MINI PROJECT REPORT

Submitted to the faculty of Engineering and Technology VI Semester B.Tech

(Autonomous Batch)

A Mini Project report on

DATA-DRIVEN MACHINE LEARNING APPROACHES FOR LITHIUM-ION BATTERIES RUL PREDICTION

BY

ADI SAI KIRAN

B22AI009

Under the Guidance of

Dr. A. Jothi Prabha

Associate Professor

Department of Computer Science & Engineering (AI&ML)

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE, WARANGAL

(An Autonomous Institute under Kakatiya University)

Warangal (Telangana State)

2024-25

ISO 9001:2015 AICTE-CII: GOLD Category Institute NAAC-'A' Grade Institute (CGPA: 3.21) NIRF-2021 Rank : 197

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE
Opp : Yerragattu Gutta, Hasanparthy (Mandal), WARANGAL - 506 015, Telangana, INDIA.
काकतीय प्रौद्योगिकी एवं विज्ञान संस्थान, वरंगल - ५०६ ०१५ तेलंगाना, भारत
కాకతీయ సాంకేతిక విజ్ఞాన శాస్త్ర విద్యాలయం, వరంగల్ - ౫౦౬ ౦౧౫ తెలంగాణ, భారతదేశము
Estd-1980
KITSW
(An Autonomous Institute under Kakatiya University, Warangal)
(Approved by AICTE, New Delhi; Recognised by UGC under 2(f) & 12(B); Sponsored by EKASILA EDUCATION SOCIETY)
website: www.kitsw.ac.in E-mail: principal@kitsw.ac.in ✆ : +91 9392055211, +91 7382564888

CERTIFICATE

This is to certify that **ADI SAI KIRAN** bearing roll no **B22AI009** of the VI Semester B.Tech. Computer Science and Engineering (AI & ML) has satisfactorily completed the Mini Project dissertation work entitled **"Data-Driven machine learning approaches for Lithium-ion Batteries RUL Prediction"**, in partial fulfillment of the requirements of the B.Tech degree during the academic year 2024-25.

Supervisor
Dr. A. Jothi Prabha
Associate Professor

Coordinator
Dr. M. Rajesh
Assistant Professor

Convener
Dr. S. Raghu
Assistant Professor

Head of the Department
Prof. S. Narasimha Reddy

ACKNOWLEDGMENT

I extend my sincere thanks to our esteemed guide, **Dr. A. Jothi Prabha, Associate Professor,** for her exemplary guidance, monitoring, and constant encouragement throughout the course at crucial junctures and for showing us the right way.

I am grateful to the respected coordinator, **Dr. M. Rajesh, Assistant Professor,** for guiding me and permitting me to utilize all the necessary facilities of the Institute.

I sincerely thank the respected convener, **Dr. S. Raghu, Assistant Professor,** for supporting me and for utilizing all the necessary facilities of the Institute.

I would like to extend thanks to our respected head of the department, **Prof. S. Narasimha Reddy,** for allowing us to use the facilities available. I would like to thank other faculty members also.

I would like to thank all the faculty members, friends, and family for the support and encouragement that they have given us during the seminar.

ADI SAI KIRAN
B22AI009

ABSTRACT

The rapid growth of electric vehicle technology has heightened the need for efficient and sustainable energy storage solutions, particularly lithium-ion batteries. Accurate prediction of the Remaining Useful life of these batteries is crucial for improving performance, safety, and cost- cost-effectiveness. This study explores the use of various machine learning methods to develop predictive models for battery health monitoring and degradation analysis. A dataset obtained from the NASA Ames Prognostics Center of Excellence will be utilized to train and evaluate the models. A systematic approach will be employed, incorporating feature selection, optimization techniques, and real-time operational factors such as temperature fluctuations, charging/discharging cycles, and load variations. The study aims to enhance predictive maintenance strategies, reduce unexpected battery failures, and improve energy management systems. The proposed methodology will contribute to prolonging battery lifespan, optimizing EV performance, and supporting the development of more sustainable and economical energy solutions.

CONTENTS

		Page No.
ABSTRACT		**i**
CONTENTS		**ii**
LIST OF FIGURES		**iii**
1	**INTRODUCTION**	**01**
1.1	Introduction	01
1.2	Objectives	01
1.3	Literature Review	02
2	**IMPLEMENTATION**	**03**
2.1	Model	03
2.2	Methodology	04
2.3	System Requirements	05
2.4	Applications	06
2.5	Advantages	07
2.6	Challenges	07
3	**EXPERIMENTATION AND RESULTS**	**08**
3.1	Experimentation	08
3.2	Code	11
3.3	Results	14
	3.31 Capacity Prediction	14
	3.3.2 Generated Output	15
4	**CONCLUSION & FUTURE SCOPE OF WORK**	**16**
REFERENCES		**17**

