

Lightweight Single-Cycle Prognostics for Li-ion Batteries: Feature Extraction and Cross-Cell Generalization

Gabriel Buzzi Sanches, Carlos Antônio Rufino Junior, Lucas Saraiva Texeira, Hudson Zanin
Universidade Estadual de Campinas
gabrielbuzzi2@gmail.com

January 20, 2026

Abstract

This paper proposes a single-cycle, feature-based approach for estimating the State of Health (SoH) and Remaining Useful Life (RUL) of lithium-ion batteries. Using the public Severson LFP dataset (124 cells), we extract 16 statistical features from voltage, current, and temperature signals of a diagnostic full cycle. Gradient boosting and random forest regressors are trained with cell-wise cross-validation and tested on 25 unseen cells. Results show a median RMSE of 0.7–0.8% for SoH table 7 and 42–60 cycles for RUL table 8, with consistent generalization across batches. Our analysis highlights the most informative features, the trade-off between feature count and accuracy, and the deployment potential of lightweight models. Limitations include the single chemistry and the need for diagnostic cycles, which future work will address.

Keywords: battery prognostics; state of health; remaining useful life; feature engineering; LFP; gradient boosting

1 Introduction

The transition to renewable energy and the rapid adoption of electric mobility have placed Lithium-ion Batteries (LiBs) at the center of modern energy storage technologies. Ensuring their safety, reliability, and economic viability requires accurate monitoring of degradation and forecasting of future performance, a field known as battery prognosis. Two key indicators are State of Health (SoH), the capacity relative to a new cell, and Remaining Useful Life (RUL), the cycles until a predefined end-of-life. Reliable estimation of these metrics is critical for optimizing battery use, enabling predictive maintenance, and extending system lifetime in applications such as electric vehicles and stationary storage systems.

Despite significant progress, existing methods for battery prognosis face important challenges. Many Machine Learning (ML) approaches rely on handcrafted features extracted from full charge–discharge cycles, which restricts their applicability to controlled datasets where continuous cycling data are available. More advanced deep learning methods have been proposed to process raw electrochemical signals directly, thereby bypassing manual feature engineering [Che et al. \[2022\]](#), [Lv](#)

et al. [2024]. However, these approaches introduce substantial computational demands and often lack interpretability, which limits their deployment in embedded battery management systems. Furthermore, most studies validate their methods on laboratory datasets collected under highly controlled conditions, often with aggressive fast-charging protocols and uniform cycling patterns Jafari and Byun [2024], Liu and Teh [2025]. Such protocols do not reflect the intermittent and partial cycling profiles seen in real-world applications, reducing the generalizability of the resulting models.

Another key limitation is the reliance on multiple consecutive cycles to extract reliable prognostic features. Operational systems rarely provide uninterrupted sequences of complete charge–discharge cycles, making such requirements impractical. This highlights the need for models capable of producing robust SoH and RUL estimates from limited information, ideally from a single diagnostic cycle, without assuming continuous access to full degradation histories.

In this work, we address these challenges by developing machine learning models trained on a large, publicly available dataset of 124 commercial LiFePO₄/graphite cells cycled under fast-charging conditions Severson et al. [2019]. Unlike prior studies that depend on sequences of multiple cycles, our methodology demonstrates that accurate inference of both SoH and RUL can be achieved using only the information contained in a single complete cycle. This approach reflects realistic diagnostic scenarios, where a dedicated cycle can be periodically performed to assess battery condition without interrupting normal operation. By extracting statistical features from voltage, current, and temperature signals at the cycle level, we provide a compact and computationally efficient representation suitable for supervised learning. Our contributions are twofold: (i) we show that reliable prognosis does not require long degradation histories for inference, thereby increasing the applicability of ML methods in real-world battery management systems, and (ii) we present a systematic evaluation of feature-based models for SoH and RUL estimation, highlighting the balance between interpretability, efficiency, and predictive performance.

2 Related Works

Several recent studies have sought to improve the accuracy, generalization, and interpretability of ML-based methods for battery prognosis. Che et al. [2022] proposed a probabilistic LSTM framework that extracts health indicators from partial Q–V curves, demonstrating strong generalization across multiple datasets of LiFePO₄ cells. While their method reduces data requirements through transfer learning, it remains specific to a single chemistry and depends on carefully chosen voltage ranges.

Jafari and Byun [2024] explored measurable features such as discharge time and temperature, integrating Explainable AI (XAI) methods to improve interpretability. Although their approach is practical for electric vehicle applications, it was validated on a very limited dataset of only four cells, restricting its generalizability.

To address the lack of physical interpretability in purely data-driven methods, Liu and Teh

[2025] proposed a hybrid model combining an incremental internal resistance aging model (IIRAM) with a gated recurrent unit (GRU) network. This approach improved robustness and provided physically meaningful health indicators but remained restricted to constant-current conditions, which rarely occur in real-world systems. Similarly, Lv et al. [2024] combined signal decomposition with CNN-BiGRU architectures to handle nonlinear degradation and regeneration effects, though the complexity of the model and reliance on small datasets raise concerns for real-time deployment.

Overall, these studies highlight the potential of ML for SoH and RUL estimation, but also reveal recurring limitations: dependence on long sequences of complete cycles, lack of robustness under realistic operating conditions, and challenges in generalization across chemistries and applications. The present work advances the field by showing that accurate predictions can be achieved using only a single diagnostic cycle, thereby relaxing the data requirements and improving practical applicability in battery management systems.

3 Dataset Descriptio

The dataset utilized in this study, originally presented in Severson et al. [2019], comprises cycling data from 124 commercial lithium iron phosphate (LFP)/graphite batteries. These batteries, manufactured by A123 Systems (model APR18650M1A), have a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V. The original study aimed to optimize fast charging for lithium-ion batteries, with all batteries cycled to failure under various fast-charging conditions in a controlled environment. In this work, the dataset was accessed using the provided MATLAB struct (.mat) files, which serve as a convenient container for individual cycle data and can be loaded in MATLAB or Python (via the h5py package). Pandas dataframes can also be generated from these structs using provided code.

The batteries were cycled in horizontal cylindrical fixtures on a 48-channel Arbin LBT potentiostat within a forced convection temperature chamber maintained at 30°C. Charging was performed using either one-step or two-step fast-charging policies, denoted as "C1(Q1)-C2," where C1 and C2 represent the constant current rates for the first and second steps, respectively, and Q1 indicates the state-of-charge (SOC, in %) at which the current switches. The second current step concluded at 80% SOC, followed by a 1C constant current-constant voltage (CC-CV) charging protocol. The upper and lower voltage cutoffs were set at 3.6 V and 2.0 V, respectively, adhering to the manufacturer's specifications. These voltage limits were enforced across all charging steps, including fast charging, and after several cycles, some batteries reached the upper voltage limit during fast charging, resulting in extended constant-voltage charging periods. All batteries were discharged at a constant current rate of 4C.

The dataset includes measurements of voltage, current, temperature, charge capacity, discharge capacity, and internal resistance for each cycle. The temperature data was collected by attaching type T thermocouples to the exposed cell cans with thermal epoxy (OMEGATHERM 201) and Kapton tape, after removing a small section of the plastic insulation. However, the reliability of temperature measurements may vary due to inconsistent thermal contact between the thermocouple

and the battery can, with some thermocouples losing contact during cycling. Internal resistance was measured during charging at 80% SOC by averaging 10 pulses of $\pm 3.6\text{C}$, with a pulse width of 30 ms for Batches 1 and 2, or 33 ms for Batch 3.

Table 1: Summary of Severson dataset batches and protocols.

Batch	Charge protocol	Discharge protocol	Notes
1	CCCV @ 1.5C (cutoff 3.6V)	CC 4C to 2.0V	45 cells, thermocouples present
2	CCCV @ 1.5C (cutoff 3.6V)	CC 4C to 2.0V	43 cells, restart artifact
3	CCCV @ 1.5C (cutoff 3.6V)	CC 4C to 2.0V	36 cells, longer rests

4 State of Health and Remaining Useful Life

The development of machine learning models for battery prognosis requires clearly defined target variables, namely State of Health (SoH) and Remaining Useful Life (RUL). These metrics serve as key performance indicators to quantify the degradation state of a battery. The following section outlines the methodology used to calculate these ground-truth labels from the experimental cycling data.

The SoH of a battery provides a snapshot of its current condition relative to its initial, "healthy" state. It is typically expressed as a percentage of the nominal capacity. For the purpose of this study, the SoH for the n th cycle, denoted as SoH_n , is calculated as the ratio of the maximum discharge capacity observed during that cycle to the rated cell capacity, as shown in eq. (1).

$$SoH_n = \frac{Q_n}{Q_{\text{rated}}} \quad (1)$$

where Q_n is the maximum discharge capacity recorded for the n th cycle, and Q_{rated} is the rated capacity of a new cell.

A critical limitation in determining the ground-truth SoH in real-world applications is that batteries rarely undergo a full charge or discharge cycle. This makes it challenging to accurately measure the true capacity at any given time. The controlled laboratory setting used to generate this dataset, where full cycles are performed, provides a consistent and reliable method for calculating this critical ground truth.

The RUL of a battery represents the number of cycles it can endure before reaching its End of Life (EoL) condition. The EoL is a predefined failure threshold, commonly set when the battery's capacity drops to 80% of its initial rated capacity. The EoL cycle is therefore defined as the cycle number at which the SoH first falls to 80% or below.

$$EoL = \min \{n \mid SoH_n \leq 0.80\} \quad (2)$$

The RUL for any given cycle n , denoted as RUL_n , is then calculated by subtracting the current cycle number from the EoL cycle number, as expressed in eq. (3) below.

$$RUL_n = EoL - n \quad (3)$$

A significant challenge associated with using cycle count to define RUL is its limited applicability to real-world scenarios, where usage is often irregular and does not consist of discrete, well-defined cycles. Furthermore, determining the ground truth RUL for a particular cycle requires having the full degradation data up to the battery's EoL, which is a common limitation when training RUL prediction models from incomplete datasets.

Below, fig. 1 shows the distribution of both SoH and RUL on the dataset from [Severson et al. \[2019\]](#). As can be seen, neither target exhibits a symmetric distribution. This skewed nature necessitates the use of machine learning models that are robust to non-normal data distributions and can effectively capture the non-linear degradation behavior of batteries.

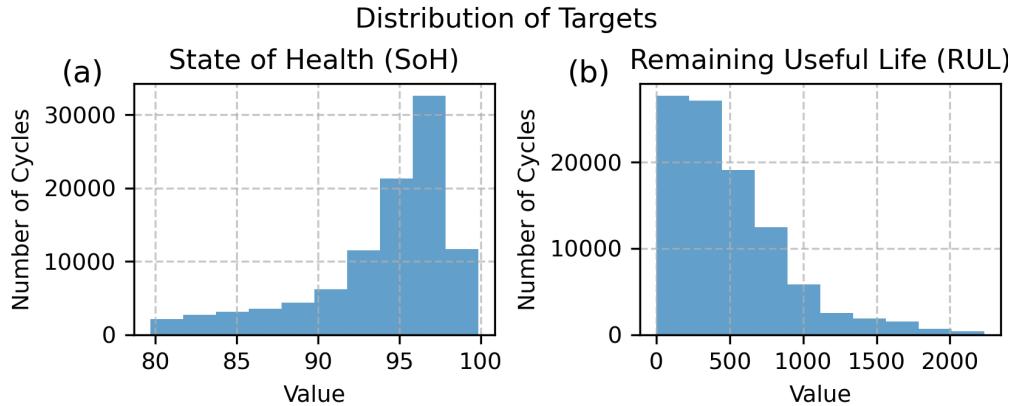


Figure 1: Distribution of State-of-Health (SOH) and Remaining Useful Life (RUL) target variables across the 124 cells in the dataset.

