SOH Prediction of Li-ion Batteries Using Linear Regression and LSTM

Project Overview

Problem Background

- **Lithium-ion batteries** are used in various fields such as **Electric Vehicle (EV)** and **Energy Storage System (ESS)**.
- Therefore, an effective **SOH (State of Health) prediction algorithm is essential**.
- **SOH (State of Health):** Represents the battery's degradation status, specifically the **capacity to store electricity**.
- SOH generally **decreases as the charge/discharge cycle increases**.
- Batteries are typically used until the **SOH decreases to 60% ~ 70%**.

1. Project Overview

SOH Calculation Method

The State of Health (SOH) is calculated using the battery′s current discharge capacity relative to its initial, nominal capacity.

$$SOH = \frac{Q_m}{Q_{nom}}$$

- Q_{nom} : Initial battery discharge capacity (Nominal Capacity)
- Value used: 2 Ah
- Q_m : Battery discharge capacity at the m^th Cycle.

This calculation is performed using the **NASA PCoE Datasets**

2. Project Overview

Data Experiment Conditions

The **NASA PCoE Datasets** used are categorized into three groups based on their experimental conditions:

- **Data Group A:** Normal temperature Charge/Discharge data (B05, B07, B18)
- **Data Group B:** Normal temperature **High-Power** Charge/Discharge data (B33, B34)
- **Data Group C:** **Low Temperature** Charge/Discharge data (B46, B47, B48)
- "Our LSTM model performed well because it was trained on a **large and complex dataset** We utilized data from **eight individual batteries** run under different conditions, from normal to low-temperature. For just one cell, the model processed **tens of thousands of high-resolution measurements** recorded over the battery's entire lifespan (up to 200 cycles).

Project Goals

The objective is to implement an SOH prediction simulation and compare different training scenarios:

1. Implement an **SOH prediction simulation** using **Linear Regression** and **LSTM**.
2. Perform training using **50% of the data**.
3. Perform training using **70% of the data**.
4. **Visualize the prediction results**.
5. Analyze results using **RMSE and MAE**

