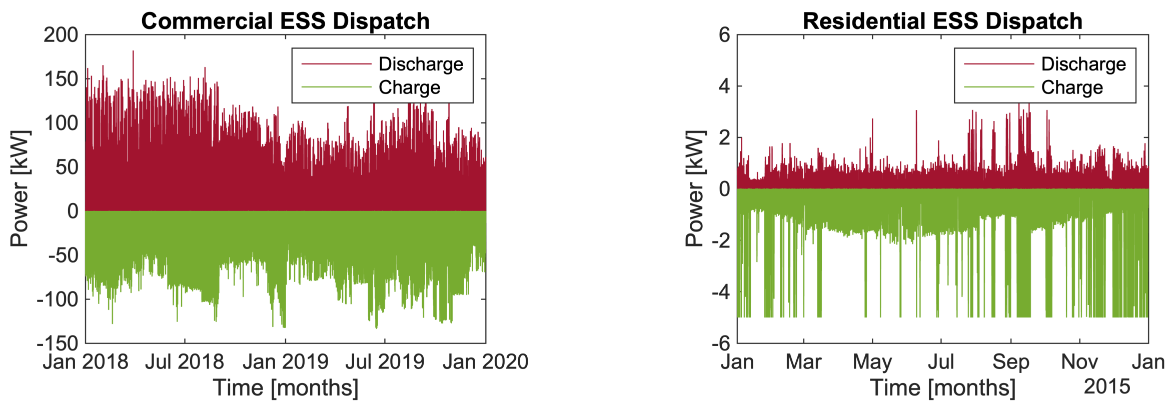
Note S1: Calculation of the equivalent grid service years

We examine two representative duty cycles for battery packs employed in commercial and residential energy storage systems (ESS), as illustrated in Figures S1(a) and S1(b), respectively [1]. The duty cycles show the ESS power dispatch in both charge and discharge for two years of commercial dispatch and one year of residential dispatch, respectively. For our analysis, these duty cycles represent the power demand that needs to be fulfilled by the SL batteries used in this work.



(a) (b)

Figure S1: **ESS power dispatch profiles in charge and discharge.** (a) Commercial ESS power dispatch profile from January 2018 to January 2020. (b) Residential ESS power dispatch profile from January 2015 to January 2016.

To calculate equivalent grid service years, we first calculate the current required by the ESS dispatch at the cell level as follows:

$$I_{cell,ESS,app}(t) = \frac{P_{app}(t)}{s_{app} \cdot p_{app} \cdot V_{cell,ESS}} \tag{1}$$

where $app \in \{comm, res\}$ denotes the commercial or residential dispatch, $P_{app}(t)$ is the total ESS power in kW at time t, s_{app} and p_{app} are the number of cells in series and parallel, respectively, and $V_{cell,ESS} = 3.63\,\mathrm{V}$ is the nominal cell voltage. Total power $P_{app}(t)$ is the cumulative sum of the total charge power and the total discharge power seen by the ESS. Then, the required Ah-throughput in the ESS is given by

$$Q_{Ah,ESS,app} = \int_{t_{0,app}}^{t_{f,app}} |I_{cell,ESS,app}(t)| dt \tag{2}$$

where $t_{0,app}$ and $t_{f,app}$ are the initial and final times of the ESS dispatch, and $Q_{Ah,ESS,app}$ represents the cumulative Ah-throughput demand. To further standardize this for our analysis, we calculate the equivalent full cycles for ESS EFC_{ESS} as follows:

$$EFC_{ESS} = \frac{Q_{Ah,ESS,app}}{2 \cdot Q_{cell,ESS}} \tag{3}$$

where $Q_{cell,ESS} = 4.85\,\mathrm{Ah}$ is the rated cell capacity used for ESS dispatch [1]. EFC_{ESS} is a dimensionless quantity and it represents the number of full cycles that a battery must go through to fulfill the ESS dispatch demand.

From Eq.(3), we get 152.25 cycles for commercial dispatch and 150 cycles for residential dispatch. It can be noted that if, instead of $Q_{cell,ESS}$, the capacity of a retired cell were to be used which would be lower than $Q_{cell,ESS}$, then EFC_{ESS} would increase indicating a higher demand from the ESS and more full cycles from the cell.

Similarly, based on the experimental campaign in this work, the equivalent full cycles for SL batteries, EFC_{SL}, is given by

$$EFC_{SL} = \frac{Q_{Ah,aging}}{2 \cdot Q_{initial,ch,C/20}} \tag{4}$$

where $Q_{Ah,aging}$ is the cumulative Ah-throughput, and $Q_{initial,ch,C/20}$ is the initial C/20 charge capacity for each cell in our dataset (see Table S2).

Finally, equivalent grid service years (EGY) represents the number of years that these SL batteries will last if they are used to provide the ESS dispatch shown in Figures S1(a) and S1(b). EGY is calculated by

$$EGY = \frac{EFC_{SL}}{EFC_{ESS}} \tag{5}$$

For cells in our dataset, $Q_{Ah,aging}$ ranges from 84,840 to 108,500 Ah which gives an equivalent EFC_{SL} from 2,095 to 2,728.8 cycles. Subsequently, by substituting these values into Eq.(5), we get EGY of approximately 14 to 18 years for both commercial and residential ESS dispatch. This analysis highlights that retired batteries that are operated in a narrow voltage range of 3 V to 4 V can theoretically be used for grid-storage applications for over a decade (up to two decades).

Note S2: Second-life battery aging campaign

The aging campaign, schematically illustrated in Fig. 1, starts with a set of RPTs to characterize the initial charge and discharge capacity from C/20 and C/40 tests, and internal resistance from HPPC tests. Afterwards, cells go through grid-like discharge/charge aging cycles followed by another set of RPTs. The number of RPTs and aging cycles for all cells are shown in Table S1.

Table S1: **Number of aging cycles for each cell at different Reference Performance Tests (RPTs)**

RPT #	Cell 1.1	Cell 1.2	Cell 1.3	Cell 1.4	Cell 2.1	Cell 2.2	Cell 2.3	Cell 2.4
1	0	0	0	0	0	0	0	0
2	223	318	212	318	207	208	202	318
3	694	820	698	824	636	677	651	835
4	1071	1221	1099	1225	1018	1058	1028	1236
5	1447	1594	1477	1602	1397	1438	1405	1637
6	1824	1969	1850	1979	1777	1819	1782	2014
7	2200	2344	2228	2358	2259	2432	2159	2389
8	2584	2720	2605	2733	2637	2808	2538	2763
9	2925	3094	2983	3252	2842	3186	2917	3462
10	-	3466	3359	3627	-	-	3294	3837
11	-	3841	-	4000	-	-	-	4212
12	-	4220	-	4319	-	-	-	4588

The C/20 test is performed with voltage derating between 3 V-4 V, and it consists of two CC-discharge/CCCV-charge cycles. The first cycle is used for pre-conditioning the cell while the second cycle is used to extract useful information e.g., charge/discharge capacity. Similarly, C/40 test is conducted at C/40 C-rate between a voltage range of 2.5 V and 4.2 V. HPPC test is also conducted with voltage derating between 3 V-4 V which means 100% SOC corresponds to 4 V and 0% SOC corresponds to 3 V for this test. A pair of pulses in discharge and charge are executed at every 10% SOC while discharging from 100% to 10% SOC and charging from 10% to 100% SOC. Before executing this protocol, a CC-discharge/CCCV-charge cycle is performed to ensure that the cell is at 4 V. All three RPTs are illustrated in Figure 1(a)-(c).

