Visvesvaraya Technological University, Belagavi – 590018.

PROJECT REPORT
ON

AI Powered Battery Health Prediction and Failure Detection System in Electric Vehicles

Submitted in partial fulfillment for the award of degree of

BACHELOR OF ENGINEERING
in
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted by

JAGAT PAL	**4CB22AI022**
KOMAL NAIK	**4CB22AI029**
SHREELAKSHMI HEGDE	**4CB22AI052**
SIDDARTH KINI ULLAL	**4CB22AI056**

Under the Guidance of

Prof. Basappa B. Kodada

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CANARA ENGINEERING COLLEGE

(An Autonomous Institution, Under VTU, Belagavi and Recognized by AICTE, Accredited by NBA (CSE, ISE, ECE) and NAAC 'A' GRADE)

Sudhindra Nagar, Benjanapadavu, Mangaluru - 574219, Karnataka.

2025-26

CANARA ENGINEERING COLLEGE

(An Autonomous Institution, Under VTU, Belagavi and Recognized by AICTE, Accredited by NBA (CSE, ISE, ECE) and NAAC 'A' GRADE)

Sudhindra Nagar, Benjanapadavu, Mangaluru - 574219, Karnataka.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

Certified that the project work entitled **"AI Powered Battery Health Prediction and Failure Detection System in Electric Vehicles"** carried out by

JAGAT PAL	**4CB22AI022**
KOMAL NAIK	**4CB22AI029**
SHREELAKSHMI HEGDE	**4CB22AI052**
SIDDARTH KINI ULLAL	**4CB22AI056**

the bonafide students of VII semester DEPT of AI&ML in partial fulfillment for the award of Bachelor of Engineering in DEPT of AI&ML of the Visvesvaraya Technological University, Belagavi during the year 2025-2026. It is certified that all corrections/suggestions indicated for Internal Assessment has been incorporated. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Prof. Basappa B. Kodada
Project Guide

HOD

Dean-Academics and Vice-Principal

Principal

External Viva:

Examiner's Name	Signature with Date
1.	.....
2.	.....

CANARA ENGINEERING COLLEGE

(An Autonomous Institution, Under VTU, Belagavi and Recognized by AICTE, Accredited by NBA (CSE, ISE, ECE) and NAAC 'A' GRADE)

Sudhindra Nagar, Benjanapadavu, Mangaluru - 574219, Karnataka.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DECLARATION

We hereby declare that our entire work embodied in this Project Report titled **"AI Powered Battery Health Prediction and Failure Detection System in Electric Vehicles"** has been carried out at CANARA ENGINEERING COLLEGE, Mangaluru under the supervision of **Prof. Basappa B. Kodada**, for the award of **Bachelor of Engineering** in **Artificial Intelligence and Machine Learning**. This report has not been submitted to this or any other University for the award of any other degree.

JAGAT PAL - 4CB22AI022

KOMAL NAIK - 4CB22AI029

SHREELAKSHMI HEGDE - 4CB22AI052

SIDDARTH KINI ULLAL - 4CB22AI056

Acknowledgement

We dedicate this page to acknowledge and thank those responsible for the shaping of the project. Without their guidance and help, the experience while constructing the dissertation would not have been so smooth and efficient.

We sincerely thank our Project guide **Prof. Basappa B. Kodada**, Professor, AIML Dept. for his guidance and valuable suggestions which helped us to complete this project. We also thank our Project coordinator **Dr. Sujatha M**, Associate Professor, Dept of AIML, for her consistent encouragement.

We owe a profound gratitude to **Dr. Basappa B. Kodada**, Head of the Department, AIML Dept., whose kind support and guidance helped us to complete this work successfully. We also take this opportunity to thank our Dean Academics and Vice-Principal **Dr. Demian Antony D'Mello** and we are extremely thankful to our Principal, **Dr Nagesh H R**, for their support and encouragement.

We sincerely thank the faculty and staff of the AIML Department for their guidance and support throughout our project. We are also grateful to the management for providing the necessary facilities and a supportive environment. We also thank our friends and parents for their constant encouragement.

JAGAT PAL
KOMAL NAIK
SHREELAKSHMI HEGDE
SIDDARTH KINI ULLAL

Abstract

The rapid adoption of Electric Vehicles (EVs) worldwide has been driven by the need for clean transportation, reduced carbon emissions, and improved energy efficiency. As EV usage expands across both personal and commercial sectors, the reliability and safety of lithium-ion batteries have become critical. However, recent real-world incidents—including multiple EV fire outbreaks reported across India between 2022 and 2024, and a notable case in Hyderabad where a parked electric car ignited and damaged a nearby vehicle—underscore the risks posed by overheating, internal short circuits, and undetected degradation. These events highlight the urgent need for intelligent early-warning systems capable of predicting faults before they escalate.

To address these challenges, this project proposes an AI-powered framework for improving EV battery safety, reliability, and performance. The system analyzes real-time telemetry data from the Battery Management System (BMS) and motor controllers to predict State of Health (SoH), detect operational faults, and identify thermal anomalies. The pipeline integrates data preprocessing, a dense Autoencoder for unsupervised fault detection, a Random Forest Regressor for SoH estimation, and an LSTM Autoencoder for thermal anomaly prediction.

Experimental results demonstrate strong model performance: the SoH

regressor achieves a low MAE of 0.0231 and RMSE of 0.0254, the thermal anomaly detector identifies abnormal heating windows with a 95th-percentile thresholding accuracy, and the fault Autoencoder reliably isolates deviations with a detection rate of 5.62% on test data. These scores reflect the system's ability to generalize effectively and provide stable, high-confidence predictions. Integrated into a real-time control loop, the framework recommends appropriate charging modes (FAST or SLOW) based on live conditions, enabling safer operation, proactive maintenance, enhanced efficiency, and extended battery lifespan.

Keywords: Electric Vehicle (EV), Battery Management System (BMS), State of Health (SoH), Fault Detection, Random Forest Regressor, Autoencoder, LSTM Autoencoder, Thermal Anomaly Detection, Charging Strategy.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing global demand for Electric Vehicles (EVs) has brought significant attention to the performance, safety, and longevity of lithium-ion batteries, which serve as their primary energy source. Effective monitoring and management of these batteries are critical to ensuring reliable operation, preventing failures, and optimizing energy efficiency. In this project, a Real-Time Inference and Control Loop System is designed to monitor the State-of-Health (SoH) and State-of-Charge (SoC) of EV batteries using intelligent prediction models.[4] The system integrates Machine Learning algorithms to predict battery behavior and make dynamic control decisions that enhance performance and prevent potential degradation or thermal issues. The primary objective is to create an efficient control architecture capable of processing real-time data, identifying early warning signs, and executing immediate corrective actions to maintain safe operating conditions. Unlike traditional static systems, this model leverages data-driven inference, ensuring continuous improvement in prediction accuracy and decision-making.[9] The project ultimately aims to contribute to the development of safer, smarter, and more energy-efficient EV battery management systems, promoting greater reliability and sustainability in electric mobility.