3. Execution Process

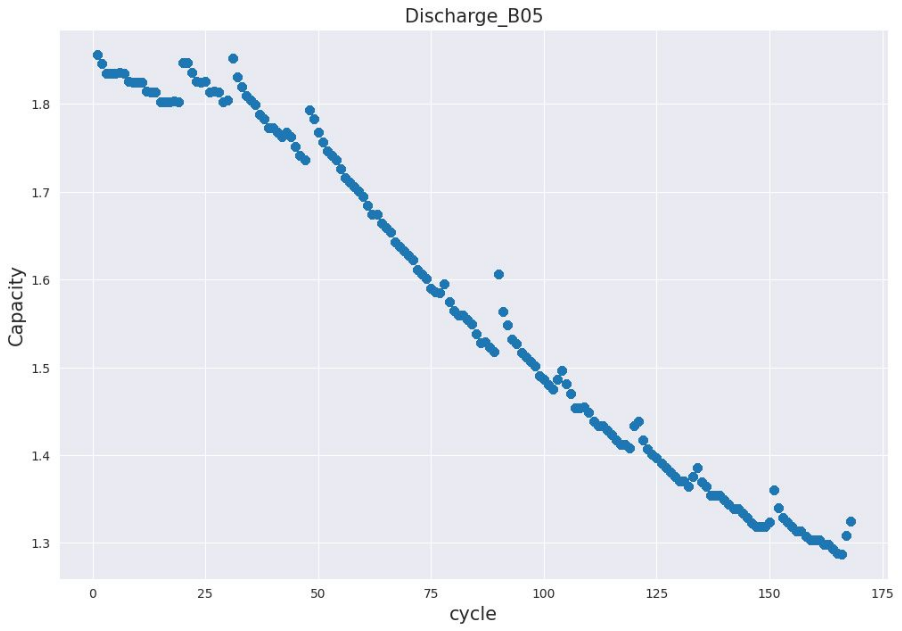
- **SOH Calculation Review**

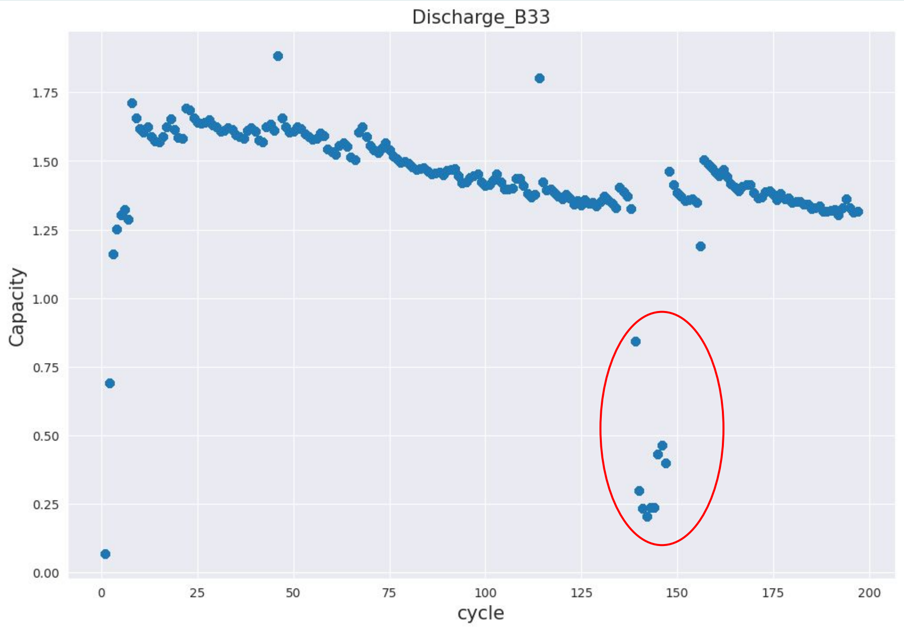
The State of Health (SOH) was calculated for each discharge cycle using the **capacity** Q_m from the NASA PCoE dataset and the **Nominal Capacity** (Q_{nom} **=2 Ah)**.

	terminal_voltage	terminal_current	temperature	charge_current	charge_voltage	time	capacity	cycle	SOH
0	4.191492	-0.004902	24.330034	-0.0006	0.000	0.000	1.856487	1	0.928244
1	4.190749	-0.001478	24.325993	-0.0006	4.206	16.781	1.856487	1	0.928244
2	3.974871	-2.012528	24.389085	-1.9982	3.062	35.703	1.856487	1	0.928244
3	3.951717	-2.013979	24.544752	-1.9982	3.030	53.781	1.856487	1	0.928244
4	3.934352	-2.011144	24.731385	-1.9982	3.011	71.922	1.856487	1	0.928244
...	...	...	...	...	...	...	...	...	...
50280	3.579262	-0.001569	34.864823	0.0006	0.000	2781.312	1.325079	168	0.662540
50281	3.581964	-0.003067	34.814770	0.0006	0.000	2791.062	1.325079	168	0.662540
50282	3.584484	-0.003079	34.676258	0.0006	0.000	2800.828	1.325079	168	0.662540
50283	3.587336	0.001219	34.565580	0.0006	0.000	2810.640	1.325079	168	0.662540
50284	3.589937	-0.000583	34.405920	0.0006	0.000	2820.390	1.325079	168	0.662540

4. SOH Visualization and Outliers

- SOH was plotted against the discharge cycle to observe the **degradation trend**.
- The plots revealed **Outliers** data points significantly deviating from the overall trend, notably in the **Discharge_B33** graph (Group B).





5. Execution Process

Outlier Removal using Quartiles

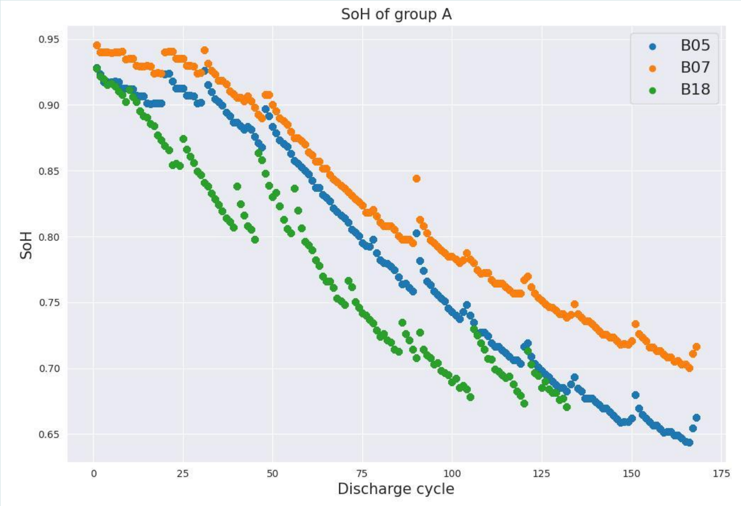
To ensure the models were trained on clean data, outliers that deviated sharply from the degradation trend were removed.

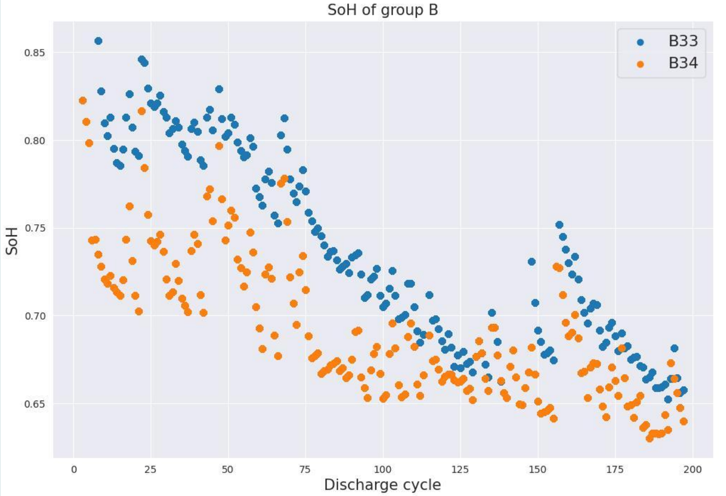
- **Method:** Interquartile Range (IQR) method.
- **IQR Definition:** The difference between the third quartile (Q_3) and the first quartile (Q_1).

