Applied Energy 378 (2025) 124820

Contents lists available at ScienceDirect

Applied Energy

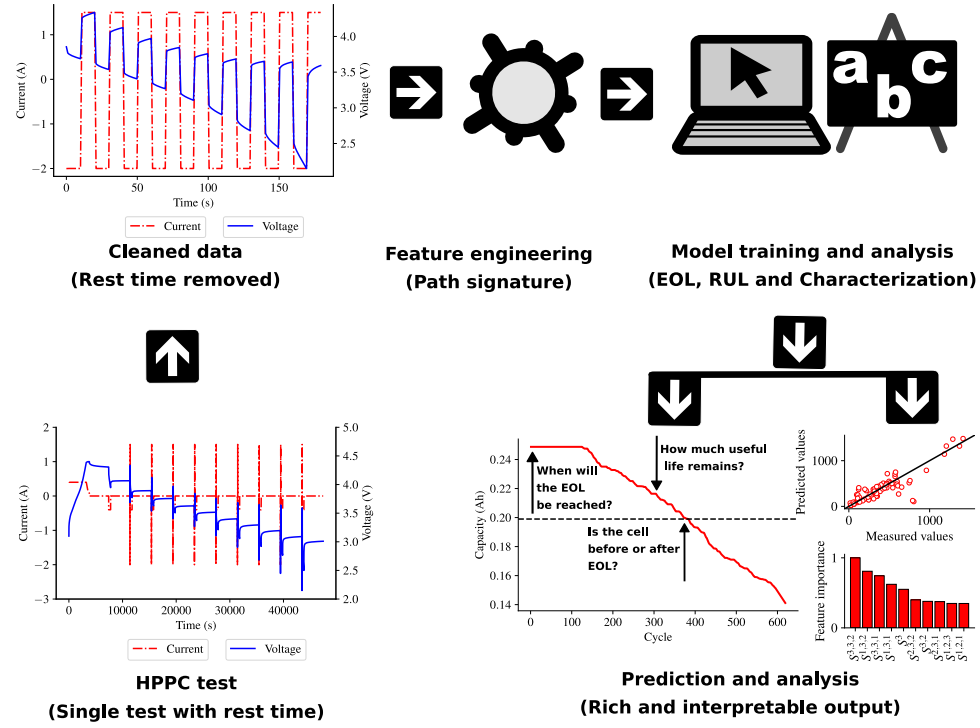
journal homepage: www.elsevier.com/locate/apenergy

Path signature-based life prognostics of Li-ion battery using pulse test data

Rasheed Ibraheem [a], Philipp Dechent [b,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, United Kingdom
[b] Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Campus Boulevard 89, Aachen, 52074, Germany
[c] Visiting Researcher at Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, United Kingdom
[d] Center for Mathematics and Applications (NOVA Math), NOVA School of Science and Technology (NOVA FCT), Campus da Caparica, Caparica, 2829-516, Portugal

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Capacity degradation
Hybrid Pulse Power Characterization testing
Path signature methodology
Lithium-ion cells

ABSTRACT

Common models predicting the End of Life (EOL) and Remaining Useful Life (RUL) of Li-ion cells make use of long cycling data samples. This is a bottleneck when predictions are needed for decision-making but no historical data is available. A machine learning model to predict the EOL and RUL of Li-ion cells using only data contained in a single Hybrid Pulse Power Characterization (HPPC) test is proposed. The model ignores

* 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.apenergy.2024.124820
Received 21 March 2024; Received in revised form 24 October 2024; Accepted 28 October 2024
Available online 8 November 2024

Explainable machine learning
Remaining useful life
End of life

the cell's prior cycling usage and is validated across nine different datasets each with its cathode chemistry. A model able to classify cells on whether they have passed EOL given an HPPC test is also developed.

The underpinning data-centric modelling concept for feature generation is the notion of 'path signature' which is combined with an explainable tree-based machine learning model and an in-depth study of the models is provided. Model validation across different SOC ranges shows that data collected from the HPPC test across a 20% SOC window suffices for effective prediction. The EOL and RUL models achieve 85 and 91 cycles MAE respectively while the classification model has an accuracy of 94% on the test data. Code for data processing and modelling is publicly available.

1. Introduction

Rechargeable batteries have been playing a pivotal role in transitioning into a net zero. They are well-adopted in the production of electric cars and also in emerging markets such as heavy-duty vehicles [1] and aviation [2]. Thus, efficient and easy-to-deploy methods are needed to study their degradation and characterization. Machine learning has been successfully deployed in battery modelling by way of training algorithms on battery health indicators to identify inherent patterns that can be used for general prediction on unseen data. This approach has been proven to have desirable features including good generalization accuracy [3], transfer learning [4,5], and (statistical) robustness [6,7]. It has also been critically reviewed in [8–10].

Machine learning models are data-driven and require clean and accurate data for training and prediction. The most common data used for data-driven battery modelling is high-throughput cell-cycling data, Electrochemical Impedance Spectroscopy (EIS), or current pulses such as the Hybrid Pulse Power Characterization (HPPC) test [11,12]. Cycling data contains records of temperature, voltage, and current over time and cycle, and has been used extensively in predicting the End of Life (EOL) [3,6,7,13], Remaining Useful Life (RUL) [9,14,15], entire capacity and Internal Resistance (IR) degradation curves [6, 7,16,17], knee-points [18–20], and building classification models for battery characterization [13,14]. EIS data has been used for capacity prediction [21] and RUL prediction [22].

Pulse resistance tests such as the HPPC test are a type of technique for accessing the performance and pre-defined features of batteries under short charge and discharge current pulses at different charge levels. During this test, current and voltage are measured, and State of Charge (SOC) and pulse resistances (also sometimes referred to as IR) are calculated. Unlike cycling and EIS data that have been utilized extensively for prediction, pulse tests are commonly used for the estimation of State of Health (SOH) [23–26], physics-based model parameter values [27], and cell-clustering by capacity [28]. This study bridges this gap by proposing machine learning models for the prediction of EOL, RUL, and failure status (classify cells on whether they have passed EOL).

From a business perspective, lifetime predictive models built on HPPC data are both economical and time-efficient. Data from long longitudinal data-collection approaches, such as monitoring usage, takes weeks to generate and requires constant monitoring, additional data storage systems and costs, and transmission. As a result, it adds to the cost of data acquisition and increases the possibility of data loss, especially for a manufacturer who relies on independent measurements without access to battery management data. To tackle these challenges, this work focuses on the use of less data (HPPC test) that can be recorded in a timescale of hours (as opposed to weeks), without any access to historical data, to develop machine learning models for battery health prognosis. The derived benefits are clear. For example, the case of a battery life evaluation for a used electric vehicle (which would ideally be done during a car listing for sale). In this scenario, a desirable method is to get data as quickly as (within one visit) possible by performing an HPPC test on the battery, making inferences with our proposed models, and finally deciding on the battery's state for valuation.

In terms of *research contributions*, there is scant literature on predictive lifetime models using pulse data such as the HPPC and this research offers a proof of concept, validated across 9 datasets, for explainable, interpretable, and low-latency models. Arguably, the modelling pipeline framework used here is classic: data cleaning, feature extraction, and model building and validation. Nonetheless, *the key technical point* here is the use of the novel 'truncated path signature' method for feature generation in combination with an explainable tree-based machine learning model. This latter choice yields business-sought interpretability and explainability of the proposed models.

Path signatures were employed for the first time in battery modelling in [7]. There, the 'path signature' method was used for the prediction of the full capacity and IR degradation curves from sparse information on the voltage profile. Path signatures originated from rough path theory [29,30] and can be likened to Fourier transform but they are much more. Fourier transforms extract signals of different frequencies from the underlying data. The signature extracts the area, segmental, and geometrical effects of the corresponding path. Its properties include time-reparametrization and translation invariance, less sensitivity to missing data, unique characterization of the path, and universal nonlinearity (full details in Section 3). In this study, the input features of the models is the signature of the path generated by time-current–voltage profiles across multiple SOCs in the HPPC test. Impurity-based and permutation importance are adopted to account for the impact of each signature term on the built models. By design, each model component can be transparently traced from output to its root features for fixing in case of any fault or model performance decay. This is in stark contrast to other no-history lifetime prediction models such as the one-cycle models [9,14,31] based on feature-free deep-learning algorithms. Lastly, for flexibility of implementation, each proposed model has been validated in situations where cells are at different levels of SOC and across different chemistries. This allows for making inferences even if the collected cells are not fully charged and reduces further the amount of data needed from the HPPC test.

The rest of this work is organized as follows: Section 2 describes the data used in this study; Section 3 provides the various approaches for feature engineering and model building; the details of the experimental results and the corresponding discussion are presented in Section 4; and conclusions were made in Section 5. The mathematical details of various algorithms used in the methodology are left to the Methods section and Supplementary Material.

2. Data: source, analysis, and preprocessing

2.1. Data source and data description

The datasets of this research are the 9 datasets (totalling 300 lithium-ion cells) used in [3] and made available in [32]. These cells were produced in the Argonne Cell Analysis, Modelling, and Prototyping (CAMP) facility and cover nine different cathode chemistries including $Li_{1.2}Ni_{0.3}Mn_{0.6}O_2$, $Li_{1.35}Ni_{0.33}Mn_{0.67}O_{2.35}$, FCG (Full concentration gradient NMC), NMC811, NMC111, NMC622, NMC532, 5Vspinel, and HE5050; see Table 1 for the number of cells per cathode chemistry used in this study.

As described in [3,32] (and further [17,31]), the datasets' cells were selected based on three criteria: the presence of graphite in the anodes as the main constituent, limitation of charging rate to $\leq$ 1C to reduce

Table 1
The breakdown of cathode groups by the number of cells. The 'Original' column refers to the actual number (300 in total) of cells in the downloaded data. The 'Processed' column corresponds to the number of cells in each group after cells that do not undergo pulse testing (9 cells), those that have reached the EOL before the first pulse test (23 cells), and those with irregular pulse voltage profiles (3 cells) have been removed (details in Section 2.2).

