



ÉCOLE CENTRALE NANTES

PROJECT REPORT  
E-PiCo+

---

# Intelligent Estimation of Battery State of Charge and Health (SOC/SOH)

---

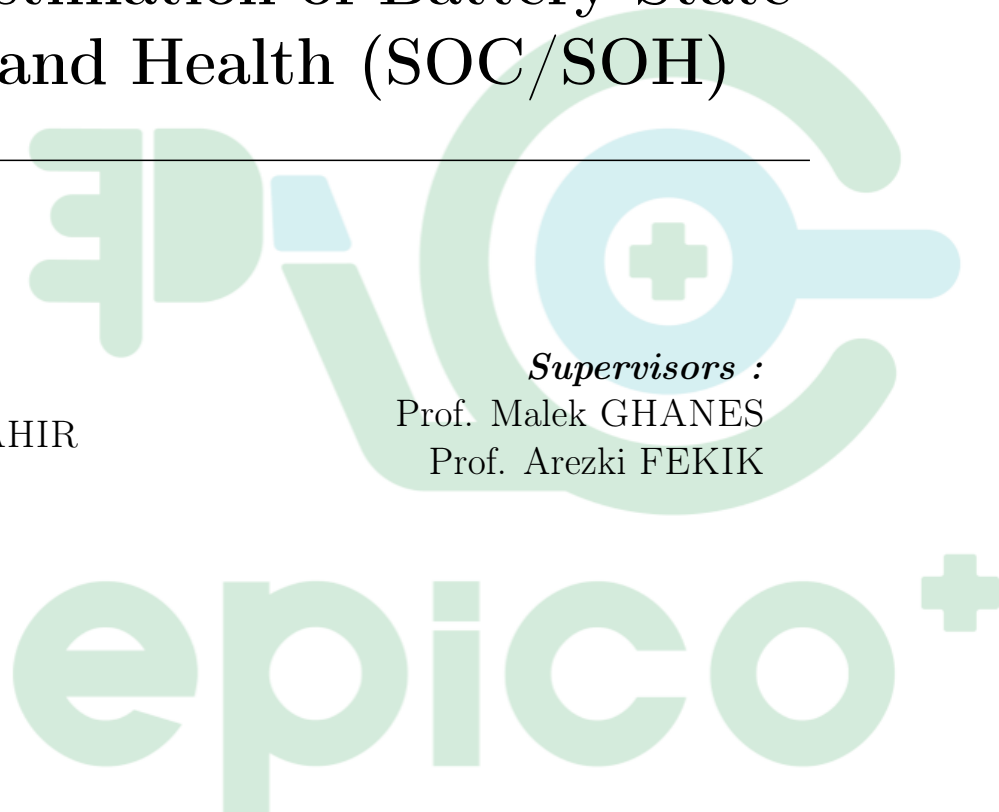
*Students :*

Muhammad Rafey TAHIR

*Supervisors :*

Prof. Malek GHANES

Prof. Arezki FEKIK



### **Abstract**

Accurate estimation of the State of Charge (SOC) and State of Health (SOH) is essential for the safe, reliable, and efficient operation of lithium-ion battery systems, particularly in electric vehicles and energy storage applications. This project presents an intelligent data-driven framework for joint SOC and SOH estimation using machine learning and internal state estimation techniques. For SOC estimation, multiple learning-based models are implemented using battery voltage, current, and temperature measurements under dynamic operating conditions. The SOC estimation methods include Fully Connected Networks (FCN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. For SOH estimation, battery internal degradation behavior is modeled using machine learning techniques, including Support Vector Machines (SVM), Random Forests, and LSTM networks. The performance of the proposed models is assessed using standard evaluation metrics such as Root Mean Square Error (RMSE) and computational efficiency. A comparative analysis is conducted to identify the most accurate and computationally efficient models for real-time battery management applications. The results demonstrate that deep learning-based approaches provide high-accuracy SOC estimation, while both classical and deep learning models effectively track battery health degradation. The study concludes with a discussion on the integration of the selected models into an embedded Battery Management System (BMS), highlighting their practical applicability and limitations.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Classical Methods for SOC and SOH Estimation . . . . .	3
1.2	Data-Driven and Modern Estimation Techniques . . . . .	4
1.3	Motivation and Objectives . . . . .	4
<b>2</b>	<b>Methodology</b>	<b>5</b>
2.1	Deep Learning for SOC Estimation . . . . .	5
2.2	Data Preprocessing and Feature Engineering . . . . .	5
2.3	LSTM Model Architecture . . . . .	6
2.3.1	Model Training . . . . .	6
2.3.2	Hyperparameter Tuning . . . . .	6
2.3.3	Algorithm Description . . . . .	7
2.4	LSTM-Based State of Health (SOH) Estimation . . . . .	7
2.4.1	SOH Definition . . . . .	8
2.4.2	Input Representation . . . . .	8
2.4.3	LSTM Degradation Modeling . . . . .	8
2.4.4	SOH Prediction . . . . .	9
2.4.5	Training Objective . . . . .	9
2.5	Dataset Description . . . . .	9
2.5.1	Battery Cell and Test Equipment . . . . .	9
2.5.2	Experimental Test Procedure . . . . .	9
2.5.3	Data Format and Recorded Signals . . . . .	10
2.5.4	Sampling Characteristics . . . . .	10
2.5.5	Battery Specifications . . . . .	10
2.5.6	Battery Physical Appearance . . . . .	10
<b>3</b>	<b>Results</b>	<b>11</b>
3.1	SOC Estimation Results . . . . .	11
3.1.1	Predicted SOC vs True SOC . . . . .	11
3.1.2	Time-Domain SOC Tracking . . . . .	13
3.1.3	Training and Convergence Analysis . . . . .	13
3.1.4	Comparative Analysis . . . . .	13
3.2	SOH Estimation Results . . . . .	15
3.2.1	SOH Prediction Over Discharge Cycles . . . . .	15
3.2.2	Time-Series SOH Prediction Performance . . . . .	15
3.2.3	Training and Validation Analysis . . . . .	16
3.2.4	Quantitative Performance Evaluation . . . . .	16
<b>4</b>	<b>Conclusion &amp; Future Work</b>	<b>17</b>
4.1	Industry Direction and Real-World Embedded BMS Software . . . . .	18

# 1 Introduction

Lithium-ion batteries are widely used in electric vehicles (EVs), renewable energy storage systems, and portable electronics due to their high energy density, long cycle life, and favorable efficiency characteristics. The performance, safety, and longevity of these battery systems strongly depend on the effectiveness of the Battery Management System (BMS). Among the key functions of a BMS, accurate estimation of the State of Charge (SOC) and State of Health (SOH) is critical for reliable operation and decision-making.

