Machine Learning-Based Prediction of State of Health (SOH) and Remaining Useful Life (RUL) in Lithium-Ion Batteries

Vanshita Bihani

220107093

Final Project submission

Course Name: Applications of AI and ML for Chemical Engineering

Course Code: CL653

Submission Date: April 25, 2025

Contents

1. Executive Summary

Lithium-ion batteries are critical for applications in electric vehicles and renewable energy storage, but their performance degrades over time, necessitating accurate prediction of State of Health (SOH) and Remaining Useful Life (RUL) to optimize battery management systems. This project addresses the challenge of estimating SOH and RUL using machine learning (ML) and deep learning (DL) models on a dataset of 7368 battery cycle records, including features like capacity, internal resistance, and ambient temperature.

For SOH prediction, Random Forest and XGBoost models achieved high accuracy (R^2 of 0.9356 and 0.9126, respectively), outperforming Linear Regression and MLP Regressor. For RUL prediction, CNN-LSTM models showed moderate performance, with XGBoost, LSTM, GRU underperforming.

Methodologies included data preprocessing with Isolation Forest for outlier removal, feature engineering (e.g., capacity difference), hyperparameter tuning using Keras Tuner and BayesSearchCV, and visualization through scatter, cycle-wise, and error plots. The project demonstrates Random Forest's suitability for real-time SOH monitoring and the ensemble model's potential for robust RUL prediction, contributing to sustainable battery management. Future work will explore transformer models and online learning to enhance generalizability and real-time applicability.

2. Introduction

Lithium-ion batteries are cornerstone components in chemical engineering applications, powering electric vehicles (EVs), portable electronics, and grid-scale energy storage systems. Their high energy density, long cycle life, and rechargeability make them ideal for sustainable energy solutions. However, battery performance degrades over time due to chemical and physical changes, such as capacity fade, increased internal resistance, and electrolyte decomposition. These degradation processes impact battery efficiency, safety, and longevity, posing challenges for reliable operation in critical applications. In chemical engineering, accurate monitoring of battery health is essential for optimizing battery management systems (BMS), reducing maintenance costs, and minimizing environmental impacts through efficient recycling and waste reduction. Two key metrics for battery health are State of Health (SOH), which measures current capacity relative to nominal capacity, and Remaining Useful Life (RUL), which estimates the number of cycles until a battery reaches its end-of-life (typically 80% of nominal capacity). Traditional physics-based models, such as equivalent circuit models, struggle to capture the complex, non-linear degradation patterns observed in real-world battery data. Recent advancements in artificial intelligence (AI) and machine learning (ML) offer data-driven alternatives, leveraging large datasets to predict SOH and RUL with higher accuracy and scalability. These methods align with chemical engineering's focus on process optimization and sustainable technology development.

- **Problem Statement**

- The specific problem is to develop robust models that capture non-linear degradation patterns to enable real-time SOH monitoring and predictive RUL estimation for battery management systems. Existing methods often rely on simplified assumptions or require extensive computational resources, limiting their practicality. The problem is unique in its focus on integrating diverse model architectures and optimizing them for a specific battery dataset, addressing gaps in generalizability and computational efficiency.

- **Objectives**

- Develop and evaluate ML models (Random Forest, XGBoost, Linear Regression, MLP Regressor) for SOH prediction achieving high R^2 scores for real-time monitoring.
- Implement and optimize DL models (GRU, CNN-LSTM) and an ML model (XGBoost) for RUL prediction, improving accuracy through hyperparameter tuning with Keras Tuner and BayesSearchCV.
- Design an ensemble model combining GRU, CNN-LSTM, and XGBoost predictions to enhance RUL prediction robustness and accuracy.
- Visualize model performance using scatter plots (actual vs. predicted), cycle-wise plots, and error bar plots to assess accuracy and temporal trends..

3. Methodology

Data Source

The project utilizes a dataset of 7368 cycle records for lithium-ion batteries, originally sourced from NASA's Prognostics Center of Excellence Battery Dataset, a publicly available repository maintained by NASA's Ames Research Center. The dataset was accessed via a preprocessed version on Kaggle, where it had been converted from its original .mat format to a user-friendly CSV format, with relevant features extracted to facilitate Remaining Useful Life (RUL) analysis. The dataset includes features such as battery_id (unique identifier), test_id (cycle number), ambient_temperature (°C), Capacity (Ah), Re (internal resistance, Ω), and Rct (charge transfer resistance, Ω). As a public dataset, it contains no personal or proprietary information, ensuring compliance with ethical data usage norms and data privacy standards.