LIST OF FIGURES

Fig.No	Title	PageNo.
3.1	Capacity vs cycles of batteries	14
3.2	Model output for input data	15

1. INTRODUCTION

The increasing demand for reliable and long-lasting energy storage systems has led to a surge in research around Lithium-ion batteries. Accurate prediction of battery's Remaining Useful Life (RUL) is critical for preventive maintenance, safety, and cost efficiency. This project focuses on leveraging operational features such as voltage, current, temperature, and cycle data to estimate battery capacity and RUL. Prognostic health management describes about the historical data and knowledge to identify the factors influencing machine failure and predicting Remaining Useful Life.

Batteries considered reaching End-of-life (EOL) is when battery capacity degrades to 70 percent of its initial capacity. The number of cycles for the battery to reach end-of-life from its current capacity is called Remaining Useful Life of battery. For machines, RUL depends on natural ageing, but for batteries it depends on cycles. The non-linear degradation pattern in lithium-ion batteries makes it difficult to predict the capacity degradation using linear models. The battery capacity degradation is because of charge and discharge cycles and temperature fluctuation during these cycles. The dataset used in this is battery dataset from NASA Ames Prognostics Center of Excellence (PCoE). Various machine learning models are employed in this project.

1.2 OBJECTIVES

The battery capacity degradation is because of charge and discharge cycles and temperature fluctuation during these cycles.Various machine learning models are employed in this project. The main objective of this project is prognostic health management of lithium-ion batteries analyzing the factors effecting the battery capacity degradation.

1. To create a model that predicts the lithium-ion batteries capacity using factors voltage, current, temperature.
2. To create a model that predicts remaining useful Life of lithium-ion batteries using charge and discharge time, initial and current capacity.
3. To analyse the relationship between battery degradation parameters and performance, thereby improving predictive accuracy and enabling early fault detection

1.3 LITERATURE SURVEY

Over the past few years, numerous research has been conducted in battery state of health field. Research conducted on capacity degradation prediction and state of health prediction of batteries using different machine learning and deep learning architectures. Three methods are employed for useful life prediction of batteries. They are model-based model, data data-driven and a hybrid method. Model-based method explains internal state of battery which changes during the electrochemical reaction during charging and discharging cycle. Mathematical formulas are used in this method, which requires knowledge of the domain. Data-driven method using the history of battery at different cycles and uses this history to develop a machine learning model.

Swain et al. [1] proposed a machine learning model for predicting RUL in EV batteries, demonstrating that ensemble methods could effectively capture non-linear patterns in degradation behavior. Tiane et al. [2] introduced a deep learning hybrid architecture that integrated Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Their work pointed to the power of deep learning to detect both spatial and temporal patterns for prognosis tasks.

Severson et al. [3], in a pioneering work, employed early-cycle data for battery lifespan prediction through data-driven methods. This was shown to have the capability of predicting cycle life prior to extensive degradation and hence minimize reliance on extended monitoring.Yang et al. [5] investigated RUL prediction with incomplete degradation history via Random Forests and demonstrated that even with incomplete information, strong predictions were possible. Earlier establishment by Dubarry et al. [6] introduced Incremental Capacity Analysis (ICA) and OCV measurements to quantify capacity fade, giving the groundwork for data-driven RUL estimation.

Zhang et al. [7] implemented an LSTM-based method for sequence modeling in battery RUL estimation, successfully addressing temporal dependencies over charge-discharge cycles. Li et al. [9] proposed a hybrid model that fuses deep neural networks with hand-designed features for enhancing prediction performance in capacity degradation.

2. IMPLEMENTATION

2.1 MODEL

A two-stage machine learning model was proposed to well predict the battery capacity and the Remaining Useful Life (RUL) of lithium-ion batteries. The hybrid model uses Random Forest Regressors in both stages, which are optimized to have high interpretability and performance in dealing with complex, nonlinear feature relationships.

Stage one of the model involves predicting the battery capacity with cycle-wise, averaged sensor values from four given batteries (B0005, B0006, B0007, B0018). Time-series values—voltage, current, and temperature when charging and discharging—were averaged using their mean values for every cycle in order to alleviate noise and reduce dimensionality. This initial estimation is important for knowing the battery's degradation pattern over its life.

The second phase uses the output of the initial model—forecast or measured capacity—or other cycle-specific characteristics of 32 batteries to predict the RUL. Charging/discharging time, min and max voltages, and cycle index are among the features employed to capture both current state and working history of the battery, which are essential in assessing how much usable life is left before capacity declines below acceptable levels.