5 Data Analysis

Building upon the dataset description provided in the previous section, this section presents an exploratory data analysis of the cycling data from 124 commercial Lithium Iron Phosphate (LFP) batteries, as detailed in [Severson et al. \[2019\]](#). The analysis focuses on characterizing the voltage, current, temperature, and capacity measurements under the fast-charging protocols described earlier, with an emphasis on their evolution over the cells' lifecycle. These insights are critical for understanding battery degradation patterns and informing the development of machine learning models for estimating State of Health (SoH) and Remaining Useful Life (RUL). The dataset, organized into three batches with varying fast-charging protocols, provides a comprehensive basis for analyzing the impact of high C-rates and full Depth of Discharge (DoD) protocols, despite the limitations noted, such as controlled laboratory conditions and inconsistent temperature measurements.

The temporal behavior of voltage, current, and temperature during a representative charge-discharge cycle is illustrated in fig. 2, using the first cycle of the 32nd cell from Batch 2 as an example. This cell follows a 5.2C(50%)-4.25C protocol, where charging involves a constant current

(CC) of 5.2C until 50% State of Charge (SoC), followed by 4.25C until 80% SoC, as described in the dataset’s experimental setup. After a 5-minute rest period, a 1C CC step continues until the upper voltage cutoff of 3.6 V, transitioning to constant voltage (CV) charging with naturally decreasing current until 100% SoC. Discharge occurs at a constant 4C current until the lower voltage cutoff of 2.0 V, followed by CV discharge until full discharge, adhering to the C/50 cutoff current for Batch 2.

The temperature profile, shown in fig. 2(c), reveals peaks during both charging and discharging, with the rate of temperature increase correlating with current intensity. During the rest period, the temperature drops, and during the 1C CC charging phase, it stabilizes, likely due to balanced heat generation and dissipation in the 30 °C chamber. This pattern is consistent with the discharge phase, where temperature rises with applied current and stabilizes during CV discharge. Although the observed correlation between temperature and the integral of current (i.e., capacity exchange) highlights that temperature can serve as a useful indicator of SoC, it should be noted that this relation partly reflects the specific experimental conditions—namely constant charge/discharge profiles under controlled ambient temperature—and may not generalize to all operating scenarios.

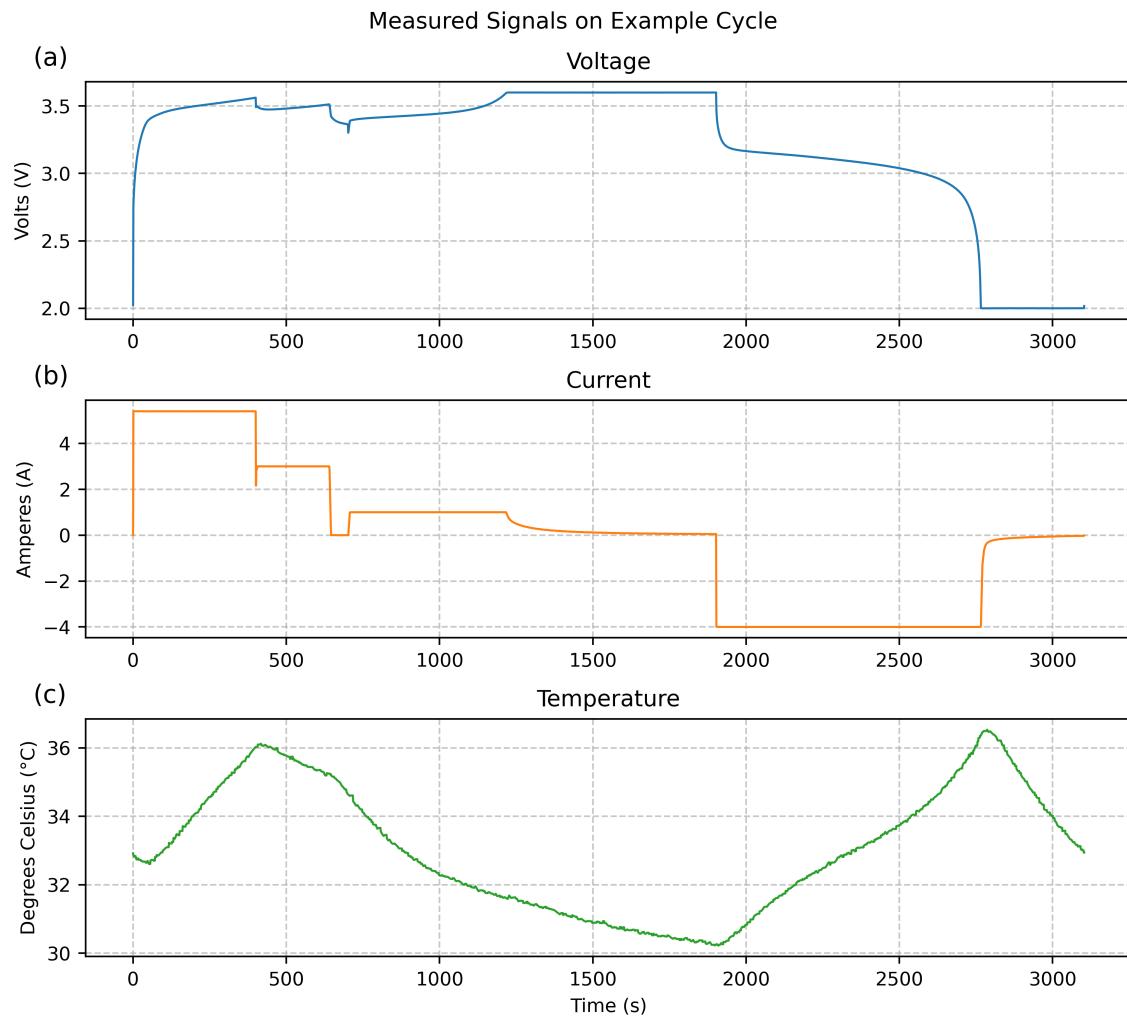


Figure 2: Temporal behavior of voltage, current, and temperature during a representative charge-discharge cycle.

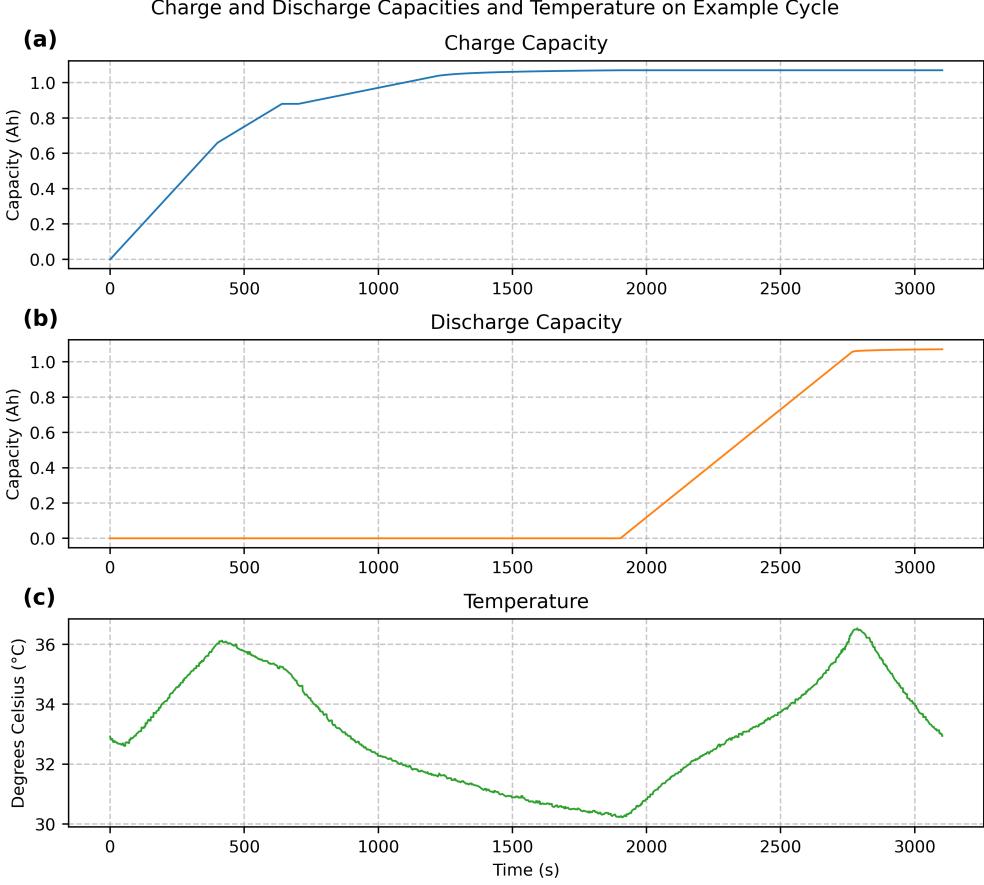


Figure 3: Temporal evolution of charge/discharge capacities and temperature throughout cell lifecycle.

The evolution of SoH across the cells' lifecycle, as defined in the dataset description, is depicted in fig. 4. Most cells start with an SoH near 100% and exhibit a gradual decline until a characteristic "knee point", after which the degradation rate accelerates. This behavior is consistent across all batches, despite variations in rest periods and cutoff currents. The cycle life distribution, shown in fig. 5, indicates that most cells reach End of Life (EoL) around 700 cycles, lower than typical for 1.1 Ah LFP cells, likely due to the aggressive 4C discharge and full DoD protocols described earlier. As noted in the dataset limitations, this accelerated aging may not reflect real-world conditions. An anomaly around cycle 250 in Batch 2, caused by an 8-hour rest period due to a computer restart, manifests as capacity spikes, introducing outliers in voltage, current, and temperature measurements. These outliers are removed during data preprocessing to ensure robust model training.

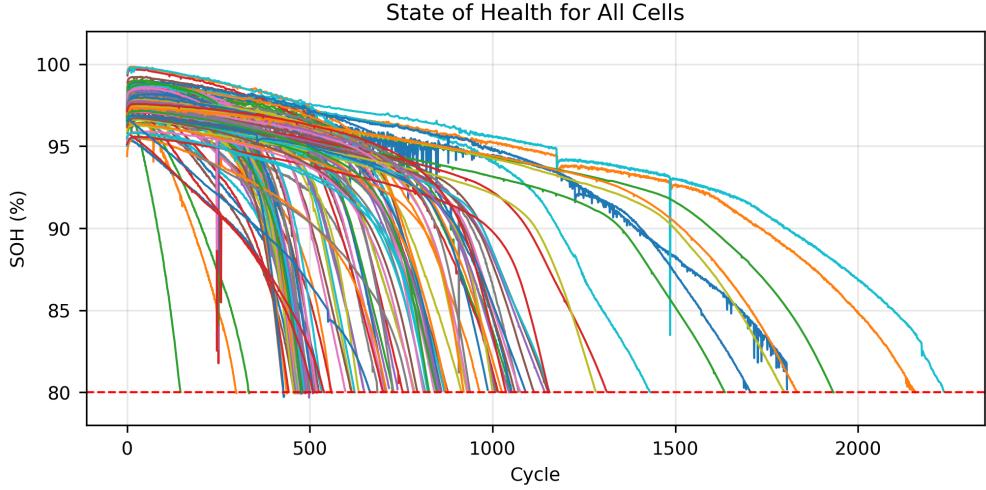


Figure 4: State of Health (SOH) evolution for all cells across Batches 1, 2, and 3.