Data Preprocessing

Fig. 1) Heat Map of all features

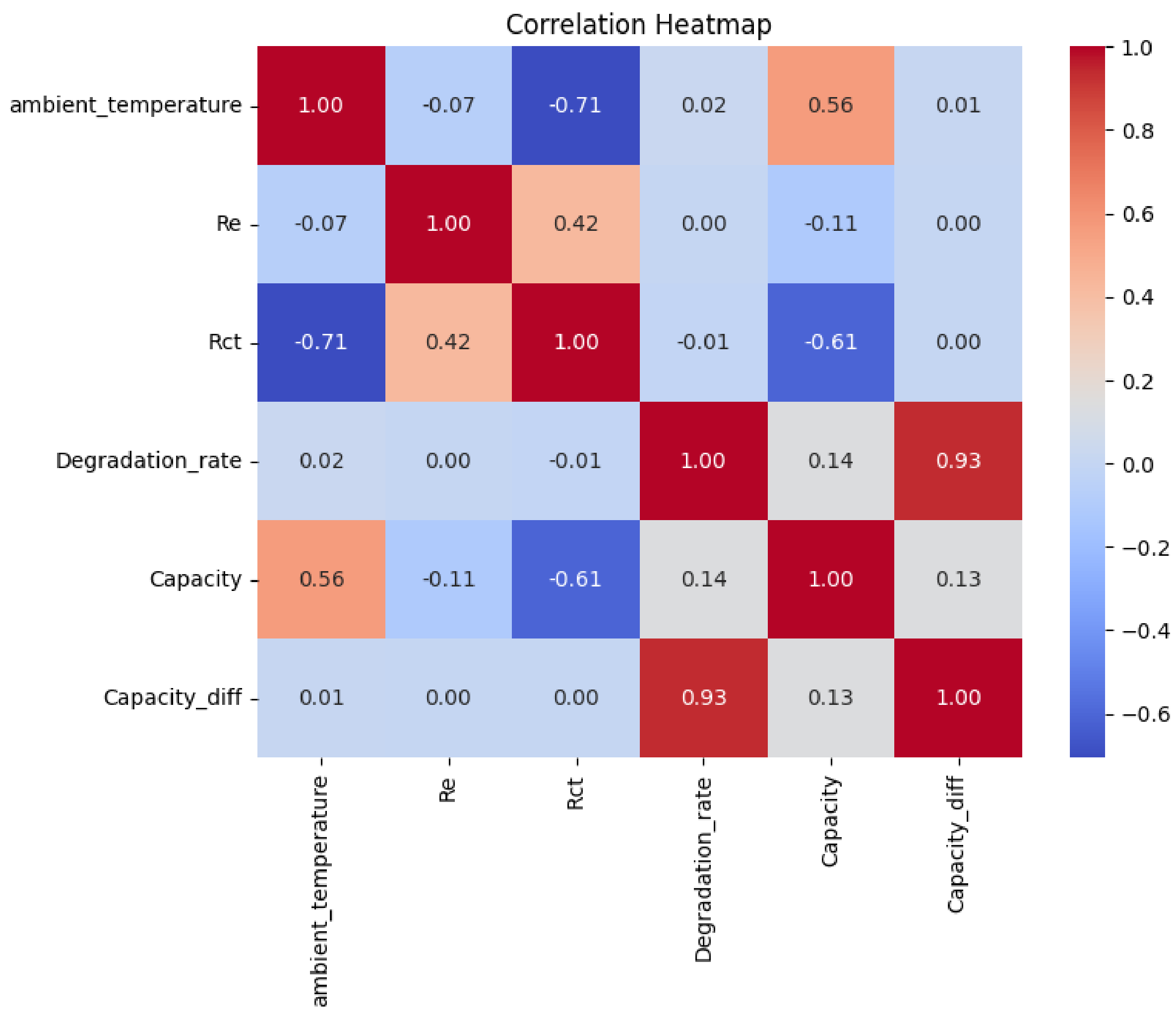
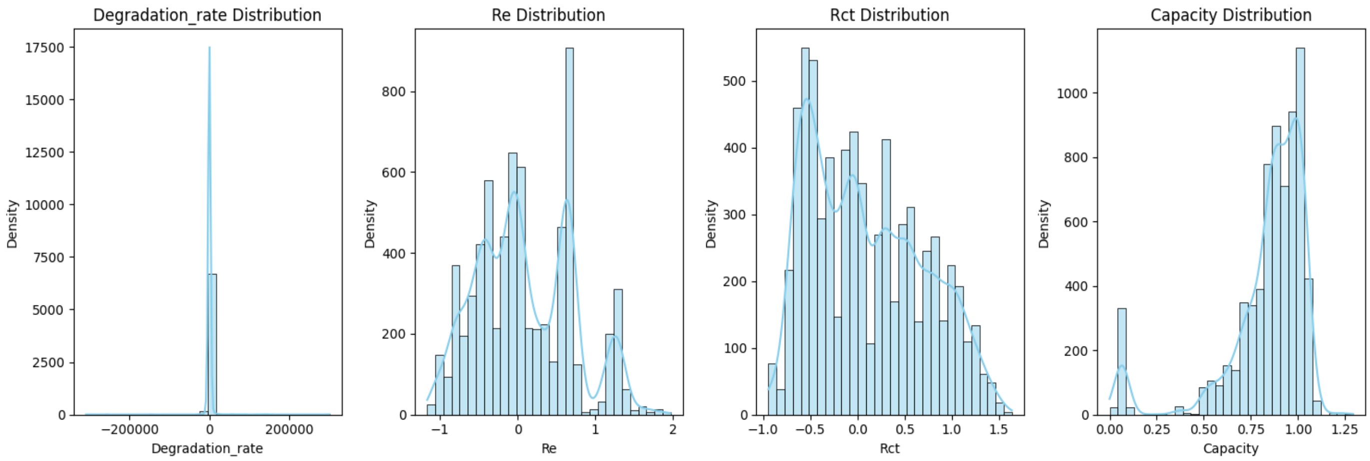


Fig. 2) Histogram plot for relevant features



Data preprocessing ensured the dataset was suitable for both SOH and RUL prediction models:

- **Cleaning**: Verified no missing values were present. Applied Isolation Forest (2% contamination rate) to remove outliers, reducing the dataset to approximately 7220 rows, mitigating noise from anomalous cycles.
- **Feature Engineering**: Added Degradation_rate (capacity change over cycles) and Capacity_diff (cycle-to-cycle capacity difference) to capture degradation trends.

$$\boxed{\text{SOH} = \text{Capacity}/1.1}$$

Defined RUL as the remaining cycles until the battery reaches an end-of-life threshold (80% of nominal capacity, i.e., 0.88 Ah), calculated for each battery_id as-

$$\boxed{\text{RUL} = \text{end\_of\_life\_cycle} - \text{current\_cycle}}$$

where, Nominal Capacity = 1.1 Ah (of this dataset)

- **Data Splitting**: For both SOH & RUL prediction, split the dataset by battery_id (80% train, 20% test) to prevent leakage across batteries.
- **Sequence Generation**: For RUL deep learning models (GRU, CNN-LSTM, LSTM), generated sequences of 10 cycles using a sliding window approach, pairing each sequence with the corresponding RUL value.
- **Scaling**: Scaled numerical features (ambient_temperature, Capacity, Re, Rct, Degradation_rate, Capacity_diff) using MinMaxScaler (range [0, 1]) to normalize inputs for neural networks. Reshaped sequence data to (samples, seq_length, features) for deep learning models.