The two-stage model hence gains from initial degradation pattern identification and better RUL estimation, thus being a resilient solution to predictive battery maintenance. For assurance of the models' reliability and generalizability, strict assessment was carried out based on key regression metrics, namely Mean Absolute Error (MAE), Mean Squared Error (MSE), and the R^2 Score.

The models showed excellent performance, with high predictive accuracy and low error rates in both stages. Moreover, GridSearchCV was used for hyperparameter tuning, enabling fine-tuning of the Random Forest parameters like the number of estimators and maximum tree depth. To avoid overfitting and enhance model robustness, early stopping and cross-validation methods were used during training. This broad modeling approach delivers a scalable and effective method apt for use in electric vehicle battery management systems and energy storage solutions.

2.2 METHODOLOGY

The implementation began by converting .mat files into Python-readable .pkl files, enabling efficient data handling and preprocessing. Charging and discharging phases data were combined per cycle according to battery ID and cycle number. Nine important features (voltage, current, temperature while charging and discharging, and time metrics) were chosen for Model 1 (Capacity Prediction). Mean values were determined to represent each cycle. The target variable was capacity of the battery, and model was trained by Random Forest Regressor for its feature of dealing with multivariate non-linear data without any need of feature scaling. The hyperparameters were tuned with GridSearchCV, and the model's performance was verified by measures like MSE, MAE, and R^2 Score.

In Model 2 (Prediction of RUL), the dataset for all 32 batteries was employed. Apart from the present capacity (using Model 1 or real data), other characteristics like max discharge voltage, min charge voltage, and total cycle time (charge + discharge) were added. The target variable, RUL, was equated as the difference between maximum cycle index and present cycle index. The same Random Forest model architecture was employed, with hyperparameter optimization and performance assessment done following the identical procedure as Model 1.

Overall, this two-stage machine learning pipeline provides a scalable, effective, and precise approach to battery health diagnostics and lifecycle estimation for more intelligent maintenance planning in electric vehicles, consumer products, and industrial energy systems. Moreover, for improving the interpretability and usability of the models developed, a convenient web interface was rolled out. This interface enables users to enter critical battery parameters like initial and current capacity, charge/discharge durations, voltage ranges, and cycle number to estimate the Remaining Useful Life (RUL) in real-time.

The application not only delivers an estimated RUL in cycles but also tabulates key features like total operation time, capacity percentage at the moment, and battery fading rate. This seamless incorporation of machine learning predictions into a usable tool highlights the applicability of the models in the real world, enabling end-users and battery managers to make informed decisions based on data-driven information

2.3. SYSTEM REQUIREMENTS

2.3.1 SOFTWARE REQUIREMENTS:

Python: The Main programming language used in preprocessing and model development. It has extensive libraries which are which are helpful in feature engineering and machine learning.

NumPy, Pandas: Powerful libraries that are used in data analysis and manipulation.

Matplotlib: For visualization of results, Matplotlib is helpful. Used in plotting the correlation matrix and plotting battery vs cycles

Pymatreader: A Library that is used to read .mat files. Enabling direct access to datasets that were originally created in MATLAB.

Scikit-learn: Open-source Python library used for machine learning. It has a wide variety of machine learning algorithms.

Jupyter Notebook: Used to create interactive documents that combine code and visualization. Used for exploratory data analysis, prototyping models, and documenting the development process interactively.

2.3.2 DATASET

Battery dataset NASA Prognostics Center of Excellence.

2.3.3 HARDWARE REQUIREMENTS

Processor: Intel Core i5 or higher (or AMD equivalent). A multi-core processor is essential to handle data preprocessing, model training, and inference tasks without performance bottlenecks.

RAM: Minimum of 8 GB RAM. Sufficient RAM is required to load large datasets into memory, support model training processes, and allow multitasking without significant slowdown.

Disk: At least 10 GB of free space. This space is needed to store datasets, model checkpoints, logs, and other temporary files generated during execution.

2.4 APPLICATIONS

Predictive analysis of lithium-ion batteries has a variety of real-world applications, such as:

Electric Vehicles (EVs): Forecasting battery degradation patterns assists in scheduling timely maintenance and replacement, hence preventing sudden failures. It enhances battery warranty management, improves residual value estimation for used EVs, and helps optimize charging behavior for extended battery life.

Energy Grids (Renewable Storage Systems): Predicting RUL is essential when dealing with bulk battery banks used for energy storage of solar and wind power. It aids in load balancing, facilitates effective dispatching of energy, and averts sudden shutdowns in intelligent grid systems. In addition, it assists with scheduling affordable replacements and enhancing grid resilience overall.

Consumer Electronics: Smartphones, laptops, and tablets rely on lithium-ion batteries; RUL monitoring can alert users when battery health degrading. This enables timely battery maintenance, improves user experience with optimized power utilization, and minimizes the dangers of overheating or unexpected power shutdown. It also fosters sustainability with wise battery recycling choices.