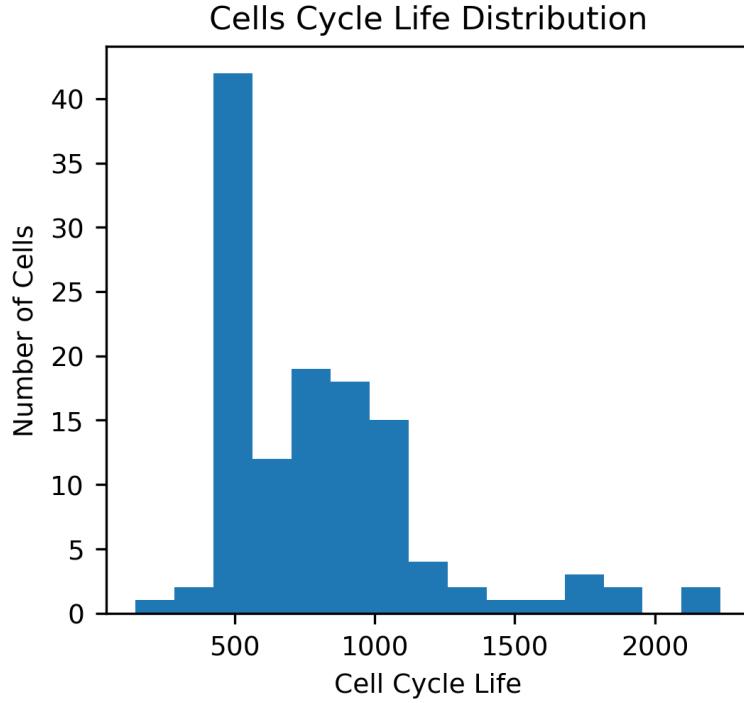


Figure 5: Distribution of cycle life (RUL) across all 124 cells in the Severson LFP dataset.

The impact of aging on voltage and current profiles is examined in fig. 6, which compares these signals at the beginning, middle, and end the life of the 32nd cell from Batch 2 as an example. At the start (fig. 6(a)), the profiles align with the defined fast-charge protocol. By mid-life (fig. 6(b)), changes are subtle, but at EoL (fig. 6(c)), increased internal resistance, measured as described in the dataset, causes the 3.6 V and 2.2 V thresholds to be reached earlier, leading to extended CV phases with reduced current. This results in shorter cycle durations at EoL, reflecting the reduced

capacity observed post-knee point in fig. 4.

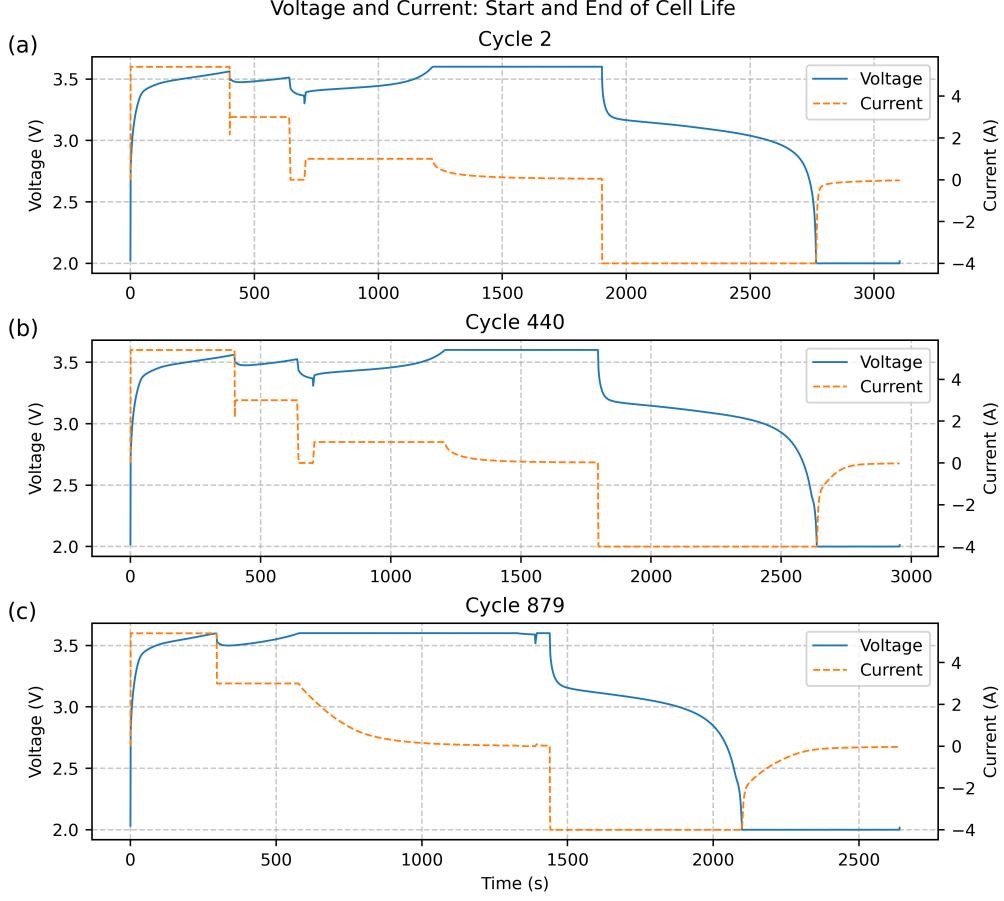


Figure 6: Voltage and current profiles at the beginning and end of battery life, showing aging effects.

A detailed visualization of the fast-charge protocol is provided in fig. 7, with color-coded regions highlighting the CC1, CC2, CC3, CV charge, discharge, and rest periods, as described in the experimental setup. For Batch 2, the protocol includes a 5-minute rest after 80% SoC and post-discharge, with C/50 cutoff currents for CV phases. This structured protocol ensures consistency across cycles, facilitating the analysis of aging effects.

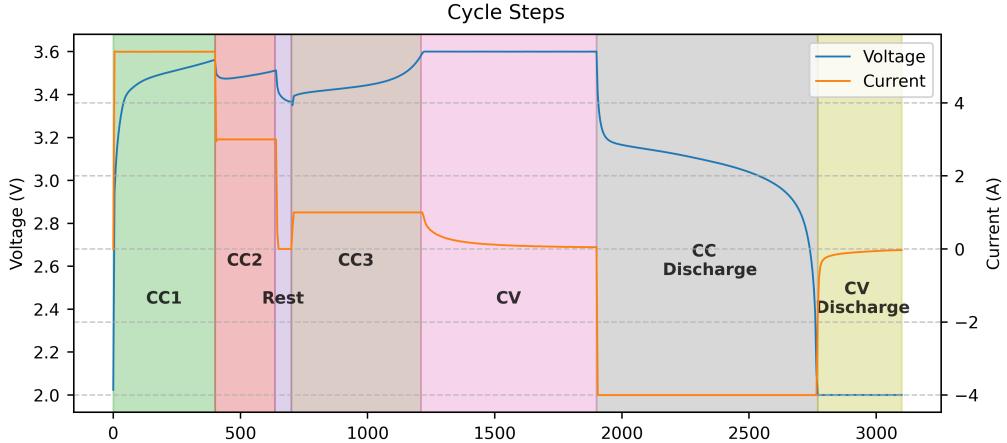


Figure 7: Charging protocol stages during constant current constant voltage (CCCV) charging phase.

The evolution of voltage, current, and temperature across multiple cycles is shown in fig. 8, using a Batch 2 cell as an example. The most significant changes occur near EoL (cycle 501), where increased internal resistance, as measured at 80% SoC, leads to earlier CV phases and higher peak temperatures during discharge, likely due to enhanced Joule heating. This aligns with the dataset's observation of thermocouple variability, which may amplify temperature measurement noise at EoL.

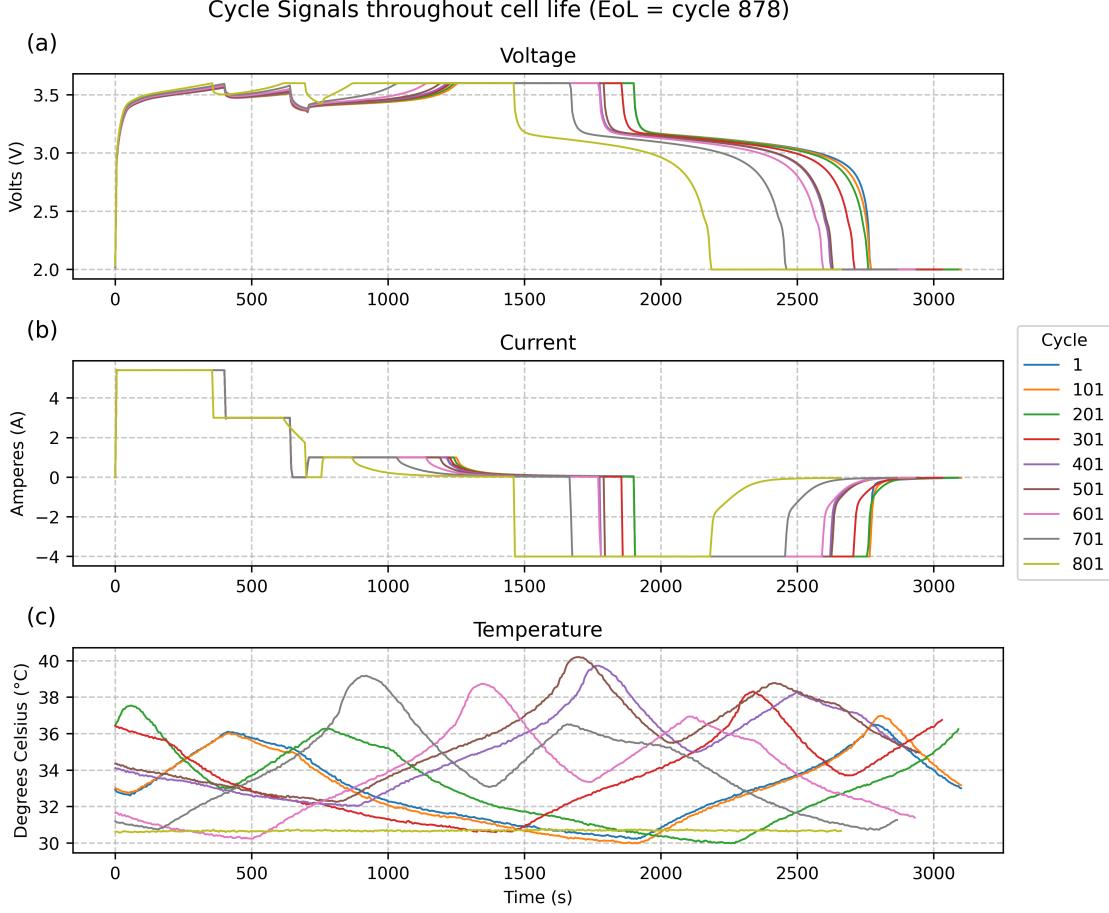


Figure 8: Evolution of voltage, current, and temperature signals across battery lifecycle.

To assess the impact of fast-charge protocols on cycle life, a correlation analysis was performed across the dataset’s range of one-step and two-step policies. Figure 9 shows a scatter plot of the mean charge C-rate versus cycle life. The mean C-rate, \bar{C} , is defined as the weighted average of the constant-current (CC) steps up to 80% SoC:

$$\bar{C} = \frac{\sum_i C_i \Delta \text{SOC}_i}{0.8}, \quad (4)$$

where C_i is the C-rate of the i -th step and ΔSOC_i is the fraction of state of charge (SOC) covered by that step. For example, a 4C(50%)-5C protocol yields $\bar{C} = (4 \times 0.5 + 5 \times 0.3)/0.8$. A Pearson correlation coefficient of $R = -0.61$ indicates a moderate negative correlation, confirming that higher charge currents, as applied across all batches, accelerate aging, consistent with the dataset’s high C-rate conditions.

