSRC through the project EP/S2026347/1, titled "Unparameterised multi-modal data, high order signature, and the mathematics of data science". T. Lyons acknowledges additionally support from the Hong Kong Innovation and Technology Commission (InnoHK-CIMDA).

Code availability

The experiments carried out in this research were written in Python programming language. All codes for data cleaning, preprocessing, feature generation, plotting, and model building can be found in https://github.com/Rasheed19/signature-project under a 'CCBY4.0'license.

Of complementary and independent interest, the Python implementation for the RRCT algorithm [26] is available from the Python package repository PyPI at https://pypi.org/project/rrct/.

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.

References

[1] Dunn B, Kamath H, Tarascon J-M. Electrical energy storage for the grid: a battery of choices. Science 2011;928–35.
[2] Nykvist B, Nilsson M. Rapidly falling costs of battery packs for electric vehicles. Nature Clim Change 2015;(5):329–32.
[3] 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.
[4] Saxena S, Ward L, Kubal J, Lu W, Babinec S, Paulson N. A convolutional neural network model for battery capacity fade curve prediction using early life data. J Power Sources 2022;542:231736.
[5] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91.
[6] 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.
[7] 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.
[8] Fermín-Cueto P, McTurk E, Allerhand M, Medina-Lopez E, Anjos MF, Sylvester J, dos Reis G. Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells. Energy AI 2020;1:100006.
[9] Li W, Sengupta N, Dechent P, Howey D, Annaswamy A, Sauer DU. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. J Power Sources 2021;482:228863.
[10] You H, Zhu J, Wang X, Jiang B, Sun H, Liu X, et al. Nonlinear health evaluation for lithium-ion battery within full-lifespan. J Energy Chem 2022;72:333–41.
[11] Greenbank S, Howey D. Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life. IEEE Trans Ind Inf 2021;18(5):2965–73.
[12] Tang X, Wang Y, Liu Q, Gao F. Reconstruction of the incremental capacity trajectories from current-varying profiles for lithium-ion batteries. iscience 2021;24(10).

[13] Ji S, Zhu J, Lyu Z, You H, Zhou Y, Gu L, Qu J, Xia Z, Zhang Z, Dai H. Deep learning enhanced lithium-ion battery nonlinear fading prognosis. J Energy Chem 2023.
[14] Shen S, Nemani V, Liu J, Hu C, Wang Z. A hybrid machine learning model for battery cycle life prediction with early cycle data. In: 2020 IEEE transportation electrification conference & expo. 2020, p. 181–4.
[15] Ma Y, Wu L, Guan Y, Peng Z. The capacity estimation and cycle life prediction of lithium-ion batteries using a new broad extreme learning machine approach. J Power Sources 2020;476:228581.
[16] Yang F, Wang D, Xu F, Huang Z, Tsui K-L. Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model. J Power Sources 2020;476:228654.
[17] 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).
[18] Kim J, Chun H, Kim M, Yu J, Kim K, Kim T, et al. Data-driven state of health estimation of li-ion batteries with RPT-reduced experimental data. IEEE Access 2019;7:106987–97.
[19] Li P, Zhang Z, Grosu R, Deng Z, Hou J, Rong Y, et al. An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries. Renew Sustain Energy Rev 2022;156:111843.
[20] Aitio A, Howey DA. Predicting battery end of life from solar off-grid system field data using machine learning. Joule 2021;5(12):3204–20.
[21] Moore P, Iliant T-M, Ion F-A, Wu Y, Lyons T. Path signatures for non-intrusive load monitoring. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing. IEEE; 2022, p. 3808–12.
[22] Lyons T, Ni H, Oberhauser H. A feature set for streams and an application to high-frequency financial tick data. In: Proceedings of the 2014 international conference on big data science and computing. BigDataScience '14, New York, NY, USA: Association for Computing Machinery; 2014.
[23] 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.
[24] 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.
[25] Ilya C, Andrey K. A primer on the signature method in machine learning. In: Signature methods in Finance. Forthcoming volume in Springer Lec. Notes in Mathematics, 2016, [in press] (arXiv:1603.03788).
[26] Tsanas A. Relevance, redundancy, and complementarity trade-off (RRCT): A principled, generic, robust feature-selection tool. Patterns 2022;3(5):100471.
[27] Strange C, Li S, Gilchrist R, Dos Reis G. Elbows of internal resistance rise curves in Li-ion cells. Energies 2021;14(4):1206.
[28] Tian J, Xiong R, Shen W, Lu J, Yang X-G. Deep neural network battery charging curve prediction using 30 points collected in 10 min. Joule 2021;5(6):1521–34.
[29] 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.
[30] Chen K-T. Integration of paths – A faithful representation of paths by noncommutative formal power series. Trans Amer Math Soc 1958;89:395–407.
[31] 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.
[32] 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).
[33] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 2020;17:261–72.
[34] Attia PM, Bills A, Planella FB, Dechent P, dos Reis G, Dubarry M, et al. Review–"knees" in lithium-ion battery aging trajectories. J Electrochem Soc 2022;169(6):060517.
[35] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM; 2016, p. 785–94.
[36] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.