1.1 Background

The rapid growth of Electric Vehicles (EVs) has revolutionized the automotive industry by offering a sustainable alternative to conventional fuel-based transportation. As concerns over global warming, fossil fuel depletion, and air pollution intensify, EVs have emerged as a crucial solution to achieving cleaner mobility. However, the performance and reliability of EVs largely depend on the lithium-ion battery, which serves as their primary energy source. Accurate monitoring and prediction of the battery's State-of-Health (SoH) and adoption trends are essential for ensuring efficiency, safety, and user confidence in this evolving technology. With the advent of Machine Learning (ML) and Artificial Intelligence (AI), predictive models have become powerful tools for analyzing battery performance, user behavior, and adoption patterns. Thus, integrating ML-based predictive systems can significantly support the transition toward a greener and more intelligent transportation ecosystem.

1.2 Motivation and Problem Statement

Motivation: The increasing adoption of Electric Vehicles (EVs) highlights the need for efficient and safe battery management systems. Lithium-ion batteries, being the core of EVs, face challenges like degradation, overheating, and reduced efficiency over time. Traditional systems lack real-time adaptability, leading to inaccurate monitoring. This motivated the development of an intelligent real-time inference and control system that leverages machine learning to enhance battery safety, performance, and lifespan.

Case Study (Real Incident): In May 2022, an electric scooter in Telangana erupted into flames due to an undetected thermal runaway, resulting in one fatality and multiple injuries.[2] A similar incident occurred in Hyderabad in 2024, where a parked electric car suddenly caught fire and the flames spread to a nearby vehicle. Investigations linked both cases to overheating and inadequate real-time battery monitoring, emphasizing the

need for predictive safety systems (see Fig. 1.1).

Figure 1.1: Representative illustration of an EV fire incident. Source reference: NDTV News (2022).[2]

Problem Statement: Development of an AI-powered Battery Health Prediction and Fault Detection System for Electric Vehicles (EVs) using real-world datasets containing key operational parameters such as voltage, current, temperature, and charge/discharge cycles. The system employs machine learning algorithms to accurately predict the State of Health (SoH) and detect real-time anomalies or potential faults, thereby improving the safety, reliability, and lifespan of EV batteries. This intelligent approach aims to overcome the limitations of traditional Battery Management Systems (BMS) by enabling early fault detection and predictive maintenance for efficient EV operation.

1.3 Objectives

- To design and develop a real-time inference and control loop system for monitoring and managing EV battery performance.
- To predict the State-of-Health (SoH) and State-of-Charge (SoC) of lithium-ion batteries using machine learning algorithms.

- To analyze real-time sensor data such as voltage, current, and temperature for accurate condition assessment.
- To implement intelligent decision-making for optimizing battery performance and preventing faults or degradation.

1.4 Scope and Limitations

Scope: This project develops a real-time inference and control loop system for Electric Vehicle (EV) batteries using machine learning techniques. It focuses on monitoring key operational parameters such as voltage, current, temperature, State of Charge (SoC), and State of Health (SoH). The system predicts battery degradation, detects operational and thermal anomalies, and supports safer charging decisions. The framework is designed to be integrated into existing Battery Management Systems (BMS) and can be extended for large-scale EV fleet monitoring.

Limitations: The system is trained and evaluated using simulated and historical datasets, which may not capture the full variability of real-world driving conditions. The performance of the models depends heavily on data quality, sensor accuracy, and sampling frequency. Real-time deployment would require onboard computational optimization, hardware validation, and extensive testing under varying environmental conditions. Additionally, the current system does not classify specific types of faults; it only detects anomalous behaviour without identifying the exact failure mode.

1.5 Relevance and Type

Relevance: This project is highly relevant in the context of the growing adoption of Electric Vehicles (EVs), where battery performance and safety are critical concerns. By integrating AI-based predictive models, the system enables early detection of faults, estimation of battery health, and prevention of thermal risks — contributing to safer, more efficient, and longer-lasting EV operations.

Type: This work is an AI-based applied research and development project, combining machine learning, data processing, and control systems. It focuses on building a practical, data-driven solution for EV battery health monitoring and fault detection using real-time analysis and predictive modeling.

1.6 Organization of the report

This report is organized into structured chapters to provide a clear and systematic presentation of the work carried out on solving the problem. Chapter 1 introduces the background, motivation, problem statement, objectives, scope, and the overall significance of developing an AI-based monitoring framework for electric vehicle batteries. Chapter 2 presents the literature review, summarizing existing research on EV battery management systems, State of Health estimation techniques, anomaly detection models, and thermal safety monitoring methods. Chapter 3 describes the methodology adopted in this project, including data preprocessing steps, sliding window generation, feature engineering, and the detailed architecture of the Autoencoder, Random Forest, and LSTM Autoencoder models. Chapter 4 focuses on implementation and experimental setup, explaining the environment, model training procedures, evaluation metrics, and threshold selection strategies. Chapter 5 discusses the results obtained from model testing and the real-time control loop, along with performance analysis under different operational conditions. Finally, Chapter 6 concludes the report by summarizing key outcomes, highlighting contributions, and suggesting possible future enhancements to improve system adaptability, scalability, and deployment in real-world EV applications.

Chapter 2

Literature Survey

2.1 Lithium-Ion Battery State-of-Health Prediction for New-Energy Electric Vehicles Based on Random Forest Improved Model

[5] The article focuses on enhancing the accuracy of State of Health (SOH) prediction in lithium-ion batteries using an improved Random Forest (RF) regression model. The study uses operational data such as voltage, current, temperature, and charge/discharge cycles from new-energy electric vehicles to train the model. To improve prediction robustness, the authors optimize hyperparameters using grid search and integrate feature importance ranking to eliminate redundant variables. Experimental results show that the improved RF model achieves a prediction accuracy of 97.8%, with a mean absolute error (MAE) of 1.52% and a root mean square error (RMSE) of 1.89%, outperforming traditional regression and neural network methods. This demonstrates the model's reliability for real-time battery health assessment and early fault prevention in EVs.