The temperature is measured during the experimental campaign via two thermocouples attached to the surface of each cell. The raw measured data is pre-processed to address three issues: outliers, noise, and missing data. For outliers, a min-max temperature limit of 15°C to 40°C is enforced to ensure real temperature data is not lost. Noise in the data is removed through a Savitzky-Golay filter which smoothens the temperature profile without loss of generality. Lastly, through sensor fusion (combining data from both sensors), temperature data is averaged when it is available from both sensors. If, for a certain time period, only one thermocouple measures the temperature, temperature data from a single thermocouple is used. Temperature data is not available for time periods where both thermocouples fail to record any data.

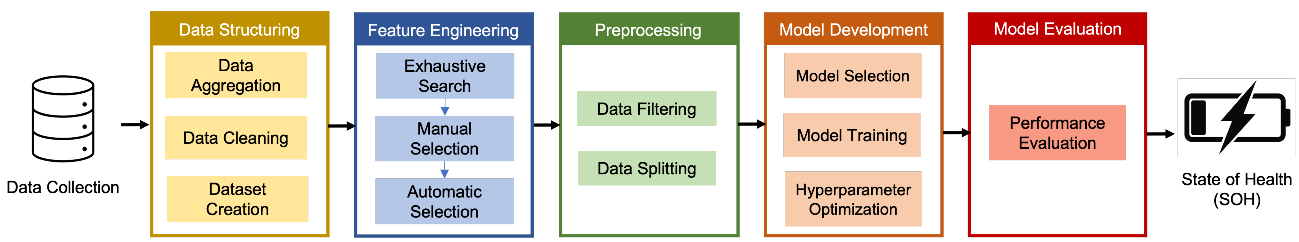


Figure S2: **Machine-learning model development pipeline.** Experimentally collected data is structured through data aggregation from different sources followed by data cleaning. Features are extracted and selected from structured data, and pre-processed to use in model development. Various machine learning models are trained and optimized, and their performance evaluated for SOH estimation to identify the best model.

Note S3: Machine learning pipeline

Data-driven model development for SOH estimation comprises of five stages: data structuring, feature engineering, data preprocessing, model development, and model evaluation, as depicted in Figure S2. Here, we elaborate on each part of the pipeline.

Data structuring

Data obtained from battery cyclers and thermocouples is stored in .xlsx and .csv format respectively. Since separate files exist for aging cycles, RPTs, and temperature measurements, raw temperature data is separated for each cell, and streamlined with the current and voltage data from aging cycles and RPTs. Through data structuring, the goal is to combine data from these different sources into a single, coherent framework that can effectively be used for downstream tasks e.g., SOH model development. As shown in Figure S3, custom MATLAB class HRAW is used to convert raw data into .mat format. Afterwards, HSUM class is used to pre-process the data, extract

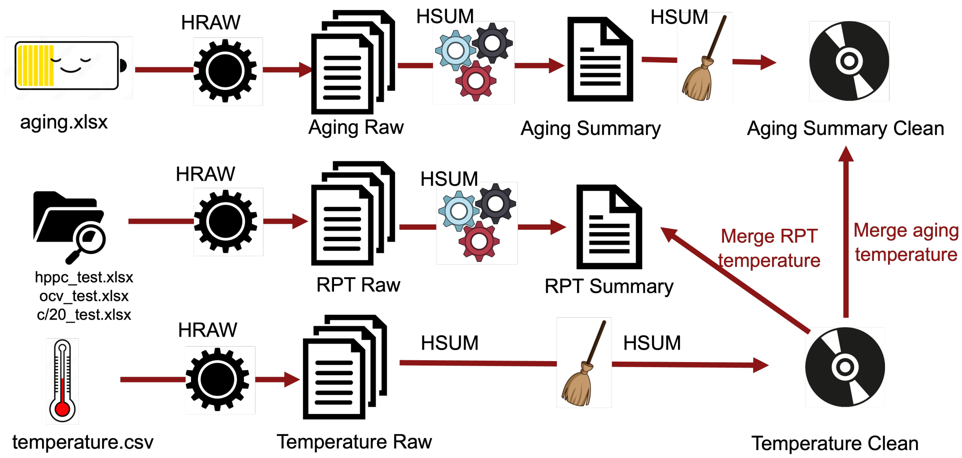


Figure S3: **Schematic of data structuring process to combine raw data for aging, RPTs, and temperature.** Raw data is first processed using HRAW class to convert it into .mat format. Processed raw data from HRAW class is further processed by HSUM class to create summary files consisting of all the features. Features are further processed to remove missing or irregular data through the HSUM class. Processed temperature data is combined with aging and RPT data in the end.

features, and combine aging and RPT temperature data. The 'summary' files contain the features extracted from the raw data and 'summary clean' files also contain the temperature data. By doing so, we minimize the need to access the original raw data repeatedly, and use the processed data for analysis and model development. Further details about other MATLAB classes and useful scripts are given in the code guide provided with the code.

Feature engineering

In this paper, the offline SOH estimation model is built upon features extracted from raw data as reported in Table S2. In total, we have 66 features from aging cycles, RPTs, and temperature data. Temperature features are based on average temperature during aging cycles and RPTs. T_{aging} is the average temperature per aging cycle, and the length of the vector is equal to the number of aging cycles from each cell. $T_{C/20}$, $T_{C/40}$, and T_{HPPC} are average temperatures during one respective RPT test, and their length is equal to the number of RPTs for each cell. Aging cycle features consist of Ah-throughput, $Q_{Ah,aging}$, and Energy-throughput, $E_{Wh,aging}$, extracted for both charge and discharge along with aging cycle energy efficiency $\eta_{E,aging}$ and aging cycle resistance $R0_{aging}$.

