Feature Extraction Analysis for BatteryML

Analysis Report

September 15, 2025

1 Feature Extractor Types

There are **4 main feature extractors** in the system:

1. `DischargeModelFeatureExtractor` - Extracts 6 statistical features
2. `FullModelFeatureExtractor` - Extracts 8 comprehensive features
3. `VarianceModelFeatureExtractor` - Extracts only 1 feature (variance)
4. `VoltageCapacityMatrixFeatureExtractor` - Extracts 2D matrix features

2 Feature Extraction Logic

2.1 DischargeModelFeatureExtractor

Features Extracted (6):

- `Minimum`: Log10 of minimum absolute value of ΔQdlin
- `Variance`: Log10 of variance of ΔQdlin
- `Skewness`: Log10 of absolute skewness of ΔQdlin
- `Kurtosis`: Log10 of kurtosis of ΔQdlin
- `Early discharge capacity`: Discharge capacity at early cycle
- `Difference between max discharge capacity and early discharge capacity`: Capacity fade

Logic:

- Uses **Severson methodology** with critical cycles [2, 9, 99]
- Computes ΔQdlin = Qdlin(late_cycle) - Qdlin(early_cycle)
- Applies **log10 transformation** to statistical measures
- Handles NaN/Inf values by setting them to 0

2.2 FullModelFeatureExtractor

Features Extracted (8):

- `Minimum`: Log10 of minimum absolute value of ΔQdlin
- `Variance`: Log10 of variance of ΔQdlin
- `Slope of linear fit to the capacity curve`: Linear regression slope
- `Intercept of linear fit to the capacity curve`: Linear regression intercept
- `Early discharge capacity`: Discharge capacity at early cycle
- `Average early charge time`: Average charge time over first 4 cycles
- `Integral of temperature over time`: Temperature integration (if available)
- `Minimum internal resistance`: Minimum IR across cycles (if available)
- `Internal resistance change`: IR change from early to late cycle (if available)

Logic:

- Extends discharge model with additional temporal and thermal features
- Uses **linear regression** on capacity fade curve
- **Temperature integration** across critical cycles
- **Internal resistance** analysis (dataset-dependent)

2.3 VarianceModelFeatureExtractor

Features Extracted (1):

- `Variance`: Log10 of variance of ΔQdlin

Logic:

- **Simplified model** using only variance feature
- Based on Severson et al. finding that variance is most predictive

2.4 VoltageCapacityMatrixFeatureExtractor

Features Extracted:

- **2D Matrix**: [num_cycles × interp_dim] where each row is ΔQdlin for a cycle

Logic:

- Creates **voltage-capacity difference matrix**
- **Interpolation**: Standardizes voltage-capacity curves to fixed dimensions
- **Smoothing**: Applies median filtering to reduce noise
- **Cycle selection**: Configurable cycle range and sampling
- **Base cycle**: Uses specified cycle as reference (typically cycle 9)

Table 1: Critical Cycles Configuration by Dataset

Dataset	Critical Cycles
HUST, MATR	[2, 9, 99] (standard)
SNL, CRUSH	[2, 9, 19] (shorter cycle life)

Table 2: Matrix Feature Parameters by Dataset

Dataset	diff_base	max_cycle	Precalculated Qdlin
HUST	9	99	Yes (some models)
SNL, CRUSH	2	19	No
MATR	8	98	Yes
MIX	8	98	Yes

3 Dataset-Specific Differences

3.1 Critical Cycles Configuration

3.2 Matrix Feature Parameters

4 Model-Feature Extractor Mapping

4.1 Sklearn Models

Table 3: Sklearn Model to Feature Extractor Mapping

Model Type	Feature Extractor
`discharge_model`	`DischargeModelFeatureExtractor`
`full_model`	`FullModelFeatureExtractor`
`variance_model`	`VarianceModelFeatureExtractor`
ridge, rf, xgb, gpr, pcr, plsr	`VoltageCapacityMatrixFeatureExtractor`

4.2 Neural Network Models

5 Key Feature Extraction Details

5.1 Qdlin Calculation

Listing 1: Qdlin Calculation Logic

```
def _get_Qdlin(I, V, Q, min_V, max_V):
    # Interpolates discharge capacity vs voltage curve
    # Filters for discharge current (I < -eps)
    # Interpolates to 1000 points between min_V and max_V
    # Reverses to get capacity vs voltage
```

5.2 Smoothing Algorithm

Table 4: Neural Network Model to Feature Extractor Mapping

Model Type	**Feature Extractor**
CNN, LSTM, MLP, Transformer	`VoltageCapacityMatrixFeatureExtractor`

Listing 2: Smoothing Algorithm

```
def smooth(x, window_size=10, sigma=3):
    # Median filtering with outlier detection
    # Replaces outliers with median values
    # Uses 3-sigma rule for outlier detection
```

5.3 Statistical Features

- **Log10 transformation** applied to all statistical measures
- **Epsilon addition** (1e-8) to prevent log(0)
- **NaN/Inf handling** by setting to 0

5.4 Temporal Features

- **Charge time calculation**: Integrates time when current ¡ 0
- **Temperature integration**: Averages temperature across cycles
- **Capacity fade analysis**: Linear regression on capacity vs cycle

6 Feature Extractor Selection Strategy

1. **Simple Models** (Linear Regression): Use statistical features
2. **Complex Models** (XGBoost, Random Forest): Use matrix features
3. **Neural Networks**: Use matrix features for spatial/temporal patterns
4. **Dataset-Specific**: Adjust critical cycles based on dataset characteristics

7 Summary

The feature extraction system is designed to be **modular** and **dataset-adaptive**, with different extractors optimized for different model types and dataset characteristics. The system provides a comprehensive framework for extracting meaningful features from battery cycling data across multiple datasets and model architectures.