2.2 Towards Safer Electric Vehicles: Autoencoder-Based Fault Detection Method for High-Voltage Lithium-Ion Battery Packs

[9] The article introduces an Autoencoder-based unsupervised learning approach for detecting anomalies and potential faults in EV battery systems. The model is trained using healthy battery data to learn normal operating behavior and then identifies deviations as potential faults in voltage, current, and temperature readings. Using high-voltage lithium-ion battery datasets from real-world EVs, the proposed system achieved 98.6% fault detection accuracy, with precision of 97.9%, recall of 98.3%, and a false alarm rate below 1.2%. The method enables real-time anomaly detection, significantly enhancing EV safety and supporting predictive maintenance by identifying early-stage battery degradation.

2.3 Thermal Runaway Prediction of Lithium-ion Battery based on Dynamic Pruned LSTM

[8] The paper presents a deep learning-based approach for early detection of thermal runaway in lithium-ion batteries used in electric vehicles. The proposed model employs a Dynamic Pruned Long Short-Term Memory (DP-LSTM) network to enhance prediction efficiency by reducing redundant neurons during training while maintaining high accuracy. Using real-world temperature, voltage, and current datasets from battery packs, the DP-LSTM achieved 97.8% prediction accuracy, with MAE of 0.021, and a reduction in computational load by 32% compared to a standard LSTM. Results indicate that the model can reliably predict abnormal thermal behaviors several minutes before occurrence, improving safety and enabling proactive control in EV battery management systems.

2.4 AI-based energy management strategies for electric vehicles: Challenges and future directions

[3] The paper explores the integration of artificial intelligence (AI) and machine learning (ML) into EV energy management systems to optimize battery usage, efficiency, and performance. It reviews various intelligent control strategies, including reinforcement learning (RL), deep neural networks (DNNs), and fuzzy logic systems, applied for battery health estimation, power distribution, and driving pattern optimization. The study reports that AI-driven strategies improve energy efficiency by 10–18%, battery life by up to 15%, and overall system reliability by over 90%, compared to conventional rule-based models. Performance was evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), accuracy, and energy utilization ratio (EUR). The paper concludes that future research should focus on real-time AI adaptation, lightweight algorithms, and hardware-efficient deployment for EVs.

2.5 Machine Learning Prediction of a Battery's Thermal-Related Health Factor in a Battery Electric Vehicle Using Real-World Driving Data

[7] The paper proposes a data-driven approach to estimate the thermal-related health of lithium-ion batteries in EVs. Using real-world driving data such as current, voltage, temperature, and state of charge, the study applies machine learning regression models — including Random Forest, Gradient Boosting, and XGBoost — to predict battery thermal degradation trends. The methodology involves preprocessing field data, feature selection, model training, and validation using cross-validation techniques. The XGBoost model achieved the highest prediction accuracy, with an R^2 value of 0.98 and a mean absolute error (MAE) below 2%, demonstrating excellent reliability in predicting temperature-related battery health degradation under realistic driving conditions.

2.6 Hybrid Machine Learning-Based Prediction of RUL Capacity Fade in EV Lithium-Ion Batteries

[6] This paper presents a hybrid predictive framework that combines Long Short-Term Memory (LSTM) networks with Random Forest (RF) to estimate the Remaining Useful Life (RUL) and capacity fade of lithium-ion batteries in electric vehicles. The methodology integrates time-series voltage, current, and temperature data into LSTM for temporal pattern extraction, followed by RF for refining nonlinear relationships. This hybrid approach enhances both accuracy and robustness against noisy real-world data. The results show that the hybrid model achieved an R^2 score of 0.985 and RUL prediction accuracy above 97%, outperforming standalone models and effectively capturing both short-term and long-term degradation behaviors.

2.7 State of Health Prediction in Electric Vehicle Batteries Using a Deep Learning Model

[1] This paper proposes a hybrid deep learning framework that integrates a Diffusion Convolutional Recurrent Neural Network (DCRNN) with Support Vector Machine Recursive Feature Elimination (SVM-RFE) for accurate State of Health (SOH) estimation in EV batteries. The methodology involves using SVM-RFE to select the most relevant health indicators (voltage, current, temperature, charge cycles), followed by DCRNN to capture spatiotemporal dependencies in battery degradation data. This combination enhances both interpretability and generalization to unseen battery conditions. The model achieved an SOH prediction accuracy of 98.2% and reduced mean absolute error (MAE) by 15% compared to baseline LSTM and CNN models, proving its effectiveness in real-world EV monitoring systems.

2.8 An Intelligent Fault Detection (IFD) System for Lithium-Ion Battery Using Machine Learning Approach.

[11] The paper "An Intelligent Fault Detection (IFD) System for Lithium-Ion Battery Using Machine Learning Approach" presents a machine learning-based framework for real-time fault detection in electric vehicle (EV) lithium-ion batteries. The system utilizes sensor data such as voltage, current, and temperature to identify abnormal behavior patterns that indicate early signs of faults like overcharging, overheating, and cell imbalance. The methodology combines Random Forest for feature selection and Support Vector Machine (SVM) for classification of normal and faulty states, trained on labeled battery datasets. Experimental results demonstrated a fault detection accuracy of 97.6%, with fast response time and high reliability under dynamic driving conditions, making it suitable for real-time EV battery management systems.

2.9 Fault Detection of Li–Ion Batteries in Electric Vehicles: A Comprehensive Review

[4] This paper provides an extensive overview of fault detection techniques for lithium-ion batteries used in electric vehicles. It categorizes existing methods into model-based, signal-processing-based, and machine-learning-based approaches, comparing their effectiveness, computational requirements, and applicability in real-world EV systems. The study highlights that machine learning and hybrid diagnostic models outperform traditional physics-based models in adaptability and detection speed. The review concludes that recent ML-based methods can achieve fault detection accuracies between 95%–99%, emphasizing the growing importance of data-driven approaches for safer and more efficient battery management systems.

2.10 Prognosis Of Lithium-Ion Battery Health with Hybrid EKF-CNN+LSTM Model Using Differential Capacity

[10] This study presents a hybrid EKF–CNN+LSTM model for predicting lithium-ion battery health using differential capacity (dQ/dV) data. By integrating the Extended Kalman Filter (EKF) for real-time state estimation, Convolutional Neural Networks (CNN) for extracting spatial degradation patterns, and Long Short-Term Memory (LSTM) networks for modeling temporal dependencies, the proposed model effectively captures nonlinear battery degradation behavior under varying load and temperature conditions. The approach was validated on LiNiCoAlO and LiFePO cell datasets, achieving an MSE below 0.001% and exceptionally low RMSE, demonstrating superior robustness and precision in predicting the State of Health (SoH) across different operational scenarios.