Aerospace & Defense: In mission-critical applications such as satellites, UAVs, and military gear, battery reliability is critical to the mission. RUL estimation guarantees that reserve power sources are in top working condition when called upon. RUL estimation guarantees that reserve power sources are in top working condition when called upon, minimizing the chances of mission failure. It also aids strategic planning for long-duration missions and enhances safety in distant or hostile environments.

Marine Systems: Submarines, autonomous underwater vehicles (AUVs), and marine sensors function in remote environments where battery failure is expensive. RUL estimation improves mission planning, safety, and energy budgeting for extended-duration marine operations. It also helps schedule dock time for battery replacement or maintenance.

2.5 ADVANTAGES

Early Detection of Battery Failure: Anticipates indications of battery degradation prior to total failure. Minimizes the risk of sudden breakdowns in vital systems like electric vehicles, aerospace equipment, or medical devices.

Cost-Effective Maintenance Planning: Enables companies to transition from reactive to predictive maintenance approaches. Avoids unnecessary battery replacements by estimating the remaining useful life with accuracy. Saves on downtime and maintenance expenses, particularly in high-scale operations such as energy grids or EV fleets.

Battery Safety: Avoids dangerous conditions like thermal runaway, swelling, or leakage by continuously tracking battery health. Maintains safety compliance in high-risk sectors such as aerospace, medical, and consumer electronics.

Flexible to Varying Battery Types and Data Sets: Machine learning and data-centric models may be retrained or fine-tuned for different lithium-ion chemistries and topologies

2.6 CHALLENGES

Noise and Inconsistency in Real-World Data: Data collected from sensors in real-world environments often contains noise due to external factors such as temperature fluctuations, sensor drift, or interference.

Varying Patterns of Degradation Across Batteries: Batteries even of the same brand and model tend to degenerate differently because of differences in usage conditions, manufacturing variability, and environmental factors.

Feature Correlation and Redundancy: Most battery metrics (e.g., voltage, current, temperature) correlate with each other, potentially leading to redundancy and multicollinearity in the data.

Scarce Labeled Data for Early-Stage Batteries: Correct labeling of RUL demands complete lifecycle information, which is generally not available for batteries that are in early or middle stages of usage.

3. EXPERIMENTATION AND RESULTS

3.1 EXPERIMENTATION

3.1.1 Data Loading and Exploration

The NASA Battery Dataset given in .mat format has cycle-wise data for 32 lithium-ion batteries subjected to different usage conditions and operational profiles. Each battery's data file includes detailed measurements recorded over its entire life cycle, such as voltage, current, temperature, capacity, and timestamps during both charge and discharge phases.

To facilitate analysis, the .mat files were programmatically loaded using the scipy.io library, specifically the loadmat function, which efficiently parses MATLAB file structures into Python-readable formats. Subsequently, the extracted data were organized into structured Pandas Data Frames for each battery. This conversion allowed for easier data manipulation, visualization, and analysis. To optimize future use and reduce loading times, the cleaned and organized Data Frames were serialized and stored in .pkl (pickle) format.

3.1.2 Data Preprocessing and Splitting

For Model 1 (Capacity Prediction), four batteries (B0005, B0006, B0007, B0018) were chosen. Mean values were computed for all time-series data (voltage, current, temperature) per cycle. Charge and discharge data were combined by battery ID and cycle number. To ensure that the model had a comprehensive view of each cycle, charge and discharge information were combined systematically, aligning data by battery ID and cycle number. This aggregation allowed the model to learn meaningful trends in capacity degradation without being overwhelmed by high-frequency fluctuations within individual cycles.

Model 2 (RUL Prediction) utilized all 32-batteries data. Derived features include max voltage on discharge, min voltage on charge, and overall time. All of the data was cleaned, missing values were managed, and proper cycle-based RUL labels were assigned. Both models used train_test_split to divide the data into training and testing sets. These derived features were chosen based on domain knowledge, considering their strong influence on battery health and degradation behaviour.

3.1.3 Model Building

Both capacity prediction and RUL prediction models were built using the Random Forest Regressor from the scikit-learn library. Random Forest is an ensemble learning algorithm that constructs multiple decision trees using random subsets of data and features. For regression tasks, it predicts the output by averaging the predictions from all trees.

This method was chosen for its ability to model nonlinear relationships, resistance to overfitting, and interpretability through feature importance scores. It handles noisy, high-dimensional data well and requires minimal preprocessing. The algorithm is particularly effective for this application due to the complex and variable nature of battery performance data.