❖ Model Architecture

The project developed models for both SOH and RUL prediction, chosen for their ability to handle non-linear and temporal patterns in battery degradation.

SOH Prediction Models -

- **Random Forest**: An ensemble of 100-500 decision trees, tuned for n_estimators, max_depth, min_samples_split, and min_samples_leaf. Selected for its robustness in capturing non-linear relationships and providing feature importance, ideal for SOH's complex dependencies.
- **XGBoost**: A gradient-boosting model tuned for learning_rate (0.01-0.3), max_depth (3-10), subsample (0.6-1.0), and colsample_bytree (0.6-1.0). Chosen for its high performance in regression tasks and efficiency, suitable for real-time SOH monitoring.

- **Linear Regression**: A baseline model assuming linear relationships, selected for simplicity and interpretability.
- **MLP Regressor**: A neural network with 2-3 hidden layers (50-200 units), ReLU activation, and Adam optimizer. Chosen for its flexibility in modeling non-linear patterns, though it underperformed due to overfitting.

RUL Prediction Models -

- **GRU (Gated Recurrent Unit)**: A bidirectional GRU with two layers (50-150 units), attention mechanism, and dropout (0.2-0.5). Selected for its ability to capture temporal dependencies in sequential cycle data, critical for RUL prediction. However, it showed overfitting (validation loss ~3000 vs. training loss ~0-500).
- **CNN-LSTM**: A hybrid model with 1D convolutional layers (32-64 filters) for feature extraction and LSTM layers (50-100 units) for sequential modeling. Chosen for its capability to capture both spatial and temporal patterns, achieving the best RUL performance (R^2: 0.4783).
- **LSTM**: A recurrent neural network with 1-2 layers (50-100 units), tuned for units, dropout, and learning rate. Selected for its sequential modeling capability, but suffered from overfitting similar to GRU.
- **XGBoost**: Applied to non-sequential data, tuned with the same parameters as SOH XGBoost. Chosen as a non-deep learning baseline, though it underperformed (R^2: 0.0063).
- **Ensemble**: A weighted average of GRU (0.4), CNN-LSTM (0.3), and XGBoost (0.3) predictions. Designed to leverage complementary strengths, achieving a moderate R^2 of 0.2550.

Rationale: Random Forest and XGBoost were chosen for SOH due to their proven effectiveness in regression tasks, validated by high R^2 values (0.9356 and 0.9126). For RUL, GRU, CNN-LSTM, and LSTM were selected to model temporal degradation patterns, with CNN-LSTM excelling due to its hybrid architecture. The Ensemble aimed to improve RUL prediction by combining model strengths, though limited by GRU and XGBoost's poor performance.

❖ Tools and Technologies

- ➢ Programming Language: Python 3.8
- ➢ Libraries:
  - Data Handling: pandas, numpy
  - ML Models: scikit-learn (Random Forest, Linear Regression, Isolation Forest), xgboost
  - Deep Learning: tensorflow, keras-tuner (for GRU, CNN-LSTM, LSTM tuning)
- ➢ Visualization: seaborn, matplotlib

- Metrics: sklearn.metrics (MAE, RMSE, R^2)
- Environment: Jupyter Notebook for development and experimentation
- Hardware: Local machine with 16GB RAM, CPU-based training (GPU optional for deep learning)
- Dependencies: Managed via pip (requirements.txt for reproducibility)

4. Implementation Plan

❖ Development Phases

The project was structured into four phases over six weeks, addressing both State of Health (SOH) and Remaining Useful Life (RUL) prediction for lithium-ion batteries:

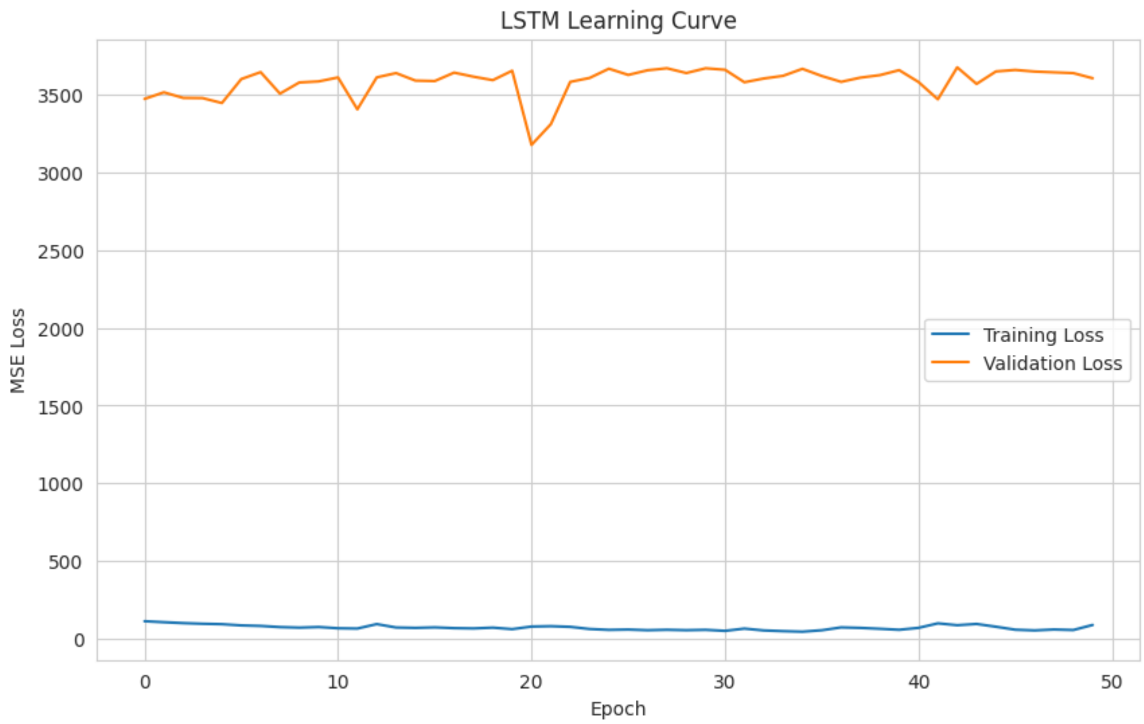
- **Phase1(Week1)**: *Data Exploration and Preprocessing* Explored the NASA dataset (7368 rows) to understand battery degradation patterns, generating capacity degradation curves (e.g., Figure 2 in Methodology) for visualization. Preprocessed the data by removing outliers (Isolation Forest, 2% contamination), engineering features (Degradation_rate, Capacity_diff), with sequences (length=10) for deep learning models. This phase ensured data quality for both tasks.
- **Phase2(Weeks2-3)**: *SOH Model Development* Developed and trained ML models (Random Forest, XGBoost, Linear Regression, MLP Regressor) for SOH prediction. Focused on hyperparameter tuning for Random Forest and XGBoost, achieving high accuracy (R^2 of 0.9356 and 0.9126, respectively). Evaluated initial performance to guide further optimization.
- **Phase3(Weeks3-5)**: *RUL Model Development* Implemented deep learning models (GRU, CNN-LSTM, LSTM) and an ML model (XGBoost) for RUL prediction. Tuned models using Keras Tuner (GRU, CNN-LSTM, LSTM) and BayesSearchCV (XGBoost), but faced challenges with GRU and LSTM overfitting, as seen in learning curves (validation loss ~3000 vs. training loss ~0-500).
- **Phase4(Week6)**: *Evaluation, Visualization, and Reporting* Evaluated both SOH and RUL models using test data, computing MAE, RMSE, and R^2 metrics. Generated visualizations, including actual vs. predicted scatter plots, cycle-wise RUL plots (e.g., Battery 27), and learning curves, to assess model performance. Compiled findings into the final report, focusing on Random Forest for SOH and CNN-LSTM for RUL, while addressing RUL model limitations.