The State of Charge (SOC) represents the remaining available capacity of a battery relative to its nominal capacity and is commonly expressed as a percentage. SOC estimation is essential for range prediction, power management, and preventing overcharge or deep discharge conditions. However, SOC cannot be directly measured and must be estimated using measurable signals such as terminal voltage, current, and temperature. Errors in SOC estimation may lead to reduced battery utilization, unexpected shutdowns, or safety risks.

The State of Health (SOH) indicates the aging condition of a battery and reflects its ability to store and deliver energy compared to its initial state. SOH degradation occurs due to electrochemical aging mechanisms such as capacity fade and internal resistance growth. Accurate SOH estimation enables predictive maintenance, remaining useful life estimation, and optimal battery replacement strategies. Inaccurate SOH estimation can result in inefficient system operation and increased operational costs.

## 1.1 Classical Methods for SOC and SOH Estimation

Traditional SOC and SOH estimation techniques are primarily model-based and rely on electrochemical or equivalent circuit representations of the battery. One of the most widely used classical SOC estimation methods is Coulomb Counting [5], which estimates SOC by integrating the battery current over time.

$$SOC(t) = SOC(t_0) - \frac{1}{Q_n} \int_{t_0}^t I(\tau) d\tau \quad (1)$$

where  $SOC(t_0)$  is the initial state of charge,  $Q_n$  is the nominal battery capacity in ampere-hours (Ah), and  $I(\tau)$  is the battery current, defined as positive during discharge. Although Coulomb Counting is simple and computationally efficient, it suffers from cumulative errors caused by sensor noise, current measurement bias, and uncertainty in the initial SOC value. Although simple and computationally efficient, Coulomb Counting suffers from cumulative errors due to sensor noise and initial SOC uncertainty.

Another classical approach is the Equivalent Circuit Model (ECM) combined with state observers such as the Kalman Filter (KF) or Extended Kalman Filter (EKF) [4]. These methods estimate SOC by modeling battery dynamics using resistor-capacitor (RC) networks and updating the SOC estimate based on voltage and current measurements. While observer-based techniques improve robustness compared to Coulomb Counting, their performance heavily depends on accurate model parameterization and may degrade under varying temperature and aging conditions.

For SOH estimation, classical approaches often rely on capacity measurements obtained from full charge–discharge cycles or internal resistance estimation. These methods are time-consuming, require controlled operating conditions, and are not always suitable for online implementation in real-world applications.

## 1.2 Data-Driven and Modern Estimation Techniques

With the increasing availability of high-resolution battery data and advances in computational capabilities, data-driven methods [6] have emerged as a powerful alternative for SOC and SOH estimation. Unlike model-based approaches, data-driven techniques learn the nonlinear relationship between measurable battery signals and internal states directly from data, reducing dependency on explicit battery models.

Machine learning algorithms such as Support Vector Machines (SVM) and Random Forests (RF) have been successfully applied to SOH estimation by capturing complex aging patterns from historical data. In recent years, deep learning models, including Fully Connected Neural Networks (FCN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks [2], have demonstrated superior performance for SOC estimation, particularly under dynamic load profiles and varying temperature conditions.

Recurrent architectures such as LSTM networks are especially well-suited for battery state estimation due to their ability to model temporal dependencies and long-term dynamics in time-series data. These modern approaches offer improved accuracy and robustness compared to classical methods, making them attractive candidates for next-generation intelligent Battery Management Systems.

## 1.3 Motivation and Objectives

Motivated by the limitations of classical model-based techniques and the promising performance of data-driven approaches, this project aims to implement and compare multiple machine learning and deep learning methods for SOC and SOH estimation. The primary objective is to evaluate the accuracy, robustness, and computational efficiency of different models and to identify suitable approaches for real-time BMS integration.

## 2 Methodology

### 2.1 Deep Learning for SOC Estimation

State of Charge (SOC) estimation for lithium-ion batteries is inherently a time-series regression problem, as the SOC at any instant depends on the historical evolution of battery voltage, current, temperature, and load conditions. Deep learning techniques are particularly well suited for this task due to their ability to learn nonlinear relationships directly from data and capture complex system dynamics without explicit physical modeling.

Several deep learning architectures have been explored in the literature for SOC estimation. Convolutional Neural Networks (CNNs) are capable of extracting local temporal patterns from time-series data and have been used effectively to identify short-term dependencies in voltage and current signals. Fully Convolutional Networks (FCNs) further extend this idea by eliminating fully connected layers and employing global pooling, resulting in fewer trainable parameters and improved generalization. However, both CNNs and FCNs primarily focus on local feature extraction and may struggle to model long-term temporal dependencies that are critical for accurate SOC estimation.

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are specifically designed to process sequential data and retain information over extended time horizons. LSTM networks address the vanishing gradient problem encountered in traditional RNNs through gated memory cells, enabling effective learning of long-range dependencies. Given the sequential nature of SOC evolution, an LSTM-based model is adopted in this work.

### 2.2 Data Preprocessing and Feature Engineering

The dataset [3] consists of time-series measurements collected from lithium-ion battery cycling experiments. The input features include voltage, current, temperature, and time, while SOC is used as the regression target.

To enhance the representational capacity of the input data, additional features are derived. Instantaneous power is computed as

$$P(t) = V(t) \cdot I(t), \quad (2)$$

where  $V(t)$  and  $I(t)$  denote voltage and current, respectively. Furthermore, coulomb-counted capacity is calculated by integrating current over time:

$$CC(t) = \sum_{k=1}^t \frac{I(k)\Delta t(k)}{3600}. \quad (3)$$

All input features are standardized using z-score normalization:

$$x' = \frac{x - \mu}{\sigma}, \quad (4)$$

where  $\mu$  and  $\sigma$  represent the mean and standard deviation of the feature. Standardization improves numerical stability and accelerates convergence during training. To ensure realistic generalization, the dataset is split at the file level into training, validation, and test sets, thereby preventing temporal leakage between splits.

## 2.3 LSTM Model Architecture

The LSTM model processes fixed-length sequences constructed using a sliding window approach. Each input sample is represented as

$$\mathbf{X} \in \mathbb{R}^{L \times d}, \quad (5)$$

where  $L$  denotes the sequence length and  $d$  is the number of input features.

The architecture consists of stacked LSTM layers with a hidden state dimension  $H$ , followed by a fully connected layer that maps the final hidden state to the SOC estimate. The output of the last timestep is used to predict SOC, enabling the model to leverage historical information within each sequence.