3.1.4 Model Training

Using the training data, the Random Forest models were trained with an initial set of hyperparameters. To further improve accuracy, a Grid Search Cross-Validation (GridSearchCV) was conducted to determine the best combination of

1. Number of estimators (n_estimators)
2. Tree depth (max_depth)
3. Minimum samples per leaf/split

Cross-Validation Procedure: The training set is divided into 5 folds. For each parameter combination, the model is trained on 4 folds and validated on the remaining fold. This process is repeated until every fold has served as a validation set once.

Selection of Best Parameters: The parameter combination yielding the best validation performance (usually the lowest mean squared error) is selected. The final model is retrained using the entire training set and these optimized parameters.

3.1.5 Model Evaluation

The trained models were evaluated using:

i. Mean Absolute Error (MAE): measures the mean of errors
ii. Mean Squared Error (MSE): penalizes large errors
iii. R^2 Score indicates goodness of fit

Mean Absolute Error (MAE): The average of the absolute differences between the predicted and actual values. Reflects the average magnitude of prediction errors.

Mean Squared Error (MSE): The average of the squared differences between the predicted and actual values. Penalizes larger errors more severely than MAE.

R2 Score: R^2 tells you how much of the variability in the target variable (e.g., battery capacity or RUL) can be explained by the model's inputs.

3.1.6 Model Testing

After training, the models were tested on unseen data from the test set. The effectiveness of the model was evaluated using the following regression metrics. The trained and validated models were tested on unseen data from the test set. Predictions for both capacity and RUL were compared against actual values using the same evaluation metrics (MAE, MSE, R^2). Visual comparisons, such as line plots and scatter plots, were used to inspect model performance and error trends.

3.1.7 Conclusion

This project successfully demonstrates the use of a two-stage machine learning pipeline for lithium-ion battery health monitoring. The first model accurately predicts battery capacity using thermal and electrical features,

while the second model estimates RUL using capacity and voltage-related statistics. The models provide valuable insights into battery degradation and remaining service time, making them highly applicable for predictive maintenance in various battery-driven applications