Ah-throughput and Energy-throughput features are also extracted from C/20 and C/40 tests. Features from C/20 tests also consist of the initial capacities $Q_{initial,ch,C/20}$ in charge and $Q_{initial,dis,C/20}$ in discharge. Incremental capacity curves are extracted from C/40 tests as shown in Figure S4. A 3^{rd}-order Savitzky-Golay filter is used for curve smoothing, and IC peak features dQ/dV_{peaks} are extracted. For all the cells, two peaks are observed between 3.7 V and 4.2 V; however, the number of IC curves per cell vary since some cells go through more rounds of RPTs than others. It is observed that Cell 2.4 and Cell 1.2 have the highest degradation since both have the smallest peak heights (peaks 1 and 2). Decrease in the height of peaks is associated with loss of active material (LAM) [2]; however, consistent with the increase in capacity for these cells, peak height also increases for most of the cells e.g., peak 2 of Cell 2.4. Maximum IC value of peak 2 $\max_{3.9 \leq V \leq 4.1} \frac{dQ_n}{dV_n}$, where n denotes the number of RPTs, is extracted as a feature from these curves.

HPPC features consist of pulse resistances extracted both during charge and discharge. Resistance is extracted

Table S2: **List of initial 66 features extracted from aging cycles, RPTs, and temperature data**

Temperature features	T_{aging}	Avg. Aging Cycle Temperature per cycle [°C]	$T_{C/20}$	Avg. C/20 Temperature per test [°C]
	T_{HPPC}	Avg. HPPC Temperature per test [°C]	$T_{C/40}$	Avg. C/40 Temperature per test [°C]
Aging cycle features	$Q_{ch,aging}$	Aging Cycle Charge Throughput [Ah]	$E_{ch,aging}$	Aging Cycle Charge Energy-throughput [Wh]
	$Q_{dis,aging}$	Aging Cycle Discharge Throughput [Ah]	$E_{dis,aging}$	Aging Cycle Discharge Energy-throughput [Wh]
	$\eta_{E,aging}$	Aging Cycle Energy Efficiency [−]	$R0_{aging}$	Aging Cycle Resistance [Ω]
	$Q_{Ah,aging}$	Aging Cycle Accumulated Ah-throughput [Ah]	$E_{Wh,aging}$	Aging Cycle Accumulated Energy-throughput [Wh]
C/20 test features	$Q_{ch,C/20}$	C/20 Charge Throughput [Ah]	$E_{ch,C/20}$	C/20 Charge Energy Throughput [Wh]
	$Q_{initial,ch,C/20}$	C/20 Initial Charge Capacity [Ah]	$Q_{initial,dis,C/20}$	C/20 Initial Discharge Capacity [Ah]
	$Q_{dis,C/20}$	C/20 Discharge Throughput [Ah]	$E_{dis,C/20}$	C/20 Discharge Energy Throughput [Wh]
	$Q_{Ah,C/20}$	C/20 Accumulated Ah-throughput [Ah]	$E_{Wh,C/20}$	C/20 Accumulated Energy-throughput [Wh]
C/40 test features	$Q_{ch,C/40}$	C/40 Charge Throughput [Ah]	$E_{ch,C/40}$	C/40 Charge Energy Throughput [Wh]
	$Q_{dis,C/40}$	C/40 Discharge Throughput [Ah]	$E_{dis,C/40}$	C/40 Discharge Energy Throughput [Wh]
	$Q_{Ah,C/40}$	C/40 Accumulated Ah-throughput [Ah]	$E_{Wh,C/40}$	C/40 Accumulated Energy-throughput [Wh]
	dQ/dV_{peaks}	C/40 Incremental Capacity (IC) Peaks [Ah/V]	dV/dV_{peaks}	C/40 Aging Derivative (AD) Peaks [−]
HPPC test features	$R0_{ch,ch,low,0s}$	Ch. Pulse in Ch. HPPC Resistance 0s CT (low SOC) [Ω]	$R0_{ch,ch,low,2s}$	Ch. Pulse in Ch. HPPC Resistance 2s CT (low SOC) [Ω]
	$R0_{ch,ch,low,3s}$	Ch. Pulse in Ch. HPPC Resistance 3s CT (low SOC) [Ω]	$R0_{ch,ch,high,0s}$	Ch. Pulse in Ch. HPPC Resistance 0s CT (high SOC) [Ω]
	$R0_{ch,ch,high,2s}$	Ch. Pulse in Ch. HPPC Resistance 2s CT (high SOC) [Ω]	$R0_{ch,ch,high,3s}$	Ch. Pulse in Ch. HPPC Resistance 3s CT (high SOC) [Ω]
	$R0_{ch,dis,low,0s}$	Dis. Pulse in Ch. HPPC Resistance 0s CT (low SOC) [Ω]	$R0_{ch,dis,low,2s}$	Dis. Pulse in Ch. HPPC Resistance 2s CT (low SOC) [Ω]
	$R0_{ch,dis,low,3s}$	Dis. Pulse in Ch. HPPC Resistance 3s CT (low SOC) [Ω]	$R0_{ch,dis,high,0s}$	Dis. Pulse in Ch. HPPC Resistance 0s CT (high SOC) [Ω]
	$R0_{ch,dis,high,2s}$	Dis. Pulse in Ch. HPPC Resistance 2s CT (high SOC) [Ω]	$R0_{ch,dis,high,3s}$	Dis. Pulse in Ch. HPPC Resistance 3s CT (high SOC) [Ω]
	$R0_{ch,SOC,low,0s}$	SOC Pulse in Ch. HPPC Resistance 0s CT (low SOC) [Ω]	$R0_{ch,SOC,low,2s}$	SOC Pulse in Ch. HPPC Resistance 2s CT (low SOC) [Ω]
	$R0_{ch,SOC,low,3s}$	SOC Pulse in Ch. HPPC Resistance 3s CT (low SOC) [Ω]	$R0_{ch,SOC,high,0s}$	SOC Pulse in Ch. HPPC Resistance 0s CT (high SOC) [Ω]
	$R0_{ch,SOC,high,2s}$	SOC Pulse in Ch. HPPC Resistance 2s CT (high SOC) [Ω]	$R0_{ch,SOC,high,3s}$	SOC Pulse in Ch. HPPC Resistance 3s CT (high SOC) [Ω]
	$R0_{dis,dis,low,0s}$	Dis. Pulse in Dis. HPPC Resistance 0s CT (low SOC) [Ω]	$R0_{dis,dis,low,2s}$	Dis. Pulse in Dis. HPPC Resistance 2s CT (low SOC) [Ω]
	$R0_{dis,dis,low,3s}$	Dis. Pulse in Dis. HPPC Resistance 3s CT (low SOC) [Ω]	$R0_{dis,dis,high,0s}$	Dis. Pulse in Dis. HPPC Resistance 0s CT (high SOC) [Ω]
	$R0_{dis,dis,high,2s}$	Dis. Pulse in Dis. HPPC Resistance 2s CT (high SOC) [Ω]	$R0_{dis,dis,high,3s}$	Dis. Pulse in Dis. HPPC Resistance 3s CT (high SOC) [Ω]
	$R0_{dis,ch,low,0s}$	Ch. Pulse in Dis. HPPC Resistance 0s CT (low SOC) [Ω]	$R0_{dis,ch,low,2s}$	Ch. Pulse in Dis. HPPC Resistance 2s CT (low SOC) [Ω]
	$R0_{dis,ch,low,3s}$	Ch. Pulse in Dis. HPPC Resistance 3s CT (low SOC) [Ω]	$R0_{dis,ch,high,0s}$	Ch. Pulse in Dis. HPPC Resistance 0s CT (high SOC) [Ω]
	$R0_{dis,ch,high,2s}$	Ch. Pulse in Dis. HPPC Resistance 2s CT (high SOC) [Ω]	$R0_{dis,ch,high,3s}$	Ch. Pulse in Dis. HPPC Resistance 3s CT (high SOC) [Ω]
	$R0_{dis,SOC,low,0s}$	SOC Pulse in Dis. HPPC Resistance 0s CT (low SOC) [Ω]	$R0_{dis,SOC,low,2s}$	SOC Pulse in Dis. HPPC Resistance 2s CT (low SOC) [Ω]
	$R0_{dis,SOC,low,3s}$	SOC Pulse in Dis. HPPC Resistance 3s CT (low SOC) [Ω]	$R0_{dis,SOC,high,0s}$	SOC Pulse in Dis. HPPC Resistance 0s CT (high SOC) [Ω]
	$R0_{dis,SOC,high,2s}$	SOC Pulse in Dis. HPPC Resistance 2s CT (high SOC) [Ω]	$R0_{dis,SOC,high,3s}$	SOC Pulse in Dis. HPPC Resistance 3s CT (high SOC) [Ω]
	$Q_{Ah,HPPC}$	HPPC Accumulated Ah-throughput [Ah]	$E_{Wh,HPPC}$	HPPC Accumulated Energy-throughput [Wh]