### 2.3.1 Model Training

The model is trained using the Mean Squared Error (MSE) loss function, defined as

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (6)$$

where  $y_i$  and  $\hat{y}_i$  denote the true and predicted SOC values, respectively.

Optimization is performed using the Adam optimizer due to its robustness to noisy gradients and adaptive learning rate behavior. Mini-batch training is employed to improve computational efficiency. Early stopping is implemented based on validation loss with a predefined patience and minimum improvement threshold to mitigate overfitting.

### 2.3.2 Hyperparameter Tuning

The performance of LSTM networks is highly sensitive to hyperparameter selection, including the number of hidden units, network depth, and learning rate. Improper choices can lead to underfitting, overfitting, or unstable training behavior. Consequently, systematic hyperparameter optimization is essential for reliable SOC estimation.

In this work, hyperparameter tuning is conducted using Optuna [1], an automated hyperparameter optimization framework based on efficient sampling strategies. The objective of the optimization process is to minimize the validation loss. The hyperparameter vector be defined as

$$\boldsymbol{\lambda} = \{H, N_l, \eta\}, \quad (7)$$

where  $H$  is the hidden layer size,  $N_l$  is the number of LSTM layers, and  $\eta$  is the learning rate.

The Optuna objective function is defined as:

$$\min_{\boldsymbol{\lambda} \in \Lambda} \mathcal{J}(\boldsymbol{\lambda}), \quad (8)$$

where  $\Lambda$  denotes the hyperparameter search space and

$$\mathcal{J}(\boldsymbol{\lambda}) = \mathcal{L}_{\text{val}}(\theta^*(\boldsymbol{\lambda})). \quad (9)$$

Here,  $\theta^*(\boldsymbol{\lambda})$  represents the optimal model parameters obtained after training the LSTM with hyperparameters  $\boldsymbol{\lambda}$ :

$$\theta^*(\boldsymbol{\lambda}) = \arg \min_{\theta} \mathcal{L}_{\text{train}}(\theta; \boldsymbol{\lambda}). \quad (10)$$

In practice,  $\mathcal{L}_{\text{val}}$  is the Mean Squared Error computed on the validation dataset after a fixed number of training epochs. Optuna iteratively evaluates  $\mathcal{J}(\boldsymbol{\lambda})$  across multiple trials and selects the hyperparameter set that yields the minimum validation loss.

The following hyperparameters are explored:

- Hidden size:  $H \in [10, 100]$
- Number of LSTM layers:  $N_l \in [1, 5]$
- Learning rate:  $\eta \in [10^{-5}, 10^{-1}]$  (log-uniform)

For each trial, an LSTM model is instantiated with sampled hyperparameters, trained for a fixed number of epochs, and evaluated on the validation set. The best-performing configuration is selected and used for final model training.

### 2.3.3 Algorithm Description

- Algorithm 1: Data Preparation
  1. Load battery cycling data files.
  2. Compute power and coulomb-counted capacity.
  3. Standardize input features.
  4. Split data into training, validation, and test sets at the file level.
  5. Generate fixed-length sequences using a sliding window.
- Algorithm 2: LSTM Training
  1. Initialize LSTM model parameters.
  2. Train the model using MSE loss and Adam optimizer.
  3. Evaluate validation loss after each epoch.
  4. Apply early stopping based on validation performance.
- Algorithm 3: Hyperparameter Optimization with Optuna
  1. Define the objective function based on validation loss.
  2. Sample hyperparameters using Optuna.
  3. Train and evaluate the LSTM model for each trial.
  4. Select the hyperparameter set yielding the lowest validation loss.

## 2.4 LSTM-Based State of Health (SOH) Estimation

State of Health (SOH) estimation aims to quantify the degradation level of a lithium-ion battery over its lifetime and is commonly defined as the ratio of the current available capacity to the nominal (rated) capacity. Unlike SOC, which varies rapidly within a single cycle, SOH evolves slowly across cycles and reflects cumulative aging effects such as capacity fade and internal resistance growth. Consequently, SOH estimation requires models capable of learning long-term temporal degradation patterns.

In this work, an LSTM-based deep learning framework is employed for SOH estimation due to its ability to capture long-range dependencies in sequential data and model nonlinear degradation dynamics.



### 2.4.1 SOH Definition

The SOH at cycle  $k$  is defined as

$$\text{SOH}(k) = \frac{Q_k}{Q_0}, \quad (11)$$

where  $Q_k$  denotes the measured discharge capacity at cycle  $k$ , and  $Q_0$  is the rated (initial) capacity of the battery.

### 2.4.2 Input Representation

For SOH estimation, the battery aging process is represented as a sequence of features extracted over charge–discharge cycles. The input sequence be defined as

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}, \quad \mathbf{x}_t \in \mathbb{R}^d, \quad (12)$$

where  $T$  is the sequence length corresponding to consecutive cycles or time windows, and  $d$  denotes the number of extracted features (e.g., voltage statistics, current profiles, temperature indicators, or capacity-related features).

All input features are normalized prior to training to ensure numerical stability and consistent gradient propagation.

### 2.4.3 LSTM Degradation Modeling

The Long Short-Term Memory (LSTM) network processes the SOH-related input sequence sequentially to capture long-term battery degradation behavior. At each time step  $t$ , the LSTM cell updates its internal states using gated nonlinear operations defined as follows:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (13)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (14)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \quad (15)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad (16)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (17)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (18)$$

where  $\mathbf{x}_t \in \mathbb{R}^d$  denotes the input feature vector at time step  $t$ ,  $\mathbf{h}_t \in \mathbb{R}^H$  is the hidden state, and  $\mathbf{c}_t \in \mathbb{R}^H$  is the cell state of the LSTM. The vectors  $\mathbf{f}_t$ ,  $\mathbf{i}_t$ , and  $\mathbf{o}_t$  represent the forget, input, and output gates, respectively, while  $\tilde{\mathbf{c}}_t$  denotes the candidate cell state.

The matrices  $\mathbf{W}_{(\cdot)} \in \mathbb{R}^{H \times d}$  and  $\mathbf{U}_{(\cdot)} \in \mathbb{R}^{H \times H}$  are learnable weight parameters associated with the input and recurrent connections, respectively, and  $\mathbf{b}_{(\cdot)} \in \mathbb{R}^H$  are bias vectors. These gated mechanisms allow the LSTM to selectively retain or discard historical information, enabling effective modeling of slow and cumulative battery degradation processes that govern SOH evolution.

For stacked LSTM architectures, the hidden state output of a lower LSTM layer serves as the input to the subsequent layer at the same time step. This hierarchical structure allows the network to learn increasingly abstract representations of battery aging dynamics across layers.