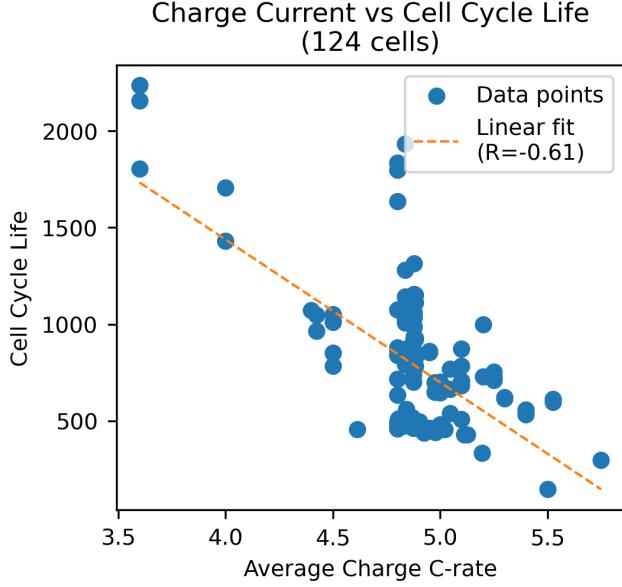


Figure 9: Scatter plot of average charge C-rate versus cycle life, with Pearson correlation coefficient $R = -0.61$.

Further analysis in fig. 10 evaluates the difference in charge "area" between the second and first constant-current (CC) steps. The goal of this analysis is to assess how different current intensities applied at different SoCs influence battery life. The area difference, ΔA , is defined as

$$\Delta A = C_2 \Delta \text{SOC}_2 - C_1 \Delta \text{SOC}_1, \quad (5)$$

where C_1 and C_2 are the C-rates of the first and second steps, respectively, and ΔSOC_1 and ΔSOC_2 are the corresponding fractions of state of charge (SOC) covered. For example, a 4C(50%)-3C protocol yields $\Delta A = 3 \times 0.3 - 4 \times 0.5$. A Pearson correlation coefficient of $R = 0.38$ indicates a weak positive correlation, suggesting that the order and duration of fast-charge steps have minimal impact on cycle life compared to the overall charge current magnitude.

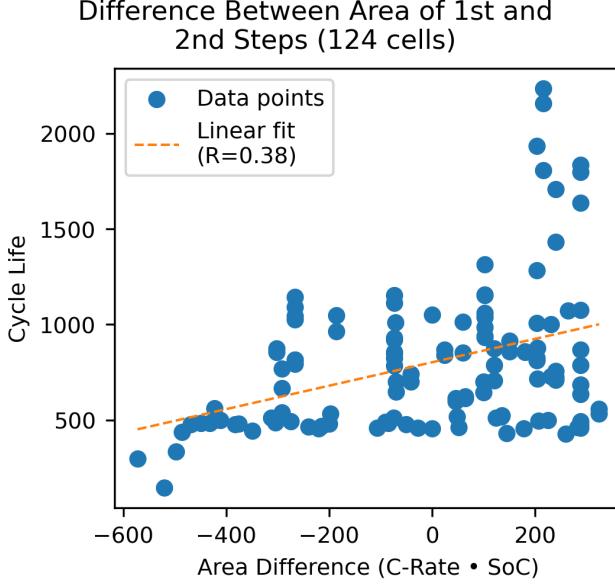


Figure 10: Differences in charge step area characteristics across battery degradation.

In summary, the exploratory data analysis, grounded in the dataset’s detailed measurements and experimental conditions, reveals key trends in LFP battery behavior under fast-charging protocols. Temperature strongly correlates with capacity exchange, supporting its use as an SoC indicator, though measurement reliability is limited by thermocouple inconsistencies. SoH degradation accelerates post-knee point, with cycle life averaging 700 cycles due to aggressive testing conditions, as noted in the dataset limitations. Aging significantly alters voltage and current profiles, with increased internal resistance driving earlier CV phases and shorter cycles at EoL. Correlation analyses confirm that higher charge currents reduce cycle life, while step order has minimal impact. These findings, despite the dataset’s constraints (e.g., single chemistry, controlled environment), provide a robust foundation for developing machine learning models to predict RUL and SoH, leveraging the observed relationships between measurable signals and degradation patterns.

6 Feature Extraction

Feature extraction is a pivotal step in estimating the State of Health (SoH) and Remaining Useful Life (RUL) of lithium-ion battery cells, particularly for cells unseen by the model during training. Since SoH and RUL evolve gradually on a cycle-by-cycle basis, analyzing raw signals sample-by-sample—especially in fast-charging scenarios—is impractical due to noise and high dimensionality. Instead, aggregating information from entire charge-discharge cycles provides a more meaningful representation of battery degradation. While one could concatenate all samples from a cycle and input them directly into a machine learning model, this risks introducing collinear features and spurious correlations from irrelevant signal variations. Alternative dimensionality reduction techniques, such as Principal Component Analysis (PCA), Partial Least Squares (PLS), or Kernel PCA from Schölkopf et al. [1998], could transform cycle samples into a lower-dimensional latent space in an un-

supervised manner, with the transformed data then used for supervised learning. Similarly, neural network auto-encoders, including those leveraging recurrent architectures or convolutional processing of time-series-turned-images, offer sophisticated transformations by encoding signals into compact embeddings. However, directly feeding raw signals into complex Neural Network (NN) models, while feasible, demands significant computational resources, hindering deployment on portable devices or electric vehicles. Consequently, the most widely adopted approach in the literature, which we employ here, involves calculating a small set of statistical metrics from voltage, current, and temperature signals per cycle. These metrics serve as features for training supervised machine learning models to estimate SoH and RUL, balancing simplicity and effectiveness.

6.1 Signals Preprocessing

Before computing statistical metrics, the voltage, current, and temperature signals require preprocessing to ensure consistency and reliability:

1. **Time gap removal:** Intermittent gaps in data collection were addressed by concatenating signals, as these discontinuities do not impact the cycle-wide statistical metrics targeted in this study.
2. **Invalid cycle filtering:** Cycles with durations below 100 seconds (approximately 1.7 minutes) or above 6000 seconds (approximately 1 hour 40 minutes) were excluded, deviating from the typical 40-to-60-minute range and likely indicating collection or annotation errors.
3. **Sampling rate standardization:** Variations in sampling rates across cycles, possibly due to hardware adjustments balancing data detail and volume, were normalized to 1Hz using linear interpolation. This ensures uniform representation of cycle segments in the statistical metrics.

6.2 Statistical Metrics

Post-preprocessing, the following statistical metrics were computed for each cycle's voltage, current, and temperature signals:

1. **Mean:** The average signal value over the cycle.
2. **Median:** The central value when the signal is sorted by magnitude.
3. **Standard Deviation (STD):** A measure of signal variability.
4. **Interquartile Range (IQR):** The range between the 25th and 75th percentiles, capturing the middle 50% spread.
5. **Kurtosis:** An indicator of the signal distribution's tailedness.

6. **Differential Entropy:** A measure of signal randomness, computed only for voltage due to infinite values arising in current and temperature calculations, possibly from near-constant segments or noise.

This yielded 16 features per cycle: 6 for voltage, 5 for current, and 5 for temperature, forming a compact yet informative representation of the battery's state.

6.3 Features Processing

Despite preprocessing, some cycles exhibited outlier feature values, evident in fig. 11. Such anomalies, common in real-world settings with unknown cells, can disrupt model training. For training data, we mitigated these issues while preserving them in test data to assess real-world robustness and avoid data leakage, once both processing steps applied relies on the feature values of the cycles around the cycle being processed. Two processing steps were applied:

1. **Spike removal:** For each feature per cell, a 10-cycle sliding window identified values below the 5th or above the 95th percentile, replacing them with the prior cycle's value to eliminate spikes.
2. **Smoothing:** A first-order Savitzky-Golay filter with a 10-cycle window was used to dampen cycle-to-cycle oscillations, which do not reflect the gradual evolution of SoH and RUL.

The processed features, shown in fig. 12, retain degradation trends while reducing noise and anomalies.

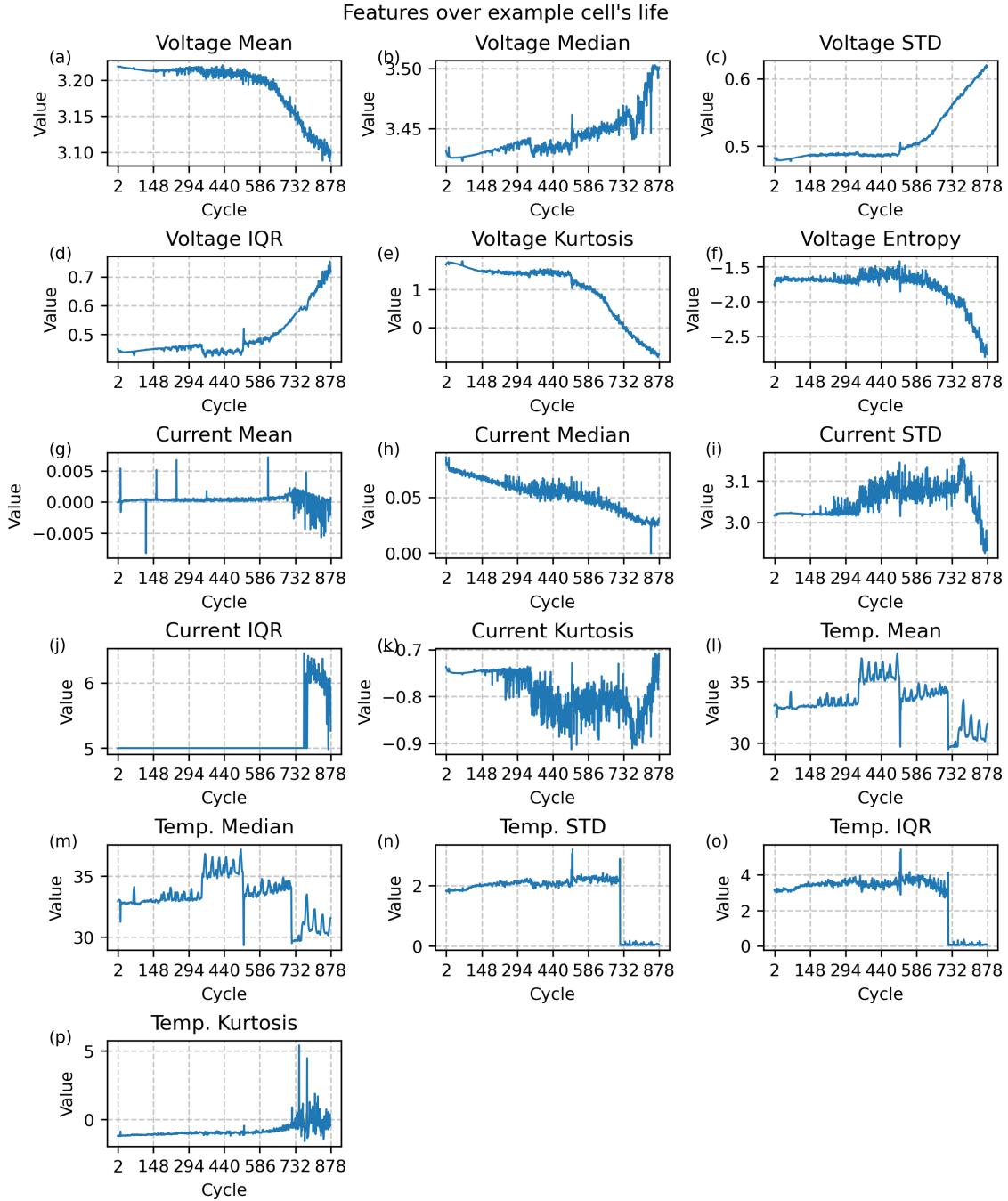


Figure 11: Distribution of raw extracted statistical features from voltage, current, and temperature signals, showing presence of outliers.