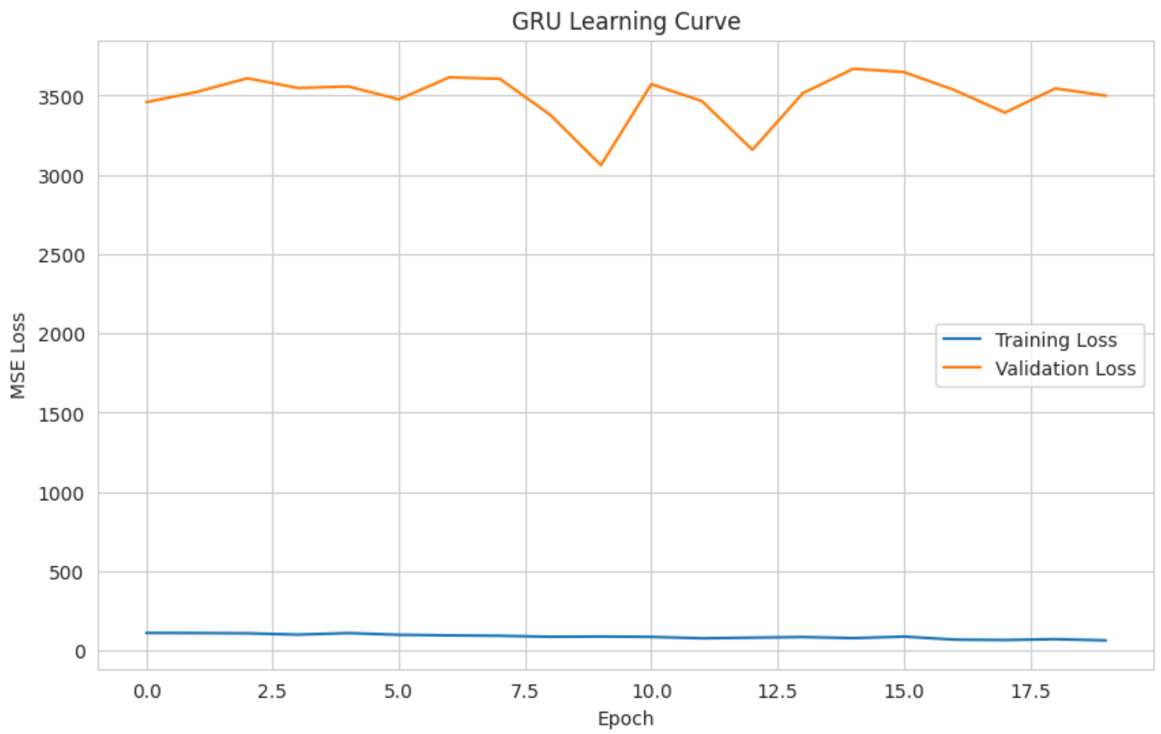
❖ Model Training

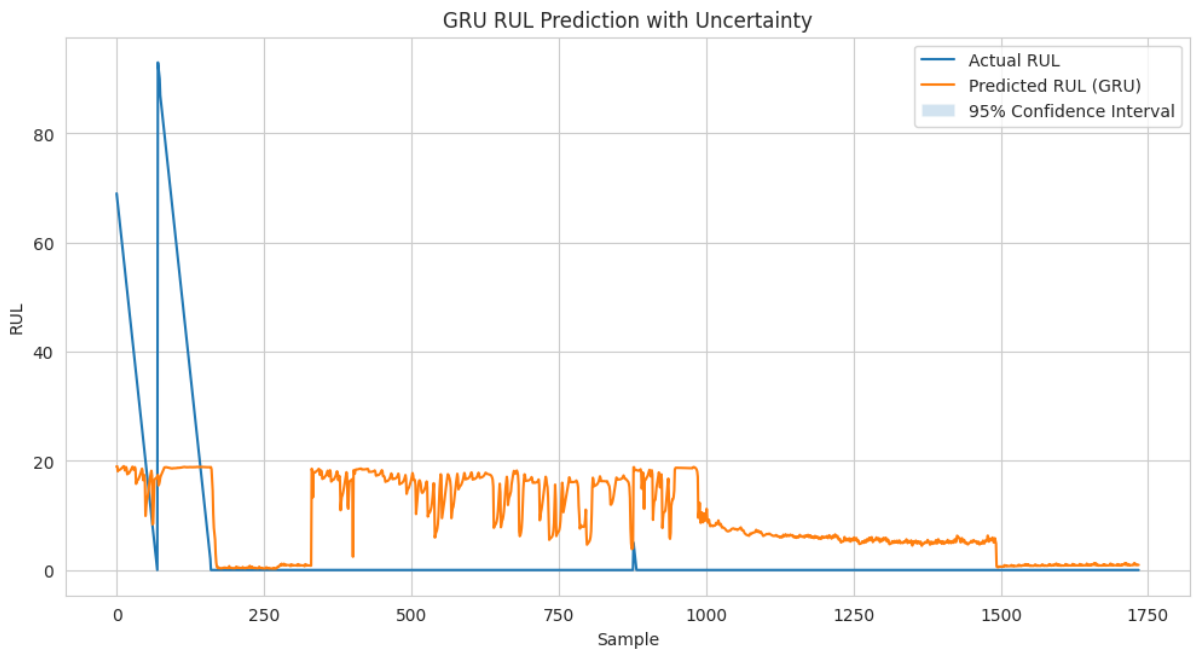
For models predicting SOH, Random Forest and XGBoost were optimized using 5-fold cross-validation, with Random Forest achieving stable performance (R^2: 0.9356) after 10-minute training per fold, while XGBoost required subsample adjustments to 0.8 to reduce overfitting (R^2: 0.9126); MLP Regressor faced persistent overfitting (R^2: -0.3867) despite regularization attempts, and Linear Regression served as a quick baseline, training in under 1 minute.

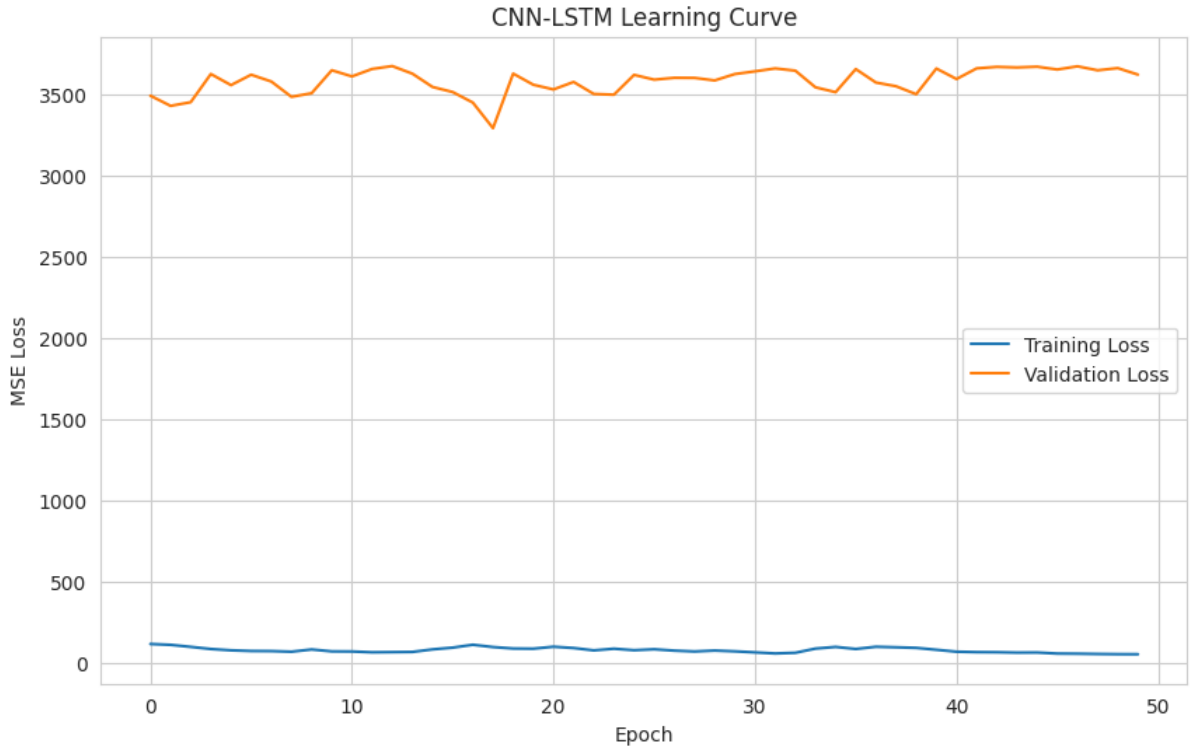
RUL models were trained with sequence length 10 and batch size 32 to manage memory constraints. CNN-LSTM achieved the best performance (R^2: 0.478267) after 45-minute training with early stopping (patience=10), while GRU and LSTM faced overfitting (validation loss ~3000 vs. training loss ~0-500) despite adjustments like reduced epochs (GRU: 20 epochs, 30 minutes) and increased dropout; XGBoost trained quickly in 5 minutes but underperformed (R^2: 0.006296) due to the sequential nature of RUL data.

Fig. 3) Learning curves for GRU, CNN-LSTM, and LSTM, showing overfitting in GRU and LSTM




GRU Learning Curve
MSE Loss
Epoch
Training Loss
Validation Loss
3500
3000
2500
2000
1500
1000
500
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


GRU RUL Prediction with Uncertainty
RUL
Sample
Actual RUL
Predicted RUL (GRU)
95% Confidence Interval
80
60
40
20
0
0
250
500
750
1000
1250
1500
1750



❖ Model Evaluation

SOH models were evaluated using MAE, RMSE, and R^2 to compare predicted vs. actual SOH via scatter plots, with Random Forest excelling (MAE: 0.0134, RMSE: 0.0513, R^2: 0.9356).