Cathode group	Number of cells	
	Original	Processed
$Li_{1.2}Ni_{0.3}Mn_{0.6}O_2$	6	6
$Li_{1.35}Ni_{0.33}Mn_{0.67}O_{2.35}$	4	2
FCG (Full concentration gradient NMC)	8	8
NMC811	15	15
NMC622	20	20
NMC532	100	89
NMC111	16	16
HE5050	78	77
5Vspinel	53	32
Total	300	265

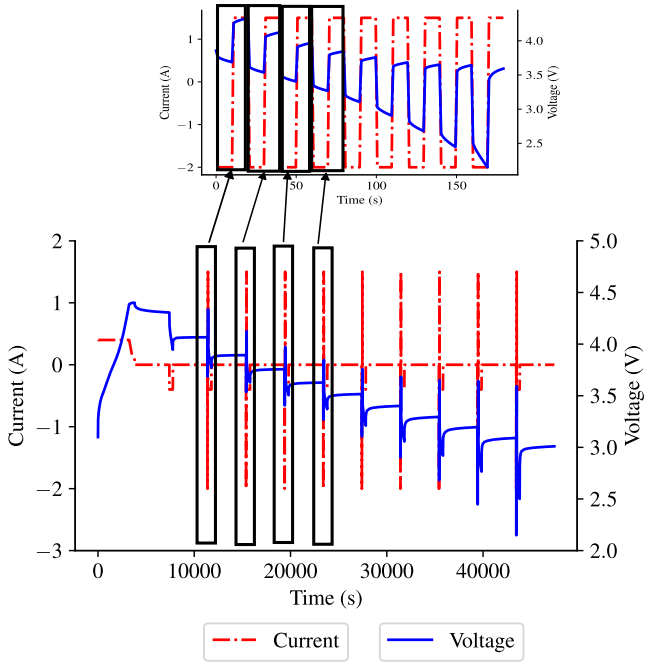


Fig. 1. Current and Voltage profiles over time for a sample cell under HPPC test. The lower plot displays the HPPC test across its full duration. Removing the rest times (from the lower figure) yields the top figure which then displays the actual regimes of interest used in this study (discharge and charge over 1 min each). The negative and positive currents indicate the discharging and charging regimes respectively.

the effect of lithium plating on battery capacity loss, and each cell lasts at least 100 cycles. These datasets were designed to have desirable characteristics including a wide range of cathode chemistries, in-chemistry variability (such as differences in electrode porosity), different suppliers (thus manufacturing variability), and various charging protocols. In terms of data components, the datasets consist of in-cycle and per-cycle measurements. In-cycle data contains measurements recorded per time (in seconds) for a given cycle number and comprises (among others) test time, voltage, and current corresponding to the HPPC test (see Fig. 1) and ordinary cycling process. The HPPC test is not carried out at every cycle but at regular cycle intervals (approximately every 50 cycles) and is described in more detail in Section 2.2 below. The per-cycle data contains measurements recorded at the end of every cycle and consists of cycle number, charge capacity, and discharge capacity. More information about these datasets can be found in [3,32] and [3, Supplementary material].

2.2. Data analysis: HPPC test and data for predicting targets

The HPPC test of this dataset. This research uses the extracted pulse testing data for the development of features for modelling and this is exactly the data that is dropped in [3] (as their focus was a prediction from standard high-throughput cell cycling data). These pulse tests, called HPPC tests, are carried out at regular intervals (roughly every 50 cycles) across the cell's life [33]. The HPPC test starts with the cell at 100% SOC and consists of a short discharge (10 s at current 2 A), a rest (40 s), and a charge (10 s at current 2 A) – a total of 1 min; after, a C/1 rate is applied to reduce the SOC by 10% and the cell is left to rest for approximately 1 h. This process is repeated until the cell reaches 0% SOC (9 times in total in decreasing increments of 10% SOC). The SOC (at any cycle) is defined using the present capacity. Thus, the amount of HPPC tests per check-up remains the same over the lifetime. From these pulse data, current and voltage profiles were retained for feature extraction leaving out qualitative logs like the state of the tester and the current state of the battery. Fig. 1 shows the current and voltage profiles under a pulse test for a sample cell, and, for this research, the rest times were dropped from the HPPC data leaving the current and voltage profiles corresponding to cropped times. The voltage profile varies between 2 V and 5 V.

Targets for the predictive models. To execute the three supervised learning tasks proposed, we define the three relevant targets (or labels) for the models to predict (from the pulse current–voltage response data): EOL, RUL, and failure status (a binary response indicating whether a cell has passed the EOL; see Table 3 below for a summary). We define EOL as the cycle number corresponding to 80% of initial capacity. Concerning the RUL, for each cell and each pulse test, the difference between the cycle at which such pulse test was carried out and EOL (i.e., EOL – pulse test cycle number) is calculated. The binary response that depicts whether a cell has passed its EOL (coded as 0) or not (coded as 1) based on each pulse test is considered. As a note, 9 out of the 300 cells did not undergo a pulse test during cycling and thus were excluded. In addition, all cells that reached EOL before the first pulse test (23 cells) and those with irregular voltage profiles (3 cells) were dropped. With these additional refinements, 265 cells were kept.

Fig. 2(**a**) shows the breakdown (by cathode chemistries) of cycles at which the first pulse test was carried out. The first pulse cycles spread between the 5th and 70th cycles, making it feasible to consider feature extraction based on the first cycles within the first 100 cycles. Fig. 2(**b**) illustrates the cycle life of cells in each cathode group. It can be seen that the data covers a wide range of EOL which is a desirable property for building a machine learning model characterized by a low generalization error.

2.3. Capacity curve data preprocessing

Critical to this research is the preprocessing or cleaning step addressing the variability of the discharge capacity curves of the cells. Since rest periods, and measurements like Reference Performance Test (RPT) or HPPC can induce immediate capacity [34,35], dips and peaks in the capacity values are present in the data (see blue curves of Fig. 3). Thus they need to be filtered appropriately before extracting the prediction targets—EOL, RUL, and binary responses.

In this research, three filtering methods were combined namely median filter, Savitzky–Golay, and isotonic regression. The mathematical description of Savitzky–Golay is given in the Methods section and those of median filter and isotonic regression are left to the Supplementary Material. The median filter of a window length of 5 was first applied to the capacity values to remove all the extreme outliers and noises. Following this, the Savitzky–Golay filter of a window length of 50 and first-degree polynomial were used to smoothen the resulting values. Here, it is emphasized that the Savitzky–Golay filter is preferable to that of line-exponential because it does not depend on the degradation pattern of the cells' capacity. Finally, isotonic regression

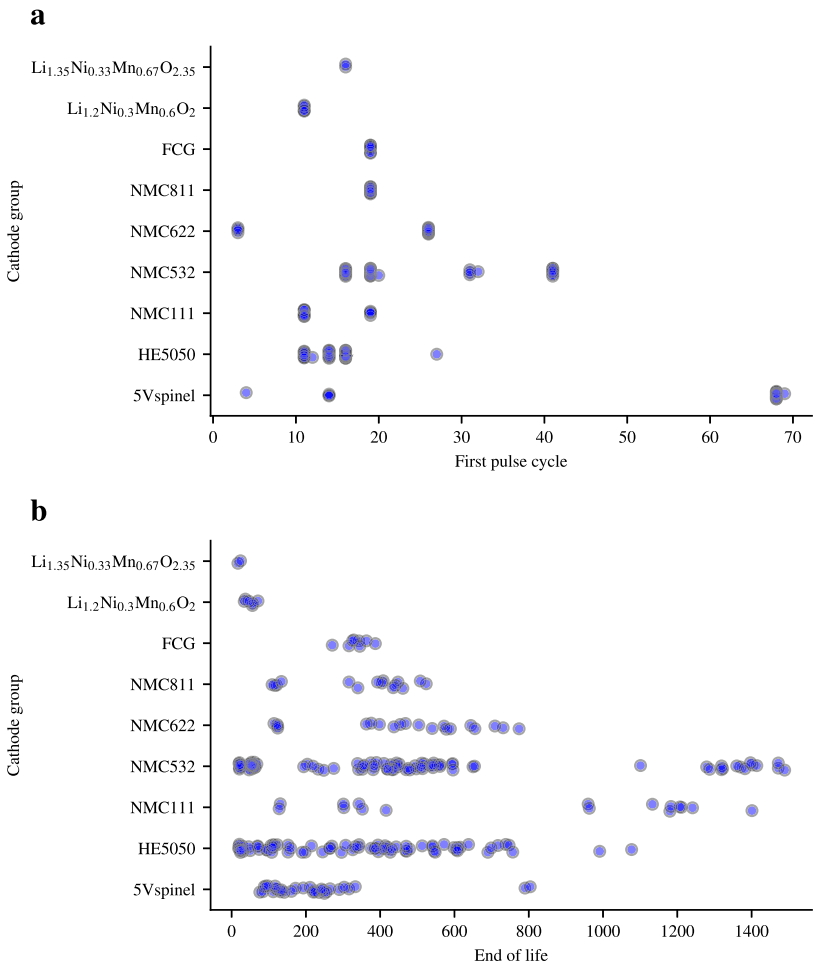


Fig. 2. (**a**) Cycle number of the first pulse test for each of the cathode chemistries. All the first pulse tests were carried out within the first 100 cycles. (**b**) The distribution of EOL of cells across each cathode chemistry. The data covers a wide range of EOLs, making it possible to build a model characterized by a low generalization error. The discrepancy in some cells' number distribution (in comparison to that of [3, Figure 1a]) can be associated with the removal of cells that did not undergo an HPPC test (see Table 1) and smoothened capacity curves (see Fig. 3). The source code of this study (including data cleaning steps) is publicly available; see Code Availability section below.

(monotone decreasing version) was fitted to the filtered values to achieve a monotone-decreasing capacity curve with respect to the cycle numbers. The initial capacity was obtained by calculating the median of the first ten values of the cleaned data. This choice was considered for two reasons: to mitigate the effect of the peaks and troughs, and to allow for the notion that it takes some cycles before the formation of the cell is finished and before an actual degradation; see Fig. 3 for the results of this filtering process for one sample cell from each cathode chemistry.

3. Modelling and methodology

3.1. An overview of path signatures

We now introduce the workhorse of our study: *path signatures*. We first introduce it as a mathematical object and then offer a comparative discussion on the distinguishing traits of path signatures in data representation. Following the description of path signatures given in [7, Section 3.2], the term 'signature' can be regarded as a transform and can be likened to the Fourier transform because it maps a stream of data (such as time series) to a new space. Contrary to the Fourier transform and other similar transforms which extract frequency information from data, signatures extract both area and segmental effects of the underlying function which generate the data under transformation. In addition, it is a universal nonlinearity in that every continuous function of a data stream under transformation can be well approximated by a linear function of its signatures. This makes signature a good choice for various machine learning problems including physical quantity encoding, data compression, and feature extraction [29,38]. It is necessary to clarify exactly what is meant by a path and iterated integral before giving a mathematical definition of a path signature.

Path. A $\mathbb{R}^d$-valued path X (or denoted by X_t to show its dependence on $t \in [a, b]$) is a continuous mapping from a real interval $[a, b]$ into a d-dimensional Euclidean space;

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

The broad use of the term 'path' is sometimes equated with the trajectory traversed by a process with well-defined start and end points. A good example can be found in the trajectory of a projectile, the stock price over a specified time, and the capacity fade curve of a battery from the nominal value to EOL.