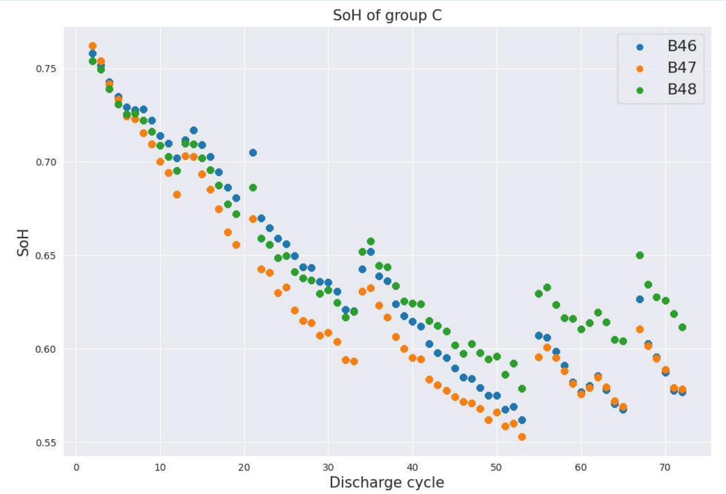
SOH Degradation Trends by Group

After data cleaning, the SOH versus Discharge Cycle was plotted for all batteries, grouped by their experimental conditions:

- **Group A (Normal Temp: B05, B07, B18):** Exhibits a relatively smooth and continuous SOH degradation trend.
- **Group B (High Power: B33, B34):** Shows a more scattered and complex degradation pattern, suggesting faster aging and more variance under high-power conditions.
- **Group C (Low Temp: B46, B47, B48):** Displays the steepest overall decline and significant fluctuation, characteristic of low-temperature degradation.


SoH of group A
B05
B07
B18
SoH
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
25
50
75
100
125
150
175
Discharge cycle


SoH of group B
B33
B34
SoH
0.85
0.80
0.75
0.70
0.65
0
25
50
75
100
125
150
175
200
Discharge cycle


SoH of group C
B46
B47
B48
SoH
0.75
0.70
0.65
0.60
0.55
0
10
20
30
40
50
60
70
Discharge cycle

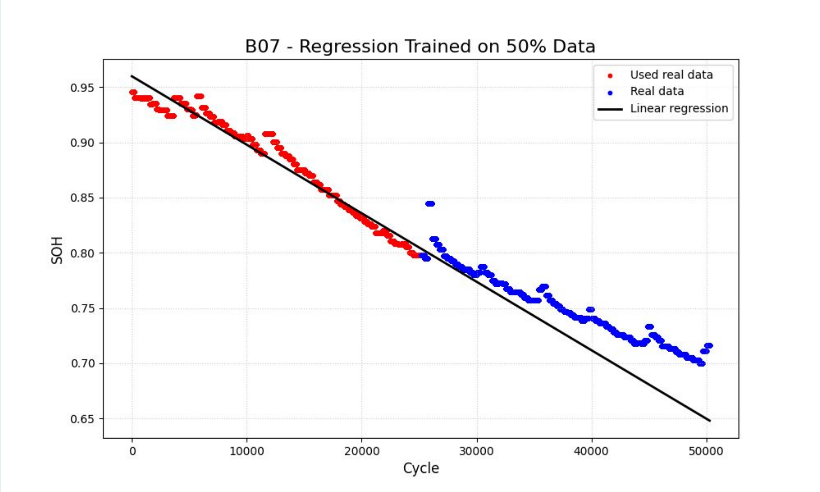
Linear Regression Model

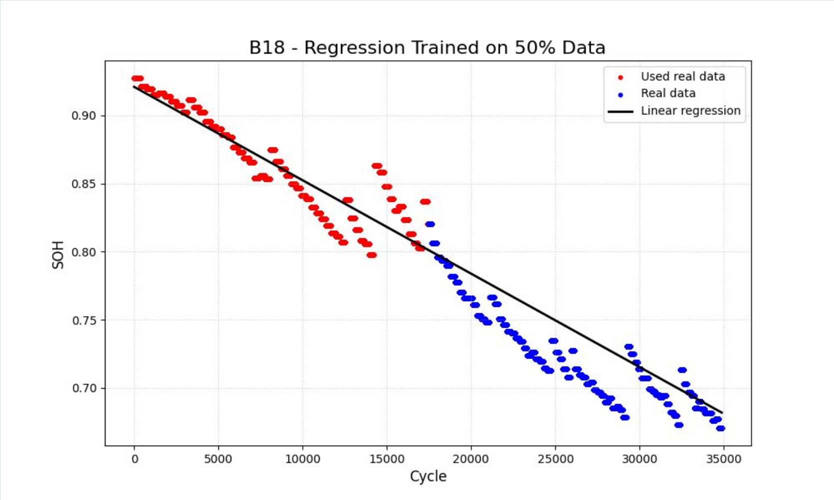
Linear regression was implemented as the **baseline model** to predict SOH based on the relationship between SOH and the discharge cycle.