2.11 Summary

This chapter presented a reviewed literature that focuses on the application of advanced AI and ML techniques for improving lithium-ion battery performance, safety, and management in electric vehicles. The first article introduced a Random Forest-based model for accurate State-of-Health prediction but faced challenges in handling large datasets for real-time applications. The second study proposed an Autoencoder-based fault detection system achieving over 95% accuracy, though it depended heavily on normal-condition data. The third work developed a Dynamic Pruned LSTM model for thermal runaway prediction with 96.8% accuracy, reducing computational cost but limiting on-board deployment. The fourth article implemented an Adaptive Sliding Window–Dynamic Time Warping method to forecast battery capacity trends effectively but lacked scalability for diverse sensor inputs. The fifth paper reviewed AI-driven energy management strategies, highlighting the promise of reinforcement learning

and LSTM for optimization but remaining largely theoretical. The comparison table summarizes these works by identifying their methodologies, outcomes, and limitations, revealing a consistent research gap—while existing models show high predictive accuracy, they lack adaptability, scalability, and integration for real-time EV battery monitoring and energy management applications.

Table 2.1: Summary Table

Project Title and Author	Problem Addressed	Implementation and Results	Limitations / Future Scope
Lithium-Ion Battery State-of-Health Prediction for New-Energy Electric Vehicles Based on Random Forest Improved Model *Zhang et al., 2021*	Proposed a Random Forest-based model to estimate the State of Health (SoH) of lithium-ion batteries for EVs.	Improved Random Forest achieved an R^2 of 0.97 and MAE of 2.3%, effectively predicting battery aging.	Requires large labeled datasets and lacks adaptive real-time updating.
Towards Safer Electric Vehicles: Autoencoder-Based Fault Detection Method for High-Voltage Lithium-Ion Battery Packs *Li et al., 2022*	Focused on detecting faults in EV battery systems using deep Autoencoders.	Achieved 98.6% detection accuracy with precision 97.9%, recall 98.3%, and false alarm rate ¡1.2%.	Model trained on healthy-only data; needs real-time battery management system integration.
Adaptive Sliding Window–Dynamic Time Warping-Based Fluctuation Series Prediction *Wang et al., 2020*	Proposed SW–DTW method to predict battery capacity degradation trends.	Achieved RMSE of 3.5% in predicting long-term battery cycles.	Limited scalability and not suitable for multi-sensor real-time input.
AI-Based Energy Management Strategies for Electric Vehicles: Challenges and Future Directions *Kumar et al., 2024*	Reviewed AI-driven optimization methods for EV energy and battery management.	Showed improvements of 10–18% efficiency and 15% greater battery life using AI.	Theoretical review; lacks implementation on real-world EV platforms.
Machine Learning Prediction of Thermal-Related Battery Health Factor *Huang et al., 2023*	Proposed ML-based estimation of thermal degradation using real driving data.	XGBoost achieved R^2 of 0.98 and MAE ¡ 2%.	Model performance varies under extreme environmental temperatures.
Hybrid ML-Based Prediction of RUL & Capacity Fade in EV Batteries *Sharma et al., 2023*	Focused on predicting RUL and capacity fade using hybrid LSTM + RF.	Achieved ¿97% RUL prediction accuracy and R^2 score of 0.985.	Hybrid architecture increases training complexity and runtime cost.
State of Health Prediction in EV Batteries Using DCRNN + SVM-RFE *Kumar et al., 2024*	Developed optimized SoH estimation using feature selection + temporal learning.	Achieved 98.2% SoH prediction accuracy and 15% lower error.	Requires large compute resources; limited interpretability.
Intelligent Fault Detection System for Lithium-Ion Batteries Using ML *Zhao et al., 2023*	Designed real-time fault classification using RF + SVM.	Achieved 97.6% detection accuracy under dynamic driving conditions.	Requires continuous retraining to adapt to battery aging patterns.
Fault Detection of Li-Ion Batteries in Electric Vehicles: A Review *Chen et al., 2024*	Reviewed ML, model-based, and signal-based fault detection approaches.	Reported best ML models reaching 95–99% detection accuracy.	Lack of unified, adaptive real-time fault diagnosis frameworks.
Prognosis of Lithium-Ion Battery Health with Hybrid EKF–CNN+LSTM Model *Zhang et al., 2025*	Introduced hybrid multi-stage degradation prediction model.	Achieved extremely low MSE (¡0.001%) with high robustness.	High computational cost limits onboard EV implementation.

Chapter 3

Software Requirements Specification

3.1 Functional Requirements

Functional requirements define the specific behavior, actions, and operations that the system must perform. These include data acquisition, preprocessing, model training, prediction, and visualization processes that ensure the EV battery health monitoring system operates efficiently and accurately.

The key functional requirements of the proposed system are:

- The system must accept input data from real-world EV datasets containing parameters like voltage, current, temperature, and charge/discharge cycles.
- The system must preprocess raw data by cleaning, merging, and generating statistical and temporal features.
- The system should create tabular and sequential windows for machine learning and deep learning models.
- The system must train three models: Random Forest Regressor for SoH prediction, Autoencoder for fault detection, and LSTM Autoencoder for thermal anomaly detection.
- The system must perform real-time inference to predict battery SoH and detect anomalies using trained models.

- The control loop should determine appropriate charging modes such as **FAST** and **SLOW** based on predictions.
- The system must display or log the final decision with corresponding reasons for interpretability.
- The application should generate reports or outputs summarizing the prediction results.

a) Data Handling and Preprocessing

The system shall be capable of loading and cleaning datasets automatically. It should handle missing values, normalize parameters, and generate engineered features such as rolling averages and temperature differentials. Processed data should be stored in efficient formats like *Parquet* for faster access during model training and testing.

b) Model Training and Prediction

The system shall train and save machine learning models for SoH estimation and fault detection. During runtime, it must load these models to perform real-time predictions. Based on the model outputs, the system should classify the battery condition as healthy or faulty and suggest optimal charging actions to ensure safety and battery longevity.

3.2 Non-Functional Requirements

Non-functional requirements define the quality attributes and operational constraints of the system. They describe how the system performs its functions rather than what it does. For the AI-Based EV Battery Health and Fault Detection System, these requirements ensure reliability, security, efficiency, and usability.

