RUL Prediction: What and How

Analysis Report

September 15, 2025

1 What We're Predicting

1.1 Target Variable: Remaining Useful Life (RUL)

- **Definition**: The number of charge-discharge cycles remaining before a battery reaches End-of-Life (EOL)
- **Type**: Integer count of cycles
- **Example**: If a battery has 500 total cycles and we're at cycle 100, RUL = 400 cycles

1.2 EOL Definition

- **Capacity-based threshold**: When discharge capacity drops to 80% (or 90% for some datasets) of nominal capacity
- **Mathematical condition**: $Q_d \leq \text{nominal\_capacity} \times \text{eol\_soh}$
- **Example**: Battery with 1.1 Ah nominal capacity reaches EOL when discharge capacity $\leq$ 0.88 Ah (80% threshold)

2 How We're Predicting It

2.1 Feature Extraction Phase

We extract features from **early cycles only** (typically cycles 2-99):

2.1.1 Statistical Features (DischargeModelFeatureExtractor)

Listing 1: Statistical Features from Severson Methodology

```
# From Severson methodology - uses Qdlin (voltage-capacity curves)
features = [
    'Minimum',              # Log10 of min absolute value of  Qdlin
    'Variance',             # Log10 of variance of  Qdlin
    'Skewness',             # Log10 of absolute skewness of  Qdlin
    'Kurtosis',             # Log10 of kurtosis of  Qdlin
    'Early discharge capacity',  # Capacity at early cycle
    'Difference between max and early discharge capacity'  # Capacity fade
]
```

2.1.2 Matrix Features (VoltageCapacityMatrixFeatureExtractor)

Listing 2: Matrix Features

```
# 2D matrix: [num_cycles   1000] where each row is  Qdlin  for a cycle
#  Qdlin  = Qdlin(late_cycle) - Qdlin(early_cycle)
# This captures the evolution of voltage-capacity curves over time
```

2.2 Label Generation Phase

For each battery, we calculate the **actual RUL** by:

Listing 3: RUL Calculation Algorithm

```
def calculate_rul(battery_data):
    label = 1  # Start counting from cycle 1
    for cycle in battery_data.cycle_data:
        label += 1
        Qd = max(cycle.discharge_capacity_in_Ah)  # Max discharge capacity
        if Qd <= nominal_capacity * eol_soh:      # EOL condition
            break  # Found EOL cycle
    return label  # This is the RUL
```

2.3 Model Training Phase

We train models to learn the mapping:

$$\text{Early Cycle Features} \rightarrow \text{RUL (number of remaining cycles)} \tag{1}$$

- **Input**: Features from cycles 2-99
- **Output**: Predicted RUL (integer number of cycles)

2.4 Prediction Process

2.4.1 Step 1: Feature Extraction

- Extract features from the first ∼100 cycles of a battery's life
- Use either statistical features or matrix features depending on model type

2.4.2 Step 2: Model Prediction

- Feed features into trained model
- Model outputs predicted RUL value

2.4.3 Step 3: Interpretation

- Predicted RUL tells us how many more cycles the battery will last
- Example: If RUL = 500, the battery will last 500 more cycles before reaching 80% capacity

3 Concrete Example

3.1 Scenario

- **Battery**: 1.1 Ah nominal capacity
- **EOL threshold**: 80% = 0.88 Ah
- **Current status**: We have data from cycles 1-100
- **Question**: How many more cycles will this battery last?

3.2 Process

3.2.1 1. Feature Extraction (from cycles 2-99)

Listing 4: Feature Extraction Example

```
# Statistical features
variance = log10(var( Qdlin )) # e.g., 2.3
minimum = log10(min(abs( Qdlin ))) # e.g., 1.8
early_capacity = 1.05 # Ah from cycle 2
# ... other features
```

3.2.2 2. Model Prediction

Listing 5: Model Prediction

```
features = [2.3, 1.8, 1.05, ...] # 6 or 8 features
predicted_rul = model.predict(features) # e.g., 400 cycles
```

3.2.3 3. Interpretation

- **Prediction**: This battery will last 400 more cycles
- **Total life**: 100 (observed) + 400 (predicted) = 500 cycles total
- **EOL cycle**: Cycle 500 (when capacity drops to 0.88 Ah)

4 Key Points

4.1 What Makes This Challenging

1. **Early prediction**: We predict total remaining life using only early cycle data
2. **Long-term forecasting**: Predicting hundreds or thousands of cycles ahead
3. **Non-linear degradation**: Battery degradation is complex and non-linear
4. **Dataset variability**: Different battery types have different degradation patterns

4.2 What We're NOT Predicting

- **Time-based RUL**: We predict cycles, not hours/days
- **Voltage-based EOL**: We use capacity, not voltage thresholds
- **Instantaneous degradation**: We predict total remaining life, not next-cycle degradation

4.3 Practical Use Case

This is like predicting how many more miles a car will last based on its performance in the first 10,000 miles. The model learns patterns from early degradation behavior to forecast the total remaining useful life.

The prediction enables:

- **Battery management systems** to plan maintenance
- **Quality control** to identify defective batteries early
- **Research** to understand degradation mechanisms
- **Manufacturing** to optimize battery design

5 Mathematical Formulation

5.1 Problem Definition

Given a battery with cycling data from cycles 1 to T_{obs}, predict the remaining useful life RUL such that:

$$RUL = T_{EOL} - T_{obs} \tag{2}$$

where T_{EOL} is the cycle when the battery reaches End-of-Life.

5.2 EOL Condition

$$Q_d(T_{EOL}) \leq Q_{nominal} \times \text{eol\_soh} \tag{3}$$

where:

- $Q_d(T_{EOL})$ is the maximum discharge capacity at cycle T_{EOL}
- $Q_{nominal}$ is the nominal capacity of the battery
- eol_soh is the End-of-Life State of Health threshold (typically 0.8 or 0.9)

5.3 Prediction Model

$$R\hat{U}L = f(\mathbf{X}_{1:T_{obs}}) \tag{4}$$

where:

- $\mathbf{X}_{1:T_{obs}}$ are the features extracted from cycles 1 to T_{obs}
- $f(\cdot)$ is the trained prediction model
- $R\hat{U}L$ is the predicted remaining useful life

6 Summary

The RUL prediction system enables early forecasting of battery degradation by learning patterns from early cycling data and predicting the total remaining useful life in terms of charge-discharge cycles. This approach is crucial for battery management, quality control, and research applications.