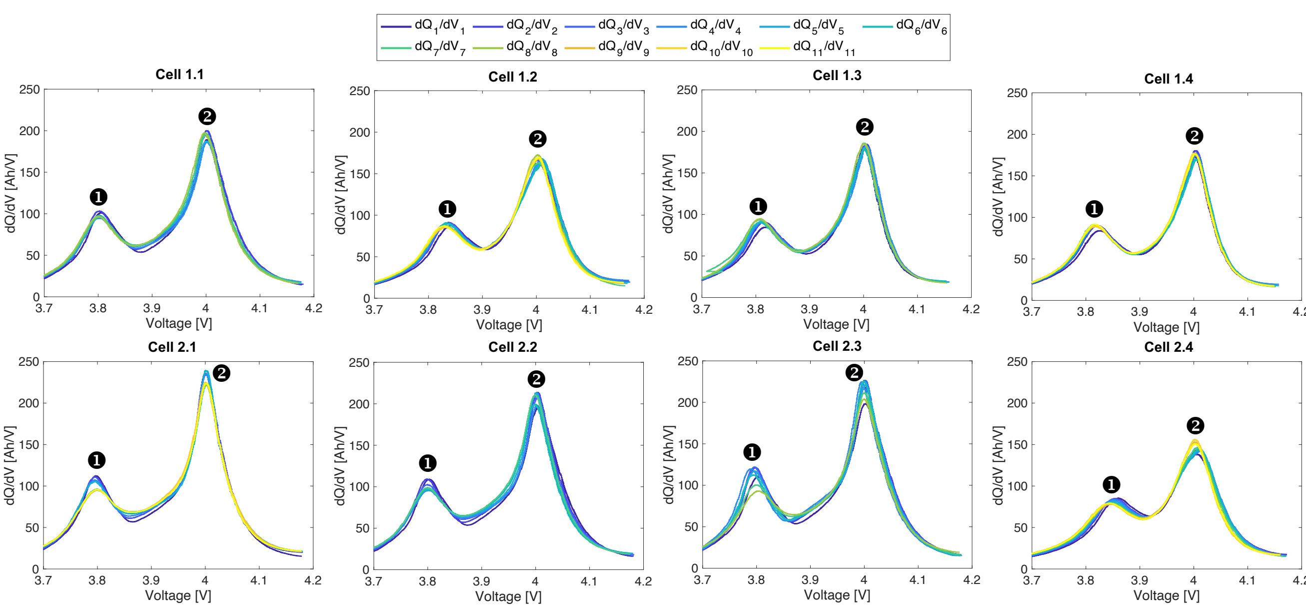


Figure S4: **IC curves for eight cells at different stages of aging.** These curves are extracted from the C/40 test, and they show two peaks marked as 1 and 2. Some cells have gone through more RPTs/cycles, such as Cell 2.4, but the curves do not show significant shift against voltage. For different cells, the change in the height of the peaks varies based on the degradation.

from charge pulse and discharge pulse (see Figure 1(b)), and from the change in current applied for a 10% SOC charge/discharge (referred to as SOC pulse in this work). Apart from calculating the resistance for the sudden voltage change when current is applied, we also calculate resistances by including 2 seconds and 3 seconds charge transfer (CT) period. We observe that, for most cases, 60% of the pulses are below 3.8 V while 40% of the pulses are above 3.8 V. We average the resistance values above (denoted as 'high') and below (denoted as 'low') 3.8 V to obtain two resistance values for each type of pulse. The naming convention for resistance features is $R0_{a,b,c,d}$ where $a \in \{ch, dis\}$ denotes the charge or discharge HPPC, $b \in \{ch, dis, SOC\}$ denotes the type of pulse used to calculate the resistance value, $c \in \{low, high\}$ denotes whether this resistance is obtained from pulses above or below 3.8 V, and $d \in \{0s, 2s, 3s\}$ denotes the CT time used to calculate the resistance value.

Using Spearman correlation [3], we obtain the heatmap shown in Figure S5. The last row of the heatmap corresponds to the target output. It can be seen that some features are strongly correlated with the target output, such as $Q_{ch,aging}$ with $|\rho| > 0.9$, while other features, such as $Q_{Ah,C/40}$, are not strongly correlated to the target output showing a value of $|\rho| < 0.2$. The latter features are those that contain minimal useful information about the target output, and they are dropped for model training. Although manual feature selection is not used for offline SOH estimation, it is used for feature selection in adaptive online SOH estimation (see Note S5).