#### 2.4.4 SOH Prediction

The SOH estimate is obtained by applying a linear regression layer to the hidden state at the final timestep:

$$\widehat{\text{SOH}} = \mathbf{W}_{\text{fc}} \mathbf{h}_T + b_{\text{fc}}, \quad (19)$$

where  $\widehat{\text{SOH}}$  denotes the predicted state of health. This sequence-to-one formulation enables the model to infer the current degradation state based on historical aging information contained within the input sequence.

#### 2.4.5 Training Objective

The LSTM-based SOH model is trained by minimizing the Mean Squared Error loss:

$$\mathcal{L}_{\text{SOH}} = \frac{1}{N} \sum_{i=1}^N \left( \text{SOH}_i - \widehat{\text{SOH}}_i \right)^2, \quad (20)$$

where  $N$  is the number of training samples.

The Adam optimizer is employed to update model parameters, and early stopping based on validation loss is used to prevent overfitting.

### 2.5 Dataset Description

The dataset [3] used in this project is obtained from the Mendeley Data repository titled “*LG 18650HG2 Battery Dataset*”. The experimental tests were conducted at McMaster University, Hamilton, Ontario, Canada, by Dr. Phillip Kollmeyer. The dataset contains high-quality experimental measurements collected from a brand-new 3 Ah LG 18650HG2 lithium-ion cell and is widely used for battery modeling, SOC estimation, and SOH analysis.

#### 2.5.1 Battery Cell and Test Equipment

All experiments were performed on a single LG HG2 cylindrical lithium-ion battery cell with a nominal capacity of 3 Ah. The cell was tested using a Digatron Firing Circuits Universal Battery Tester (75 A, 5 V) with voltage and current measurement accuracy of 0.1% of full scale. The tests were carried out inside an 8 cu.ft. thermal chamber to ensure accurate temperature control during operation.

#### 2.5.2 Experimental Test Procedure

A comprehensive series of charge, discharge, pulse, and drive-cycle tests were conducted at six different ambient temperatures: 40 °C, 25 °C, 10 °C, 0 °C, −10 °C, and −20 °C. After each test sequence, the battery was recharged at a 1C constant-current/constant-voltage (CC–CV) rate to 4.2 V with a 50 mA cutoff current, ensuring consistent initial conditions.

The performed test procedures include:

- Hybrid Pulse Power Characterization (HPPC) tests at multiple discharge (1C, 2C, 4C, and 6C) and charge rates, conducted at various SOC levels from 100% down to 0%.

- C/20 charge and discharge tests for capacity characterization.
- Constant-current discharge tests at 0.5C, 1C, and 2C rates.
- Standard automotive drive cycles including UDDS, HWFET, LA92, and US06.
- Eight mixed drive cycles consisting of random combinations of standard drive profiles to emulate real-world electric vehicle usage.

For ambient temperatures below 10°C, regenerative current limits were reduced to prevent premature cell degradation. Drive-cycle tests were repeated until approximately 95% of the available 1C discharge capacity at the given temperature was utilized.

### 2.5.3 Data Format and Recorded Signals

The dataset is provided in both MATLAB (.mat) and CSV formats. Each file contains synchronized time-series measurements of key battery parameters. The recorded data channels include:

- Time (seconds)
- Timestamp (date and time of measurement)
- Terminal voltage (V)
- Battery current (A)
- Accumulated ampere-hours (Ah)
- Accumulated watt-hours (Wh)
- Instantaneous power (W)
- Battery case temperature (°C)

The battery temperature was measured at the center of the cell casing using an AD592 temperature sensor with an accuracy of  $\pm 1^\circ\text{C}$ .

### 2.5.4 Sampling Characteristics

The sampling rate of the dataset is non-uniform. High-dynamic tests such as drive cycles were recorded at a 0.1 s time step, while charging phases and rest periods were logged at lower sampling rates. As a result, resampling or interpolation is required to obtain a uniform time base for machine learning model training.

### 2.5.5 Battery Specifications

Table 1 summarizes the main specifications of the LG 18650HG2 battery cell used in this dataset.

### 2.5.6 Battery Physical Appearance

Figure 1 shows the physical appearance and dimensions of the LG 18650HG2 lithium-ion battery cell used in the experiments.

Table 1: LG HG2 Battery Cell Specifications

Parameter	Specification
Chemistry	LiNiMnCoO <sub>2</sub> (NMC) / Graphite + SiO
Nominal Voltage	3.6 V
Nominal Capacity	3.0 Ah
Charge Method	CC-CV, 4.2 V cutoff
Maximum Discharge Current	20 A
Energy Density	240 Wh/kg

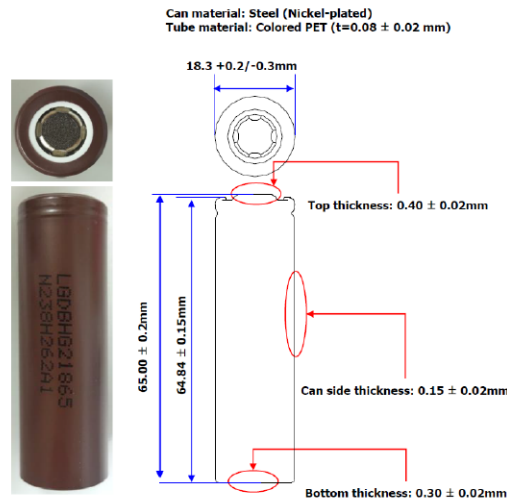


Figure 1: LG HG2 lithium-ion battery cell used in the dataset

### 3 Results

This section presents the results obtained for State of Charge (SOC) estimation using different data-driven models, namely Fully Connected Networks (FCN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. The models were evaluated using unseen test drive cycles to assess their generalization capability under realistic operating conditions.

#### 3.1 SOC Estimation Results

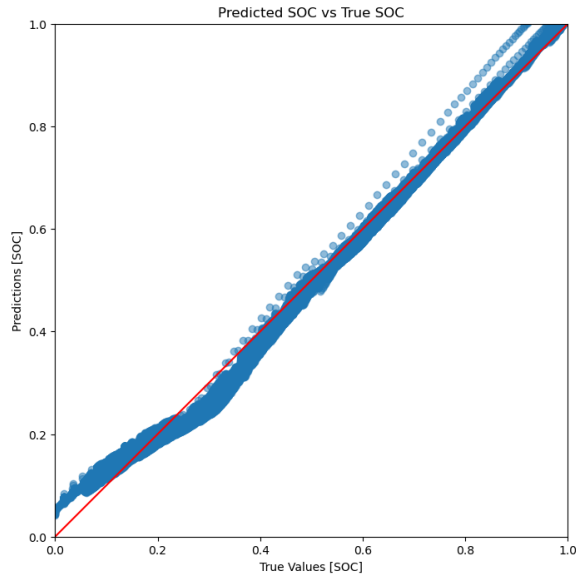
The SOC estimation performance of the proposed data-driven models is evaluated using multiple unseen test drive cycles, including UDDS, US06, HWFET, and mixed driving profiles. Both predicted-versus-true SOC scatter plots and time-domain SOC trajectories are used to assess point-wise accuracy and temporal tracking capability.