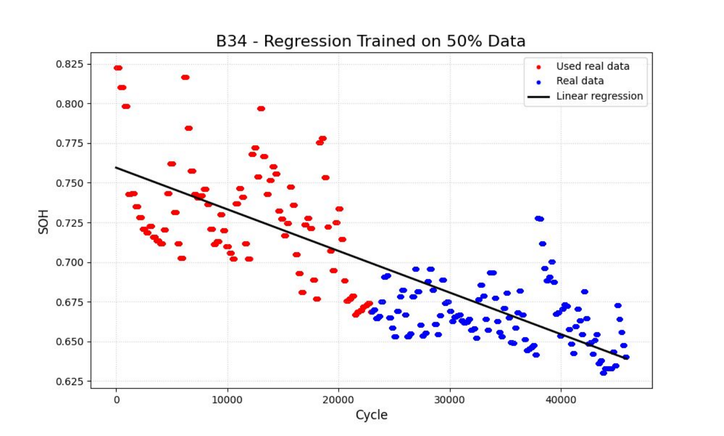
$$\hat{y}_i = a_1 x_i + a_0$$

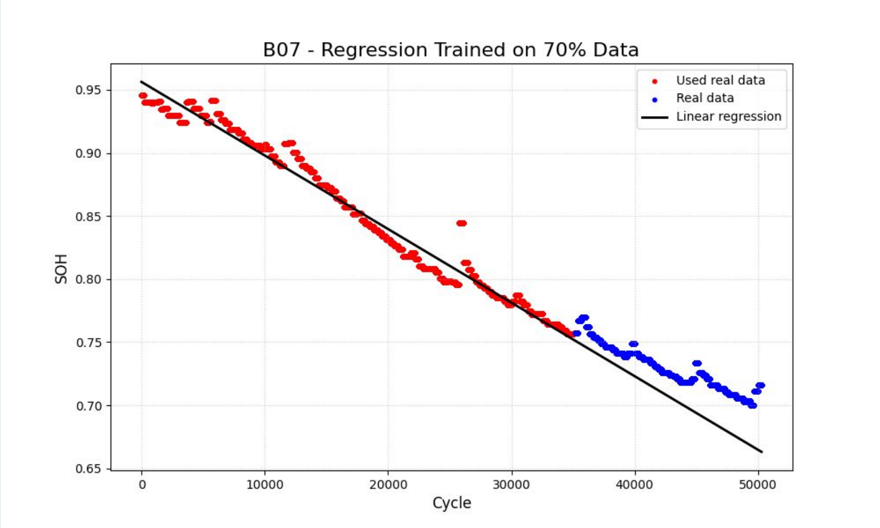
$$E \equiv E_2(a_0, a_1) = \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2$$

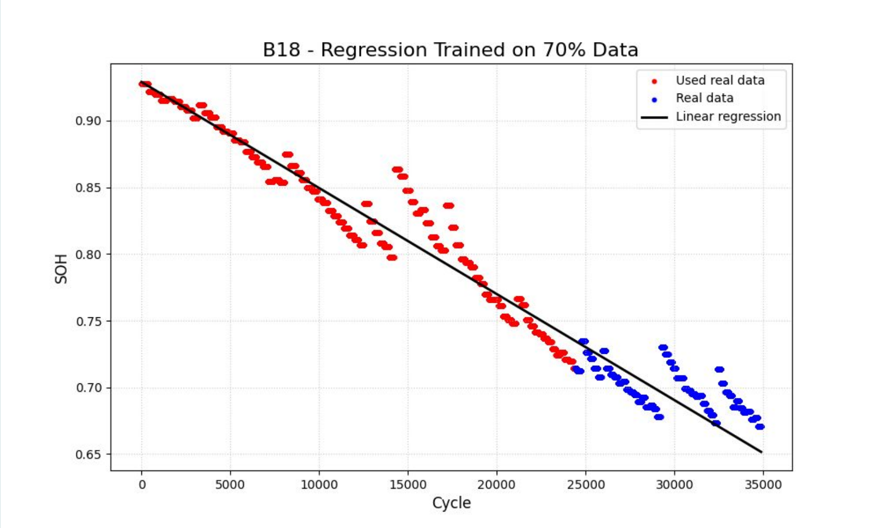
$$a_0 = \frac{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} x_i y_i \sum_{i=1}^{m} x_i}{m \sum_{i=1}^{m} x_i^2 - \left(\sum_{i=1}^{m} x_i\right)^2}, \qquad a_1 = \frac{m \sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m \sum_{i=1}^{m} x_i^2 - \left(\sum_{i=1}^{m} x_i\right)^2}.$$