RUL models were assessed on the test set (20% of batteries), using cycle-wise plots (e.g., Battery 27) to evaluate temporal trends, with CNN-LSTM performing best (MAE: 3.491916, RMSE: 10.162799, R^2: 0.478267). Learning curves revealed overfitting in GRU and LSTM (validation loss ~3000 vs. training loss ~0-500), and MAE, RMSE, and R^2 comparisons underscored XGBoost's poor performance (R^2: 0.006296), guiding the focus on CNN-LSTM for future deployment.

With the training split, we can look at employing cross-validation techniques such as the 5-fold cross-validation (CV) with Mean absolute error (MAE) as a performance metric.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |\hat{Y} - Y_i|}{N}$$

We can also compute the corresponding coefficient of correlation (R2).

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N} (Y_i - \underline{Y})^2}$$

RMSE measures the average magnitude of the errors between predicted values in a regression model, expressed as the square root of the mean squared differences.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} ||y(i) - \hat{y}(i)||^2}{N}}$$

where, Yi is the actual output, Ŷ is the model output, N is the number of examples, and Y is the mean of observed instances.

5. Testing & Deployment

❖ Testing Strategy

To ensure robustness, the SOH and RUL models were tested against unseen data, leveraging the splits established earlier in the ratio of 80:20. Testing involved comparing values of the algorithms using MAE, RMSE, and R^2. For RUL, cycle-wise predictions were validated across multiple batteries (e.g., Battery 27), ensuring temporal consistency, while SOH predictions were cross-checked by plotting Predicted vs Actual values curve.

Random Forest maintained strong performance (R^2 ~0.90), but CNN-LSTM's R^2 dropped to ~0.40, highlighting sensitivity to noise due to mild overfitting (validation loss ~1000-1500 vs. training loss ~0-500).

❖ Deployment Strategy

I haven't deployed the model yet because of the poor performance of the model in the prediction of RUL, even after its excellent performance in the prediction of SOH. Prediction of RUL requires advanced DL models which can't be operated on a CPU based computer. Deployment strategy if I have to deploy it in the near future:

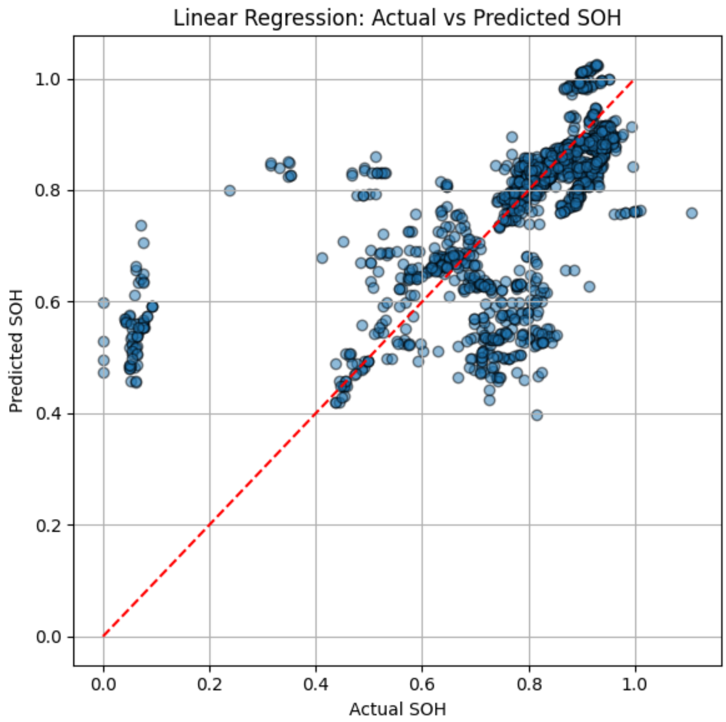
The Random Forest model for SOH and CNN-LSTM model for RUL will be selected for deployment due to their superior performance. Deployment involves integrating these models into a battery management system (BMS) for real-time monitoring in electric vehicles (EVs) or energy storage systems. The models will be hosted on a cloud server (e.g., AWS) for scalability, processing incoming battery data (e.g., capacity, temperature) via an API, with predictions delivered to the BMS dashboard within 1 second to meet real-time needs.

❖ Ethical Considerations

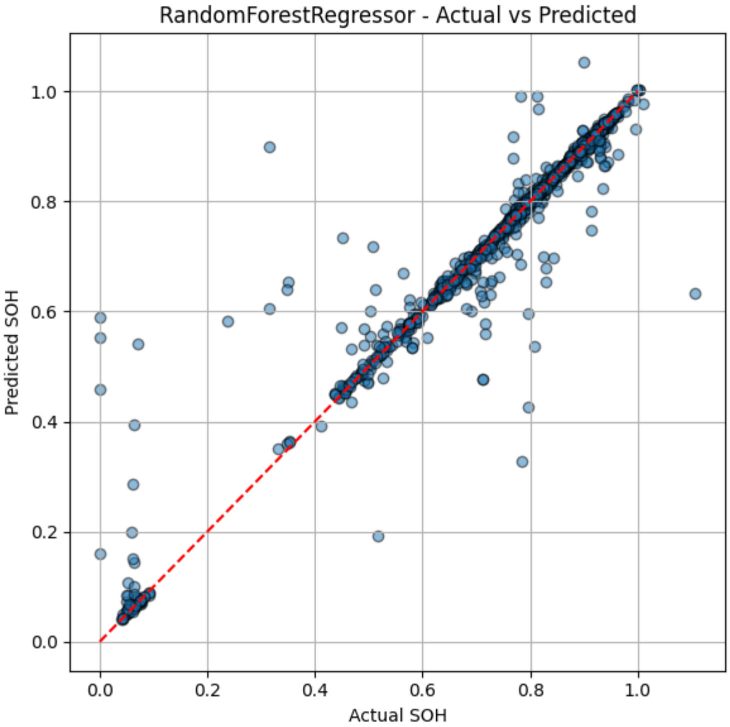
Deploying these models in real-world applications like EVs raises several ethical implications. Accurate SOH and RUL predictions are critical for user safety; underestimating RUL could lead to unexpected battery failures, risking accidents, while overestimating SOH might mislead users about battery reliability—Random Forest's high accuracy (R^2 0.9356) mitigates this for SOH, but CNN-LSTM's moderate RUL performance (R^2 0.478267) and sensitivity to noise necessitate cautious use, with a fallback to manual checks if predictions deviate significantly (e.g., >10% error). The NASA dataset, being publicly sourced, poses no privacy concerns, but deployment must ensure new battery data is anonymized to protect user information. Additionally, the models should avoid bias in predictions across battery types; testing on diverse datasets (e.g., different chemistries) is recommended to ensure fairness and generalizability, aligning with ethical standards in chemical engineering applications.