Iterated integral. Suppose X_t is a real d-dimensional path where each dimension depends on the parameter $t \in [a, b]$ as given in Eq. (1). The kth level iterated integral of X_t is given by

$$S(X_t)_{a,t}^{i_1,\dots,i_k} := \int_{a<t_k<t} \cdots \int_{a<t_1<t_2} dX_{t_1}^{i_1} \dots dX_{t_k}^{i_k}, \tag{2}$$

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

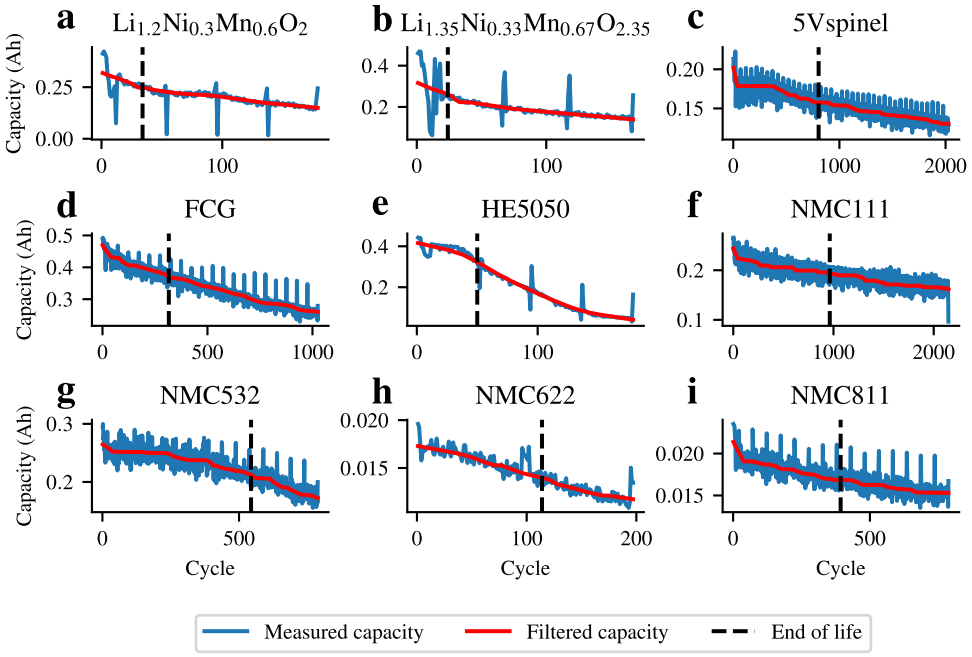


Fig. 3. The filtered capacity curves for a random cell in each cathode chemistry. A three-step filtering process was considered: the median filter with a window 5, Savitzky–Golay filter [36] of window 50 and polynomial of order 1, and isotonic regression (monotone decreasing) [37]. EOL marked as 80% nominal capacity.

Having defined a path and iterated integral, the following gives the technical definition of the signature of a path according to [29].

Signatures. The path signature of X_t over $[a,b]$ is defined as a real sequence of numbers where each of its terms is an iterated integral defined in Eq. (2) with superscripts 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. In notation, this sequence can be written as

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

A well-known example of a path signature is that of a 2-dimensional path. Its first few terms have been associated with *Lévy* area and statistical moments [29]. In general, suppose $X_t = \{X_t^1, X_t^2\} = \{f(t), g(t)\}$ with $dX_t = \{dX_t^1, dX_t^2\} = \{f'dt, g'dt\}$ where f and g are differentiable functions in the interval $[a,b]$. The first few terms of $S(X_t)_{a,b}$, for $t \in [a,b]$, are calculated as follows

$$S(X)^1_{a,b} = \int_{a<t<b} dX_t^1 = \int_a^b f'\,dt = f(b) - f(a),$$

$$S(X)^2_{a,b} = \int_{a<t<b} dX_t^2 = \int_a^b g'\,dt = g(b) - g(a),$$

$$S(X)^{1,1}_{a,b} = \int_{a<t_1<t_2<b} dX^1_{t_1}\,dX^1_{t_2} = \int_a^b \left[\int_a^{t_2} f'(t_1)\,dt_1\right] f'(t_2)dt_2,$$

$$S(X)^{1,2}_{a,b} = \int_{a<t_1<t_2<b} dX^1_{t_1}\,dX^2_{t_2} = \int_a^b \left[\int_a^{t_2} f'(t_1)\,dt_1\right] g'(t_2)\,dt_2,$$

$$S(X)^{2,1}_{a,b} = \int_{a<t_1<t_2<b} dX^2_{t_1}\,dX^1_{t_2} = \int_a^b \left[\int_a^{t_2} g'(t_1)\,dt_1\right] f'(t_2)\,dt_2,$$

$$S(X)^{2,2}_{a,b} = \int_{a<t_1<t_2<b} dX^2_{t_1}\,dX^2_{t_2} = \int_a^b \left[\int_a^{t_2} g'(t_1)dt_1\right] g'(t_2)dt_2,$$

$$S(X)^{1,1,1}_{a,b} = \int_{a<t_1<t_2<t_3<b} dX^1_{t_1}\,dX^1_{t_2}\,dX^1_{t_3} = \int_a^b \left[\int_a^{t_3}\left[\int_a^{t_2} f'(t_1)\,dt_1\right] f'(t_2)\,dt_2\right] f'(t_3)\,dt_3,$$

and so on.

3.2. Desirable properties and comparison with literature

The concept of path signature was originally described in [39] in which it was applied to piecewise smooth paths. Then in [40], it was broadened to paths characterized by finite length. Path signature, as a feature extraction technique, has characteristics that make it more powerful and versatile (see Table 2 for a comparison with popular feature extractors) – we point the reader to [29] for an intuitive introduction to path signatures methods. Because the path signature consists of iterated path integrals, it inherits their properties of translation and reparametrization invariance. For instance, removing the rest times and extracting measured profiles (e.g., voltage and current) over an SOC range during an HPPC test, where we do not care about the absolute start and end times, both translation and reparametrization invariance are useful properties.

Multi-dimensional time series data can always be represented and encoded in a more compact and representative way by a fixed-length vector. For instance, in this study information from three time series (pulse time, voltage, and current) was encoded in a fix-length feature vector instead of having a separate set of features for each time series. In this application, samples in the time series need not be evenly spaced in time and samples can also be taken using large time steps; see [7] for using signatures for subsampled data.

Another interesting property that makes signature a good set of features for machine learning tasks is the fact that the full set of signatures of a path X_t (i.e., when the depth $k \to \infty$) uniquely characterizes X_t up to translations and reparameterizations [40].

It is also a universal nonlinearity, a good characteristic for any machine learning problem. This property comes from the fact that linear functionals on the signature $S(X_t)$ of the path X_t are dense in the functional space of all functions of X_t. This means that if f represents the function to be learned during a supervised machine learning task and maps X_t to a set of labels y, then the universal nonlinearity [41] states that for any $\epsilon > 0$, there exists a linear function L such that

$$\left|f(X_t) - L\big(S(X_t)\big)\right| < \epsilon.$$

Table 2
Comparison of path signatures to popular transforms and feature extractors in the literature through the lens of versatility, representation ability and invariance properties.

Versatility	Path signatures are natural for capturing dependencies (area, segmental and geometrical effects), in general, from any time series data and also preserve these dependencies in the time domain (unlike Fourier transform [44] that maps data into the frequency domain and thus time-domain information is lost). They are versatile, with rich theoretical guarantees such as universal non-linearity [41], and not bound to a specific data regime (unlike specially designed, field-informed, classical methods targeting particular data regions such as data under constant-current discharging [6]).
Data representation	Signature is capable of encoding multi-dimensional time series in a more compact and representative way by a fixed-length vector. For instance, information from current and voltage profiles from an HPPC test can be obtained from a path with these profiles as components (stacked profiles) instead of profile-to-profile feature extraction. This is advantageous over popular chemistry-informed features employed in [3,13] where features are *independently* derived from profiles such as voltage, current, temperature, and capacity while completely ignoring their *joint interaction*.
Invariance	Path signatures are guaranteed with both time parameterization and translation invariance. This means that the extracted features will be robust to preprocessing such as reparameterizing times (as in the case of removing rest times from the HPPC test) and shifting measured profile values by a constant value. This is not the case for auto-encoders [16] and transformers [31]. Auto-encoders and transformers are deep learning techniques which are sensitive to changes in time steps or input data shifting.

This property further distinguishes the path signature from Fourier and wavelet basis because it establishes a relationship *between the function of* X_t (and not directly X_t) and a linear function of $S(X_t)$. This is particularly useful in supervised machine learning (both regression and classification) because the labels y are always written as a function of the features (i.e., $y = f(X_t)$).

In this study, a discrete path was considered and a signature can also be calculated on such data. Because transforming a large amount of data stream using signature transform can be highly computationally intensive (especially in the case of a high dimensional path or wanting many terms of the sequence), efficient packages including *iisignature* [42] and *signatory* [43] (implemented in Python) have been developed to handle signature calculation.

3.3. Feature extraction

The steps in generating features for models involve three processes: data extraction, data cleaning, and construction of path signature. As for data extraction, pulse test data corresponding to both charge and discharge regimes were loaded from the 265 cells using the *battery-data-toolkit* Python library.[1] For data cleaning, time series data for current and voltage were selected. Rest periods within each profile were removed before feature extraction (see Fig. 1). This ensured consistency across pulses. Concerning setting up path signature, a 3-dimensional time-augmented path $X_t = \{t, I(t), V(t)\}$ was defined where t, $I(t)$ and $V(t)$ are the cropped pulse time, current and voltage response respectively. The path signature of X_t was then calculated up to depth or level k using the *iisignature* Python library [42], where k was determined via hyperparameter tuning:

$$S(X_t)_{t_0,t_f} = \left(S(X_t)^1_{t_0,t_f}, S(X_t)^2_{t_0,t_f}, S(X_t)^{1,1}_{t_0,t_f}, S(X_t)^{1,2}_{t_0,t_f}, S(X_t)^{2,1}_{t_0,t_f}, \ldots, S(X_t)^{i_1,\ldots,i_k}_{t_0,t_f}\right),$$

where t_0, t_f are the initial and final times for a given pulse snapshot respectively; $i_1, \ldots, i_k \in \{1, 2, 3\}$.