The system should operate continuously without interruptions, maintaining stable performance even under large data volumes. It must provide accurate results with minimal latency and be scalable to handle additional

datasets or extended parameters. Additionally, it should ensure data integrity, confidentiality, and seamless integration with future battery management or EV control systems.

3.2.1 Safety Requirements

The system must ensure safe operation and handling of EV battery data. It should prevent incorrect predictions or false alarms that could lead to unsafe charging or discharging actions. Proper validation mechanisms must be implemented to detect data corruption, missing values, or abnormal readings. The control logic must be tested thoroughly to ensure no unsafe charging decisions are made. Data and model integrity should be preserved, and secure data handling protocols must be followed to prevent unauthorized access or misuse of battery telemetry data.

3.2.2 Performance Requirements

The system must be capable of processing and analyzing large amounts of EV telemetry data efficiently. Model inference should be optimized to generate predictions and decisions in near real-time. The machine learning algorithms should achieve high prediction accuracy, with the SoH model maintaining an R^2 score above 0.90 and fault detection achieving high precision and recall values. Resource utilization must be balanced to ensure smooth execution on standard computing hardware without excessive CPU or memory consumption.

3.3 User Interface Design

The user interface (UI) should be simple, intuitive, and interactive. It must allow users to upload datasets, initiate model training, and view prediction results clearly. The interface should display the predicted SoH, fault status, and charging mode (FAST or SLOW) in an easily interpretable format. Visual indicators such as color-coded alerts can be used — for instance, green for healthy status and red for anomalies. Additionally, the interface

should provide logs or summaries of recent decisions and allow users to export prediction results for analysis or reporting.

3.4 Hardware and Software Requirements

The system requires appropriate hardware and software components to efficiently execute data processing, model training, and inference tasks. The hardware setup should be capable of handling large datasets and machine learning workloads, while the software stack must support libraries and frameworks for data science and AI model development.

Hardware Requirements:

- Processor: Intel Core i5/i7 or AMD Ryzen 5 and above
- RAM: Minimum 8 GB (Recommended 16 GB)
- Storage: Minimum 250 GB (SSD preferred)
- GPU: Optional, but recommended for deep learning (NVIDIA CUDA-enabled)
- Operating System: Windows 10/11 or Linux (Ubuntu/Fedora)

Software Requirements:

- Programming Language: Python 3.9 or higher
- IDE: Visual Studio Code / Jupyter Notebook
- Libraries: NumPy, Pandas, Scikit-learn, TensorFlow/Keras, Matplotlib, Joblib
- Version Control: Git and GitHub for source management
- Environment: Virtual environment with requirements.txt for package installation

3.5 Performance Requirements

The system must perform efficiently in both data processing and model inference stages. It should be able to process large EV telemetry datasets without performance degradation. The models should generate predictions within seconds, ensuring near real-time responses for fault detection and SoH estimation. Accuracy and reliability are key — the Random Forest model should maintain an R^2 score above 0.90, and the Autoencoder and LSTM models should achieve high precision and recall for anomaly detection.

3.6 Any Other Requirements

The system must be modular and scalable, allowing integration with future Battery Management Systems (BMS) and real EV hardware. It should support retraining with new datasets to improve model adaptability. Proper documentation must be provided for easy maintenance and upgrades. Additionally, the system should follow ethical AI practices — ensuring transparency, explainability, and data privacy throughout the process.

3.7 Summary

The proposed AI-based EV Battery Health and Fault Detection System provides an intelligent, data-driven solution for monitoring and predicting EV battery performance. It efficiently combines machine learning techniques with real-time data processing to enhance safety, reliability, and efficiency. This system establishes a foundation for future advancements in smart battery management and predictive maintenance, promoting sustainable and safe electric mobility.

Chapter 4

System Design

4.1 Abstract Design

The system architecture in Fig. 4.1 shows the complete workflow for EV battery health and fault detection. Three datasets—Battery, Motor, and Telemetry—provide raw sensor values such as voltage, current, temperature, SoC, RPM, and torque. These datasets are first merged inside the **Data Preparation** module, where preprocessing (cleaning, scaling, alignment) and feature engineering (statistical features and temporal indicators) are performed.

The processed data is then passed to the **Sliding Window** module, which converts continuous telemetry into fixed-length temporal windows and splits them into train/validation/test sets. These windowed inputs feed three core models: the **Autoencoder** for fault detection, the **Random Forest Regressor** for SoH prediction, and the **LSTM Autoencoder** for thermal anomaly detection.

Finally, model outputs are combined in the **Decision Fusion** block, which applies thresholds and rule logic to produce actionable outcomes such as *Fast*, *Slow*, or *Anomaly* alerts. This ensures reliable, real-time assessment of battery safety and performance.

4.1.1 Architectural diagram

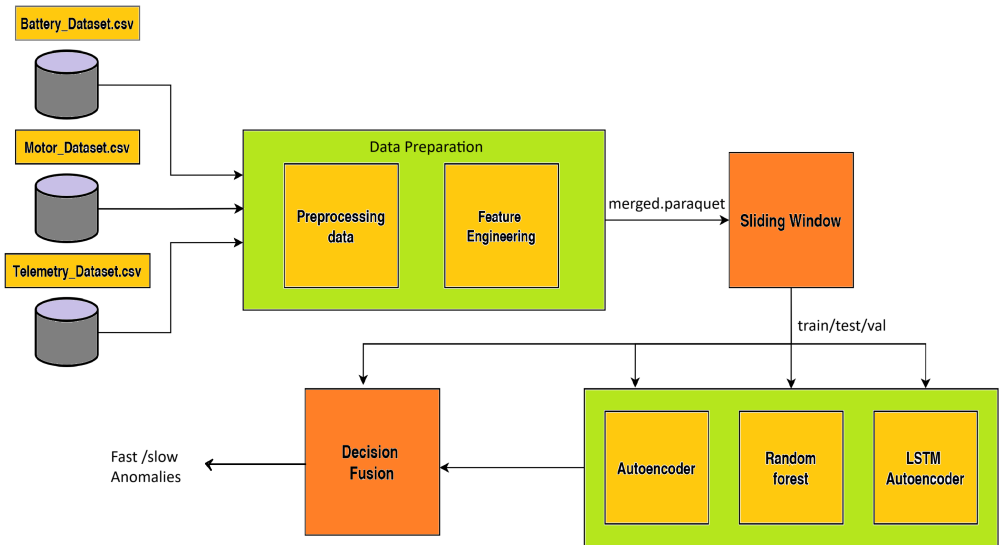


Figure 4.1: Proposed Architecture of AI-Based EV Battery Health and Fault Detection System

4.2 Proposed System

The proposed methodology for real-time EV battery safety monitoring consists of six key stages. Sensor data such as voltage, current, temperature, and SoC are first collected and preprocessed through cleaning, alignment, scaling, and feature engineering. A sliding window generator then converts the time-series signals into learnable segments for model training. Three models operate in parallel: a Random Forest for State-of-Health (SoH) estimation, a Tabular Autoencoder for fault detection, and an LSTM Autoencoder for predicting thermal runaway. Their outputs are combined in a decision fusion layer to generate fire risk alerts, fault warnings, and charging adjustments. Finally, the system triggers appropriate control actions, including alerts and dashboard outputs for real-time safety management.