3.2 CODE

```
import os
import numpy as np import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor
from xgboost import XGBRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.pipeline import make_pipeline

import warnings warnings.filterwarnings('ignore')

def split_cell(df):
    df_charge = df[df['type'] == 'charge']
    df_discharge = df[df['type'] == 'discharge']
    df_impedance = df[df['type'] == 'impedance']

    df_charge = df_charge.reset_index(drop=True)
    df_discharge = df_discharge.reset_index(drop=True)
    df_impedance = df_impedance.reset_index(drop=True)

    return df_charge, df_discharge, df_impedance

data_folder = "../Dataset/data"
battery_ids = os.listdir(data_folder)
processed_dfs = []

for battery_id in battery_ids:
    file_path = os.path.join(data_folder, f"{battery_id}")

    with open(file_path, "rb") as f: battery_data = pickle.load(f)
    battery_data['battery_id'] = battery_id[:5]

    df_charge, df_discharge = preprocess_data(battery_data)
    df_charge['battery_id'] = battery_id[:5]
    df_discharge['battery_id'] = battery_id[:5]

    df_processed = merge_cycles(df_charge, df_discharge)
    df_processed['Total time'] = df_processed['charge_time'] + df_processed['discharge_time']

    initial_capacity = df_processed['Capacity'].iloc[0]
    df_processed['initial_capacity'] = initial_capacity

    df_processed['current_capacity_percent'] = (df_processed['Capacity'] / initial_capacity) * 100
```

```
    df_processed['battery_fading_percent'] = 100 - df_processed['current_capacity_percent']
    last_cycle = df_processed['cycle'].max()
    df_processed['RUL'] = last_cycle - df_processed['cycle']

    processed_dfs.append(df_processed)
    df_all = pd.concat(processed_dfs, ignore_index = True)
    df_all['battery_id_encoded'] = LabelEncoder().fit_transform(df_all['battery_id')

    df_all = pd.concat(processed_dfs, ignore_index = True)
    df_all['battery_id_encoded'] = LabelEncoder().fit_transform(df_all['battery_id'])

features = ['battery_id_encoded', 'initial_capacity', 'Capacity', 'current_capacity_percent',
'battery_fading_percent', 'cycle', 'charge_time', 'min_voltage_charge', 'discharge_time',
'max_voltage_discharge', 'Total time']

target = 'RUL'

X = df_all[features]
y = df_all[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

param_grid = {
'C': [0.1, 1, 10, 100, 1000],
'gamma': ['scale', 0.001, 0.01, 0.1, 1],
'epsilon': [0.01, 0.1, 0.2, 0.5] }

grid_search = GridSearchCV( SVR(kernel = 'rbf'), param_grid, cv = 5, scoring = 'r2', verbose = 1, n_jobs = -
1)
grid_search.fit(X_train_scaled, y_train)

best_svr = grid_search.best_estimator_
svr_preds = best_svr.predict(X_test_scaled)

svr_r2 = r2_score(y_test, svr_preds)
svr_mse = mean_squared_error(y_test, svr_preds)

print("Best Parameters:", grid_search.best_params_)
print(f"SVR R² Score: {svr_r2:.3f}")
print(f"SVR Mean Squared Error: {svr_mse:.3f}")


plot_df = pd.DataFrame({ 'Actual RUL': y_test.values, 'Predicted RUL': preds })

plt.figure(figsize=(8, 6))
plt.scatter(plot_df['Actual RUL'], plot_df['Predicted RUL'], alpha=0.7, color='royalblue', edgecolor='black')
plt.plot([plot_df['Actual RUL'].min(), plot_df['Actual RUL'].max()],
```

```
[plot_df['Actual RUL'].min(), plot_df['Actual RUL'].max()], color='red', linestyle='--', label='Perfect
Prediction')
plt.title("Predicted vs Actual Remaining Useful Life")
plt.xlabel("Actual RUL (Cycles)")
plt.ylabel("Predicted RUL (Cycles)")
plt.legend() plt.grid(True)
plt.tight_layout()
plt.show()

models = {
"Support vector Regressor": make_pipeline(StandardScaler(), SVR(C=1000, epsilon=0.2, gamma=1)),
"Random Forest Regressor": RandomForestRegressor( n_estimators=300, max_depth=20,
max_features='sqrt', min_samples_split=2, min_samples_leaf=1, random_state=42),
"KNN Regressor": make_pipeline(StandardScaler(), KNeighborsRegressor(n_neighbors=3)), "Gradient
Boosting": GradientBoostingRegressor(
n_estimators=200, learning_rate=0.1, random_state=42),
"Decision Tree": DecisionTreeRegressor(max_depth=10, random_state=42), "Linear Regression":
make_pipeline(StandardScaler(), LinearRegression())
}

class bcolors:
        HEADER = '\033[95m'
OKBLUE = '\033[94m'
OKCYAN = '\033[96m'
OKGREEN = '\033[92m'
WARNING = '\033[93m'
FAIL = '\033[91m'
ENDC = '\033[0m'
BOLD = '\033[1m'

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    preds = model.predict(X_test)

    r2 = r2_score(y_test, preds)
    mse = mean_squared_error(y_test, preds)

    results[name] = {"r2": r2, "mse": mse}
    sorted_results = sorted(results.items(), key = lambda x: x[1]['r2'], reverse=True)

    print(f"\n{bcolors.BOLD}{'Model': <25} {'R² Score':<15} {'MSE':<15}{bcolors.ENDC}")
    print(f"{'-'*55}")

for name, metrics in sorted_results:
        print(f"{bcolors.HEADER}{name:<25}{bcolors.ENDC}"
        f"{bcolors.OKGREEN}{metrics['r2']:<15.3f}{bcolors.ENDC}"
        f"{bcolors.OKBLUE}{metrics['mse']:<15.3f}{bcolors.ENDC
```