For the model that predicts the EOL, only the first pulse test carried out within the first 100 cycles was used for signature calculation for each of the cells, i.e, the feature matrix $\boldsymbol{F}^{\text{EOL}}$ in this case is given by

$$\boldsymbol{F}^{\text{EOL}} := \begin{bmatrix} S^1_{c_1} & S^2_{c_1} & S^{1,1}_{c_1} & S^{1,2}_{c_1} & S^{2,1}_{c_1} & \cdots & S^{i_1,\ldots,i_k}_{c_1} \\ S^1_{c_2} & S^2_{c_2} & S^{1,1}_{c_2} & S^{1,2}_{c_2} & S^{2,1}_{c_2} & \cdots & S^{i_1,\ldots,i_k}_{c_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ S^1_{c_N} & S^2_{c_N} & S^{1,1}_{c_N} & S^{1,2}_{c_N} & S^{2,1}_{c_N} & \cdots & S^{i_1,\ldots,i_k}_{c_N} \end{bmatrix} \in \mathbb{R}^{N \times \frac{3(3^k-1)}{2}}, \tag{3}$$

where $S^{(\cdot)}_{c_i}$ is understood as a signature component of $S(X_t)_{t_0,t_f}$ corresponding to cell c_i. In the case of the model that predicts RUL, signatures were calculated for each cell and every pulse test before the EOL. The classification model that predicts whether a cell has passed its EOL mark or not has the same matrix structure as that of the RUL model except for the fact that no restriction on the cycle at which the pulse test is carried out. Thus the feature matrix $\boldsymbol{F}^{\text{RUL-CM}}$ in both cases is given by

$$\boldsymbol{F}^{\text{RUL-CM}} := \begin{bmatrix} S^1_{c_{1,1}} & S^2_{c_{1,1}} & S^{1,1}_{c_{1,1}} & S^{1,2}_{c_{1,1}} & S^{2,1}_{c_{1,1}} & \cdots & S^{i_1,\ldots,i_k}_{c_{1,1}} \\ S^1_{c_{1,2}} & S^2_{c_{1,2}} & S^{1,1}_{c_{1,2}} & S^{1,2}_{c_{1,2}} & S^{2,1}_{c_{1,2}} & \cdots & S^{i_1,\ldots,i_k}_{c_{1,2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ S^1_{c_{1,n_1}} & S^2_{c_{1,n_1}} & S^{1,1}_{c_{1,n_1}} & S^{1,2}_{c_{1,n_1}} & S^{2,1}_{c_{1,n_1}} & \cdots & S^{i_1,\ldots,i_k}_{c_{1,n_1}} \\ S^1_{c_{2,1}} & S^2_{c_{2,1}} & S^{1,1}_{c_{2,1}} & S^{1,2}_{c_{2,1}} & S^{2,1}_{c_{2,1}} & \cdots & S^{i_1,\ldots,i_k}_{c_{2,1}} \\ S^1_{c_{2,2}} & S^2_{c_{2,2}} & S^{1,1}_{c_{2,2}} & S^{1,2}_{c_{2,2}} & S^{2,1}_{c_{2,2}} & \cdots & S^{i_1,\ldots,i_k}_{c_{2,2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ S^1_{c_{2,n_2}} & S^2_{c_{2,n_2}} & S^{1,1}_{c_{2,n_2}} & S^{1,2}_{c_{2,n_2}} & S^{2,1}_{c_{2,n_2}} & \cdots & S^{i_1,\ldots,i_k}_{c_{2,n_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ S^1_{c_{N,1}} & S^2_{c_{N,1}} & S^{1,1}_{c_{N,1}} & S^{1,2}_{c_{N,1}} & S^{2,1}_{c_{N,1}} & \cdots & S^{i_1,\ldots,i_k}_{c_{N,1}} \\ S^1_{c_{N,2}} & S^2_{c_{N,2}} & S^{1,1}_{c_{N,2}} & S^{1,2}_{c_{N,2}} & S^{2,1}_{c_{N,2}} & \cdots & S^{i_1,\ldots,i_k}_{c_{N,2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ S^1_{c_{N,n_N}} & S^2_{c_{N,n_N}} & S^{1,1}_{c_{N,n_N}} & S^{1,2}_{c_{N,n_N}} & S^{2,1}_{c_{N,n_N}} & \cdots & S^{i_1,\ldots,i_k}_{c_{N,n_N}} \end{bmatrix} \in \mathbb{R}^{\sum_{i=1}^{N} n_i \times \frac{3(3^k-1)}{2}}, \tag{4}$$

where $S^{(\cdot)}_{c_{i,j}}$ is an element of the sequence $S(X_t)_{t_0,t_f}$ corresponding to cell c_i at pulse test j; $n_1, \ldots, n_N$ are the number of pulse tests for the cells $c_1, \ldots, c_N$ respectively.

3.4. Machine learning models and techniques

The Extreme Gradient Boosting (XGBoost) [45] was chosen as a machine learning model in this research. An overview of the mathematical details of this algorithm is presented in the Supplementary Material. The scikit-learn [46] implementation of the algorithm was used for all the experiments carried out in this study. Three models were proposed, namely *EOL model*, *RUL model*, and *classification model*, which predict the EOL, RUL, and binary responses (a cell has passed the EOL or not), respectively; see Table 3 for model description.

To train these models, 182 cells were set aside for training and 83 for testing and generalization error estimation. This approximately corresponds to a 70%–30% train–test split strategy. Each split maintains the percentage of cells in the cathode groups such that every cathode chemistry has cells represented in the splits (except for $Li_{1.35}Ni_{0.33}Mn_{0.67}O_{2.35}$ which has 1 cell each in the train and test sets). The feature matrix corresponding to the training set was standardized by subtracting each column mean from the column values and dividing the centred results by the respective standard deviation. The calculated

[1] https://pypi.org/project/battery-data-toolkit/.

Table 3
Description of the machine learning models considered in this study. Voltage and Current profiles are denoted by V and I respectively.

Model name	Model input	What is predicted
EOL model	V, I profiles of a single set of pulse tests carried out within the first 100 cycles; see Eq. (3) for the corresponding feature matrix	EOL
RUL model	V, I profiles of a single set of pulse tests carried out *at any cycle n before the EOL*; see Eq. (4) for the corresponding feature matrix	RUL = EOL − n
Classification model	V, I profiles of a single set of pulse tests carried out *at any cycle n*; see Eq. (4) for the corresponding feature matrix	Passed EOL (0); not passed EOL (1)

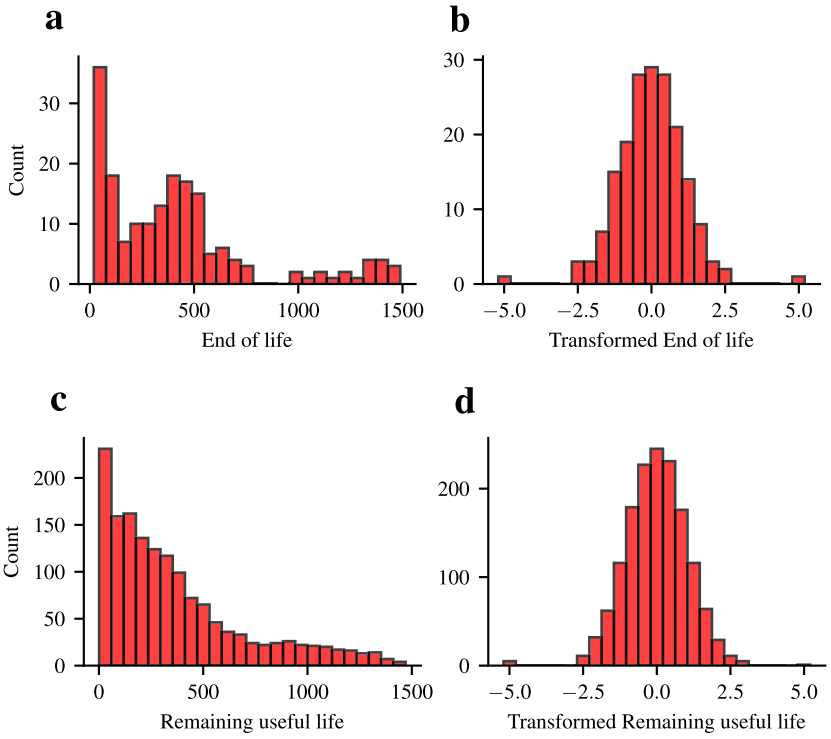


Fig. 4. (**a**) EOL distribution and (**b**) its transformation via quantile transform. (**c**) RUL distribution and (**d**) its quantile transformation. Data are shown for all the cells in the training set. The transformation ensures that skewness has a minimal effect on the generalization error of the machine learning model.

mean and standard deviation (on the training features) were used to scale the feature matrices corresponding to the test set. In the case of EOL and RUL models, targets were transformed using the Quantile Transformation (details of this algorithm are presented in the Supplementary Material). These were considered to mitigate the effect of target skewness on the models; see Fig. 4(**a-d**) for the transformation results.

Hyperparameter values for each of the models were chosen via cross-validated grid search implemented via `GridSearchCV` class of the scikit-learn. This involves the creation of a parameter space for the selected hyperparameters, fitting an XGBoost model using each of the combinations on the cross-validation training set, and evaluating the resulting fitted model on the validation set using an error metric. The combination of parameter values that produce the lowest error was kept for fitting the final model. Hyperparameters were categorized into two: data and model parameters. Data parameter controls the level or depth k of signature used for feature matrix creation; the higher it is, the more information extracted and the more complex the resulting model. On the other hand, model parameters are XGBoost parameters that were tuned to achieve robust and high-accuracy models. A list of such parameters considered for tuning includes the number of estimators or boosted trees, learning rate, maximum tree depth, and regularization parameter. Full definitions of these parameters and how they affect model performance can be found in the XGBoost documentation.[2] Evaluation metrics for selecting the best set of hyperparameters are, see Eq. (6), the mean absolute error (MAE) for the EOL and RUL models and the F_1-score for the classification model.

Model generalization error was estimated using the test set. For the EOL and RUL models, MAE and root mean squared error (RMSE) were used for the estimation. Precision, recall, F_1-score, the area under the receiver operating characteristic curve (AUC ROC), and accuracy were employed in the case of the classification model. The mathematical definitions of these metrics are provided in Eq. (6). In addition, confidence intervals were constructed for each metric using the notion of bootstrap pivotal confidence intervals [47,48]; see the Supplementary Material for its mathematical description. Furthermore, cross-validation via ordinary and stratified k-fold was also explored to further test the accuracy and robustness of the various models proposed in this study.

Feature importance analysis via impurity-based technique [45] and permutation approach [49] was carried out to show the impact of the signature components on the performance of the proposed models. Details of these feature importance algorithms are provided in the Supplementary Material.

[2] https://xgboost.readthedocs.io/en/stable/parameter.html.

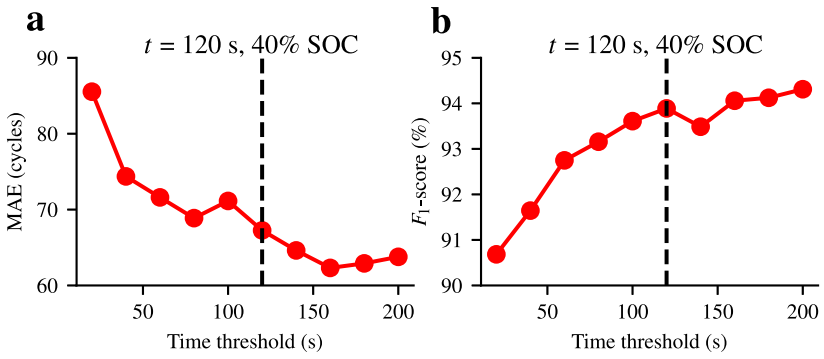


Fig. 5. Cross-validation results from (**a**) RUL and (**b**) classification models obtained under different time thresholds of the HPPC test. At $t = 120$ s (40% SOC for a fully charged cell at the start of an HPPC test), there is a less significant increase in model accuracy and thus this time mark is adopted for any further analysis on fully charged cells.

