RUL Label Annotator Analysis for BatteryML

Analysis Report

September 15, 2025

1 RUL Definition and Purpose

RUL (Remaining Useful Life) represents the **number of cycles remaining** before a battery reaches its End-of-Life (EOL) condition. It's a **regression target** that predicts how many more charge-discharge cycles a battery can undergo before it degrades to a specified capacity threshold.

2 RUL Calculation Algorithm

2.1 Core Algorithm

Listing 1: RUL Calculation Logic

```
def process_cell(self, cell_data: BatteryData) -> torch.Tensor:
    label, found_eol = 1, False
    for cycle in cell_data.cycle_data:
        label += 1
        Qd = max(cycle.discharge_capacity_in_Ah)
        if Qd <= cell_data.nominal_capacity_in_Ah * self.eol_soh:
            found_eol = True
            break

    if not found_eol:
        label = label + 1 if self.pad_eol else float('nan')

    if label <= self.min_rul_limit:
        label = float('nan')

    return torch.tensor(label)
```

2.2 Step-by-Step Process

1. **Initialize**: Start with `label = 1` (counting from cycle 1)
2. **Iterate through cycles**: For each cycle in the battery data:
   - Increment label counter (`label += 1`)
   - Extract maximum discharge capacity: `Qd = max(cycle.discharge_capacity_in_Ah)`
   - Check EOL condition: `Qd <= nominal_capacity * eol_soh`
3. **EOL Detection**: If capacity drops below threshold, mark `found_eol = True` and break
4. **Handle incomplete data**: If EOL not reached, add padding or mark as NaN
5. **Quality filtering**: Remove batteries with RUL $\leq$ `min_rul_limit`

3 Key Parameters

3.1 EOL SOH Threshold (`eol_soh`)

Table 1: EOL SOH Threshold by Dataset

Dataset	**Threshold**	**Capacity %**
SNL, CRUSH	0.9	90% of nominal capacity
Others (HUST, MATR, etc.)	0.8	80% of nominal capacity

3.2 Padding Strategy (`pad_eol`)

- **Default**: `True`
- **Behavior**:
  - If EOL not reached: `label = label + 1` (add one more cycle)
  - If `False`: `label = NaN` (mark as invalid)

3.3 Minimum RUL Limit (`min_rul_limit`)

- **Default**: 100.0 cycles
- **Purpose**: Filters out batteries with very short life (likely defective)
- **Effect**: RUL $\leq$ 100 cycles $\rightarrow$ `NaN` (excluded from training)

4 What Exactly Do We Predict?

4.1 Target Variable Characteristics

- **Type**: Continuous integer (number of cycles)
- **Range**: Typically 100-2000+ cycles (dataset dependent)
- **Unit**: Charge-discharge cycles
- **Interpretation**: "This battery has X cycles left before reaching EOL"

4.2 Prediction Task

- **Input**: Battery features extracted from early cycles (typically cycles 2-99)
- **Output**: Predicted number of remaining cycles until 80% capacity degradation
- **Problem Type**: Regression (not classification)

5 Dataset-Specific RUL Characteristics

5.1 RUL Distribution by Dataset

5.2 EOL Threshold Variations

- **SNL, CRUSH**: 90% capacity (stricter EOL)
- **Others**: 80% capacity (standard EOL)

Table 2: RUL Distribution Across Datasets

Dataset	Mean RUL	Std RUL	Range
CALCE	566±106	106	∼400-700
MATR	823±368	368	∼400-1200
HUST	1899±389	389	∼1500-2300
HNEI	248±15	15	∼230-260
RWTH	658±64	64	∼600-700
SNL	1256±1321	1321	∼0-2500+
UL_PUR	209±50	50	∼150-250

6 Label Transformation Pipeline

6.1 Sequential Transformations

1. **Log Scale Transformation**: log(RUL) - Compresses the wide range of RUL values
2. **Z-Score Normalization**: $\frac{\log(\text{RUL})-\mu}{\sigma}$ - Standardizes for model training

6.2 Mathematical Flow

$$\text{Raw RUL} \rightarrow \log(\text{RUL}) \rightarrow \frac{\log(\text{RUL}) - \mu}{\sigma} \rightarrow \text{Model Input} \tag{1}$$

7 Quality Control and Data Filtering

7.1 Exclusion Criteria

1. **Short-lived batteries**: RUL $\leq$ 100 cycles (likely defective)
2. **Incomplete data**: Batteries that don't reach EOL (if `pad_eol=False`)
3. **Invalid capacity data**: Cycles with missing or invalid discharge capacity

7.2 Data Validation

- **Capacity check**: `Qd = max(cycle.discharge_capacity_in_Ah)` must be valid
- **EOL detection**: Must find cycle where capacity drops below threshold
- **Range validation**: RUL must be within reasonable bounds

8 Practical Example

8.1 Scenario

- **Battery**: Nominal capacity = 1.1 Ah
- **EOL threshold**: 80% = 0.88 Ah
- **Cycle data**: Cycles 1-500 with decreasing capacity

8.2 RUL Calculation

Listing 2: RUL Calculation Example

```
Cycle 1: Qd = 1.05 Ah > 0.88 Ah    Continue
Cycle 2: Qd = 1.02 Ah > 0.88 Ah    Continue
...
Cycle 150: Qd = 0.87 Ah    0.88 Ah    EOL reached!
RUL = 150 cycles
```

8.3 After Transformations

$$\text{Raw RUL} : 150 \tag{2}$$

$$\log(\text{RUL}) : \log(150) \approx 5.01 \tag{3}$$

$$\text{Normalized} : \frac{5.01 - \mu}{\sigma} \rightarrow \text{Model input} \tag{4}$$

9 Key Insights

1. **RUL is cycle-based**: Not time-based, but cycle-count based
2. **Capacity-driven EOL**: Based on discharge capacity degradation, not voltage
3. **Early prediction**: Uses features from early cycles to predict total remaining life
4. **Quality filtering**: Excludes defective or incomplete batteries
5. **Log transformation**: Handles the wide range of RUL values across datasets
6. **Dataset adaptation**: Different EOL thresholds for different battery types

10 Summary

The RUL prediction task is essentially asking: **"Given what I observe in the first ~100 cycles of this battery's life, how many more cycles will it last before degrading to 80% of its original capacity?"**

This approach enables early prediction of battery degradation, which is crucial for:

- Battery management systems
- Predictive maintenance
- Quality control in manufacturing
- Research and development of battery technologies