3.2 RESULTS

3.2.1 CAPACITY PREDICTION

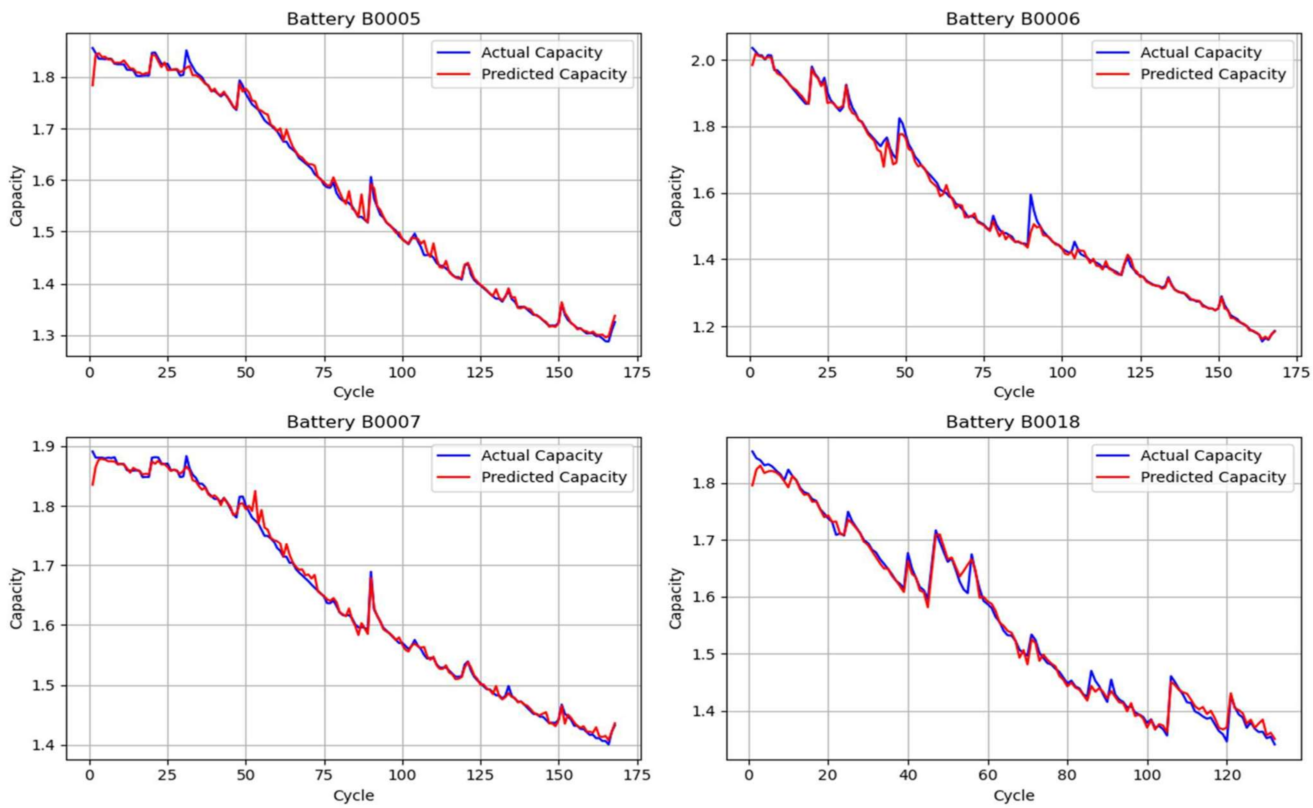

Fig No 3.1 Capacity vs cycles of batteries

The battery capacity prediction model demonstrated excellent performance, achieving a Mean Absolute Error (MAE) of 0.0073, a Mean Squared Error (MSE) of 0.00011, and a high R^2 Score of 0.98, indicating strong predictive accuracy. The model successfully captured the underlying degradation trends in lithium-ion battery capacity across charge-discharge cycles. Predictions were closely aligned with actual values, reflecting the model's ability to generalize well on unseen test data. Additionally, the model maintained robustness across different cycles, with only minimal deviations observed. Any slight underfitting or overfitting at the cycle extremes was effectively mitigated through hyperparameter tuning using grid search

3.2.2 RUL PREDICTION OUTPUT:

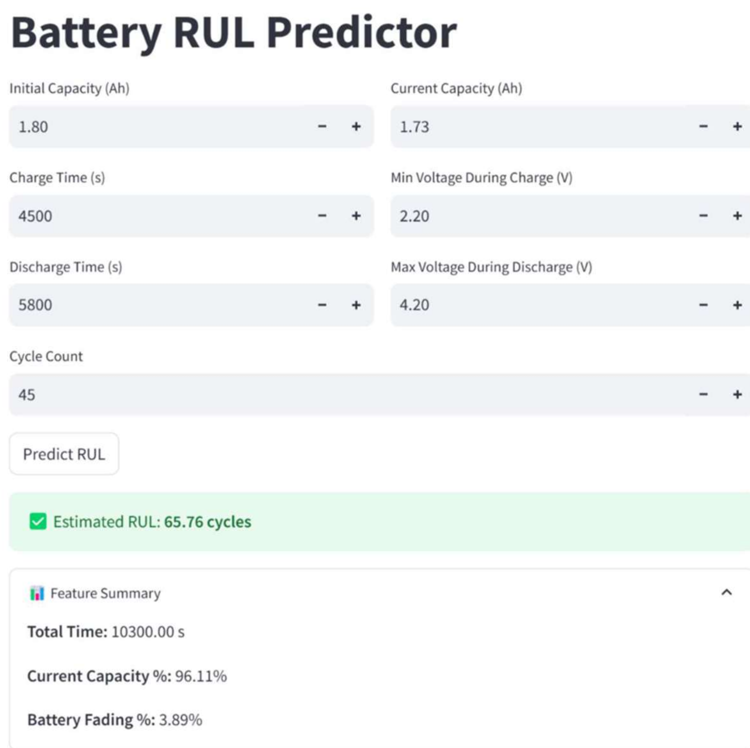

Fig No 3.2 output Image

The Battery RUL Predictor is an easy-to-use interface for estimating the Remaining Useful Life (RUL) of lithium-ion batteries given important input parameters like initial capacity, current capacity, charge/discharge time, voltage range, and cycle count. After values are inputted, the tool provides the forecasted RUL in this case 65.76 cycles as well as a summary of detailed features such as total operating time, current capacity percentage (96.11%), and battery fading percentage (3.89%).