3.5. Choosing the time threshold for HPPC data extraction

As described in Section 2.2, each complete discharge and charge profile of the HPPC took approximately 20 s. In this study, we aim to use as little cross-sectional data as possible. To accomplish this, RUL and the classification models were cross-validated on the training data using different time thresholds for feature extraction. In particular, $t \in \{20i : i = 1, 2, \ldots, 10\}$ were considered, where t is in seconds and the largest time was chosen such that it covers all the HPPC profile across cells; note that each i corresponds to an SOC state (e.g., $i = 1$ corresponds to 90% SOC, $i = 5$ is 50% SOC). Since EOL is a particular case of RUL (by using the definition RUL = EOL – n, and choosing n to be 1; i.e., the first cycle), the optimal threshold for the RUL model was chosen for the EOL model.

The results of this experiment are presented in Fig. 5. As expected of the proposed models, an increase in t approximately corresponds to an improved metric value (a decrease in the MAE for the RUL model and an increase in F_1-score for the case of the classification model). This trend results from the fact that more information is available for the model to train and make accurate predictions as more data is extracted from the HPPC test. Interestingly, the improvement looks rather subtle after the $t = 120$ s mark (which corresponds to a 40% SOC for a cell with 100% SOC at the beginning of an HPPC test). Thus this time threshold was adopted for any further experimentation in this research.

4. Results and discussion

4.1. Remaining useful life and end of life prediction models

The summary of the performance of both the EOL and RUL models is provided in Table 4. EOL was predicted with an MAE and RMSE of approximately 85 and 148 cycles respectively on the test set. As for RUL, the generalization error estimated on the test set yielded an MAE and RMSE of approximately 91 and 134 cycles respectively. On the 95% confidence intervals obtained from the bootstrap pivotal confidence interval, it was discovered that the resulting length of each constructed interval is moderately short, indicating confidence in the accuracy of each metric value. Another noticeable trend is the difference between the accuracy of the EOL and RUL models on both the train and test sets. The MAE and RMSE on EOL are generally lower (except for the RMSE values on the test set) in comparison to that of RUL. This is because the target distributions are different; see Fig. 4(**a-d**). Target values for the RUL model are obtained at different stages of the life cycle of each cell (in contrast to that of EOL which is restricted to the first 100 cycles) and thus introduce variability to the distribution (as a result of variability in the chemical reactions in the cell).

Table 5 provides the summary of the cross-validation results obtained from the application of a k-fold algorithm on the training set. For each model, a 5-fold cross-validation was considered. It was discovered that the standard deviation of MAE and RMSE across folds were approximately 19 and 44 cycles respectively for the case of EOL; and approximately 3 and 9 cycles respectively for the case of RUL. These low cross-validated standard deviations reflect a well-generalized, stable, and reliable model. Fig. 6 shows the parity plots of the predictions from both the EOL and RUL models on the test set. It can be seen that each model approximately has the same performance in predicting low and extreme cycle numbers. In addition, the embedded histogram of residuals suggests that the prediction errors are not skewed (a balance between over-prediction and under-prediction). Most of the residuals fall in the bin containing zero, indicating a reduction in the deviation of predicted values from the true values.

Table 4
Performance metrics of the individual models that predict the EOL and RUL. The 95% confidence interval (CI) is obtained via bootstrap pivotal confidence intervals.

		MAE (cycles)		RMSE (cycles)	
		Value	CI	Value	CI
EOL	Train	1.98	[1.49, 2.43]	3.82	[2.83, 4.86]
	Test	85.16	[57.31, 108.89]	147.98	[97.14, 202.36]
RUL	Train	6.69	[6.25, 7.12]	10.80	[10.00, 11.58]
	Test	91.47	[83.69, 98.87]	133.75	[121.28, 146.13]

Table 5
Cross-validation results of the proposed EOL and RUL models. The mean ($\bar{x}$) and standard deviation (σ) of the performance metrics were calculated for each model cross-validated on the training set.

	MAE (cycles)		RMSE (cycles)	
	$\bar{x}$	σ	$\bar{x}$	σ
EOL	110.54	19.15	168.16	44.05
RUL	68.50	2.75	101.92	9.34

4.2. Classification model for cell characterization

The summary of the performance of the classification model is given in Table 6 and Fig. 7. From these results, it was discovered that there was a balance between the precision and recall scores (92.94% and 93.08% respectively on the test set) indicating that the model performed well in terms of minimizing false positives and false negatives. This is also evident in the F_1-score (93.01% on the test set) and also suggests that the model is robust and consistent in its predictions as no preference is given to one class over the other. These are all desirable qualities because there are costs associated with wrongly predicting a cell has passed its EOL and vice versa. The former would result in underutilizing the corresponding cell as it will be channelled to other low-power-consuming use cases (such as lighting) or recycled, while the latter would lead to a risk of power failure.

The high AUC ROC score (with a generalization score of 98.36%) implies a good discrimination ability of the model in differentiating

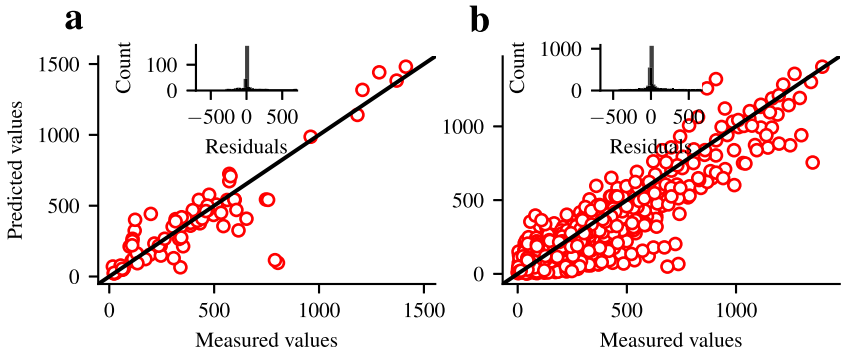


Fig. 6. Parity plots alongside the histograms of residuals. These compare the predicted to the measured values for the case of (**a**) EOL and (**b**) RUL predictions on test data. Histograms portray centrality and no evidence of skew.

Table 6
Performance metrics of the classification model that predicts whether a cell has passed its EOL or not based on pulse test data. The 95% confidence interval (CI) was calculated using the concept of bootstrap pivotal CI.

	Train		Test	
	Value (%)	CI	Value (%)	CI
Precision	100	[100, 100]	92.94	[91.01, 95.03]
Recall	100	[100, 100]	93.08	[91.19, 95.12]
F_1-score	100	[100, 100]	93.01	[91.63, 94.5]
AUC ROC	100	[100, 100]	98.36	[97.85, 98.93]
Accuracy	100	[100, 100]	94.44	[93.37, 95.63]

Table 7
Cross-validation results of the proposed classification model that predicts whether a cell has passed its EOL based on a single pulse test. The mean ($\bar{x}$) and standard deviation (σ) of the performance metrics were calculated for each model cross-validated on the training set.

	$\bar{x}$ (%)	σ (%)
Precision	94.87	0.53
Recall	93.05	1.62
F_1-score	93.95	1.03
AUC ROC	98.65	0.36
Accuracy	94.63	0.87

between the positive and negative classes. This is also corroborated by the confusion matrix of Fig. 7(**a**) where the misclassification for "passed EOL" and "not passed EOL" are only 4.66% and 6.92% respectively. In addition, this high AUC ROC score also suggests that the model has a good true positive rate (also called sensitivity) while keeping a low false positive rate (also known as 1 – specificity). This also implies that the model is robust to threshold variations as the area under the ROC curve is obtained by evaluating the model performance across different probability thresholds for classification. All these are evident in the ROC curve of Fig. 7(**b**) where the proposed XGBoost model is compared to a classification model that produces true positives as much as false positives.

The results of the stratified 5-fold cross-validation are provided in Table 7. The stratification approach was considered to preserve the proportion of each class across folds. One of the important deductions from these results is that the mean of each metric is similar to the generalization scores obtained from the test set; see Table 6. This further shows the consistency of the model in predicting each class. Another noticeable trend is the small standard deviation of each metric. This implies model consistency, generalization, and reliability. In other words, model performance metrics did not vary significantly when different subsets of data were used for training and testing; and the model has a high chance of generalizing well on unseen data.

4.3. Feature importance analysis

In this study, impurity-based [45] and permutation importance [49] are employed for feature analysis. The former shows the significance of each signature component in building trees for prediction while the latter illustrates how the model performance would change if there is an alteration (through permutation) in the feature-to-target bond of the fitted model. Because signature depth was part of the hyperparameters of the proposed model pipelines, the optimal values (obtained via hyperparameter tuning) were 3, 6, and 6 for the EOL, RUL, and classification models respectively. The optimal depth for the EOL model is different from the rest of the models because of the difference in the dimension of their corresponding feature matrix; it was shown in Eqs. (3) and (4) that F^{EOL} has a smaller dimension in comparison to $F^{\text{RUL-CM}}$, and thus a lengthier signature basis would be required to capture the underlying function that links the features to their corresponding targets. Using the shuffle product property of the signature [29], a non-linear function of signatures can be written as a linear combination of iterated integrals (see Section 3.1 for its definition); this is akin to basis function expansion and the more terms (or higher depth), the more representative is the underlying function—striking a balance between efficacy and computational costs is also needed.