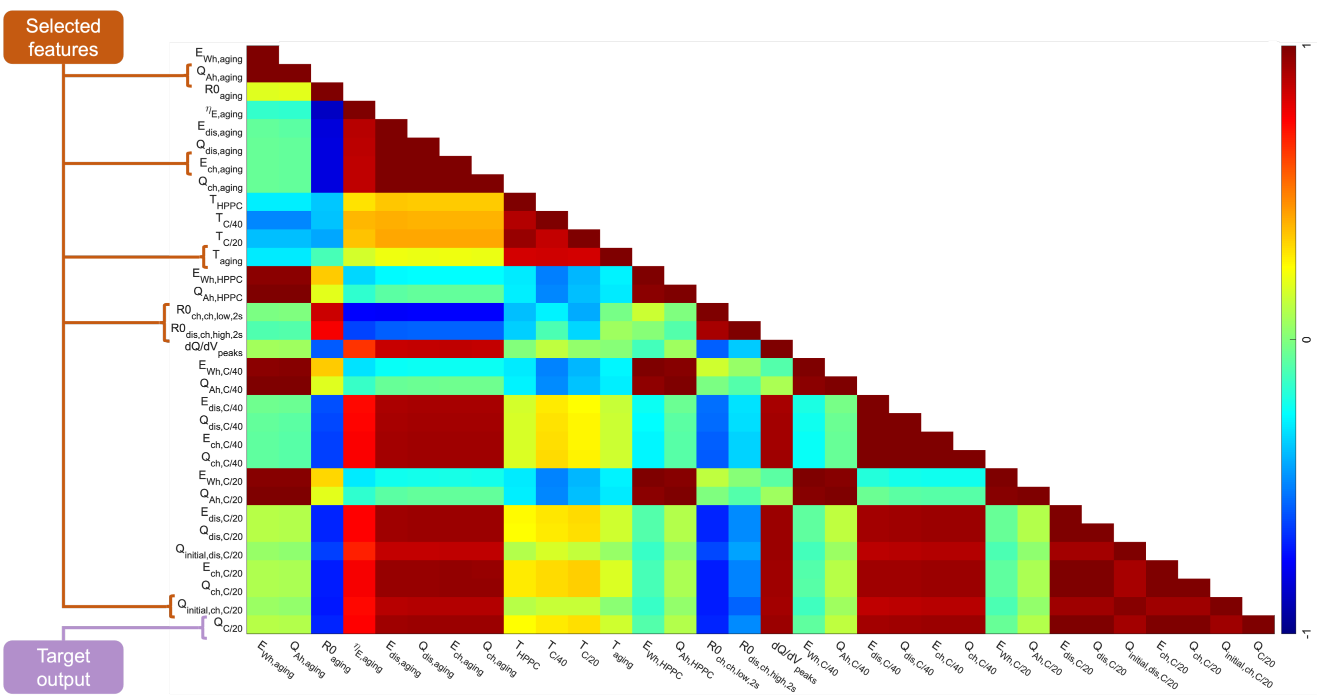


Figure S5: **Heatmap showing combined Spearman correlation for all eight cells.** The last row of the heatmap shows the correlation of the target output $Q_{C/20}$ with all the other features. For ease of readability, a subset of the HPPC resistance features are included in this heatmap. The selected features for offline SOH estimation are shown on the left side of the figure (both horizontal and vertical axis have the same features).

Data preprocessing

For SOH estimation model, selected input features are from aging cycles and the model output is $Q_{ch,C/20}$. For model training, the length of input features and model output should be the same. As shown in Figure S6, this is done by averaging the feature values of the last 20 aging cycles before an RPT. By doing so, the input feature vectors become the same length as $Q_{ch,C/20}$, which also reduces the amount of input data that needs to be processed for model training.

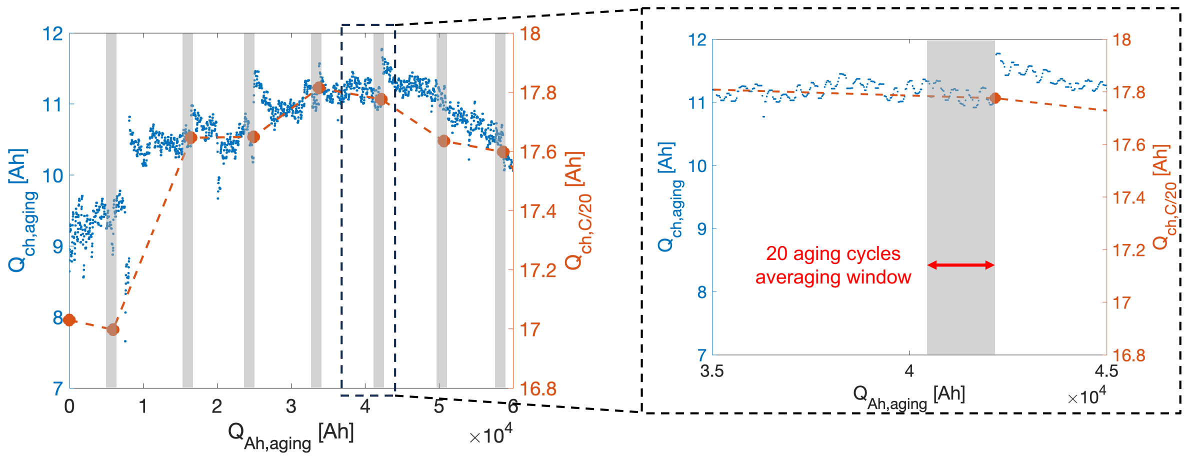


Figure S6: **Example of data pre-processing to match the lengths of input features and model output.** For every RPT, $Q_{ch,aging}$ is averaged for the last 20 cycles before the RPT to make the length of the input features and output the same.

Machine learning algorithms

Dataset partition

Offline SOH estimation models are trained on six cells and tested on two cells. In this paper, four combinations of test sets are used: (1.1,2.1),(1.2,2.2),(1.3,2.3),(1.4,2.4). For each test set combination, the remaining cells are used to train the model (see Note S4 for model performance). In the case of clustering-based adaptive estimation algorithm, seven cells are instead used for training, while one cell is used for testing.

Elastic-Net Regression

A regularized regression method that linearly combines L_1 and L_2 penalties of the Lasso and Ridge methods [4]. ENR assumes an affine relationship between the features and SOH indicators $Y = X\beta + \beta_0$. The slope and intercept parameters, β and β_0, respectively, can be obtained by solving the following optimization problem

$$\hat{\beta} = \underset{\beta_0,\beta,\lambda}{\operatorname{argmin}} \|Y - X\beta - \beta_0 \mathbf{1}_{n\times 1}\|_2 \\ + \lambda(1-\alpha)\|\beta\|_2^2 + \lambda\alpha\|\beta\|_1, \quad (6)$$

where $Y \in \mathbb{R}^n$ is a vector of measured SOH values, $X \in \mathbb{R}^{n\times m}$ is a matrix of m input features with n observations, $\beta \in \mathbb{R}^m$, $\beta_0 \in \mathbb{R}^n$, $\mathbf{1}_{n\times 1} \in \mathbb{R}^n$ is a vector of 1's, and $\lambda, \alpha \in \mathbb{R}^+$ are hyperparameters. For our model, a value of $\alpha = 0.2$ is chosen while λ is tuned on the training set using grid search and 5-fold cross-validation.