The RUL prediction model, trained using data from 32 batteries, achieved strong performance metrics on the test set. It recorded a Mean Absolute Error (MAE) of 17.2 cycles, a Mean Squared Error (MSE) of 37.28, and a coefficient of determination (R^2 Score) of 0.982, indicating a high level of accuracy and generalization across diverse battery profiles. The model effectively predicted the number of cycles remaining before the battery reaches its end-of-life threshold.

4. CONCLUSION & FUTURE SCOPE OF WORK

In this study, we implement a two-stage machine learning pipeline aimed at predicting the capacity and Remaining Useful Life (RUL) of lithium-ion batteries. In the first stage, a regression model was trained using input features such as voltage, current, temperature, and cycle count to accurately estimate the battery's current capacity. In the second stage, another regression model utilized the predicted capacity, along with additional operational features like charging time, discharging time, and voltage range, to predict the RUL. The models demonstrated high accuracy and good generalization when evaluated on unseen data, with performance metrics such as low Mean Absolute Error (MAE) and high R^2 scores. This confirms the reliability of the models for practical applications.

The current work lays a strong foundation for battery health prediction, and there is significant scope for further advancement. Future enhancements may include the integration of deep learning models such as LSTMs or Transformers to better capture temporal degradation patterns, and real-time deployment of the system into Battery Management Systems (BMS) for continuous monitoring. Transfer learning approaches can be explored to adapt the models across different battery types and usage conditions. Additionally, incorporating explainable AI techniques like SHAP can help interpret the model's predictions, making them more transparent and trustworthy. Expanding the feature set to include parameters like internal resistance, state-of-charge (SOC), and ambient environmental data may further improve accuracy. A hybrid approach combining physics-based models with data-driven methods could offer robustness in uncertain conditions. Lastly, cloud integration and validation on real-world datasets would make the system scalable and industry-ready for large-scale applications in electric vehicles, renewable energy storage, and portable electronics.

REFERENCES

[1] D. Swain, M. Kumar, A. Nour, K. Patel, A. Bhatt, B. Acharya, and A. Bostani, "Remaining useful life predictor for EV batteries using machine learning," IEEE Access, 2024.

[2] A. Tiane, C. Okar, M. Alzayed, and H. Chaoui, "Comparing hybrid approaches of deep learning for remaining useful life prognostic of lithium-ion batteries," IEEE Access, vol. 12, pp. 70334–70344, 2024.

[3] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, et al., "Data-driven prediction of battery cycle life before capacity degradation," Nature Energy, vol. 4, no. 5, pp. 383– 391, 2019.

[4] V. Safavi, A. Mohammadi Vaniar, N. Bazmohammadi, J. C. Vasquez, and J. M. Guerrero, "Battery remaining useful life prediction using machine learning models: A comparative study," Information, vol. 15, no. 3, p. 124, 2024.

[5] N. Yang, H. Hofmann, J. Sun, and Z. Song, "Remaining useful life prediction of lithium-ion batteries with limited degradation history using random forest," IEEE Trans. Transp. Electrific., vol. 10, no. 3, pp. 5049–5060, 2024.

[6] M. Dubarry, V. Svoboda, R. Hwu, and B. Y. Liaw, "Incremental capacity analysis and close- to-equilibrium OCV measurements to quantify capacity fade in commercial rechargeable lithium batteries," Electrochem. Solid-State Lett., vol. 9, no. 10, pp. A454–A457, 2006.

[7] Y. Zhang, Y. Jiang, C. Zhang, et al., "Long short-term memory for RUL prediction of lithium- ion batteries," IEEE Access, vol. 5, pp. 21227–21238, 2017.

[8] K. A. Severson, P. M. Attia, N. Jin, et al., "Data-driven prediction of battery cycle life before capacity degradation," Nature Energy, vol. 4, pp. 383–391, 2019.

[9] Y. Li, Z. Zhang, D. Wu, and Q. Wang, "A hybrid deep learning model for accurate prediction of lithium-ion battery capacity degradation," Appl. Energy, vol. 251, p. 113378, 2019.

[10] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS), 2017, pp. 4765–4774.

B22AI009_mini_project_removed.pdf

Kakatiya Institute of Technology and Science

Document Details

Submission ID

trn:oid:::3618:92672540

Submission Date

Apr 24, 2025, 1:09 PM GMT+5:30

Download Date

Apr 24, 2025, 1:10 PM GMT+5:30

File Name

B22AI009_mini_project_removed.pdf

File Size

960.0 KB

14 Pages

3,538 Words

21,317 Characters

18% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

67 Not Cited or Quoted 18%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

10% Internet sources
9% Publications
15% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.