The histograms of Fig. 8(**a-f**) show the first 10 most important features according to the two algorithms considered. Importance values were scaled to be in an interval $[0, 1]$ for ease of interpretation (with 0 and 1 as the lowest and largest importance values respectively). For both algorithms, the signature components involving all the path functions (namely $S^{2,3,2}$ and $S^{3,3,1}$) were discovered to be the most important for the EOL model. These features correspond to the definitions

$$\begin{cases} S^{2,3,2} = \int_{t_0}^{t_f} \left[\int_{t_0}^{t_3} \left[\int_{t_0}^{t_2} I'(t_1)\, dt_1 \right] V'(t_2)\, dt_2 \right] I'(t_3)\, dt_3; \\ S^{3,3,1} = \int_{t_0}^{t_f} \left[\int_{t_0}^{t_3} \left[\int_{t_0}^{t_2} V'(t_1)\, dt_1 \right] V'(t_2)\, dt_2 \right] dt_3; \end{cases}$$

where t_0 and t_f are the initial and final pulse times respectively. These definitions suggest that the signature components capture the segmental effect of both the pulse current and the voltage response profile. The rationale behind their importance can be linked to the fact that the battery voltage under the HPPC test gives a good indication of its strength to withstand various real use cases, and the magnitude of the pulse current determines the geometry of the current–voltage path. As

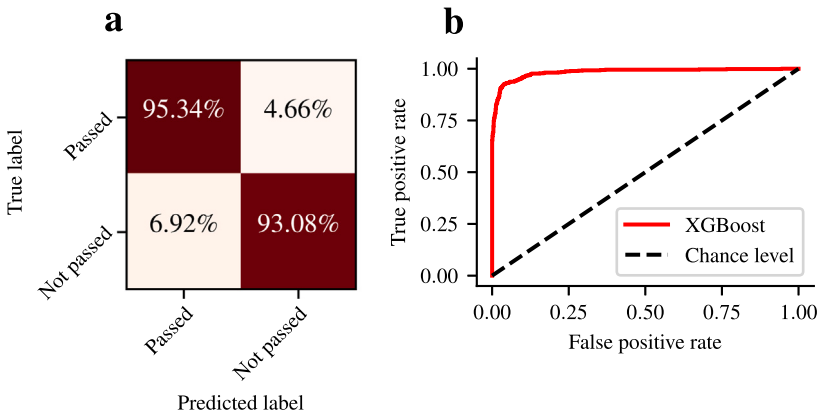


Fig. 7. Performance of the classification model. The confusion matrix and ROC curve are provided in (**a**) and (**b**) respectively. The confusion matrix indicates there is a balance of performance in correctly labelling each class and the ROC suggests that the proposed classification model is far from a random guess.

for the model that predicts the RUL, the signature component $S^{1,3,3,2,1,3}$ and $S^{3,1,1,3,2,2}$ were selected by the impurity-based and permutation importance algorithms respectively. These correspond to the definitions

$$\begin{cases} S^{1,3,3,2,1,3} = \int_{t_0}^{t_f}\left[\int_{t_0}^{t_6}\left[\int_{t_0}^{t_5}\left[\int_{t_0}^{t_4}\left[\int_{t_0}^{t_3}\left[\int_{t_0}^{t_2} dt_1\right] V'(t_2)\,dt_2\right] \times V'(t_3)\,dt_3\right] I'(t_4)\,dt_4\right] dt_5\right] V'(t_6)dt_6. \\ S^{3,1,1,3,2,2} = \int_{t_0}^{t_f}\left[\int_{t_0}^{t_6}\left[\int_{t_0}^{t_5}\left[\int_{t_0}^{t_4}\left[\int_{t_0}^{t_3}\left[\int_{t_0}^{t_2} V'(t_1)\,dt_1\right] dt_2\right] dt_3\right] \times V'(t_4)\,dt_4\right] I'(t_5)dt_5\right] I'(t_6)\,dt_6. \end{cases}$$

As for $S^{1,3,3,2,1,3}$ and $S^{3,1,1,3,2,2}$, they do not only show the importance of voltage and current load but also the HPPC time. The duration of the pulses plays a significant role in the extent of information derived from the HPPC test. The longer the test, the more information would be available for model consumption. The intensity of the applied pulse current determines the overall geometry of both the voltage and current profiles, where one of the characteristics of signatures is their sensitivity to the geometric shape of a path [29]. With regards to the classification model, both feature importance techniques indicated that $S^{1,3,1,3,3,3}$ is the most important signature component which is exactly defined as

$$S^{1,3,1,3,3,3} = \int_{t_0}^{t_f}\left[\int_{t_0}^{t_6}\left[\int_{t_0}^{t_5}\left[\int_{t_0}^{t_4}\left[\int_{t_0}^{t_3}\left[\int_{t_0}^{t_2} dt_1\right] V'(t_2)\,dt_2\right] dt_3\right] \times V'(t_4)\,dt_4\right] V'(t_5)dt_5\right] V'(t_6)\,dt_6,$$

which can also be explained in the light of $S^{3,3,1}$ above.

The results of the effect of the active HPPC time threshold on the feature importance are displayed in Fig. 8(**g-i**). For brevity, only the impurity-based algorithm was considered. This experiment is motivated by knowing how feature relevance (to model) changes when different data volumes (controlled by active HPPC time in seconds) are used for model building. For this purpose, the Jaccard similarity (which measures the ratio of the cardinality of the intersection of two sets to that of their union) was adopted. Overall, there is a significant feature similarity across time thresholds for the EOL model whereas similarity scores are not pronounced in both the RUL and classification models. The EOL model uses data restricted to within the first 100 cycles, so it is expected to have similar features under different time thresholds. On the other hand, the RUL and classification models draw features from several cycle numbers (with different battery chemical changes); thus features are expected to have dissimilarity under different HPPC times.

4.4. Model performance under different SOC ranges

In practice, a user might not have a fully charged battery at 100% SOC and still want to make predictions with the current level of SOC. To address this and further investigate the robustness of the proposed models, data corresponding to SOC ranges of 20% and 10% differences were extracted from the whole HPPC test. Concretely, the current–voltage profiles of three different SOC ranges (100%–80%, 80%–60%, and 60%–40% in the case of 20% difference) and five ranges (100%–90%, 90%–80%, 80%–70%, 70%–60%, and 60%–50% for the 10% difference) were extracted. Each range corresponds to two blocks of the HPPC test for the 20% difference and one block for the 10%; see Fig. 1 for a sample visualization of a block. Using this data, the models described in Table 3 were trained and cross-validated.

Tables 8 and 9 show the 5-fold cross-validation results of the resulting models. The performance metrics obtained under each SOC range are competitive (however marginally worse) when compared to those of Tables 5 and 7 (for an SOC range of 100%–40%). This shows that data obtained over SOC ranges with windows of 20% and 10% are capable of building an effective model for making inferences on new data. Between Tables 8 and 9, the results in the former regarding EOL and RUL are slightly better than those in the latter. The differences at the level of the classification algorithm are marginal.

4.5. Comparison with the past literature

In this section, a comparison between the performance of the proposed EOL model and that of the literature is presented. To the best of our knowledge, the studies that used the data considered in this work for EOL prediction are those of [3,31]. In [3], two *feature-based machine learning models* namely Extratrees [50] and NuSVR [51] were proposed for the prediction of EOL from one cycle (first cycle) and the first 100 cycles of battery cycling data. A total of 396 features were extracted from the current, voltage, SOH, charge time, SOC, capacity, and IR. Overall, an MAE of 103 and 73 cycles on test data were recorded in their paper when predictions were made from one cycle and the first 100 cycles respectively. In addition, a leave-one-group-out validation was performed on the proposed models by leaving a cathode group out in turn for testing and using the rest for training models (excluding the cells that live more than 1500 cycles for cycle distribution consistency).

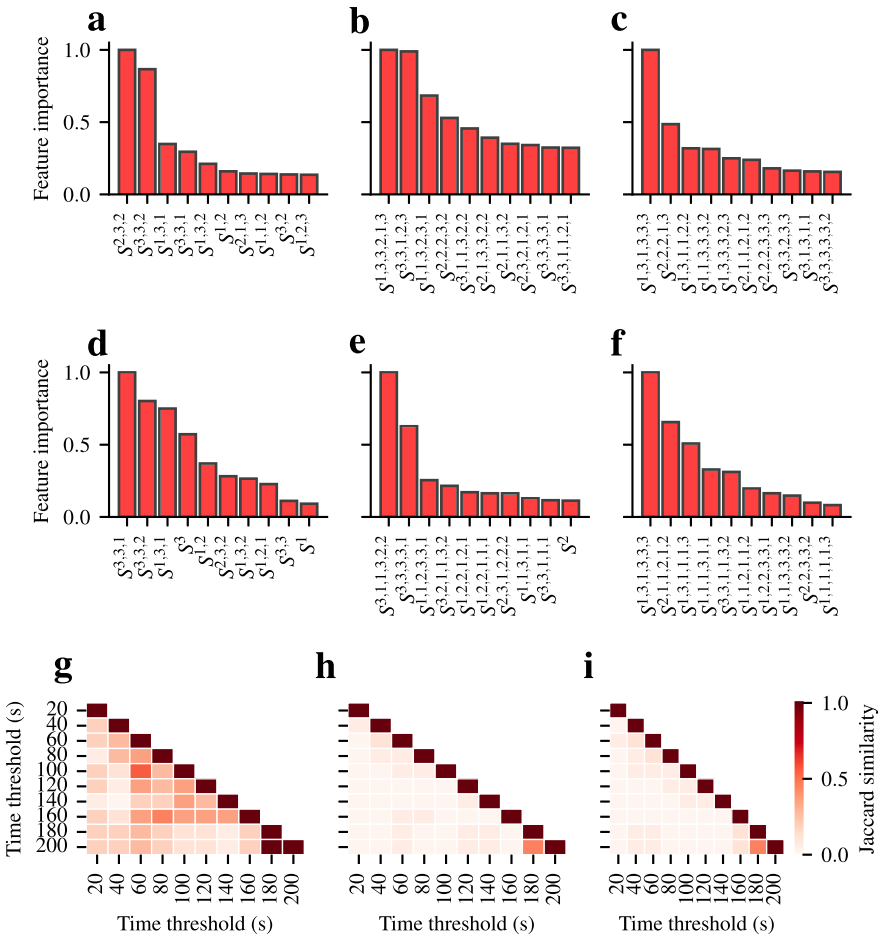


Fig. 8. XGBoost impurity-based feature importance analysis for (**a**) EOL, (**b**) RUL, and (**c**) classification models. Permutation feature importance analysis carried out on the training set for (**d**) EOL, (**e**) RUL, and (**f**) classification models. In the case of impurity-based, the importance value depicts how frequently the corresponding feature was used in splitting tree nodes during model training; whereas in the case of permutation, the importance value shows the degree of the effect of the corresponding feature on the fitted model when its values are shuffled. The importance values are scaled to be in the interval $[0, 1]$ for ease of interpretation. Jaccard similarity of top 10 most important features selected under different time thresholds of the HPPC test for (**g**) EOL, (**h**) RUL, and (**i**) classification models. For brevity, the impurity-based feature importance algorithm was considered for this analysis. Similarity scores under RUL and classification models are similar and both are distinguished from that of EOL.

Table 8
Cross-validation results from EOL, RUL, and classification models obtained under different time ranges of the HPPC test. The window considered here ensures *that there are two complete discharge and charge regimes* in each range. This experiment is motivated by using data under different SOCs, which applies to situations where non-fully charged cells are supplied for making predictions.

	Time range (s)	SOC range (%)	MAE (cycles)	
			$\bar{x}$	σ
EOL	0–40	100–80	118.75	20.88
	40–80	80–60	117.09	25.87
	80–120	60–40	124.72	25.15
RUL	0–40	100–80	75.37	6.41
	40–80	80–60	79.10	3.16
	80–120	60–40	72.01	1.16
			F_1-score (%)	
			$\bar{x}$	σ
Classification	0–40	100–80	91.25	1.37
	40–80	80–60	92.34	1.32
	80–120	60–40	92.68	0.73

Table 9
Cross-validation results from EOL, RUL, and classification models obtained under different time ranges of the HPPC test. The window considered here ensures *that there is one complete discharge and charge regime* in each range. This experiment is motivated by using data under different SOCs, which applies to situations where non-fully charged cells are supplied for making predictions.