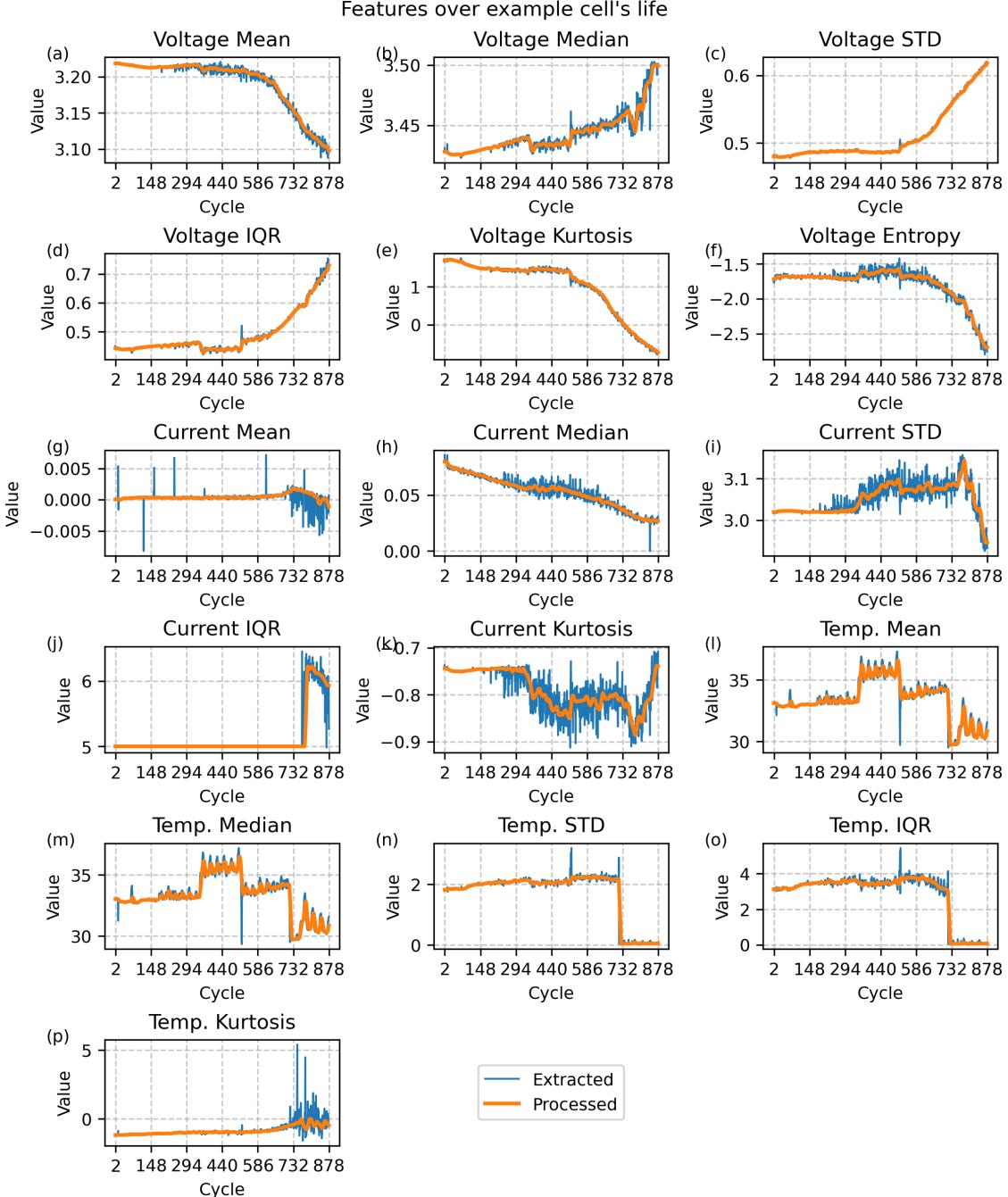


Figure 12: Distribution of extracted features after preprocessing (spike removal and smoothing) applied to eliminate outliers.

6.4 Features Analysis

The processed feature trajectories in fig. 12 reveal distinct aging patterns. Voltage mean decreases while median increases, likely due to heightened variability (STD and IQR) from rising internal resistance. Voltage kurtosis and differential entropy decline, possibly as extended constant voltage phases in aged cells yield more uniform signals. Current mean remains near zero, reflecting balanced

charge-discharge durations, while median decreases due to prolonged constant-voltage charging with lower currents. Current STD drops as aged cells handle high currents less variably, and IQR rises then falls, possibly tied to impedance shifts. Temperature features generally increase, driven by greater heat from internal resistance, though kurtosis shows a rise-then-fall pattern, meriting further study.

Pearson correlation coefficients between features and targets (SoH and RUL), shown in fig. 13, highlight strong relationships, especially among voltage features, making them key candidates for modeling.

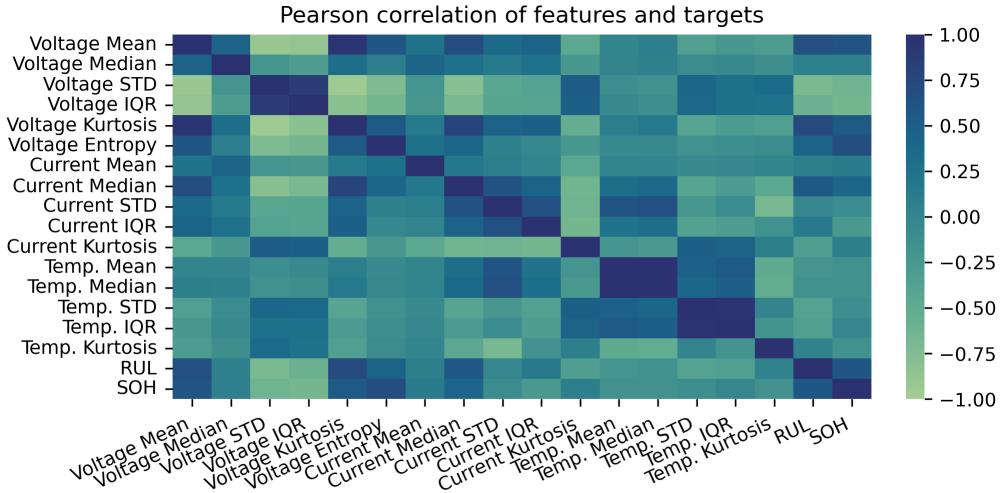


Figure 13: Pearson correlation heatmap between extracted features and target variables (SOH and RUL).

6.5 Feature Selection

Given the 16 extracted features, not all contribute equally to the estimation of SoH and RUL. Some may introduce noise or redundancy, particularly in models that lack built-in feature selection mechanisms. While bivariate correlation analysis can identify individual linear relationships with the target, it fails to capture more complex, multivariate interactions.

To address this, we applied a feature selection strategy based on feature importance derived from a Random Forest model. Specifically, we used a Random Forest with 10 estimators — a lightweight yet sufficiently robust configuration that balances stability and computational efficiency. This choice helps mitigate the sensitivity of single decision trees to data splits, providing more reliable importance estimates without incurring high computational cost.

We further enhanced the robustness of the feature importance ranking by using Group K-Fold cross-validation with 10 folds, where each group corresponds to an individual cell. The training set consists of 99 unique cells, allowing for a meaningful and well-balanced 10-fold split. This ensures that the training and validation sets remain group-independent, preventing data leakage between folds. Using 10 folds allows each cell to be evaluated multiple times across different validation splits, increasing the reliability of the aggregated feature importance scores.

In each fold, we trained the Random Forest on the training partition and recorded the feature importance. These importances were then summed across all folds and averaged to produce a final ranking. The top-ranked features were selected for subsequent modeling. This process helps retain the most informative degradation indicators while discarding less relevant or redundant ones, ultimately improving the predictive performance of downstream models by accounting for non-linear and multi-feature interactions.

6.6 Machine Learning Models

In this study, we evaluated four machine learning regression models on our dataset to predict the target variable: the Tweedie Regressor [Tweedie, 1984, Jørgensen, 1997], K-Nearest Neighbors (KNN) Regressor [Fix and Hodges, 1951, Hastie et al., 2009], Extra Trees Regressor [Geurts et al., 2006], and LightGBM Regressor [Ke et al., 2017]. Each model was trained and tested using standard cross-validation techniques to ensure robustness. In the following, we describe each model and its mathematical formulation.

6.6.1 Tweedie Regressor

The Tweedie Regressor is a generalized linear model tailored for response variables following a Tweedie distribution [Tweedie, 1984, Jørgensen, 1997], ideal for non-negative, continuous data that may include a mass at zero, because it generalizes distributions like Poisson, gamma, and compound Poisson–gamma. Although the SoH distribution does not strictly follow a Tweedie distribution, we include Tweedie regression as a comparative model to evaluate its performance relative to other regression approaches, exploring its flexibility and robustness under different data distributions. It models the expected value of the response variable y given features \mathbf{x} as:

$$\mu(\mathbf{x}) = \exp(\mathbf{x}^T \boldsymbol{\beta}),$$

where $\boldsymbol{\beta}$ is the coefficient vector, and the model optimizes the Tweedie deviance loss function:

$$\text{Deviance} = 2 \sum_{i=1}^n \left[\frac{y_i^{2-p}}{(1-p)(2-p)} - \frac{y_i \mu_i^{1-p}}{1-p} + \frac{\mu_i^{2-p}}{2-p} \right],$$

with p (where $1 < p < 2$) controlling the Tweedie distribution’s variance-power relationship. This loss balances the fit for zero and non-zero values, making it effective for mixed distributions.

6.6.2 K-Nearest Neighbors (KNN) Regressor

The KNN Regressor [Fix and Hodges, 1951, Hastie et al., 2009] predicts the target value for a test instance \mathbf{x} by averaging the target values of its k nearest neighbors in the feature space:

$$\hat{y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} y_i,$$

where $\mathcal{N}_k(\mathbf{x})$ is the set of k nearest neighbors based on a distance metric (e.g., Euclidean distance, $d(\mathbf{x}, \mathbf{x}_i) = \sqrt{(\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)}$). The prediction is a simple average, making KNN intuitive but sensitive to the choice of k and feature scaling.

6.6.3 Extra Trees Regressor

The Extra Trees Regressor [Geurts et al., 2006] is an ensemble method that constructs multiple decision trees with randomized splits and aggregates their predictions. For a test instance \mathbf{x} , the prediction is:

$$\hat{y}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T h_t(\mathbf{x}),$$

where T is the number of trees, and $h_t(\mathbf{x})$ is the prediction from the t -th tree. Each tree is built by selecting random features and split thresholds, reducing variance through averaging while maintaining computational efficiency compared to standard Random Forests.

6.6.4 LightGBM Regressor

The LightGBM Regressor [Ke et al., 2017] is a gradient boosting framework that constructs an additive model of weak learners (decision trees):

$$\hat{y}(\mathbf{x}) = \sum_{m=1}^M f_m(\mathbf{x}),$$

where $f_m(\mathbf{x})$ is the m -th tree, and M is the number of boosting iterations. The model minimizes a loss function (e.g., mean squared error, $L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$) by iteratively adding trees that reduce the gradient of the loss. LightGBM uses histogram-based splitting and leaf-wise growth for efficiency, making it scalable for large datasets.

7 Model Training and Optimization

This section describes the methodology for training and optimizing machine learning models to estimate the State of Health (SoH) and Remaining Useful Life (RUL) of lithium-ion batteries, utilizing the features extracted from the dataset detailed in prior sections. Machine learning models are mathematical frameworks that identify patterns in data to predict outcomes, here mapping cycle features to SoH or RUL. Separate models were trained for SoH and RUL to avoid the complexity of joint estimation, focusing on generalization to cells not included in the training data, a critical requirement for applications such as battery management systems in electric vehicles or energy storage.

7.1 Data Preparation

To ensure models generalize to unseen cells, the dataset of 124 cells was partitioned into training and test sets by randomly assigning entire cells, preventing any cell’s data from appearing in both sets. This approach mitigates data leakage, where models could exploit specific cell behaviors, leading to overly optimistic performance estimates. As presented in table 2, the training set comprises 99 cells (80,618 cycles), and the test set includes 25 cells (18,480 cycles), adhering to an approximate 80:20 split, a standard practice to balance training data availability with robust evaluation.

Partition	No. of Cells	Total Samples (Cycles)
Training	99	80618
Test	25	18480
Total	124	99098

Table 2: Division of cells and total samples (cycles) into training and test sets.