Other models

Three other regressions models used in this paper include Support Vector Regression (SVR) [5], Gaussian Process Regression (GPR) [6], and Random Forest Regression (RFR) [7]. SVR aims to find a hyperplane that minimizes the prediction error by transforming the input features into higher-dimensional space. GPR is a non-parametric, kernel-based probabilistic model that assumes the model output follows the Gaussian distribution. It uses a kernel function K to model the structure of the data and gives confidence intervals on the model predictions. For this work, K is selected as the squared exponential function. Lastly, RFR uses an ensemble of multiple decision trees using Bagging (bootstrap aggregation), and averages the predictions of all the decision trees to make the final prediction. The depth of the tree and the number of leaves (branches) – both hyperparameters – determine the computational cost of the model training.

Note S4: Robustness analysis for the data-driven model

Statistical significance of data-driven models

Due to the limited size of the dataset, a statistical significance test is performed to check the performance of SVR, RFR, and GPR models against the ENR model. Models are trained on six cells and evaluated on two test cells. To ensure generalizability in model performance, eight different combinations of test cells are chosen while the remaining cells are used for training. The set of test cells (TS) is given by

$$TS = \{(1.1, 2.1), (1.2, 2.2), (1.3, 2.3), (1.4, 2.4), (1.1, 2.4), (1.2, 2.3), (1.3, 2.2), (1.4, 2.1)\}$$

All four models are evaluated on each combination of the test cells, and based on the results, an RMSE value is calculated. For the significance test, the null hypothesis H_0 states that RMSE of ENR is similar to the RMSE of the other three models while the alternate hypothesis H_1 states that RMSE of ENR is smaller than the other models. To test our hypotheses, we choose a significance level ρ of 5%, which is equivalent to a probability $p = 0.05$. When $p < 0.05$, H_0 can be rejected in favor of H_1; otherwise, H_0 cannot be rejected and no claim can be made about H_1. Since we consider eight combinations of test sets, we only have the sample mean instead of the population mean. In such cases, the t-test [8] is used. A t-value is calculated by

$$t = \frac{\bar{y}_{MOD} - \mu_{ENR}}{s_{ENR}/\sqrt{n}} \tag{7}$$

where $\bar{y}_{MOD}$ is the sample mean of RMSE for model $MOD \in \{SVR, RFR, GPR\}$ that we want to test, μ_{ENR} is the mean of RMSE of ENR, s_{ENR} is the sample standard deviation of RMSE of ENR, and $n = 8$ is the number of test sets. Table S3 shows the RMSE for all the four models for all eight test sets.

It can be seen that apart from RFR, all models have at least one or more test sets which provide the least RMSE value. For model selection, from the performance of the four models on all test sets, it can be seen that ENR consistently gives better performance than the other models. As shown in Figure S7, on test sets $(1.3, 2.3)$ and $(1.3, 2.2)$, the RMSE from SVR model is significantly larger than other models while RFR also has RMSE$> 1\,\mathrm{Ah}$ for four test sets.

The calculated t-values for the four models are shown in Table S3. Since ENR is our reference model, its t-value is zero. The t-value is used with the t-distribution [8] which is similar to normal distribution, but with heavier tails. From the t-distribution table, a threshold t-value $t_{threshold} = 2.145$ is obtained that corresponds to $p = 0.05$ and

Table S3: **RMSE and t-value for all four data-driven models for eight test sets**

	RMSE [Ah]			
Test sets	ENR	SVR	RFR	GPR
$(1.1, 2.1)$	0.6669	**0.4851**	1.1656	0.6035
$(1.2, 2.2)$	0.5295	**0.4799**	0.5649	0.7901
$(1.3, 2.3)$	0.7636	3.9538	0.7762	**0.6521**
$(1.4, 2.4)$	**0.3237**	0.4709	1.0936	0.6078
$(1.1, 2.4)$	**0.5665**	0.5871	1.1363	0.5792
$(1.2, 2.3)$	0.7972	**0.6981**	0.6910	0.7951
$(1.3, 2.2)$	0.6088	3.0228	0.5844	**0.5728**
$(1.4, 2.1)$	**0.4020**	0.5517	1.2789	0.7094
Sample mean μ	**0.5823**	1.2812	0.9114	0.6638
Sample std. s	0.1645	1.3868	0.2872	**0.0907**
t-value	0	12.0190	5.6596	1.4013

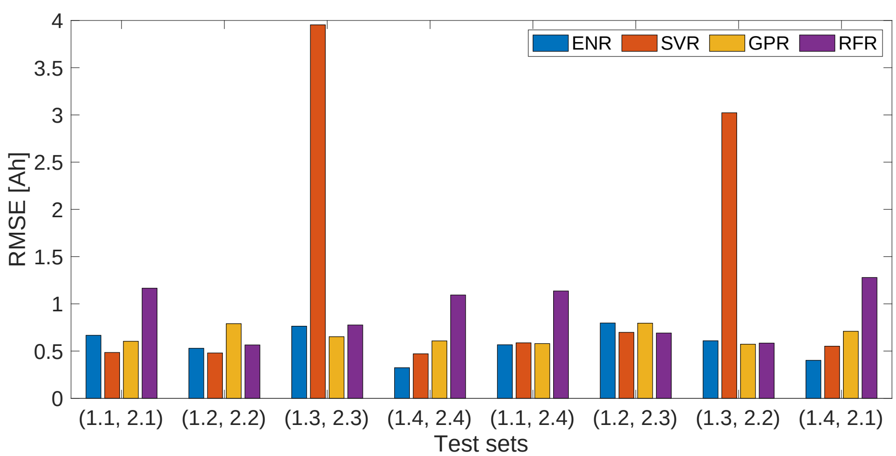


Figure S7: RMSE of the four data-driven models for eight different tests sets. ENR and GPR consistently give RMSE < 1 Ah while SVR has significantly large errors for test sets $(1.3, 2.3)$ and $(1.3, 2.2)$.

degrees of freedom $df = n + n - 2 = 8 + 8 - 2 = 14$. By comparing our t-values obtained from the models to $t_{threshold}$, we can reject H_0 if $t > t_{threshold}$. Both SVR and RFR have t-values higher than the threshold t-value, but for GPR, the t-value is lower than the threshold. Based on this, we can claim that the H_0 can be rejected for SVR and RFR models, which means RMSE of ENR is smaller than SVR and RFR with statistical significance. For GPR, we cannot reject H_0, which means RMSE of both models is comparable. With ENR performing better than 2 out of 3 models based on the significance test, it is selected as the preferred SOH estimation model in this work.