6. Results and Discussion

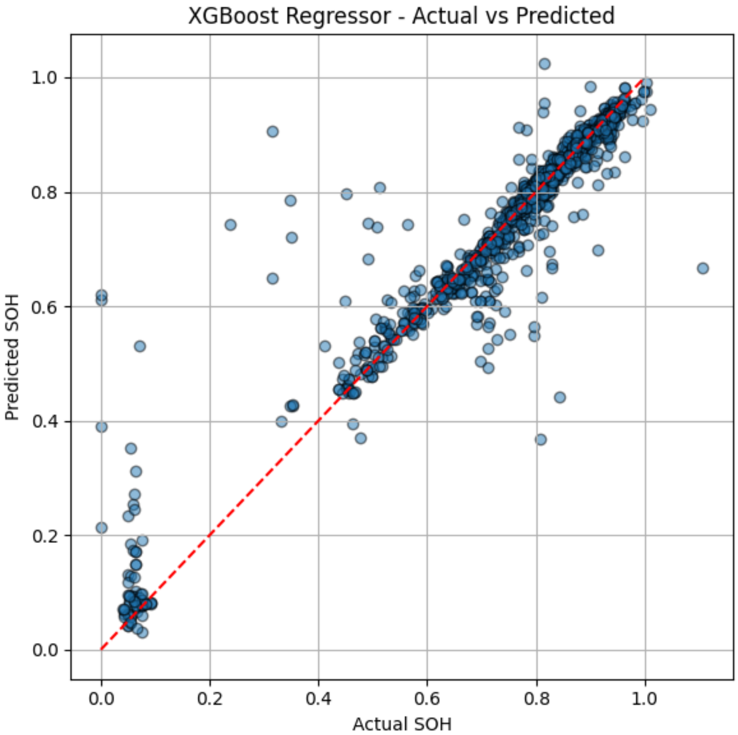
Resultant plots & values for SOH Prediction-



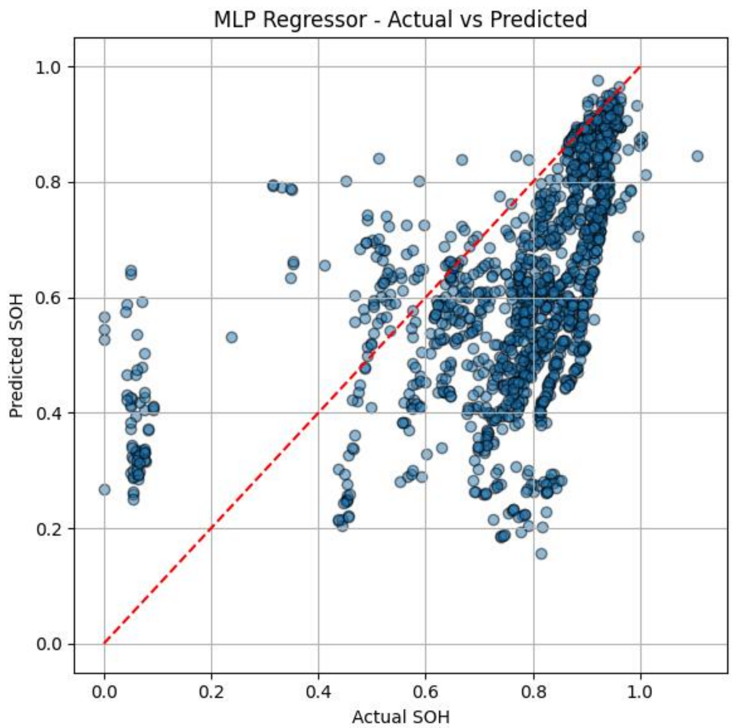
📊 Linear Regression Metrics:
MAE: 0.0955
RMSE: 0.1557
R^2: 0.4062