##### 3.1.1 Predicted SOC vs True SOC

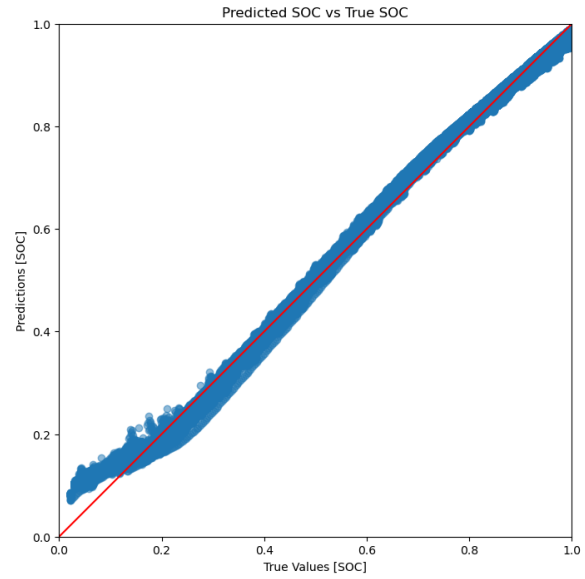
Figure 2 presents the predicted SOC versus true SOC scatter plots for different models. The diagonal line represents the ideal  $y = x$  relationship. All models demonstrate the

ability to learn the nonlinear mapping between battery input signals and SOC; however, the density and dispersion of prediction points vary significantly across models.

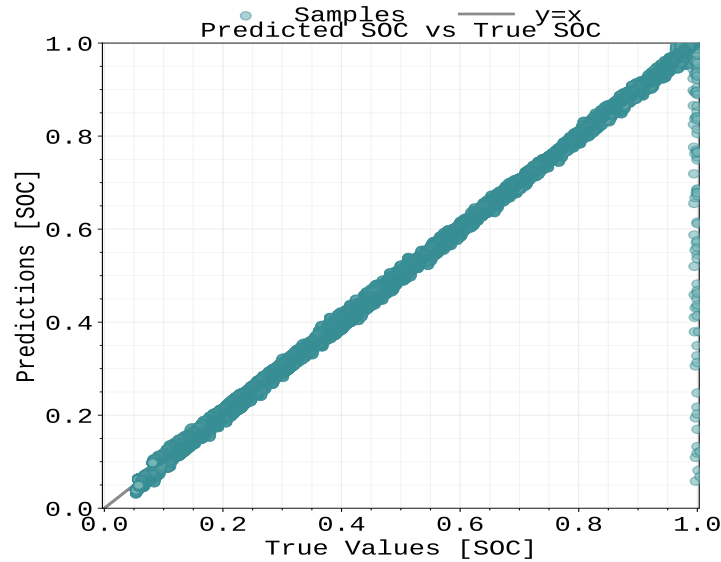
The FCN model exhibits noticeable dispersion around the ideal line, particularly in low and mid SOC regions, indicating limited robustness. The CNN model reduces this dispersion by capturing local temporal patterns. The LSTM model shows the tightest clustering along the diagonal, confirming superior prediction accuracy across the entire SOC range.



(a) FCN



(b) CNN



(c) LSTM

Figure 2: Predicted SOC versus true SOC for different models

### 3.1.2 Time-Domain SOC Tracking

Figures 3 and 4 illustrate SOC estimation performance in the time domain for various drive cycles. These plots provide insight into the models' ability to track SOC evolution under realistic and dynamic operating conditions.

Across all test cases, the LSTM model closely follows the true SOC trajectory across all drive cycles, maintaining high accuracy even during aggressive current fluctuations and extended discharge durations. In contrast, the FCN model shows small but accumulating deviations over long discharge periods while The CNN model improves temporal smoothness but still exhibits minor lag during rapid SOC transitions.