Robustness of the ENR model

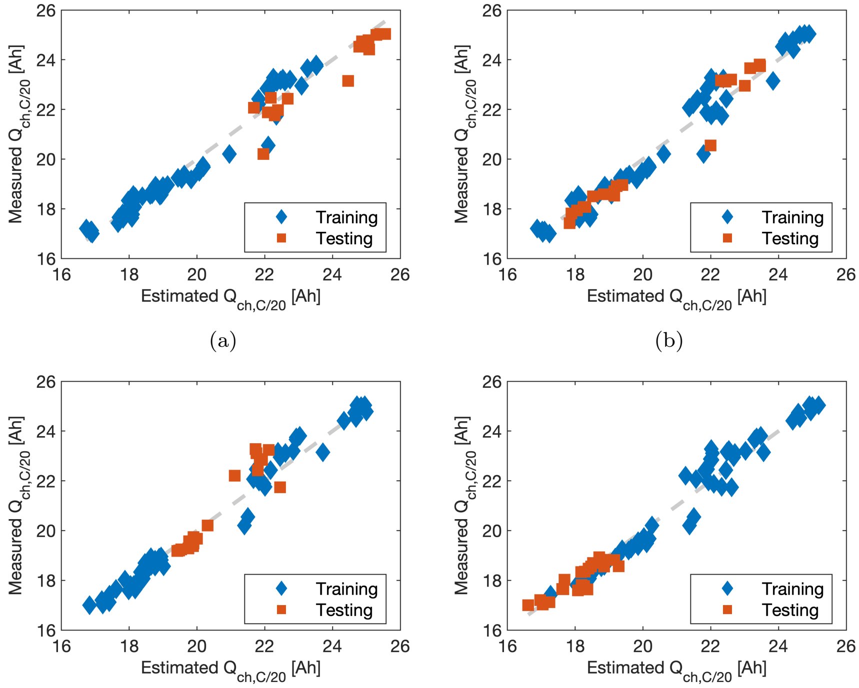
To examine the robustness of the ENR model, we test the performance of the model on four different test sets. Parity plots of estimated and measured capacity reveal three distinct patterns of training-testing dataset splits, as illustrated in Figure S8. It can be seen that despite the presence of the training data at either ends of the measured capacity range, as seen in Figures S8(a) and S8(d), the model is still able to give good estimation results on the test sets. This shows that the model can perform well on the edge cases of the dataset. This is further supported by the mean absolute percentage error (MAPE) shown in Figure S9. Only in the case of Cells 1.3 and 2.3, the test set error is significantly higher. We attribute this to the presence of anomalous data in the extracted features which originate from errors in the measured data for these cells. Furthermore, Figure S10 depicts the distribution histogram of pointwise capacity estimation percentage errors (PCEPE) between the measured and estimated capacities. All estimations remain within a $\pm 6\%$ error bound, and most fell within a $\pm 3\%$ error bound. These results indicate that ENR model is robust and maintains its performance over various combinations of the test cells.

Robustness of the adaptive model

To validate the robustness of the adaptive method, we perform leave-one-out validation on the entire dataset by training the model on seven cells and testing it on one cell. The root mean squared percentage error (RMSPE) of the offline ENR model is 3.40%, while the RMSPE of the adaptive model is 3.27%. The RMSPE for each training-testing set split is given in Table S4. The adaptive estimation improves the estimation accuracy for Cells 1.1, 1.2, 1.4, 2.3, and 2.4. However, for Cells 1.3, 2.1, and 2.2, adding the clustering-based estimation leads to a degradation in accuracy. This is attributed to the low correlation between the distances in the existing aging-cycle features and the distances in SOH for these cells. Overcoming this limitation could potentially involve fine-tuning the clustering method, revising the distance metric, extracting additional features, or training the clustering method with a larger dataset of SL-battery aging data.

Table S4: **Comparison of RMSPE from adaptive model and offline ENR model**

RMSPE [%]	Cell 1.1	Cell 1.2	Cell 1.3	Cell 1.4	Cell 2.1	Cell 2.2	Cell 2.3	Cell 2.4
Adaptive model	2.93	1.22	2.62	0.86	2.26	3.82	7.27	0.95
Offline ENR model	2.95	1.47	2.41	1.82	1.45	3.27	7.82	1.58



(a) (b)

(c) (d)

Figure S8: **Estimated and measured $Q_{ch,C/20}$ for four different combinations of training and test sets.** (a) Test set $(1.1, 2.1)$, (b) Test set $(1.2, 2.2)$, (c) Test set $(1.3, 2.3)$, and (d) Test set $(1.4, 2.4)$.

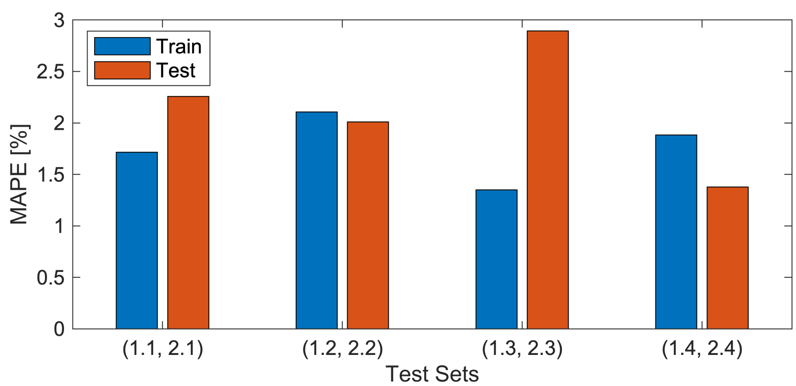


Figure S9: Comparison of the mean absolute percentage error (MAPE) between four different combinations of training and test sets.