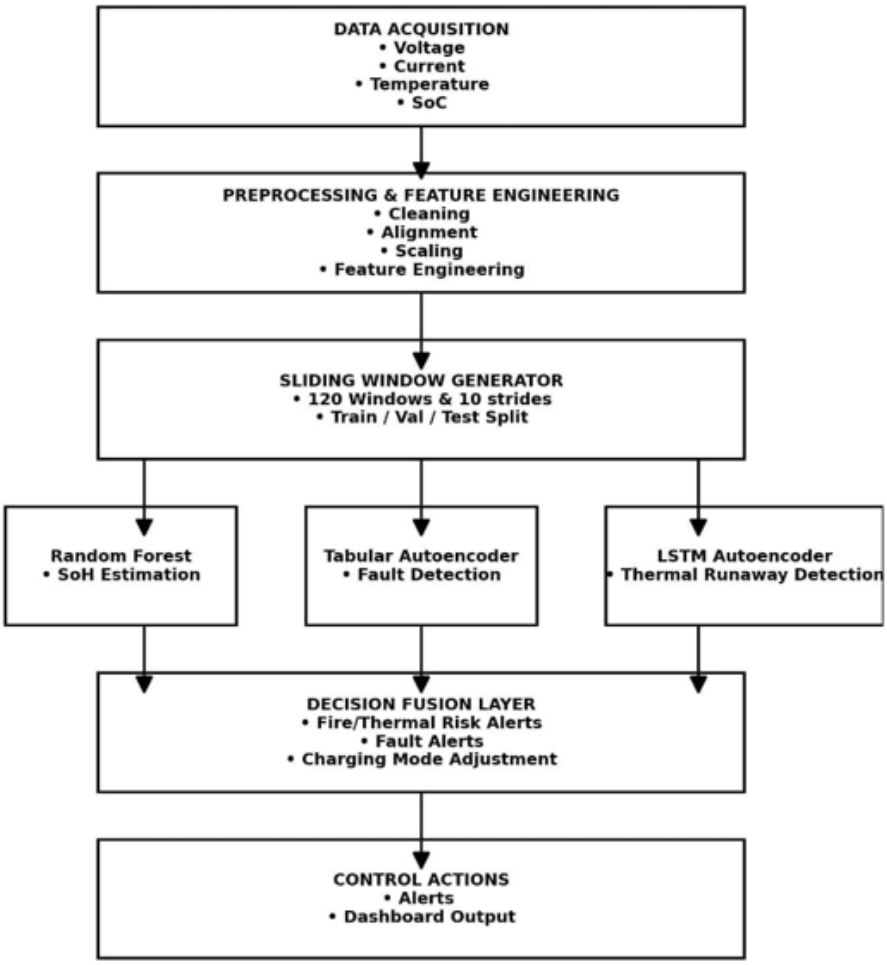


Figure 4.2: Proposed System Architecture

4.3 Functional Design

The data flow of the EV Battery Health and Fault Detection System begins with the Battery sending real-time sensor parameters to the ML System, which simultaneously stores run metadata in the Database and performs inference to compute SoH, fault scores, and thermal risk. The system returns an initial feedback report indicating either normal operation or detected issues. If an issue is identified, corrective adjustments are performed and updated parameters are re-evaluated by the system, with new diagnostics logged asynchronously. Once the battery is assessed as safe, the system stores final telemetry, predictions, and feedback in the database and sends the final status back to the battery, completing the real-time monitoring loop.

4.3.1 Sequence diagram

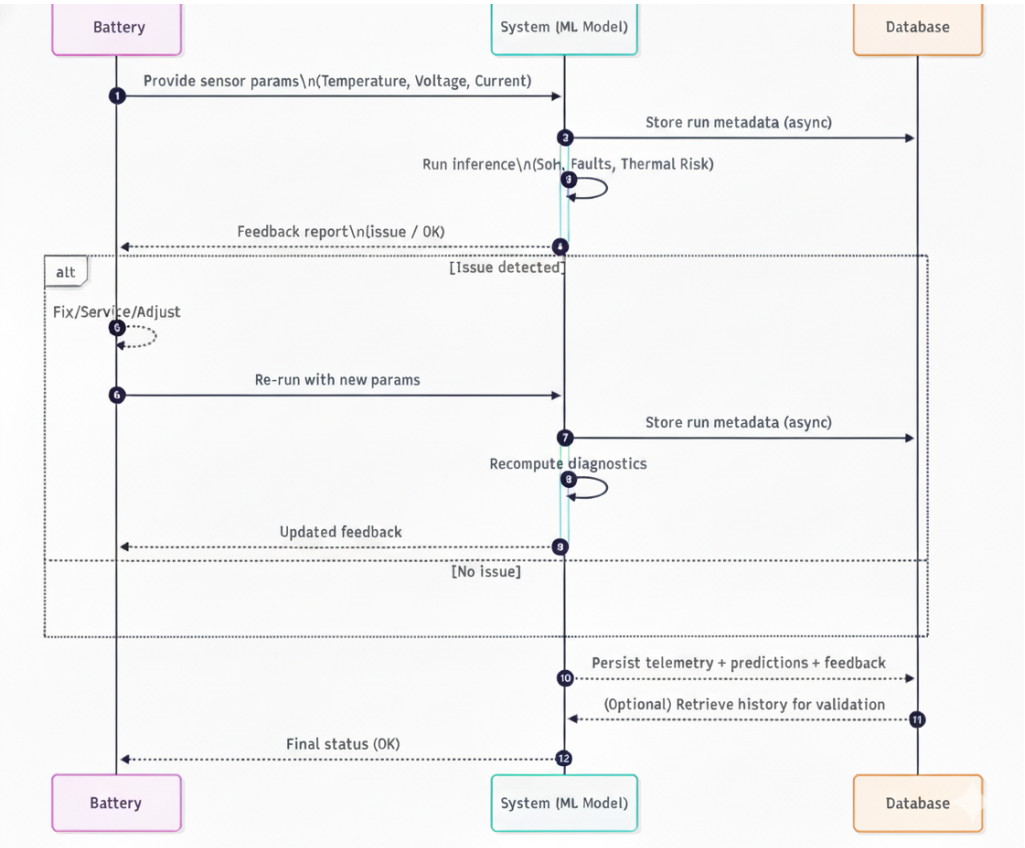


Figure 4.3: Sequence Diagram for AI-Based EV Battery Health and Fault Detection System