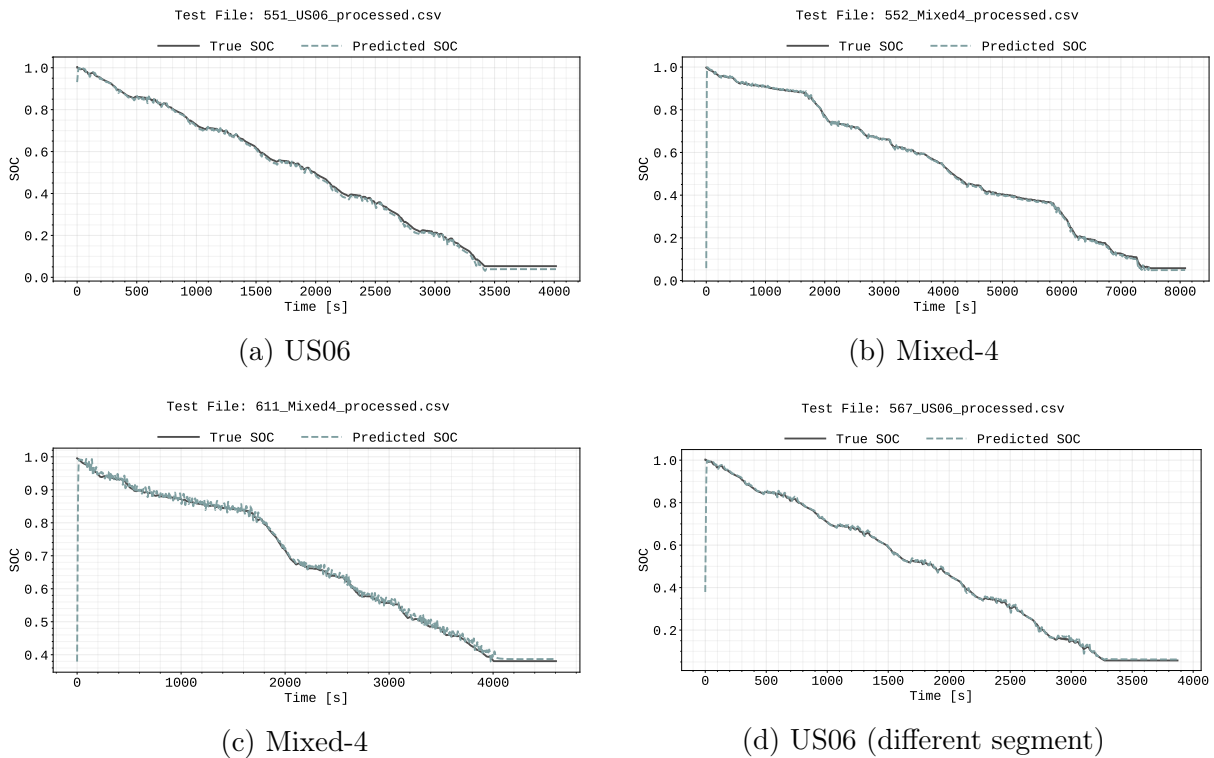


Figure 3: SOC estimation over time for different test drive cycles

### 3.1.3 Training and Convergence Analysis

Figure 5 shows the training and validation loss curves for the SOC estimation models. All models exhibit rapid convergence during the initial epochs, followed by stable loss values, indicating effective learning and absence of significant overfitting. The LSTM model achieves the lowest validation loss, confirming its superior generalization capability. This stable convergence behavior demonstrates that the selected network architectures and training procedures are well suited for SOC estimation tasks in real-world Battery Management Systems.

### 3.1.4 Comparative Analysis

A qualitative comparison of the SOC estimation methods is summarized in Table 2. While FCN and CNN models offer lower computational complexity, the LSTM model provides