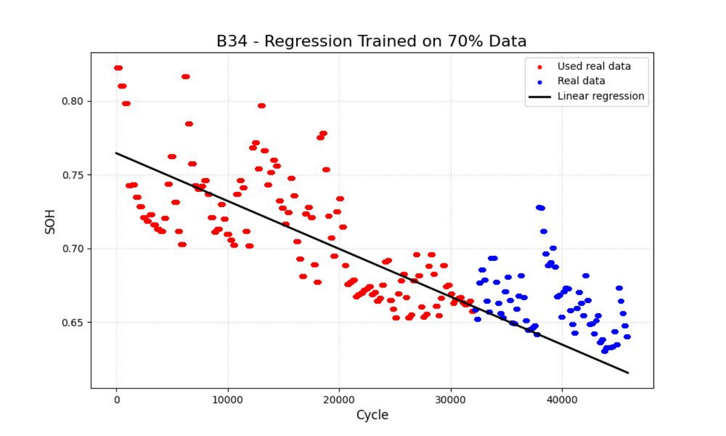

B07 - Regression Trained on 50% Data
Used real data
Real data
Linear regression
SOH
Cycle
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
10000
20000
30000
40000
50000


B18 - Regression Trained on 50% Data
Used real data
Real data
Linear regression
SOH
Cycle
0.90
0.85
0.80
0.75
0.70
0
5000
10000
15000
20000
25000
30000
35000

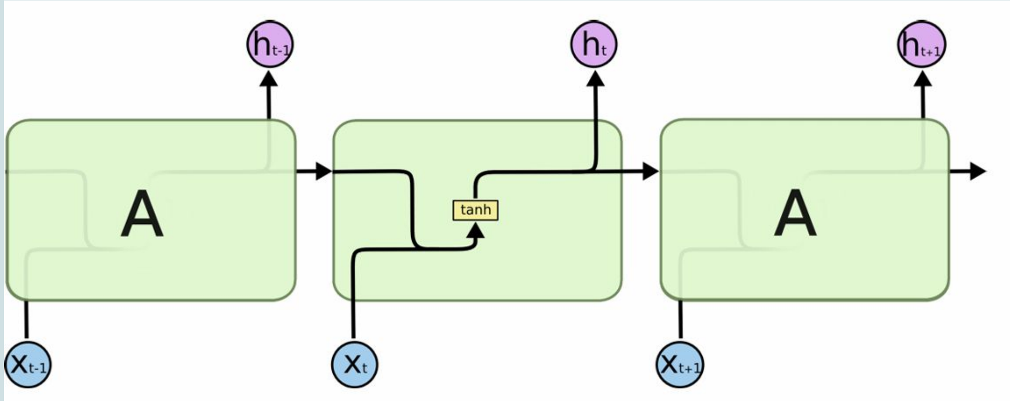

B34 - Regression Trained on 50% Data
Used real data
Real data
Linear regression
SOH
Cycle
0.825
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0.625
0
10000
20000
30000
40000


B07 - Regression Trained on 70% Data
Used real data
Real data
Linear regression
SOH
Cycle
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
10000
20000
30000
40000
50000


B18 - Regression Trained on 70% Data
Used real data
Real data
Linear regression
SOH
Cycle
0.90
0.85
0.80
0.75
0.70
0.65
0
5000
10000
15000
20000
25000
30000
35000

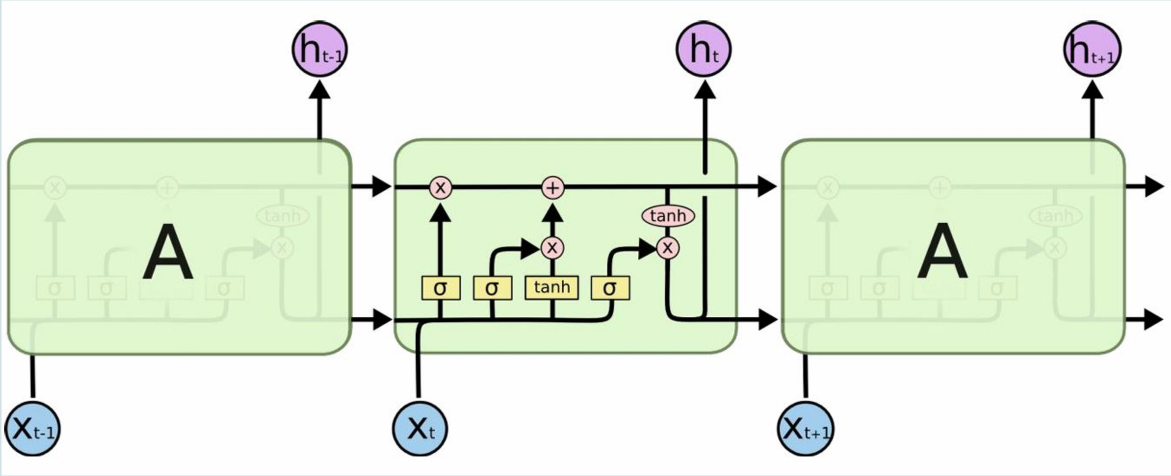

B34 - Regression Trained on 70% Data
Used real data
Real data
Linear regression
SOH
Cycle
0.80
0.75
0.70
0.65
0
10000
20000
30000
40000