Features were standardized using Standard Scaling, transforming each feature to have a mean of zero and a Standard Deviation (STD) of one, based on statistics derived from the training set. Standardization ensures that features with different scales (e.g., voltage mean in volts vs. unitless kurtosis) contribute equally to model training, preventing bias toward larger-magnitude features. The scaling parameters from the training set were applied to the test set to maintain consistency, mirroring real-world scenarios where new data are processed using established parameters.

7.2 Model Training and Evaluation

Model training entails optimizing internal parameters (weights) to minimize prediction errors on the training data. In this study, models learn to map cycle features (e.g., voltage mean, temperature STD) to SoH or RUL, minimizing the Mean Squared Error (MSE), which quantifies the average squared difference between predicted and actual values, penalizing larger errors more heavily. Ground truth SoH and RUL values from all cycles of the 99 training cells were utilized, leveraging the dataset’s full-life data. In practical applications, full-life data may be unavailable, necessitating alternative approaches: (1) personalized models trained on early cycles of a specific cell for its future predictions, or (2) general models trained on early cycles from multiple cells, applicable to any cell at any stage. These methods are challenging, as early-cycle data may not capture aging patterns, a significant hurdle in battery research. Given the availability of full-life data, this study uses all cycles from training cells to assess feature and model performance for unseen cells, aligning with the objective of generalization.

Model performance was evaluated by generating predictions for the test set (25 unseen cells) and comparing them to ground truth SoH and RUL values using three metrics:

- **Mean Absolute Error (MAE):** The average absolute difference between predicted and

actual values, expressed in the target’s units (e.g., percentage for SoH, cycles for RUL).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

- **Root Mean Squared Error (RMSE):** The square root of MSE, emphasizing larger errors while maintaining the target’s units.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

- **Coefficient of Determination (R^2):** A metric from 0 to 1, indicating the proportion of variance in the target explained by the model, with 1 representing perfect predictions.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (8)$$

For inference, models require voltage, current, and temperature signals from a single full charge-discharge cycle, sampled at 1 Hz or higher, processed into the 15 features described previously.

7.3 Hyperparameter Optimization

Machine learning models rely on hyperparameters—configurable settings that define their structure or learning process, such as the number of trees in a Random Forest or the learning rate in a neural network. Unlike model weights, hyperparameters are set prior to training and typically determined empirically. To automate this process, the Tree-structured Parzen Estimator (TPE) [Bergstra et al. \[2011\]](#), a Bayesian optimization algorithm, was employed. TPE iteratively samples from a predefined hyperparameter space, balancing exploration of new configurations with exploitation of previously successful ones.

To balance predictive accuracy and model generalization, a custom cost function was adopted, combining validation error and the generalization gap. The objective to minimize is defined as:

$$\mathcal{L} = \text{RMSE}_{\text{val}} + \frac{|\text{RMSE}_{\text{train}} - \text{RMSE}_{\text{val}}|}{\text{RMSE}_{\text{val}}} \quad (9)$$

where RMSE_{val} is the root mean square error on the validation folds, as defined by eq. (7), and the second term, $\frac{|\text{RMSE}_{\text{train}} - \text{RMSE}_{\text{val}}|}{\text{RMSE}_{\text{val}}}$, represents the relative generalization gap between training and validation errors. This formulation penalizes both high validation errors and large discrepancies between training and validation performance, effectively discouraging overfitting while prioritizing predictive accuracy.

Trials were evaluated using 5-fold Group K-Fold cross-validation. The 99 training cells were partitioned into 5 non-overlapping groups (based on cell ID), ensuring that all cycles from a given cell remained within the same fold to prevent data leakage. The final loss score is computed as

the average of the defined loss across the 5 validation folds. fig. 14 below illustrates the 5-fold cross-validation process adopted.

To improve optimization efficiency and prevent unnecessary trials, an early stopping strategy was employed. If the cost does not improve for a fixed number of consecutive trials (patience = 10), the optimization process is halted. This ensures that exploration ceases when no meaningful improvement is observed.

All Cells					
Training Cells					Test Cells
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Training 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Training 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Training 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Training 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Training 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Figure 14: Diagram of data splitting in K-Fold cross-validation. On each training green folds are used to train, and the purple fold (validation set) is used to evaluate the trained model. Test cells are left out of the process.

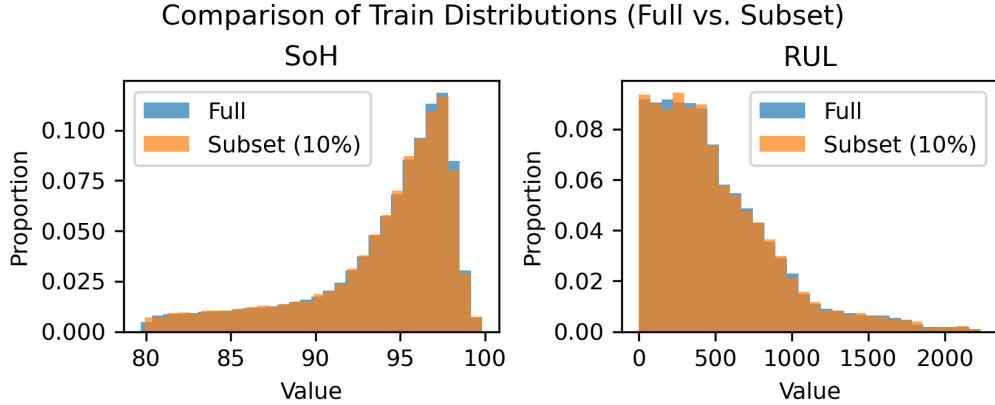


Figure 15: Distribution of SOH and RUL in the 10% optimization subset compared to the full dataset.

This methodology leverages full-life data from training cells to develop models that generalize to unseen cells, addressing the challenges of SoH and RUL estimation. Through standardized features, multi-object optimization and hyperparameter optimization via TPE and cross-validation,

the approach ensures robust predictions avoiding artificially inflated metrics due to overfitting.

8 Results: Optimization Phase

8.1 Hyperparameter Optimization Performance

The tables 3 and 4 below show the results of the optimization process for both State of Health (SoH) and Remaining Useful Life (RUL), respectively. It can be seen that the models produced good values of validation Root Mean Squared Error (RMSE), which is the average of the RMSE on each fold of cross validation, while maintaining low relative gaps which is a good indicative of non-overfitting models. For RUL models, although it was harder to keep the relative gap values low, which indicates models for that target with the features extracted on this work are more prone to overfitting the training data.

Table 3: Mean training and validation RMSE and relative gap across folds for machine learning models predicting State of Health (SOH) using 4 or 16 features, evaluated via 5-fold cross-validation.

Model	Features	RMSE		Relative Gap
		Train	Validation	
LGBMRegressor	16	0.52	0.94	0.44
	4	0.86	1.22	0.29
ExtraTreesRegressor	16	0.78	1.07	0.27
	4	1.23	1.40	0.12
KNeighborsRegressor	16	0.80	1.28	0.38
	4	1.15	1.50	0.24
TweedieRegressor	16	1.66	1.77	0.06
	4	2.03	2.06	0.01

Table 4: Mean training and validation RMSE and relative gap across folds for machine learning models predicting Remaining Useful Life (RUL) using 4 or 16 features, evaluated via 5-fold cross-validation.

Model	Features	RMSE		Relative Gap
		Train	Validation	
ExtraTreesRegressor	16	31.38	103.64	0.70
	4	52.33	110.00	0.52
LGBMRegressor	16	18.51	106.14	0.83
	4	22.12	109.09	0.80
KNeighborsRegressor	16	56.22	143.10	0.61
	4	51.19	112.09	0.54
TweedieRegressor	16	367.01	367.01	0.00
	4	337.10	340.03	0.01

The hyperparameter optimization process successfully identified optimal configurations for all machine learning models across both SoH and RUL prediction tasks. tables 3 and 4 present the cross-validation performance metrics obtained during the optimization phase, revealing distinct patterns in model behavior and generalization capabilities.

For SoH prediction (table 3), the LGBMRegressor demonstrated superior performance with the lowest validation RMSE of 0.94% when utilizing all 16 features, while maintaining a moderate relative gap of 0.44. This indicates a balanced trade-off between predictive accuracy and overfitting resistance. The ExtraTreesRegressor achieved comparable performance with a validation RMSE of 1.07% and notably lower relative gap of 0.27, suggesting better generalization properties. The TweedieRegressor, while exhibiting excellent generalization characteristics ($relativegap \leq 0.06$), showed substantially higher validation errors, indicating insufficient model complexity for capturing the underlying SoH patterns.

RUL prediction (table 4) proved more challenging, with all models exhibiting higher relative gaps compared to SoH estimation, suggesting increased susceptibility to overfitting. The ExtraTreesRegressor achieved the best validation performance with an RMSE of 103.64 cycles using 16 features, though with a concerning relative gap of 0.70. The LGBMRegressor, despite lower training error (18.51 cycles), showed an even larger relative gap of 0.83, indicating significant overfitting tendencies. The TweedieRegressor again demonstrated perfect generalization ($relativegap \approx 0.00$) but at the cost of substantially higher prediction errors.

8.2 Optimal Hyperparameters

The tables 5 and 6 show the optimal values of the hyperparameters for SoH and RUL, respectively.

The optimal hyperparameters (tables 5 and 6) reveal consistent patterns that provide insights

into the underlying data characteristics and model requirements. For tree-based methods, the optimal configurations favored relatively shallow trees ($\text{max_depth} \leq 10$) and small leaf sizes ($\text{min_samples_leaf} \leq 4$), suggesting that the battery aging patterns can be captured without excessive model complexity. The LGBMRegressor consistently selected low learning rates (0.007–0.11) paired with moderate numbers of estimators (400–661), indicating a preference for gradual learning to avoid overfitting.

Interestingly, the KNeighborsRegressor optimal configurations showed preference for relatively large neighborhood sizes ($n_{\text{neighbors}} = 21\text{--}35$) and uniform weighting, suggesting that local averaging over substantial regions of the feature space is beneficial for battery prognostics. The consistent selection of Manhattan distance ($p = 1$) over Euclidean distance ($p = 2$) for most configurations may indicate that the standardized features exhibit more meaningful relationships under L1 norm.

8.3 Impact of Feature Reduction

The transition from 16 to 4 features resulted in predictable but manageable performance degradation across all models. For SoH prediction, validation RMSE increased by 20–30% for tree-based methods (LGBMRegressor: 0.94% → 1.22%, ExtraTreesRegressor: 1.07% → 1.40%), while simultaneously reducing overfitting tendencies as evidenced by decreased relative gaps. This trade-off suggests that the four most important features capture the majority of SoH-relevant information, with additional features contributing primarily to model complexity rather than fundamental predictive power.

The feature reduction impact on RUL prediction was less pronounced, with validation RMSE changes typically within 10% (ExtraTreesRegressor: 103.64 → 110.00cycles). Notably, the reduced feature set generally improved generalization capabilities, with several models showing decreased relative gaps. This pattern indicates that RUL prediction may be more robust to feature dimensionality reduction, possibly due to the inherent noise in cycle-life estimation making additional features less beneficial.

9 Results: Test Set Evaluation

9.1 Test Metrics and Performance

The results of each model type using the optimal hyperparameter set are shown on tables 7 and 8 for SoH and RUL, respectively. The performance drop between using 16 and 4 features were relatively low indicating models with only the most important features identified by the Tree-based ensemble described on section 6 is feasible.

The overall performance of all models is satisfactory with RMSE values below 0.80% for SoH using a LGBMRegressor for 75% of the test cells and below 59.91 for RUL using an ExtraTreesRegressor for 75% of the test cells. R^2 values for 75% of the test cells were above 0.97 and 0.96 for SoH and RUL, respectively, indicating high accordance between the models' estimations and reference values.

Table 7: Performance metrics (RMSE and R²) for machine learning models predicting State of Health (SoH) using 4 or 16 features, evaluated via 5-fold cross-validation.