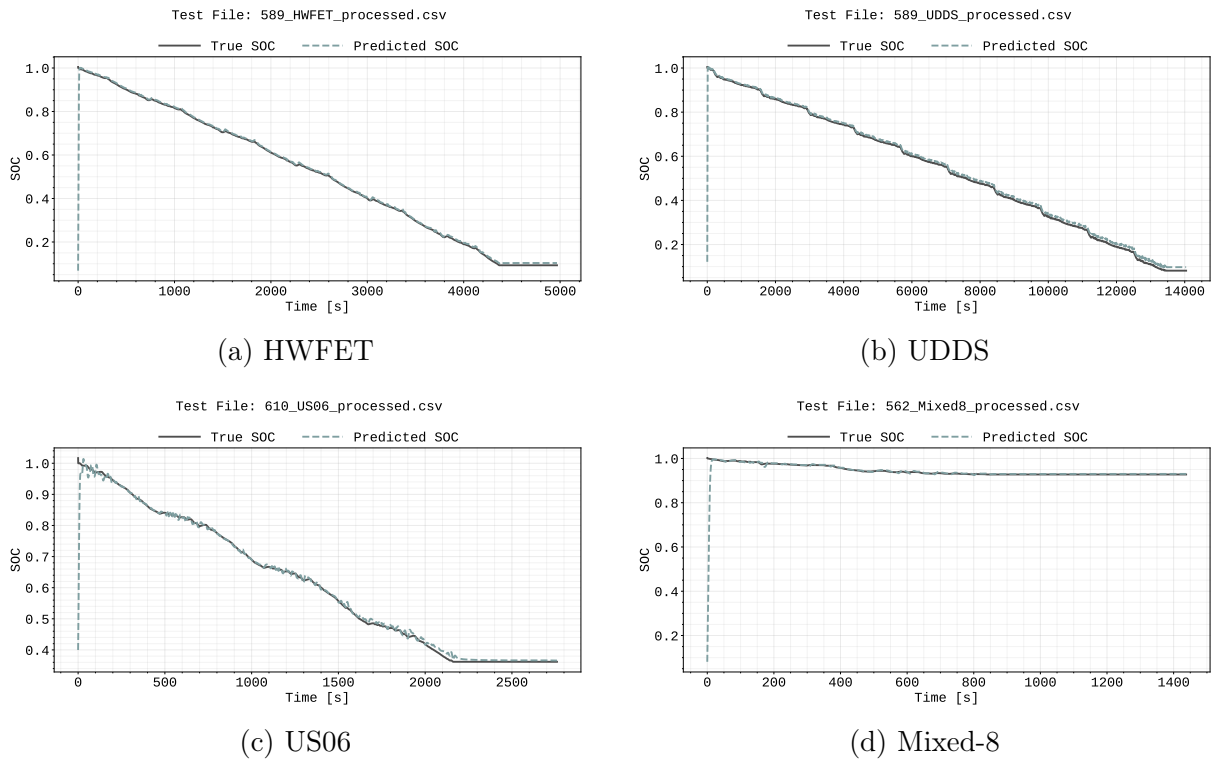


Figure 4: Additional SOC tracking results under dynamic operating conditions

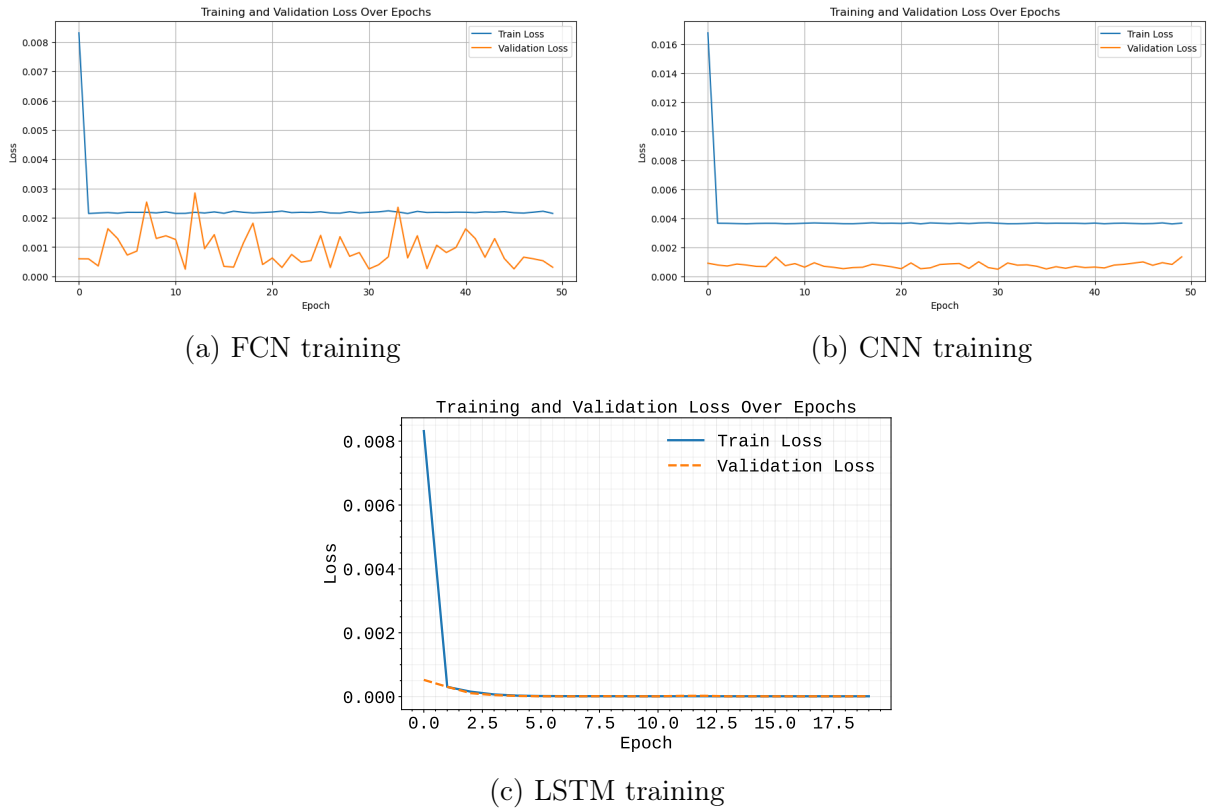


Figure 5: Training and validation loss curves for SOC estimation models

the highest estimation accuracy and robustness under dynamic operating conditions, making it the most suitable candidate for real-time BMS applications.

Table 2: Comparison of SOC Estimation Methods

Model	MSE	MAE	Key Observations
FCN	0.0004	0.017	Simple structure, higher prediction dispersion
CNN	0.0006	0.018	Improved accuracy, captures local patterns
LSTM	0.0004	0.009	Best SOC tracking, smooth and robust predictions

Overall, the results confirm that incorporating temporal information significantly enhances SOC estimation accuracy. Among the evaluated methods, the LSTM-based approach achieves the most reliable performance and demonstrates strong potential for integration into intelligent Battery Management Systems.

## 3.2 SOH Estimation Results

This subsection presents the results obtained for State of Health (SOH) estimation using a data-driven approach based on a long short-term memory (LSTM). The SOH is defined as the ratio between the current maximum capacity of the battery and its nominal capacity, and it is evaluated over successive discharge cycles to capture long-term degradation behavior.

### 3.2.1 SOH Prediction Over Discharge Cycles

Figure 6 illustrates the comparison between the real SOH values and the LSTM-predicted SOH for battery C08 as a function of discharge cycles. The model is trained using historical degradation data and evaluated on unseen test samples. The predicted SOH curve closely follows the overall downward trend of the real SOH, indicating that the LSTM model successfully captures the long-term degradation pattern of the battery.

Although minor deviations can be observed at certain cycles, particularly during abrupt changes in SOH, the predicted values remain consistent with the real measurements. This demonstrates the ability of the LSTM model to generalize well and provide smooth SOH estimates, which is desirable for practical Battery Management System (BMS) applications.

### 3.2.2 Time-Series SOH Prediction Performance

Figure 7 shows the predicted SOH compared to the real SOH values on the test dataset in the time domain. The LSTM prediction accurately tracks the gradual decrease in SOH while maintaining a smooth profile. This behavior indicates robustness against measurement noise and confirms the suitability of the proposed approach for online SOH monitoring.



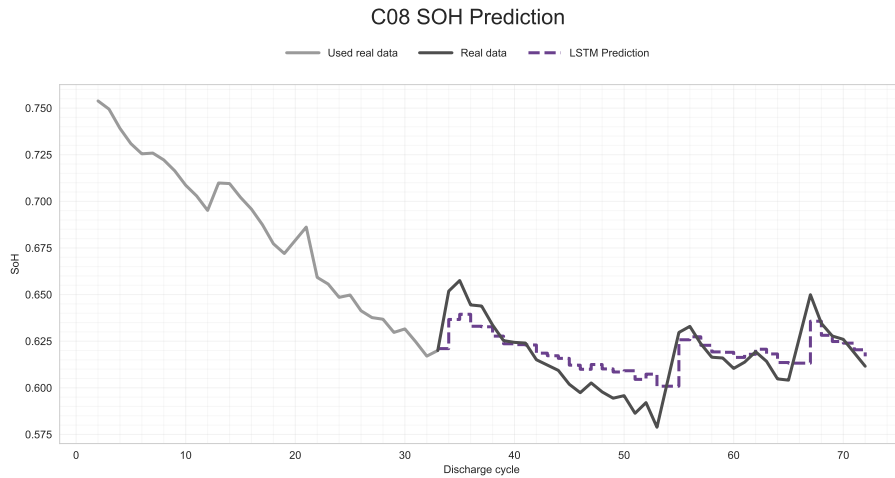


Figure 6: LSTM-based SOH prediction versus real SOH values over discharge cycles

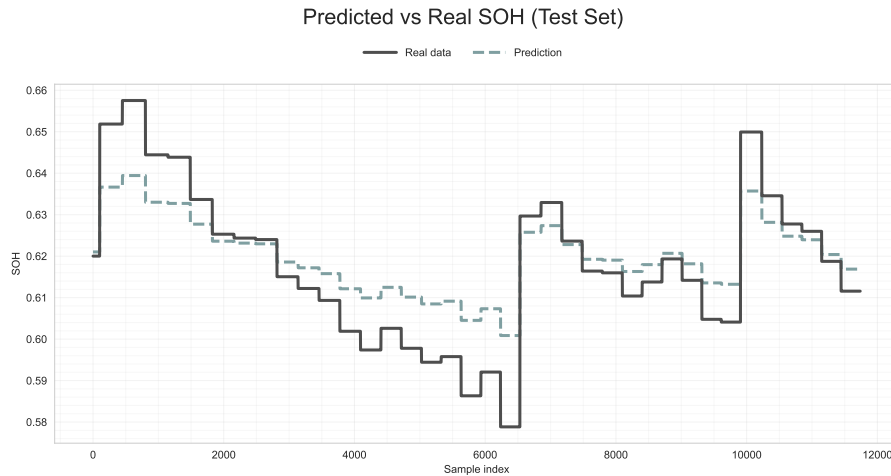


Figure 7: Predicted versus real SOH values on the test dataset

### 3.2.3 Training and Validation Analysis

The training and validation loss curves are presented in Figure 8. The results show rapid convergence within the first few epochs, followed by stable loss values throughout the training process. The small gap between training and validation losses indicates good generalization performance and absence of significant overfitting.