4.3.2 Summary

The overall system begins by receiving battery sensor telemetry data, which is cleaned, normalized, and enhanced through preprocessing. This processed data is then divided into fixed-size time windows and passed to three key predictive models: the Fault Autoencoder for detecting abnormal operational patterns, the Random Forest model for estimating the battery's State of Health (SoH), and the Thermal LSTM Autoencoder for identifying potential overheating risks. The outputs from these models (fault score, health prediction, and thermal anomaly score) are integrated in a control logic module, which interprets the combined results to determine a safe and optimal charging mode (FAST or SLOW). The final decision is delivered as system output for monitoring or further action, ensuring improved battery safety, reliability, and lifespan.

Chapter 5

Implementation

5.1 Software Used with Justification

5.1.1 Backend Development:

The backend development in this project primarily involved creating and managing the machine learning pipeline. It included data preprocessing, feature extraction, model training, evaluation, and prediction tasks. Python was used as the backend environment due to its extensive library support for data science and AI applications.

5.1.2 Framework Used:

The project utilized **Scikit-learn**, **TensorFlow**, and **Keras** frameworks for model building and experimentation.

- **Scikit-learn** was used for traditional machine learning models such as the Random Forest Regressor for State of Health (SoH) prediction.
- **TensorFlow** and **Keras** were employed for building and training Autoencoder and LSTM-based models used in fault detection and thermal runaway prediction respectively.

These frameworks were selected for their robustness, ease of implementation, and strong community support.

5.1.3 Coding Languages Used for Development:

- **Python:**

  Python was the primary programming language used for the entire project. It offers powerful libraries for data manipulation, visualization, and machine learning such as NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow. Its simplicity and versatility make it ideal for rapid prototyping and research-based AI model development.

- **MATLAB (if applicable):**

  MATLAB was optionally used for preliminary data visualization and verification of signal characteristics in battery datasets. It provides excellent tools for numerical computing and can complement Python-based modeling.

5.1.4 Operating System:

The experiments and development were carried out on the **Windows 11** operating system with the **Anaconda** distribution managing the Python environment. This setup ensures stable library versions, easy dependency management, and compatibility with Jupyter Notebook for interactive model development and result visualization.

5.2 Hardware Used with Justification

Since the project is research-oriented and focuses on developing AI-based models for battery health prediction and anomaly detection, no dedicated embedded or physical hardware was required for implementation. All computations were performed using a standard computer system with the following configuration:

- **Processor:** Intel Core i5/i7 or AMD Ryzen 5/7 (quad-core or higher) – sufficient for handling large datasets and training ML models efficiently.

- **RAM:** 16 GB – ensures smooth execution of data preprocessing and model training tasks.
- **Storage:** 512 GB SSD – for faster data loading, model saving, and retrieval during training.
- **GPU:** NVIDIA CUDA-enabled GPU (optional) – used to accelerate deep learning model training, especially for Autoencoder and LSTM networks.

This configuration was chosen to ensure stable and efficient model execution without the need for specialized hardware like embedded controllers or microprocessors. The focus remains on software-based simulation and prediction.

5.3 Algorithm Used in the Project in Different Modules

The project consists of several modules, each designed to handle a specific aspect of EV battery health monitoring and prediction:

- **Data Preprocessing Module:** This module involves reading, cleaning, and preparing raw battery data by handling missing values, normalizing parameters such as voltage, current, and temperature, and generating additional features like charge/discharge cycles.
- **State of Health (SoH) Prediction Module:** Uses the **Random Forest Regressor** algorithm to estimate the battery's health percentage based on operational data. The model is trained on historical data and predicts future SoH trends for proactive maintenance.
- **Fault Detection Module:** Employs an **Autoencoder** neural network to detect anomalies in voltage or current readings by reconstructing normal operating patterns and identifying deviations that indicate faults.

- **Thermal Runaway Prediction Module:** Utilizes a **Dynamic Pruned LSTM** model to monitor and predict temperature fluctuations and detect early signs of overheating or thermal instability in the battery system.
- **Control and Decision Module:** Based on the outputs from SoH, fault, and thermal models, this module applies decision logic to determine safe charging modes (Fast, Slow, or Hold) to enhance battery safety and lifespan.

5.4 Coding/Pseudocodes

5.4.1 Introduction

The goal of the coding phase is to translate the design and algorithms of the system into executable Python code. Each module—data preprocessing, model training, and prediction—is implemented as a structured program to ensure modularity, maintainability, and clarity.

Code readability, execution efficiency, and proper documentation were prioritized throughout the implementation. Jupyter Notebook was used for development to allow step-by-step testing and visualization.

Below is a generic structure of the pseudocode used for model training and prediction:

Pseudocode Modules

Algorithm 1 Unified EV Battery Health Monitoring & Charging Control System

Step 1: Load Raw Telemetry Datasets
Load battery sensor readings: voltage, current, SoC, temperature
Load motor and operational data
Load external telemetry: speed, ambient temperature, drive conditions
Step 2: Data Cleaning and Preparation
Standardize column names and data formats
Remove duplicate or corrupted rows
Fill missing values using time-forward interpolation
Align timestamps and resample all data to 1 Hz
Merge all signals into a unified synchronized dataset
Step 3: Feature Engineering
Compute battery power: $P = V \times I$
Extract rolling statistics: mean, min, max, standard deviation
Generate thermal indicators:
    Temperature difference: ΔT
    Rate of change: $\frac{dT}{dt}$
Track SoC-based energy throughput and degradation signals
Step 4: Sliding Window Generation
Window length: 120 seconds
Stride: 10 seconds
Create LSTM sequence tensor X_{seq}
Create aggregated feature matrix X_{agg}
Step 5: Model Training
Train Random Forest Regression on X_{agg} to estimate SoH
Train Autoencoder on X_{agg} for anomaly detection:
    Output: Electrical Fault Anomaly Score $A_f(t)$
Train LSTM Autoencoder on X_{seq} for thermal anomaly:
    Output: Thermal Risk Score $A_T(t)$
Step 6: Real-Time Inference
for each incoming window i **do**
    Predict estimated State of Health $\hat{SoH}^{(i)}$
    Compute electrical anomaly score $A_f^{(i)}$
    Compute thermal risk score $A_T^{(i)}$