	Time range (s)	SOC range (%)	MAE (cycles)	
			$\bar{x}$	σ
EOL	0–20	100–90	132.86	21.05
	20–40	90–80	127.21	17.23
	40–60	80–70	145.56	25.18
	60–80	70–60	126.38	16.22
	80–100	60–50	140.92	46.94
RUL	0–20	100–90	85.54	5.89
	20–40	90–80	86.85	5.80
	40–60	80–70	85.95	2.93
	60–80	70–60	90.34	2.76
	80–100	60–50	83.89	2.54
			F_1-score (%)	
			$\bar{x}$	σ
Classification	0–20	100–90	90.68	1.60
	20–40	90–80	89.64	0.65
	40–60	80–70	90.27	1.08
	60–80	70–60	90.65	0.88
	80–100	60–50	92.18	0.98

Their NuSVR model generally outperforms the Extratrees algorithm with the best MAE of 91 cycles when tested on 5Vspinel. In [31], a *deep learning* approach called multi-variate *transformer architecture* was adopted for the prediction of EOL using a single cycle of quantities including discharge capacity, discharge energy, coulombic efficiency,

Table 10
Comparison of the performance of our proposed model in predicting EOL. For comparison purposes, feature-based models built on the cycling data belonging to [3,31] were considered. In places where numbers are marked with a dagger (†) and an asterisk (∗), it means the metric is obtained from the Extratrees and NuSVR of [3] respectively. Meaning of symbols: measured voltage at time t (V), current (I), capacity (Q), discharge energy (E), coulombic efficiency (C_{eff}), energy efficiency (E_{eff}), internal resistance (IR), state of health (SOH), state of charge (SOC), and charge time (t_c).

Papers	Data used from [3]	Data regime	Cycles used	Test MAE (cycles)
This work	V, I	Single HPPC test	1	85.16
[3]	I, V, SOH, t_c, SOC, Q, IR	Regular cycling	1 100	$103^{\dagger}$, 114^{*} $78^{\dagger}$
[31]	Q, E, C_{eff}, E_{eff}, IR	Regular cycling	1	231

Table 11
Comparison of the performance of our proposed model in predicting EOL. The model was trained on all but one cathode chemistry and then tested on the left-out cathode group. In places where numbers are marked with a dagger (†) and an asterisk (∗), it means the metric is obtained from the Extratrees and NuSVR of [3] respectively. Meaning of symbols: measured voltage at time t (V), current (I), capacity (Q), internal resistance (IR), state of health (SOH), state of charge (SOC), and charge time (t_c).

Cathode group	MAE (cycles)	
	This work	[3, Table 2a, 2b]
NMC111	175.04	$507^{\dagger}$, 287^{*}
NMC532	297.61	$221^{\dagger}$, 152^{*}
NMC622	199.66	$156^{\dagger}$, 117^{*}
NMC811	129.44	$98^{\dagger}$, 104^{*}
HE5050	249.84	$181^{\dagger}$, 276^{*}
5Vspinel	283.80	$318^{\dagger}$, 91^{*}
Data used	V, I	I, V, SOH, t_c, SOC, Q, IR
Data regime	Single HPPC test	100-cycle data (regular cycling)

energy efficiency, and IR drop. The recorded prediction accuracy of their proposed model on the test data is an MAE of 231 cycles.

In this study, a different and novel approach was adopted for the prediction of EOL. The current and voltage profiles of a single HPPC test carried out within the first 100 cycles were utilized for feature extraction via path signature. An XGBoost model was trained on the extracted feature matrix and an MAE of approximately 85 cycles was achieved on the test set. This model attained a competitive accuracy when compared to the one-cycle and 100-cycle modes of [3,31] and less data (in terms of measured profiles and data regime) is needed for model input; see Table 10 for more detail. In the leave-one-group-out experiment (cells that lived more than 950 cycles were dropped for distribution consistency; see Fig. 2(**b**)), our proposed models have a competitive accuracy and even out-performed that of [3] in the case of NMC111 (for NuSVR and Extratrees), HE5050 (for Extratrees) and 5Vspinel (for Extratrees); see Table 11 for a detailed comparison.

5. Conclusion

The characterization of cells and the prediction of EOL and RUL using current and voltage profiles of a single active HPPC test have been investigated. The proposed models achieved a performance on par with existing literature while strictly ignoring the cell's general historical usage, taking much less data as input and work consistently across datasets with varying cathode chemistry. Additionally, each proposed model has been validated in situations where cells are at different levels of SOC and it is shown that data covering SOC ranges with a window of 20% suffices for effective prediction. This allows for making inferences even if collected cells are not fully charged which is valuable for manufacturers and end users of batteries who seek a quick, explainable, and easy-to-use model for cell characterization and prognostics.

The models developed are transparent, using XGBoost and path signatures for explainability and interpretability. All model components are traceable for easy practical maintenance and the data processing and model implementation are publicly available.

A potential future study is the investigation of the magnitude/frequency of pulse loads as they uniquely determine the geometry of the corresponding profiles (and the extracted signatures), thus, conditional on relevant data being available, the proposed modelling procedure should be tested on different types of pulse tests. This question aligns with the lack of standardization of pulse tests, even though there are proposed standards such as the HPPC. Nonetheless and unequivocally, the HPPC data of the study contains sufficient information that, once combined with the expressive power of path signatures, allows for competitive predictive lifetime models. While the methods presented in this research cannot be used as a quick tool to assess the battery in minutes (such as real-time prediction while an electric vehicle is in use), multiple pulse tests like the HPPC offer measurements in real-life environments. The technique takes at least a couple of hours which is still fast compared to other methods requiring data loggers for days or even weeks. Future research should focus on finding a good trade-off between the accuracy of the prediction and the time it takes to obtain such predictions.

Methods

Savitzky–golay filter

Savitzky–Golay filter is one of the filtering techniques used in digital signal processing (DSP) to remove noise and smoothing signals or data streams. It makes use of the least-square smoothing where a polynomial p of degree N is repeatedly fitted to sub-samples (each of size w called the window size) of the original data obtained by shifting the window to every point in the data; i.e, for every sub-sample $\boldsymbol{x}^s \in \mathbb{R}^w$ of a signal or data vector $\boldsymbol{x} \in \mathbb{R}^n$, a symmetric set $t \in \{-M, -M+1, \dots, M-1, M\}$, $M = (w-1)/2$ is generated and a polynomial p of degree N is fitted by minimizing the unconstrained optimization problem:

$$\sum_{t=-M}^{M} \left[p(t) - \boldsymbol{x}^s(t)\right]^2; \quad p(t) = \sum_{k=0}^{N} \alpha_k t^k; \quad \alpha \in \mathbb{R}, \tag{5}$$

where by $\boldsymbol{x}^s(t)$ we mean the component of $\boldsymbol{x}^s$ that coincides with the discretized value t. For every fit, the filter will replace the median of the corresponding sub-sample with the best fit value (i.e., the fitted polynomial evaluated at the central point or median of the fitted interval). Because of the symmetricity of the interval about its median, this is equivalent to evaluating $p(t)$ at $t = 0$ and this is the same as

$$p(t = 0) = \alpha_0.$$

Thus, only the constant of the polynomial is needed to be solved. For instance, if $w = 5, N = 2$, we have

$$p(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2; \quad M = (5-1)/2 = 2; \quad t \in \{-2, -1, 0, 1, 2\}.$$

Solving the Least-Square Problem (5) to obtain the first coefficient α_0,

$$\alpha_0 = \frac{1}{35}\left[-3\boldsymbol{x}^s(-2) + 12\boldsymbol{x}^s(-1) + 17\boldsymbol{x}^s(0) + 12\boldsymbol{x}^s(1) - 3\boldsymbol{x}^s(2)\right],$$

which is the resulting digital filter for every sub-sample $\boldsymbol{x}^s$.

As in the case of the median filter, the boundaries of the signal need extra treatment: one way is to pad the boundaries with some constant,

say zero, and another approach (which was adopted in this research) is to evaluate a degree N polynomial, which is fitted to the last w values of the edges, on the last $\lfloor w/2 \rfloor$ output values. The SciPy [52] implementation was used in this work via the function `savgol_filter` in the `signal` module. Following this implementation, if w is even, then the evaluation set is given by $t \in \{-(w/2)-0.5, -(w/2)+0.5, \ldots, (w/2)-1.5, (w/2)-0.5\}$. In this case, the central point is the median which is also zero.

Model performance metrics

For the regression problems (EOL and RUL predictions), mean absolute error (MAE) and root mean squared error (RMSE) were used. The classification model that predicts whether a cell has passed its EOL was evaluated using precision, recall, F_1-score, the area under the receiver operating characteristic curve (AUC ROC), and accuracy. Each of these metrics is defined below:

$$\begin{cases} \text{MAE}(y,\hat{y}) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}|y_i-\hat{y}_i|, \\ \text{RMSE}(y,\hat{y}) = \sqrt{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}, \\ \text{precision} = \dfrac{TP}{TP+FP}, \\ \text{recall} = \dfrac{TP}{TP+FN}, \\ F_1\text{-score} = 2\times\dfrac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}, \\ \text{AUC ROC} = \text{area under ROC curve} \\ \text{accuracy} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\mathbb{1}(y_i-\hat{y}_i) \end{cases} \tag{6}$$

where y_i and $\hat{y}_i$ are the actual and predicted values for sample i respectively; TP, FP, and FN are the number of true positives, false positives, and false negatives respectively; n is the number of samples.

CRediT authorship contribution statement

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

Code availability

The data cleaning and modelling codes for this research can be found in this GitHub repository: https://github.com/Rasheed19/pulse-project.

Funding

This project was funded by an industry-academia collaborative grant *EPSRC EP/R511687/1* awarded by *EPSRC*, UK & *the University of Edinburgh*, UK 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).

P. Dechent is supported by the Deutsche Forschungsgemeinschaft (DFG), Germany (project number 511349305).

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

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.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.apenergy.2024.124820.

Data availability

The datasets of this research are the 300 refined Lithium-ion cells used in Paulson et al. (2022) [3] and made available in Ward et al. (2023) [32].

References

[1] Sripad S, Viswanathan V. Performance metrics required of next-generation batteries to make a practical electric semi truck. ACS Energy Lett 2017;2(7):1669–73. http://dx.doi.org/10.1021/acsenergylett.7b00432.

[2] Fredericks WL, Sripad S, Bower GC, Viswanathan V. Performance metrics required of next-generation batteries to electrify vertical takeoff and landing (vtol) aircraft. ACS Energy Lett 2018;3(12):2989–94. http://dx.doi.org/10.1021/acsenergylett.8b02195.

[3] 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.