### 3.2.4 Quantitative Performance Evaluation

To quantitatively evaluate the performance of the LSTM-based SOH estimation model, standard regression metrics were computed on the test dataset. The obtained results are summarized in Table 3. The low RMSE and MAE values indicate high prediction accuracy, while the coefficient of determination ( $R^2$ ) confirms that a significant portion of the SOH variance is captured by the model.

Overall, the LSTM-based SOH estimation approach demonstrates strong capability in capturing battery degradation trends with high accuracy and stability. These results

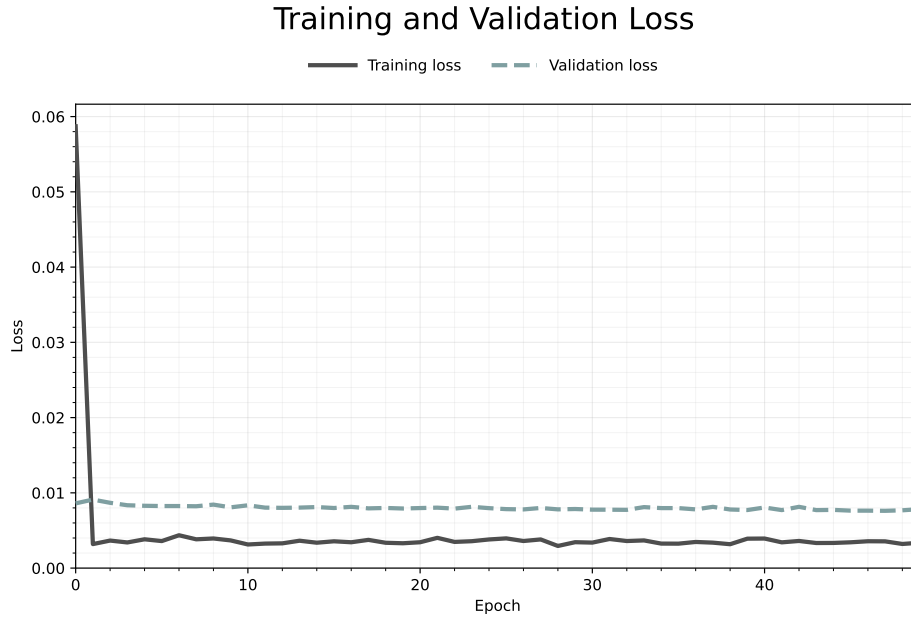


Figure 8: Training and validation loss curves for the LSTM-based SOH model

Table 3: SOH Estimation Performance Metrics

Metric	Value
Mean Squared Error (MSE)	0.0003
Root Mean Squared Error (RMSE)	0.010
Mean Absolute Error (MAE)	0.008
$R^2$ Score	0.728

highlight the effectiveness of data-driven methods for SOH estimation and their potential for integration into real-time battery health monitoring systems.

## 4 Conclusion & Future Work

This project investigated data-driven estimation of two key Battery Management System (BMS) states: State of Charge (SOC) and State of Health (SOH). For SOC estimation, multiple deep learning architectures (FCN, CNN, and LSTM) were implemented and evaluated under realistic automotive drive cycles. The qualitative comparison using predicted-vs-true scatter plots and time-domain SOC tracking indicates that sequence-aware models provide the most reliable performance in dynamic conditions, with the LSTM demonstrating the best overall tracking behavior due to its ability to model long-term temporal dependencies.

For SOH estimation, a hybrid CNN–LSTM approach was used to capture degradation-related patterns and their temporal evolution across cycles. The achieved test metrics (low RMSE/MAE and a reasonably strong  $R^2$ ) show that the model can capture the overall degradation trend while remaining stable during inference. Overall, the results confirm that incorporating temporal information (through LSTM-style recurrence) significantly improves estimation robustness for both SOC and SOH in comparison to purely feedforward approaches.

## 4.1 Industry Direction and Real-World Embedded BMS Software

Modern EV and battery companies increasingly treat BMS as a software-defined system, combining model-based estimation with data-driven learning where beneficial. Examples of organizations and directions relevant to embedded or production-grade deployment include:

- **Eatron Technologies (AI-powered BMS software):** Eatron publicly discusses AI-enabled BMS capabilities (state estimation, RUL, diagnostics) and has collaborated with Infineon on AI-BMS proof-of-concepts intended for real-time state estimation [Blog].

These references show two important trends: (i) production systems often combine physics/model-based ideas with learning, and (ii) embedded deployment focuses heavily on deterministic memory, quantization, and runtime efficiency.

A practical next step is deploying the best-performing models (e.g., SOC-LSTM and SOH CNN-LSTM) on embedded BMS hardware. TensorFlow Lite for Microcontrollers (TFLM) is well-suited for this setting because it is designed for resource-constrained systems and uses a pre-allocated *tensor arena* for deterministic memory usage. Performance can be improved on Arm Cortex-M targets using CMSIS-NN optimized kernels integrated with TFLM, which can significantly reduce inference latency and energy per inference.

The following improvements are recommended to extend this project into a more deployment-ready BMS solution:

- **Quantization:** Convert models to fully-integer (int8) to reduce flash/RAM footprint and improve speed. This is usually the single biggest enabler for MCU deployment.
- **Operator selection:** Prefer operators with optimized kernels (e.g., via CMSIS-NN on Cortex-M).
- **Memory planning:** Size and verify the tensor arena using TFLM memory planning guidance and profiling tools.
- **Closed-loop validation:** Integrate SOC/SOH predictions into a BMS control loop (charge limiting, power derating) and validate behavior on hardware-in-the-loop (HIL) or a lab cyclor.
- **Safety and compliance considerations:** For automotive-grade systems, evaluate functional safety requirements and fallbacks (e.g., revert to observer-based estimation when ML confidence is low).

## References

- [1] Takuya Akiba et al. “Optuna: A next-generation hyperparameter optimization framework”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.
- [2] Ephrem Chemali et al. “Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries”. In: *IEEE Transactions on Industrial Electronics* 65.8 (2018), pp. 6730–6739. DOI: 10.1109/TIE.2017.2787586.
- [3] Philip Kollmeyer et al. “LG 18650HG2 Li-ion battery data and example deep neural network xEV SOC estimator script”. In: *Mendeley Data* 3.2020 (2020).
- [4] Xingtao Liu et al. “Data-driven state of charge estimation for power battery with improved extended Kalman filter”. In: *IEEE Transactions on Instrumentation and Measurement* 72 (2023), pp. 1–10.
- [5] Kong Soon Ng et al. “Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries”. In: *Applied energy* 86.9 (2009), pp. 1506–1511.
- [6] Shahid Gulzar Padder et al. “Data-driven approaches for estimation of EV battery SoC and SoH: A review”. In: *IEEE Access* (2025).