Model	Features	RMSE		R ²	
		Median	Q3	Median	Q1
LGBMRegressor	16	0.71	0.80	0.97	0.96
	4	0.96	1.25	0.95	0.90
KNeighborsRegressor	16	0.69	1.14	0.98	0.90
	4	1.04	1.32	0.94	0.88
ExtraTreesRegressor	16	0.83	0.97	0.97	0.94
	4	0.96	1.32	0.95	0.89
TweedieRegressor	16	1.36	2.33	0.88	0.70
	4	1.63	2.24	0.81	0.70

Table 8: Performance metrics (RMSE and R²) for machine learning models predicting Remaining Useful Life (RUL) using 4 or 16 features, evaluated via 5-fold cross-validation.

Model	Features	RMSE		R ²	
		Median	Q3	Median	Q1
ExtraTreesRegressor	16	41.63	59.91	0.96	0.90
	4	45.21	68.41	0.96	0.91
LGBMRegressor	16	49.46	76.74	0.95	0.89
	4	59.92	69.59	0.93	0.91
KNeighborsRegressor	16	54.17	95.94	0.93	0.79
	4	63.85	100.44	0.91	0.81
TweedieRegressor	16	63.52	102.68	0.90	0.71
	4	89.41	124.98	0.85	0.60

9.2 Model Validation and Prediction Analysis

figs. 16 to 19 below show in detail the scatter plots of all reference vs. estimated values of all test cells for the best model in each case: SoH and RUL and using all 16 features and only 4 features.

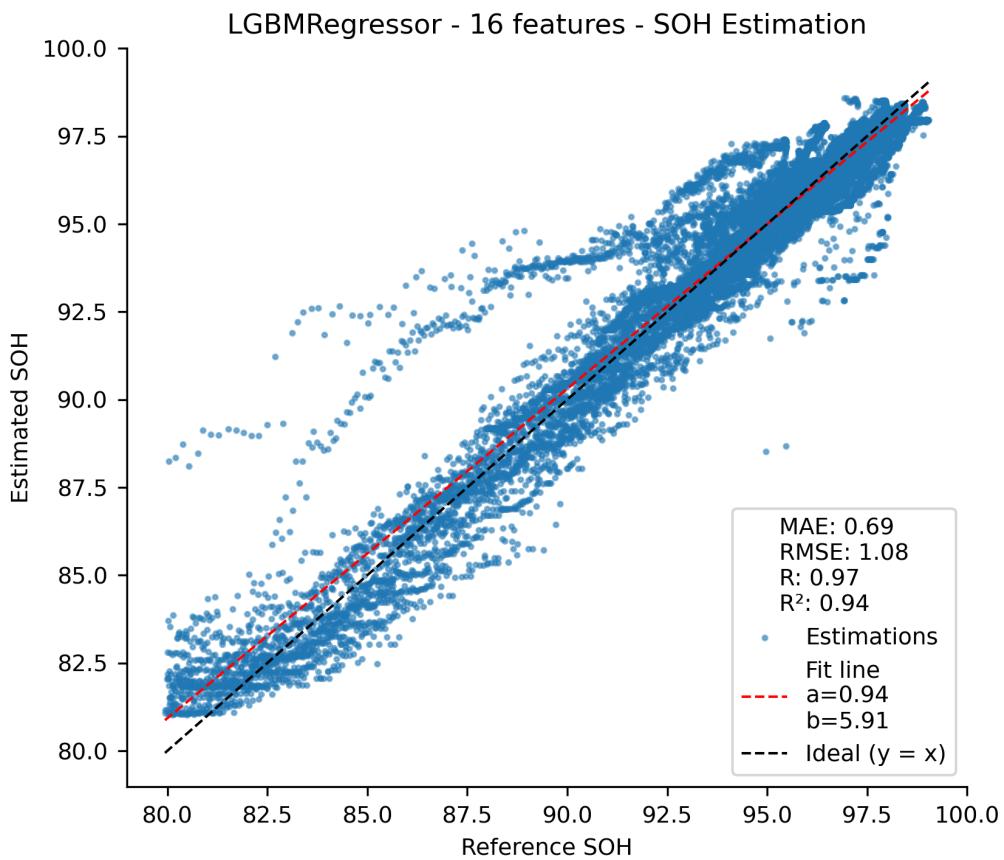


Figure 16: Predicted vs. true SoH for test cells using 16 features. The dashed line shows the ideal $y = x$ fit. Performance metrics (RMSE, MAE, R^2) are reported per panel.

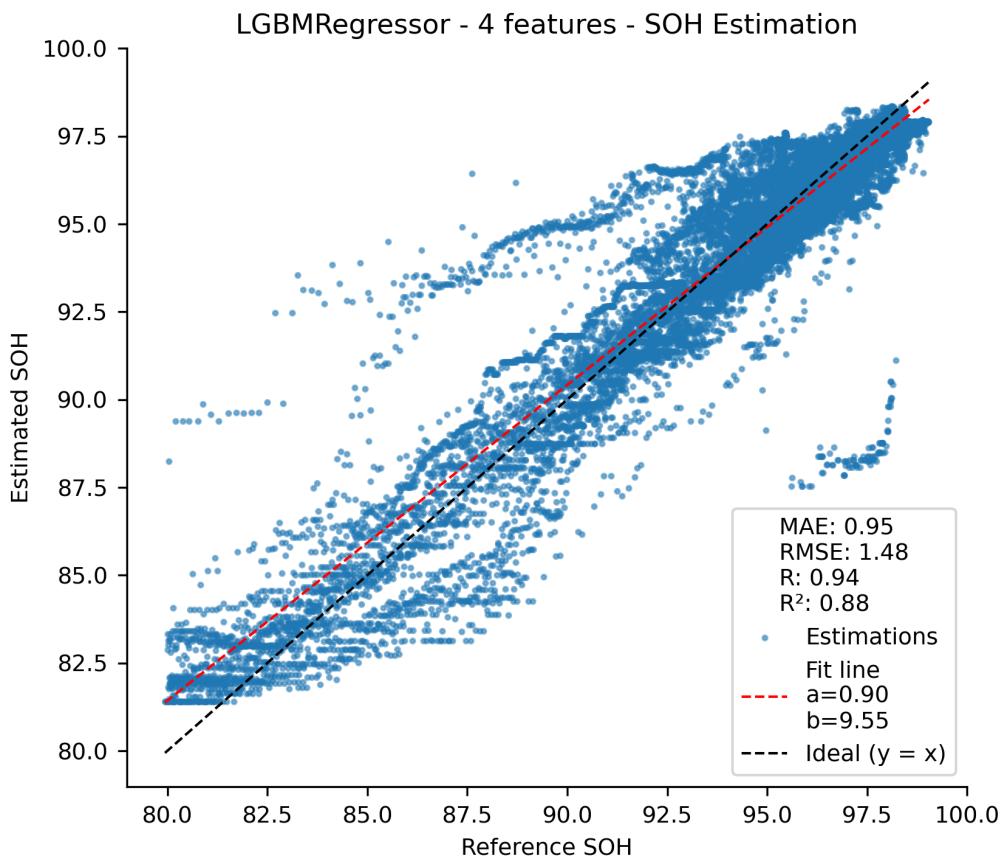


Figure 17: Predicted vs. true SoH for test cells using 4 features. The dashed line shows the ideal $y = x$ fit. Performance metrics (RMSE, MAE, R^2) are reported per panel.

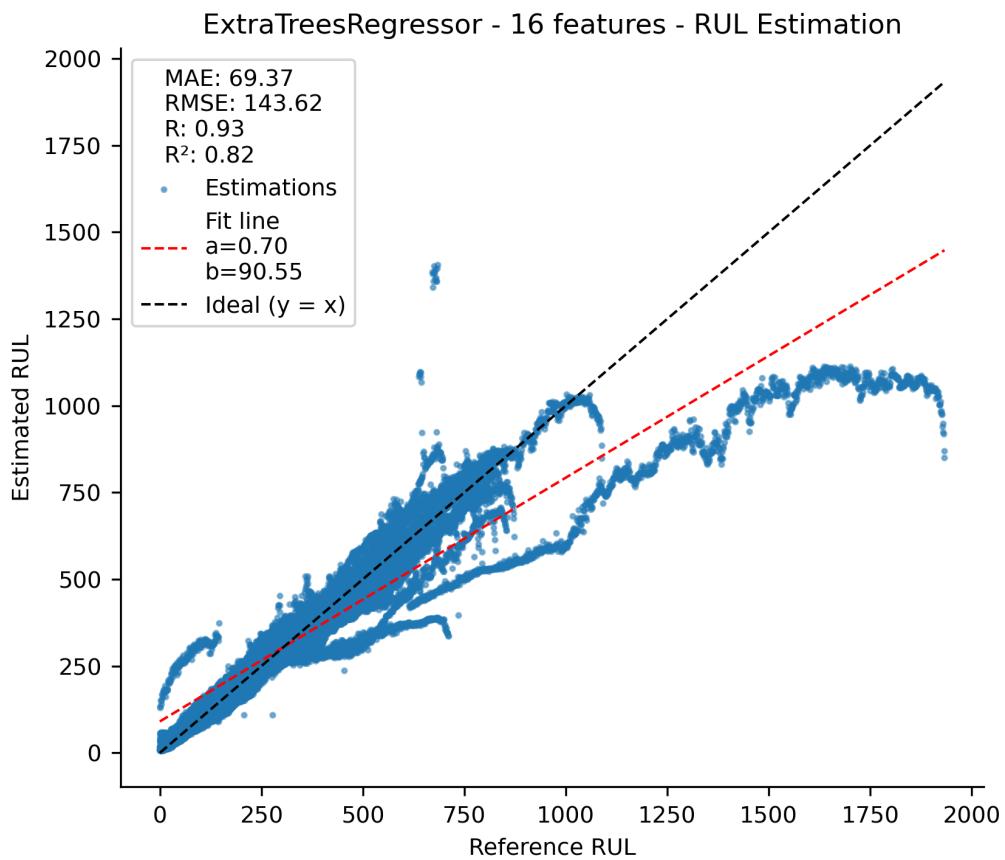


Figure 18: Predicted vs. true RUL for test cells using 16 features. The dashed line shows the ideal $y = x$ fit. Performance metrics (RMSE, MAE, R^2) are reported per panel.

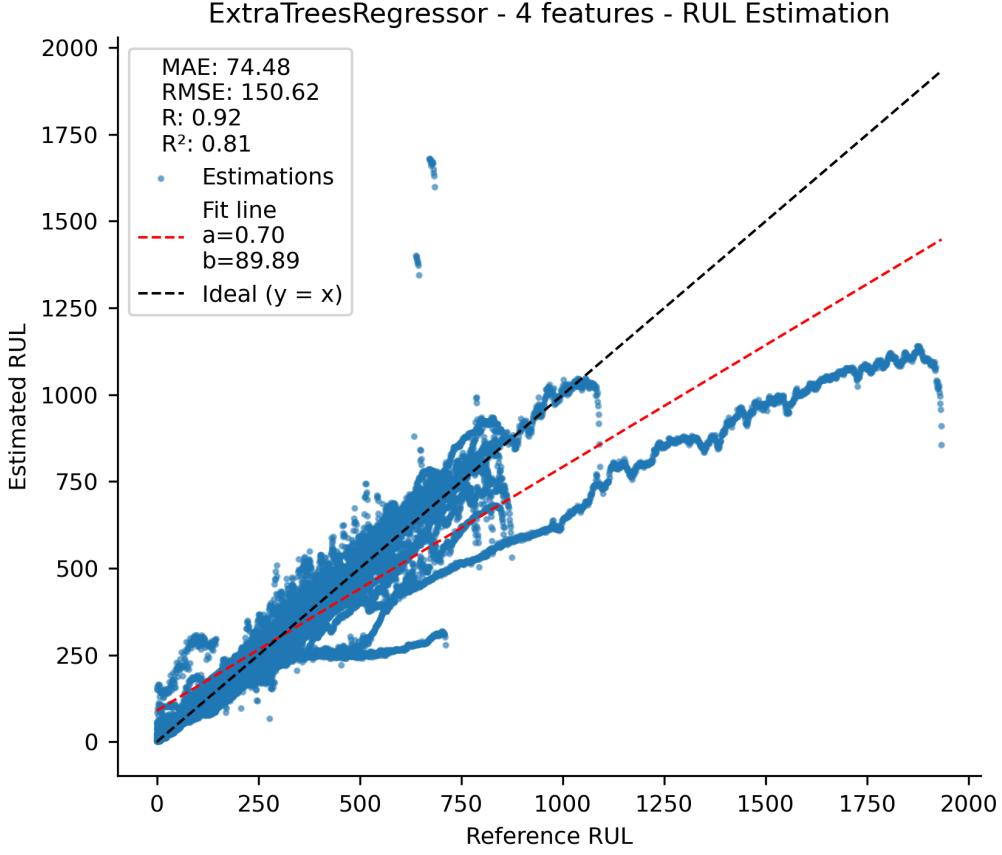


Figure 19: Predicted vs. true RUL for test cells using 4 features. The dashed line shows the ideal $y = x$ fit. Performance metrics (RMSE, MAE, R^2) are reported per panel.