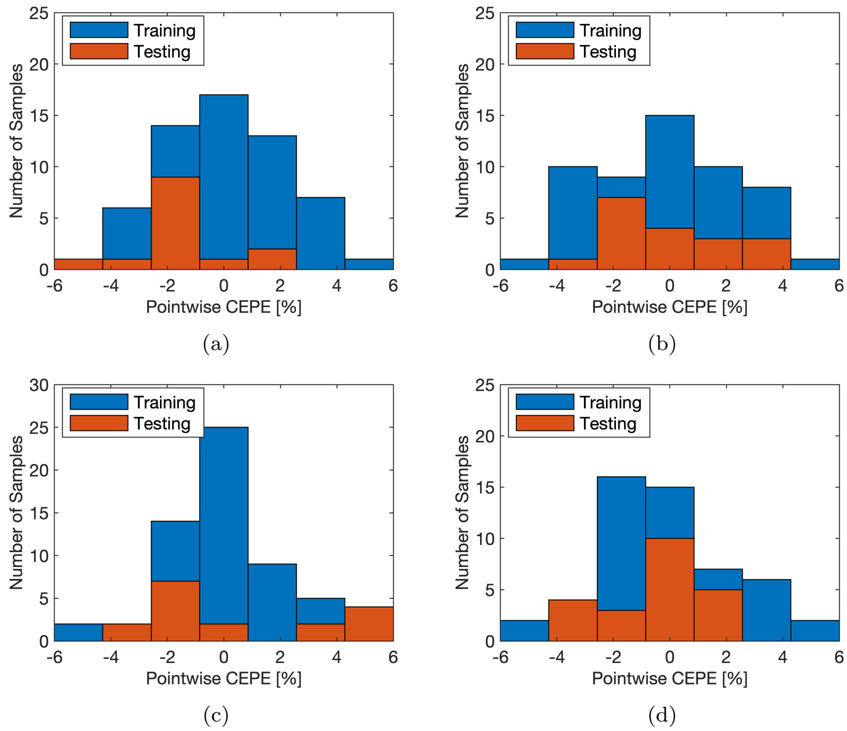


Figure S10: **Histogram of pointwise capacity estimation percentage errors (CEPE) between measured and estimated $Q_{ch,C/20}$.** (a) Test set $(1.1, 2.1)$, (b) Test set $(1.2, 2.2)$, (c) Test set $(1.3, 2.3)$, and (d) Test set $(1.4, 2.4)$.

Note S5: Online adaptive SOH estimation

In this algorithm, we leverage the idea of clustering to assess the proximity of one or more input features, in the feature space, based on a distance metric, and identify a cell from the training set that behaves similarly to the test cell [9]. By doing so, knowledge of SOH from the known cell can be used to improve the SOH estimation on the test cell. The key assumption of this algorithm is that if two cells have close features in the feature vector space, then the corresponding SOH values should be close as long as the features have a high correlation to the SOH. As mentioned in Note S3, seven cells are used for training which means there are seven different clusters. For BMS_2, as new data becomes available, the algorithm repeatedly checks the distance between the test cell and seven clusters, and uses SOH information from the cluster with the smallest distance. Naturally, this suggests that one test cell can belong to a different cluster at different points in time. The results are shown in Figure 6 for Cell 2.4.

From Figure S5, we can see that $Q_{ch,aging}$ has high correlation with $Q_{ch,C/20}$, which suggests that closeness of $Q_{ch,aging}$ for two cells implies closeness of their SOH. Using the L_2 distance metric, the distance between $Q_{ch,aging}$ of Cell z and Cell k is defined as

$$\text{dist}(Q^z_{ch,aging}, Q^k_{ch,aging}) = \sqrt{\sum_{i=1}^{N} \left(Q^z_{ch,aging}(Ah(i)) - Q^k_{ch,aging}(Ah(i))\right)^2}, \tag{8}$$

where z is the test cell, k is a cell in the training set, $i = 1, 2, 3, ..., N$ and $Ah(i)$ is the Ah-throughput value at which the L_2 distance is calculated. $Q^z_{ch,aging}(Ah(i))$ and $Q^k_{ch,aging}(Ah(i))$ are discrete-time trajectories which means as i increases, longer feature trajectories are used to calculate the distance. This also implies that knowledge of measurement history is incorporated into the online SOH estimation.

Once distance between test cell z and all the training cells 1 to k is calculated, the cluster S^z with the minimum distance is given by

$$S^z(Ah(n)) = \underset{1\leq k\leq K}{\operatorname{argmin}} \left(\text{dist}\left(Q^z_{ch,aging}(Ah(1),\ldots,Ah(n)), Q^k_{ch,aging}(Ah(1),\ldots,Ah(n))\right)\right) \tag{9}$$

where $K = 7$ is the total number of clusters and $S^z = k^*$ is the cluster with the minimum distance to Cell z at $Ah(n)$. To obtain the estimated C/20 charge capacity of Cell z $\hat{Q}^z_{ch,C/20}$, we use a linear combination of estimated C/20 charge capacities $\bar{Q}^k_{ch,C/20}$ from each cluster with different weights λ_k [9] given by

$$\hat{Q}^{z,ct}_{ch,C/20}(Ah) = Q^z_{initial,ch,C/20}\sum_{k=1}^{K}\lambda_k\bar{Q}^k_{ch,C/20}(Ah) \tag{10}$$

The final online SOH estimation $\hat{Q}^z_{ch,C/20}$ is a weighted combination of the result from regression model (offline ENR) $\hat{Q}^{z,rg}_{ch,C/20}$ and the clustering-based model $\hat{Q}^{z,ct}_{ch,C/20}$

$$\hat{Q}_{ch,C/20} = (1 - w(Ah))\hat{Q}^{rg}_{ch,C/20} + w(Ah)\hat{Q}^{ct}_{ch,C/20} \tag{11}$$

$$w(Ah) = \alpha Ah, \tag{12}$$

where the weight $0 \leq w(Ah) \leq 0.5$ [9] is a linear function of Ah-throughput, and α [9] is a hyperparameter that controls the relative contribution of the two models to the online SOH estimation. Even though this algorithm works in an open-loop manner, the algorithm guarantees that the error on the estimates be bounded [9].

References

[1] K. Moy and S. Onori, "Synthetic Grid Storage Duty Cycles for Second-Life Lithium-Ion Battery Experiments," *SAE Technical Papers*, pp. 1–9, 2023. [Online]. Available: https://doi.org/10.4271/2023-01-0516

[2] D. Anseán, V. M. García, M. González, C. Blanco-Viejo, J. C. Viera, Y. F. Pulido, and L. Sánchez, "Lithium-ion battery degradation indicators via incremental capacity analysis," *IEEE Transactions on Industry Applications*, vol. 55, no. 3, pp. 2992–3002, 2019. [Online]. Available: https://doi.org/10.1109/TIA.2019.2891213

[3] C. Wissler, "The spearman correlation formula," *Science*, vol. 22, no. 558, pp. 309–311, 1905. [Online]. Available: https://doi.org/10.1126/science.22.558.309

[4] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, apr 2005. [Online]. Available: https://doi.org/10.1111/j.1467-9868.2005.00503.x

[5] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 2004. [Online]. Available: https://doi.org/10.1023/B:STCO.0000035301.49549.88

[6] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018. [Online]. Available: https://doi.org/10.1016/j.jmp.2018.03.001

[7] M. R. Segal, "Machine Learning Benchmarks and Random Forest Regression," apr 2004. [Online]. Available: https://escholarship.org/uc/item/35x3v9t4

[8] T. K. Kim, "T test as a parametric statistic," *Korean journal of anesthesiology*, vol. 68, no. 6, pp. 540–546, 2015. [Online]. Available: https://ekja.org/journal/view.php?doi=10.4097/kjae.2015.68.6.540

[9] X. Cui, M. A. Khan, and S. Onori, "Online Adaptive Data-driven State-of-health Estimation for Second-life Batteries with BIBO Stability Guarantees," *arXiv preprint arXiv:2401.04734*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.04734