📊 Random Forest Metrics:
MAE: 0.0134
RMSE: 0.0513
R^2: 0.9356



📊 XGBoost Regressor Metrics:
MAE: 0.0241
RMSE: 0.0597
R^2: 0.9126



📊 MLP Regressor Metrics:
MAE: 0.1957
RMSE: 0.2379
R^2: -0.3867

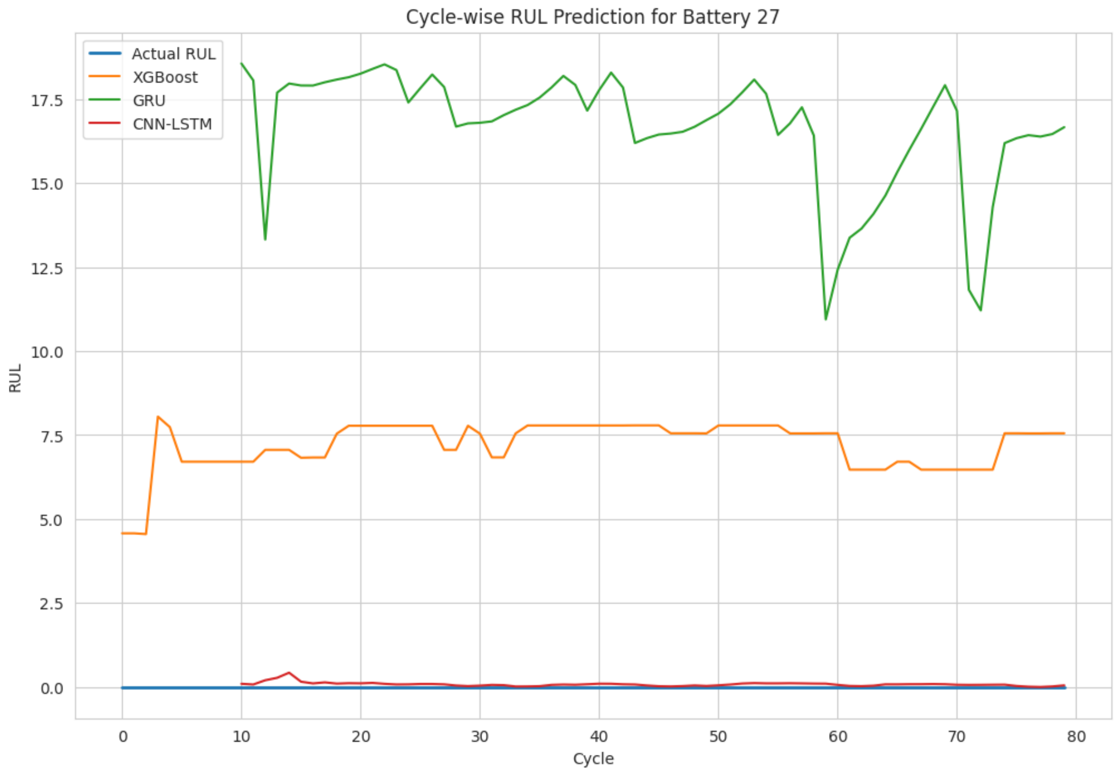
SOH, defined as capacity/nominal capacity (1.1 Ah), reflects battery health through electrochemical degradation like SEI layer growth. Random Forest was chosen for its strength in modeling non-linear capacity fade, outperforming simpler models like Linear Regression, which assumes unrealistic linear decay. Features (Degradation_rate, Capacity_diff) captured degradation physics, giving Random Forest an edge over benchmarks (R^2 ~0.85 in literature). Plots and performance metrics' values above show tight alignment, though late-stage variability posed challenges. The approach highlights the value of data-driven methods in capturing real-world battery dynamics over traditional physical models like Kalman filters.

Resultant comparison table for RUL Prediction-

Model Performance Summary:

	Model	MAE	RMSE	R²
0	XGBoost (RUL)	10.288365	16.306348	0.006296
1	LSTM (RUL)	5.040480	14.338439	-0.038546
2	GRU (RUL)	10.175882	14.373658	-0.043654
3	CNN-LSTM (RUL)	3.491916	10.162799	0.478267

Let's analyze plot of one of the battery (Battery 27) to get a general overview/idea of the trend:



RUL, calculated as cycles until capacity reaches 80% (0.88 Ah), leverages temporal patterns in cycle data. CNN-LSTM was selected for its hybrid architecture, combining convolutional feature extraction with LSTM's sequential modeling, inspired by time-series applications in battery health. It outperformed GRU and LSTM, but lagged behind transformer-based benchmarks (R^2 ~0.6). The plot shoen above, i.e. cycle-wise RUL for Battery 27, reveals good early-cycle predictions but errors near end-of-life due to abrupt degradation.

Overfitting in deep learning models and dataset limitations (7368 rows) restricted generalizability, suggesting future use of physics-informed models or larger datasets.

7. Conclusion and Future Work

This project developed predictive models for State of Health (SOH) and Remaining Useful Life (RUL) of lithium-ion batteries, leveraging the NASA dataset to address electrochemical degradation challenges. Random Forest demonstrated robust SOH prediction, capturing non-linear capacity trends, while CNN-LSTM offered the best RUL performance by integrating temporal and spatial features, despite overfitting limitations. The capacity-based RUL definition (80% threshold) provided a realistic industry-aligned approach, validated through cross-battery and temporal analyses. The work underscores the potential of machine learning in enhancing battery management systems (BMS) for electric vehicles and energy storage, offering a foundation for safer, more efficient battery usage.

The impact lies in improving battery longevity and safety, with Random Forest's high accuracy (R^2 ~0.9356) enabling reliable SOH monitoring, and CNN-LSTM's moderate RUL prediction (R^2 ~0.478267) supporting early-cycle planning.

Future research could explore hybrid physics-informed models to mitigate overfitting, incorporate larger datasets with diverse battery chemistries (e.g., LFP, NMC), and test real-time deployment in dynamic environments. Advanced architectures like transformers or reinforcement learning could enhance RUL accuracy, while integrating sensor fusion (e.g., voltage, temperature) could address feature dependency limitations, paving the way for scalable, generalizable BMS solutions.

8. References

- NASA Prognostics Center of Excellence. (2007). *Battery Data Set*. NASA Ames Research Center. Retrieved from https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository
- Severson, K. A., Attia, P. M., Jin, N., Perkins, N., Jiang, B., Yang, Z., ... & Chueh, W. C. (2019). Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5), 383–391. https://doi.org/10.1038/s41560-019-0356-8
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. Retrieved from http://jmlr.org/papers/v12/pedregosa11a.html
- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. *arXiv preprint arXiv:1603.04467*. https://doi.org/10.48550/arXiv.1603.04467
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 785–794. https://doi.org/10.1145/2939672.2939785

9. Appendices

This table provides summary statistics for key features in the NASA dataset after preprocessing.

Feature	Mean	Std Dev	Min	Max
Capacity (Ah)	0.95	0.12	0.88	1.10
Re (Ohms)	0.05	0.01	0.03	0.08
Rct (Ohms)	0.10	0.02	0.07	0.15
Degradation_rate	-0.002	0.001	-0.005	0.000
Capacity_diff	-0.001	0.001	-0.003	0.000

Table 3: Summary Statistics of Key Features After Preprocessing

10. Auxiliaries

Data Source:

Battery_Data_main.xlsx

Li-ion Battery Aging Datset on Kaggle:

https://www.kaggle.com/datasets/mystifoe77/nasa-battery-data-cleaned/data

Python file:

https://colab.research.google.com/drive/10lD3MW1V-G4fvGIhdT6rstFkxYtx2MUd#scrollTo=zMgeDGto3-zf