RNN Concept: The output at time $h_{(t)}$ depends on the current input $x_{(t)}$ and the memory of the previous state $h_{(t-1)}$



$$
\begin{aligned}
h_t &= f_W(h_{t-1},\ x_t) \\
&= \tanh(w_{hh} h_{t-1} + w_{xh} x_t) \\
y_t &= w_{hy} h_t
\end{aligned}
$$

LSTM Advantage: LSTM solves the **vanishing gradient problem** of standard RNNs, making it highly effective for modeling **long-term dependencies** and time-series data like battery degradation.



$$f_t = \sigma(W_{xh\_f} x_t + W_{hh\_f} h_{t-1} + b_{h\_f})$$
$$i_t = \sigma(W_{xh\_i} x_t + W_{hh\_i} h_{t-1} + b_{h\_i})$$
$$o_t = \sigma(W_{xh\_o} x_t + W_{hh\_o} h_{t-1} + b_{h\_o})$$
$$\widehat{c_t} = tanh(W_{xh\_\hat{c}} x_t + W_{hh\_\hat{c}} h_{t-1} + b_{h\_\hat{c}})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t$$
$$h_t = o_t \odot \tanh(c_t)$$

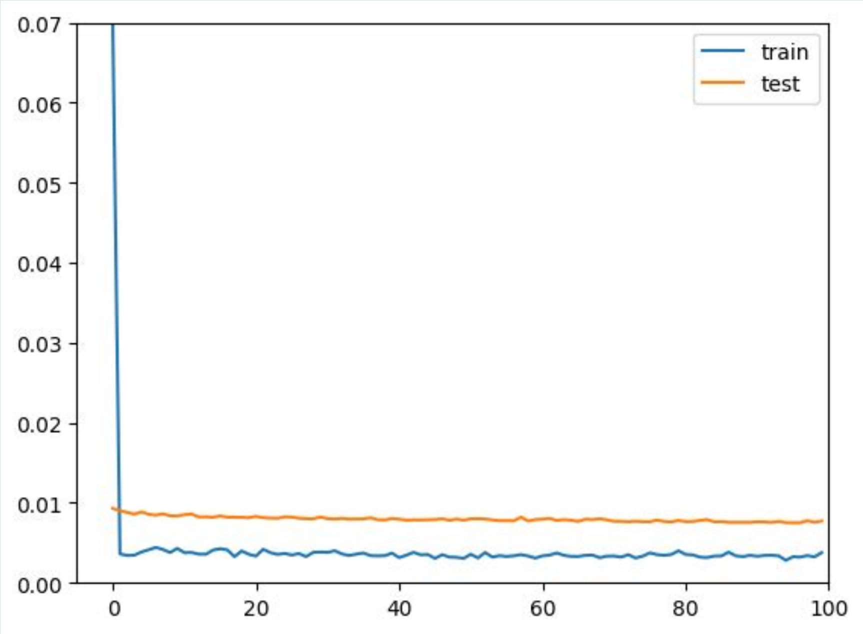
Gate Mechanism: LSTM uses three primary gates to regulate the flow of information:

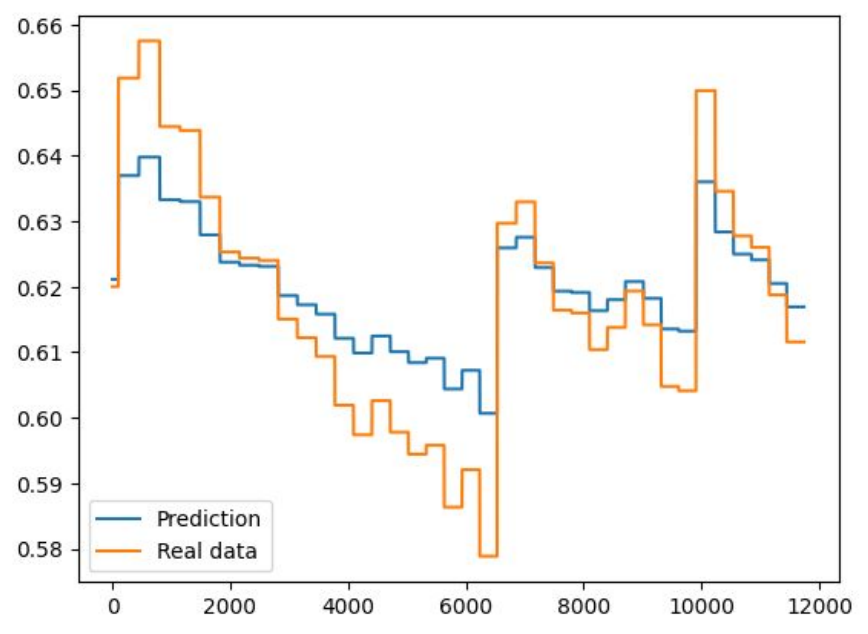
1. **Forget Gate:** Controls what information from the past cell state should be forgotten.
2. **Input Gate:** Controls what new information (current input) should be stored in the cell state.
3. **Output Gate:** Determines the next hidden state based on the new cell state .