It can be confirmed that the model works really well for most data while some points are more distant from the ideal fit line. For SoH models, predictions show both under- and over-estimation patterns with a balanced distribution around the ideal fit line. In contrast, RUL models exhibit a tendency toward under-estimation, particularly evident in the high-cycle region where one test cell with approximately 2000 cycles consistently deviates from the ideal regression line. This divergence likely stems from the scarcity of such long-lived cells in the training dataset, highlighting a limitation in model generalization to extreme cycle-life scenarios.

For the RUL models with 16 and 4 features it can be noticed that a similar set of points diverge from the ideal regression line. We identified that these points correspond to the same cell of the test set. From the plot it can be seen that this cell is the only one with almost 2000 cycles cycle life, and this divergence of the model might be happening because data with RUL values that high are more scarce on the training set as well due to the fact few cells cycle that long on the dataset used.

9.3 Test Set Performance and Model Validation

The test set results (tables 7 and 8) demonstrate that the optimized models successfully generalize to unseen cells, with performance metrics closely matching the cross-validation estimates. For SoH prediction, the LGBMRegressor achieved an RMSE below 0.80% for 75% of test cells, with Coefficient of Determination (R^2) values above 0.97, indicating high accordance between model estimations and reference values. Similarly, for RUL prediction, the ExtraTreesRegressor maintained RMSE values below 59.91 cycles for 75% of test cells, with R^2 values above 0.96.

9.4 Model-Specific Performance Characteristics

The superior performance of tree-based ensemble methods (LGBMRegressor and ExtraTreesRegressor) aligns with their inherent ability to capture non-linear relationships and feature interactions without explicit modeling. Battery aging involves complex, non-linear degradation mechanisms that these methods can naturally accommodate. The gradient boosting approach of LGBMRegressor appears particularly well-suited for SoH estimation, where sequential refinement of predictions can effectively model the gradual capacity fade patterns.

The consistently poor performance of TweedieRegressor, despite its excellent generalization properties, suggests that linear models—even with flexible error distributions—are insufficient for capturing the complexity of battery degradation patterns encoded in the extracted features. This finding reinforces the necessity of non-linear modeling approaches for battery prognostics applications.

9.5 Practical Implications and Deployment Considerations

From a practical battery management perspective, the results demonstrate that effective SoH and RUL estimation can be achieved using a minimal feature set derived from single charge-discharge cycles sampled at 1 Hz or higher. The 4-feature models, while showing modest performance reduction, offer significant computational advantages for embedded applications where processing power and memory are constrained.

However, a critical limitation for real-world deployment concerns data availability. The feature extraction methodology requires complete constant current constant voltage (CCCV) charge-discharge cycle data, which may not be readily available during normal battery operation in applications such as electric vehicles or grid energy storage systems. Such complete cycling data would typically only be obtainable during dedicated diagnostic procedures with the load disconnected from the battery, limiting the frequency of prognostic updates and potentially compromising the timeliness of health assessments. This constraint necessitates careful consideration of diagnostic scheduling and may require development of alternative feature extraction methods that can operate on partial cycle data or operational load profiles.

Despite this limitation, the observed performance characteristics suggest different deployment strategies for SoH and RUL estimation. SoH models demonstrate robust performance across the

full range of battery conditions, making them suitable for periodic diagnostic monitoring applications. RUL models, while showing excellent performance for typical battery lifespans, may require additional training data or specialized handling for batteries exhibiting exceptional longevity, and their deployment would be similarly constrained by the need for complete cycle data.

10 Conclusion

This work investigated machine learning techniques for estimating the State of Health (SoH) and Remaining Useful Life (RUL) of Lithium-ion Batteries (LiBs) using the large-scale dataset provided by Severson et al. [Severson et al. \[2019\]](#). The dataset, comprising 124 commercial LiFePO₄/graphite cells cycled under various fast-charging conditions, enabled a systematic exploration of degradation patterns and their relationship with measurable signals such as voltage, current, and temperature. Through comprehensive preprocessing and the extraction of statistical features, we established a compact yet informative representation of cell behavior that balances interpretability and computational efficiency.

The exploratory analysis confirmed well-known degradation phenomena, including the characteristic knee point in SoH evolution, accelerated capacity fade under high charging rates, and the influence of internal resistance growth on voltage and temperature profiles. These findings provide a robust foundation for supervised learning models by linking observable features to battery aging mechanisms.

Nevertheless, the dataset’s constraints must be acknowledged. The exclusive focus on one chemistry (LFP/graphite), highly controlled laboratory conditions, and continuous full charge–discharge cycles limit the direct applicability of the models to real-world scenarios such as electric vehicles or grid storage, where cycling is partial, irregular, and influenced by environmental variations. The inconsistencies in temperature measurements and experimental irregularities further highlight the need for robustness in model design.

Despite these limitations, the results demonstrate that statistical feature representations of single cycles can capture essential information for battery prognosis. This finding points toward a practical diagnostic framework in which SoH and RUL can be estimated from periodic diagnostic cycles, without requiring continuous monitoring of long cycle histories. Future work will extend this approach to heterogeneous datasets, incorporate physics-informed features to enhance generalization across chemistries and conditions, and evaluate the deployment of the proposed methodology in realistic battery management system environments.

A limitation of the present approach is the requirement of a dedicated full diagnostic cycle, which may not be available in continuous operation. Future work will evaluate the feasibility of periodic diagnostics (e.g., monthly) and the extension to multiple chemistries such as NMC and NCA.

List of Acronyms

DoD Depth of Discharge

EoL End of Life

IQR Interquartile Range

LFP Lithium Iron Phosphate

LiBs Lithium-ion Batteries

MAE Mean Absolute Error (The average of absolute differences between predicted and actual values, used to assess prediction accuracy.)

ML Machine Learning

MSE Mean Squared Error (The average of squared differences between predicted and actual values, used to measure model accuracy.)

NN Neural Network

PCA Principal Component Analysis

PLS Partial Least Squares

R² Coefficient of Determination (A metric indicating how well a model explains the variability of the target variable, ranging from 0 to 1.)

RMSE Root Mean Squared Error (The square root of the mean squared error, providing a measure of prediction error in the same units as the target.)

RUL Remaining Useful Life

SoC State of Charge

SoH State of Health

STD Standard Deviation

XAI Explainable AI

References

Yunhong Che, Yusheng Zheng, Yue Wu, Xin Sui, Pallavi Bharadwaj, Daniel-Ioan Stroe, Yalian Yang, Xiaosong Hu, and Remus Teodorescu. Data efficient health prognostic for batteries based on sequential information-driven probabilistic neural network. *Appl. Energy*, 323(119663):119663, October 2022.

Kai Lv, Zhiqiang Ma, Caijilahu Bao, and Guangchen Liu. Indirect prediction of lithium-ion battery RUL based on CEEMDAN and CNN-BiGRU. *Energies*, 17(7):1704, April 2024.

Sadiqa Jafari and Yung Cheol Byun. Accurate remaining useful life estimation of lithium-ion batteries in electric vehicles based on a measurable feature-based approach with explainable AI. *J. Supercomput.*, 80(4):4707–4732, March 2024.

Wei Liu and Jiashen Teh. Remaining useful life prediction of lithium-ion batteries based on an incremental internal resistance aging model and a gated recurrent unit neural network. *Energy (Oxf.)*, 333(137527):137527, October 2025.

Kristen Severson, Peter Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael Chen, Muratahan Aykol, Patrick Herring, Dimitrios Fraggedakis, Martin Bazant, Stephen Harris, William Chueh, and Richard Braatz. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4:383–391, 03 2019. doi: 10.1038/s41560-019-0356-8.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Maurice C.K. Tweedie. An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions*, pages 579–604, 1984.

Bent Jørgensen. *The Theory of Dispersion Models*. CRC Press, 1997.

E. Fix and J.L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. Technical report, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 2006.

Guolin Ke, Qiwei Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qi Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyperparameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cf12577bc2619bc635690-Paper.pdf.

Table 5: Optimal hyperparameters for SOH prediction

Model	Num. Features	Hyper Param	Optimal Value
ExtraTreesRegressor	16	n_estimators	87
		max_depth	7
		min_samples_split	3
		min_samples_leaf	1
		criterion	squared_error
	4	n_estimators	87
		max_depth	7
		min_samples_split	3
		min_samples_leaf	1
		criterion	squared_error
KNeighborsRegressor	16	n_neighbors	35
		weights	uniform
		p	1
	4	n_neighbors	35
		weights	uniform
		p	1
	16	n_estimators	661
		learning_rate	0.007
		num_leaves	18
		max_depth	5
		min_child_samples	13
		subsample	0.865
		colsample_bytree	0.819
		reg_alpha	0.125
		reg_lambda	0
		n_estimators	661
LGBMRegressor	4	learning_rate	0.007
		num_leaves	18
		max_depth	5
		min_child_samples	13
		subsample	0.865
		colsample_bytree	0.819
		reg_alpha	0.125
		reg_lambda	0
		n_estimators	661
		learning_rate	0.007
TweedieRegressor	16	num_leaves	18
		max_depth	5
		min_child_samples	13
		subsample	0.865
	4	colsample_bytree	0.819
		reg_alpha	0.125
		reg_lambda	0
		power	1.378
GammaRegressor	16	alpha	0
		fit_intercept	True
		tol	0
		power	1.378
GammaRegressor	4	alpha	0
		fit_intercept	True
		tol	0
		power	1.378

Table 6: Optimal hyperparameters for RUL prediction

Model	Num. Features	Hyper Param	Optimal Value
ExtraTreesRegressor	16	n_estimators	62
		max_depth	10
		min_samples_split	4
		min_samples_leaf	4
	4	criterion	squared_error
		n_estimators	110
		max_depth	10
		min_samples_split	9
KNeighborsRegressor	16	min_samples_leaf	2
		criterion	squared_error
		n_neighbors	35
	4	weights	uniform
		p	1
		n_neighbors	21
	4	weights	uniform
		p	2
LGBMRegressor	16	n_estimators	563
		learning_rate	0.029
		num_leaves	17
		max_depth	7
		min_child_samples	9
		subsample	0.533
		colsample_bytree	0.974
		reg_alpha	0.531
	4	reg_lambda	0.029
		n_estimators	400
		learning_rate	0.112
		num_leaves	24
TweedieRegressor	16	max_depth	9
		min_child_samples	15
		subsample	0.777
		colsample_bytree	0.675
	4	reg_alpha	0.001
		reg_lambda	0
		power	1.378
		alpha	0
	16	fit_intercept	True
		tol	0
	4	power	1.761
		alpha	0
	4	fit_intercept	True
		tol	0