[4] Zhou KQ, Qin Y, Yuen C. Transfer-learning-based state-of-health estimation for lithium-ion battery with cycle synchronization. IEEE/ASME Trans Mechatronics 2023;28(2):692–702. http://dx.doi.org/10.1109/tmech.2022.3201010.

[5] Wang Q, Ye M, Cai X, Sauer DU, Li W. Transferable data-driven capacity estimation for lithium-ion batteries with deep learning: A case study from laboratory to field applications. Appl Energy 2023;350:121747. http://dx.doi.org/10.1016/j.apenergy.2023.121747.

[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] Li AG, West AC, Preindl M. Towards unified machine learning characterization of lithium-ion battery degradation across multiple levels: A critical review. Appl Energy 2022;316:119030. http://dx.doi.org/10.1016/j.apenergy.2022.119030.

[9] 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.

[10] Ji S, Zhu J, Yang Y, Dos Reis G, Zhang Z. Data-driven battery characterization and prognosis: Recent progress, challenges, and prospects. Small Methods 2024;2301021.

[11] dos Reis G, Strange C, Yadav M, Li S. Lithium-ion battery data and where to find it. Energy AI 2021;5:100081. http://dx.doi.org/10.1016/j.egyai.2021.100081.

[12] Hasib SA, Islam S, Chakrabortty RK, Ryan MJ, Saha DK, Ahamed MH, Moyeen SI, Das SK, Ali MF, Islam MR, Tasneem Z, Badal FR. A comprehensive review of available battery datasets, rul prediction approaches, and advanced battery management. IEEE Access 2021;9:86166–93. http://dx.doi.org/10.1109/ACCESS.2021.3089032.

[13] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, Chen MH, Aykol M, Herring PK, Fraggedakis D, Bazant MZ, Harris SJ, Chueh WC, Braatz RD. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91. http://dx.doi.org/10.1038/s41560-019-0356-8.

[14] 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.

[15] 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.

[16] Li W, Sengupta N, Dechent P, Howey D, Annaswamy A, Sauer DU. One-shot battery degradation trajectory prediction with deep learning. J Power Sources 2021;506:230024. http://dx.doi.org/10.1016/j.jpowsour.2021.230024.
[17] 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. http://dx.doi.org/10.1016/j.jpowsour.2022.231736.
[18] 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. http://dx.doi.org/10.1016/j.egyai.2020.100006.
[19] You H, Zhu J, Wang X, Jiang B, Sun H, Liu X, Wei X, Han G, Ding S, Yu H, Li W, Sauer DU, Dai H. Nonlinear health evaluation for lithium-ion battery within full-lifespan. J Energy Chem 2022;72:333–41. http://dx.doi.org/10.1016/j.jechem.2022.04.013.
[20] Attia PM, Bills A, Brosa Planella F, Dechent P, dos Reis G, Dubarry M, Gasper P, Gilchrist R, Greenbank S, Howey D, Liu O, Khoo E, Preger Y, Soni A, Sripad S, Stefanopoulou AG, Sulzer V. Review—knees in lithium-ion battery aging trajectories. J Electrochem Soc 2022;169(6):060517. http://dx.doi.org/10.1149/1945-7111/ac6d13.
[21] Jones PK, Stimming U, Lee AA. Impedance-based forecasting of lithium-ion battery performance amid uneven usage. Nature Commun 2022;13(1). http://dx.doi.org/10.1038/s41467-022-32422-w.
[22] Zhang Y, Tang Q, Zhang Y, Wang J, Stimming U, Lee AA. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. Nature Commun 2020;11(1). http://dx.doi.org/10.1038/s41467-020-15235-7.
[23] Han H, Xu H, Yuan Z, Shen Y. A new soh prediction model for lithium-ion battery for electric vehicles. In: 2014 17th international conference on electrical machines and systems. ICEMS, IEEE; 2014, p. 997–1002. http://dx.doi.org/10.1109/icems.2014.7013631.
[24] Ran A, Cheng M, Chen S, Liang Z, Zhou Z, Zhou G, Kang F, Zhang X, Li B, Wei G. Fast remaining capacity estimation for lithium-ion batteries based on short-time pulse test and Gaussian process regression. Energy Environ Mater 2022;6(3). http://dx.doi.org/10.1002/eem2.12386.
[25] Meng J, Cai L, Luo G, Stroe D-I, Teodorescu R. Lithium-ion battery state of health estimation with short-term current pulse test and support vector machine. Microelectron Reliabil 2018;88–90:1216–20. http://dx.doi.org/10.1016/j.microrel.2018.07.025.
[26] Cai L, Meng J, Stroe D-I, Peng J, Luo G, Teodorescu R. Multiobjective optimization of data-driven model for lithium-ion battery soh estimation with short-term feature. IEEE Trans Power Electron 2020;35(11):11855–64. http://dx.doi.org/10.1109/tpel.2020.2987383.
[27] Lu D, Scott Trimboli M, Fan G, Zhang R, Plett GL. Nondestructive pulse testing to estimate a subset of physics-based-model parameter values for lithium-ion cells. J Electrochem Soc 2021;168(8):080533. http://dx.doi.org/10.1149/1945-7111/ac1cfa.
[28] Ran A, Liang Z, Chen S, Cheng M, Sun C, Ma F, Wang K, Li B, Zhou G, Zhang X, Kang F, Wei G. Fast clustering of retired lithium-ion batteries for secondary life with a two-step learning method. ACS Energy Lett 2022;7(11):3817–25. http://dx.doi.org/10.1021/acsenergylett.2c01898.
[29] Chevyrev I, Kormilitzin A. A primer on the signature method in machine learning. 2016, http://dx.doi.org/10.48550/ARXIV.1603.03788.
[30] Lyons TJ, Caruana M, Lévy T. Differential equations driven by rough paths. Lecture notes in mathematics, vol. 1908, Berlin: Springer; 2007, lectures from the 34th Summer School on Probability Theory held in Saint-Flour, July (2004) 6–24, With an introduction concerning the Summer School by Jean Picard.
[31] Paulson NH, Kubal J, Babinec SJ. Multivariate prognosis of battery advanced state of health via transformers. Cell Rep Phys Sci 2024;5(5). http://dx.doi.org/10.1016/j.xcrp.2024.101928.
[32] Ward L, Kubal J, Babinec SJ, Lu W, Dunlop A, Trask S, Polzin B, Jansen A, Paulson NH. Dataset of nmc battery tests from camp. 2023, http://dx.doi.org/10.18126/FDXQ-7YUL, 2023 release.
[33] Christophersen JP. Battery test manual for electric vehicles, revision 3. United States: Idaho National Lab. (INL); 2015, http://dx.doi.org/10.2172/1186745, available at https://www.osti.gov/biblio/1186745.
[34] Lewerenz M, Dechent P, Sauer DU. Investigation of capacity recovery during rest period at different states-of-charge after cycle life test for prismatic li(ni1/3mn1/3co1/3)o2-graphite cells. J Energy Storage 2019;21:680–90. http://dx.doi.org/10.1016/j.est.2019.01.004.
[35] Burrell R, Zulke A, Keil P, Hoster H. Communication—identifying and managing reversible capacity losses that falsify cycle ageing tests of lithium-ion cells. J Electrochem Soc 2020;167(13):130544. http://dx.doi.org/10.1149/1945-7111/abbce1.
[36] Savitzky A, Golay MJE. Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 1964;36(8):1627–39. http://dx.doi.org/10.1021/ac60214a047.
[37] Barlow RE, Brunk HD. The isotonic regression problem and its dual. J Amer Statist Assoc 1972;67(337):140–7. http://dx.doi.org/10.1080/01621459.1972.10481216.
[38] 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.
[39] Chen K-T. Integration of paths–a faithful representation of paths by noncommutative formal power series. Trans Amer Math Soc 1958;89(2):395. http://dx.doi.org/10.2307/1993193.
[40] 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.
[41] Kidger P, Bonnier P, Perez Arribas I, Salvi C, Lyons T. Deep signature transforms. Adv Neural Inf Process Syst 2019;32.
[42] 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.
[43] Kidger P, Lyons T. Signatory: differentiable computations of the signature and logsignature transforms, on both cpu and gpu. 2020, http://dx.doi.org/10.48550/ARXIV.2001.00706.
[44] Sangiri JB, Kulshreshtha T, Ghosh S, Maiti S, Chakraborty C. A novel methodology to estimate the state-of-health and remaining-useful-life of a li-ion battery using discrete fourier transformation. J Energy Storage 2022;46:103849. http://dx.doi.org/10.1016/j.est.2021.103849.
[45] 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. KDD '16, New York, NY, USA: ACM; 2016, p. 785–94. http://dx.doi.org/10.48550/ARXIV.1603.02754.
[46] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Mueller A, Nothman J, Louppe G, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in python. J Mach Learn Res 2012. http://dx.doi.org/10.48550/ARXIV.1201.0490.
[47] Young GA. Bootstrap: More than a stab in the dark? Statist Sci 1994;9(3). http://dx.doi.org/10.1214/ss/1177010383.
[48] Davison AC, Hinkley DV, Young GA. Recent developments in bootstrap methodology. Statist Sci 2003;18(2). http://dx.doi.org/10.1214/ss/1063994969.
[49] Breiman L. Random forests. Mach Learn 2001;45(1):5–32. http://dx.doi.org/10.1023/a:1010933404324.
[50] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63(1):3–42. http://dx.doi.org/10.1007/s10994-006-6226-1.
[51] Platt J, et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classif 1999;10(3):61–74.
[52] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, Vijaykumar A, Bardelli AP, Rothberg A, Hilboll A, Kloeckner A, Scopatz A, Lee A, Rokem A, Woods CN, Fulton C, Masson C, Haggstrom C, Fitzgerald C, Nicholson DA, Hagen DR, Pasechnik DV, Olivetti E, Martin E, Wieser E, Silva F, Lenders F, Wilhelm F, Young G, Price GA, Ingold G-L, Allen GE, Lee GR, Audren H, Probst I, Dietrich JP, Silterra J, Webber JT, Slavic J, Nothman J, Buchner J, Kulick J, Schoenberger JL, de Miranda Cardoso JV, Reimer J, Harrington J, Rodríguez JLC, Nunez-Iglesias J, Kuczynski J, Tritz K, Thoma M, Newville M, Kuemmerer M, Bolingbroke M, Tartre M, Pak M, Smith NJ, Nowaczyk N, Shebanov N, Pavlyk O, Brodtkorb PA, Lee P, McGibbon RT, Feldbauer R, Lewis S, Tygier S, Sievert S, Vigna S, Peterson S, More S, Pudlik T, Oshima T, Pingel TJ, Robitaille TP, Spura T, Jones TR, Cera T, Leslie T, Zito T, Krauss T, Upadhyay U, Halchenko YO, Vázquez-Baeza Y. Scipy 1.0: fundamental algorithms for scientific computing in python. Nat Methods 2020;17(3):261–72. http://dx.doi.org/10.1038/s41592-019-0686-2.