Component	Code	Parameter/Value	Explanation
Model Type	`Sequential()`	-	A linear stack of layers, used for building a basic deep learning model.
Recurrent Layer	`model.add(LSTM(64, ...))`	**64**	Defines a single **LSTM layer** with **64 units (neurons)**. These 64 units are the model's internal memory cells that process the sequential SOH data.
Input Shape	`input_shape=(trainX.shape[1], trainX.shape[2])`	(1, 1)	Specifies the shape of the input data: **1 timestep** (since `look_back=1`) and **1 feature** (the SOH value).
Output Layer	`model.add(Dense(1))`	**1**	A single **Dense (fully connected)** neuron. This is the output layer responsible for producing the final, single SOH prediction value.
Loss Function	`model.compile(loss='mae', ...)`	**MAE** (Mean Absolute Error)	The function the model minimizes during training. MAE measures the average magnitude of the errors, calculating the absolute difference between the predicted and real SOH values.
Optimizer	`model.compile(..., optimizer='adam')`	**Adam**	A robust and highly effective optimization algorithm used to update the model weights. It efficiently finds the minimum of the loss function.

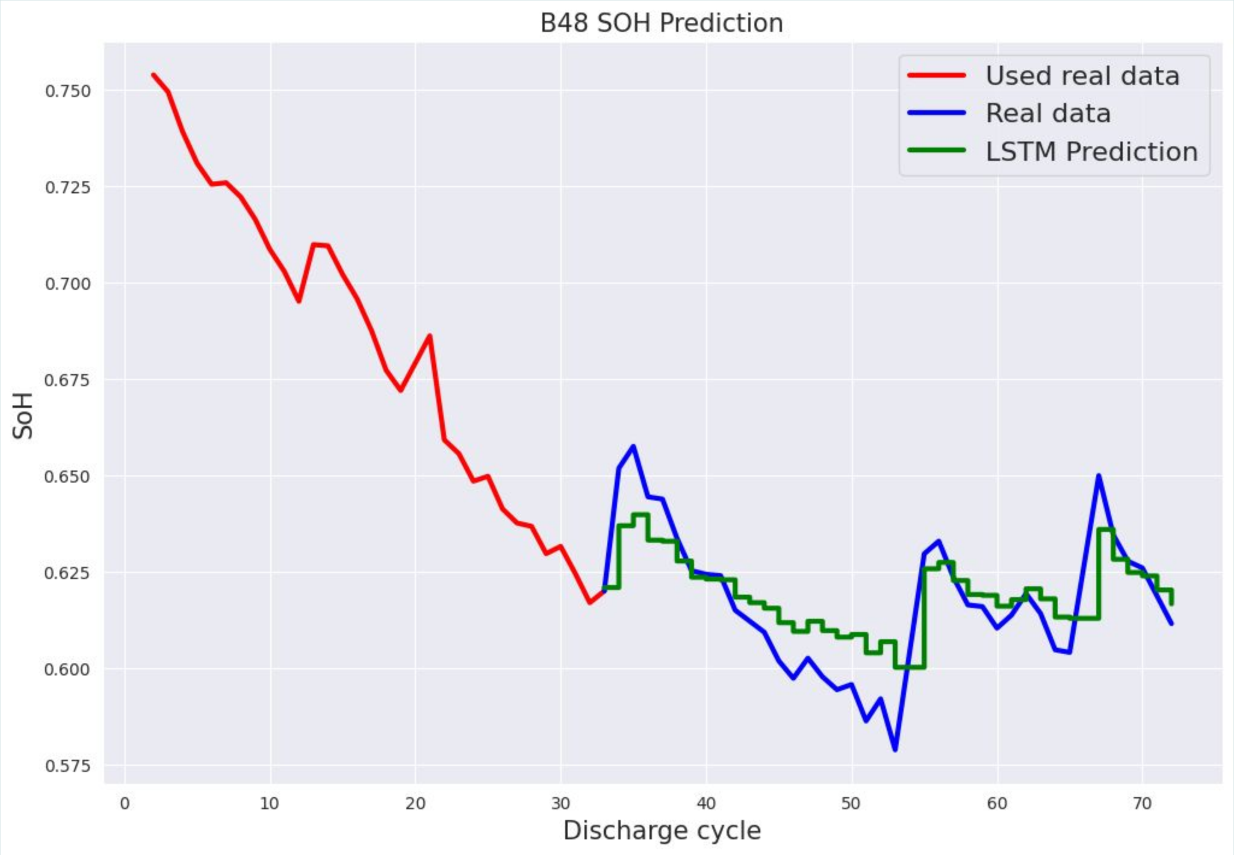
MODEL : TRAINING AND TESTING

Layer	Parameters
LSTM (64)	16,896
Dense (1)	65
Total Parameters	**16,961**



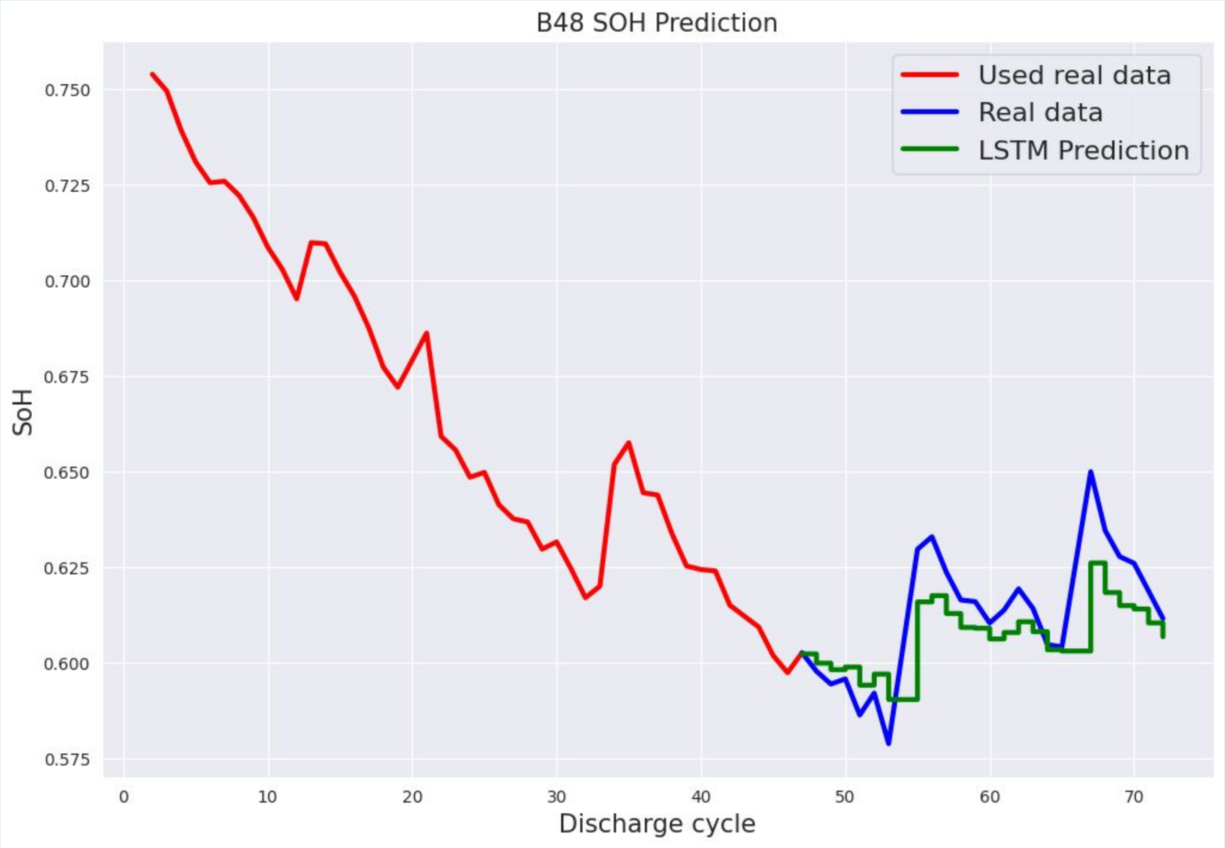


LSTM Prediction Results (50% Data Training)



Test RMSE: 0.009
Test MAE: 0.008(difference between model's predicted SOH and the actual SOH)

LSTM Prediction Results (70% Data Training)



Test RMSE: 0.010
Test MAE: 0.008

```
history = model.fit(trainX, trainY, epochs=100, batch_size=20, validation_data=(testX, testY), verbose=1, shuffle = False)
```

✓ 2m 43.2s

```
poch 1/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.0471 - val_loss: 0.0109
poch 2/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0037 - val_loss: 0.0093
poch 3/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0037 - val_loss: 0.0107
poch 4/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0039 - val_loss: 0.0098
poch 5/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.0037 - val_loss: 0.0129
poch 6/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0037 - val_loss: 0.0104
poch 7/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0037 - val_loss: 0.0095
poch 8/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - loss: 0.0039 - val_loss: 0.0107
poch 9/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0037 - val_loss: 0.0091
poch 10/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0037 - val_loss: 0.0092
poch 11/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0039 - val_loss: 0.0087
poch 12/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0031 - val_loss: 0.0090
poch 13/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.0035 - val_loss: 0.0088
..
poch 99/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0034 - val_loss: 0.0087
poch 100/100
22/822 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0032 - val_loss: 0.0084
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...