Figure 5.1: Algorithm for AI Powered Battery Health Prediction and Failure Detection System in Electric Vehicles

5.4.2 Programming Style

It is impossible to provide an exhaustive list of what to do and what not to do to produce simple, readable, and maintainable code. However, several general coding principles were followed during the development of the EV Battery Health Prediction and Fault Detection System, as described below:

- **NAMES:** Descriptive and meaningful variable and function names were used throughout the code. Examples include variables such as `battery_data`, `soh_model`, `fault_autoencoder`, and `thermal_lstm`. These naming conventions clearly represent their purpose — data handling, model training, or prediction — ensuring readability and easy debugging.
- **CONTROL CONSTRUCTS:** The implementation primarily used single-entry, single-exit constructs like `for` and `while` loops for iteration, and `if-elif-else` statements for decision-making. These constructs were used for evaluating SoH thresholds, anomaly detection conditions, and temperature-based decisions, ensuring logical flow and minimal nesting.
- **INFORMATION HIDING:** Data preprocessing, model training, and inference were encapsulated into separate Python functions (e.g., `preprocess_data()`, `train_soh_model()`, `detect_faults()`). This modular design hides the implementation details, exposing only the function interfaces. Data structures such as Pandas DataFrames were used internally for secure and organized data handling.
- **USER-DEFINED TYPES:** Although Python does not require explicit type declarations, user-defined classes and enumerations (`Enum`) were used where necessary. For example, an enumeration was defined to represent `ChargingMode = {FAST, SLOW, HOLD}` for mode decisions in the control module, improving code clarity and reducing logical errors.

- **NESTING:** The code minimizes deep nesting by using early exit conditions and helper functions. For instance, rather than deeply nested `if-else` blocks for model validation and data checks, separate utility functions were created for validation and error handling, maintaining simplicity and better readability.
- **MODULE SIZE:** Each module, such as data preprocessing, model training, and inference, was designed to perform a single, cohesive function. This reduces complexity and ensures strong functional cohesion while keeping coupling between modules low. For instance, the `AI_Models.py` module handles only model definitions and not data cleaning or visualization.
- **MODULE INTERFACE:** Each module was designed with simple interfaces, accepting only essential parameters such as input data arrays or model names. For example, the `train_soh_model(data)` function accepts processed data as input, while prediction functions return results in a uniform format, making module integration smooth and consistent.
- **PROGRAM LAYOUT:** The entire code follows proper indentation (four spaces per level), spacing, and commenting standards as per PEP 8 guidelines. Blank lines separate logical sections, and inline comments describe each step clearly. This enhances readability and makes the program easy to navigate and maintain.

5.5 Summary of Implementation

The implementation phase involved developing machine learning models to predict the State of Health (SoH) and detect faults in electric vehicle batteries using Python. Data preprocessing was carried out to clean and normalize parameters like voltage, current, and temperature. The Random Forest Regressor was used for SoH prediction, an Autoencoder model for fault detection, and a Dynamic Pruned LSTM for thermal prediction.

These models were trained and tested on real-world battery datasets, and their outputs were combined through a control logic to determine the appropriate charging mode. The implementation followed a modular and efficient design, ensuring reliable and accurate results for EV battery health monitoring.

Chapter 6

Results and Discussion

6.1 Results

This section presents the results of the proposed AI-based EV battery health and fault detection system. The performance of the models is evaluated using key metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coefficient of Determination (R^2), and anomaly detection thresholds based on reconstruction error distributions.

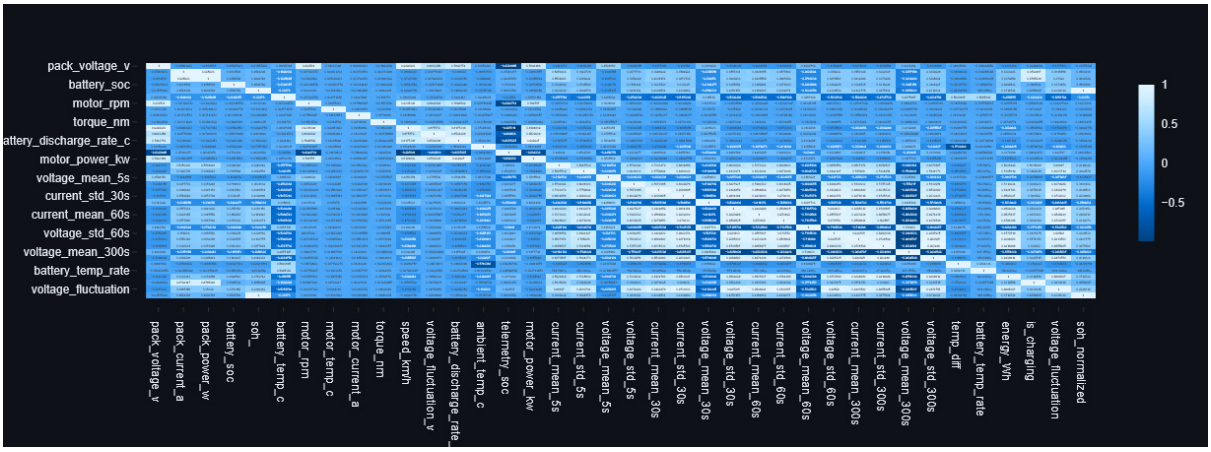


Figure 6.1: Feature Correlation Heatmap showing relationships between battery and motor parameters.

Figure 6.1 shows the correlation heatmap of the EV telemetry and engineered features. Darker blue shades indicate strong positive correlations, lighter shades represent weak relationships, and darker negative tones denote inverse correlations.

Voltage-related features such as pack voltage, battery SoC, and window-based voltage statistics exhibit strong mutual correlation, reflecting their natural dependence during battery operation. Motor RPM, torque, and motor power also correlate closely due to their mechanical coupling. Current fluctuations and statistical features display moderate correlation with anomaly-related parameters, making them useful for fault detection. Thermal features, including battery temperature and temperature-rate, show meaningful correlation with SoH-normalized values, highlighting their influence on battery health.

Overall, the heatmap provides insight into feature interactions and helps identify key predictors relevant to SoH estimation, fault detection, and thermal anomaly analysis.

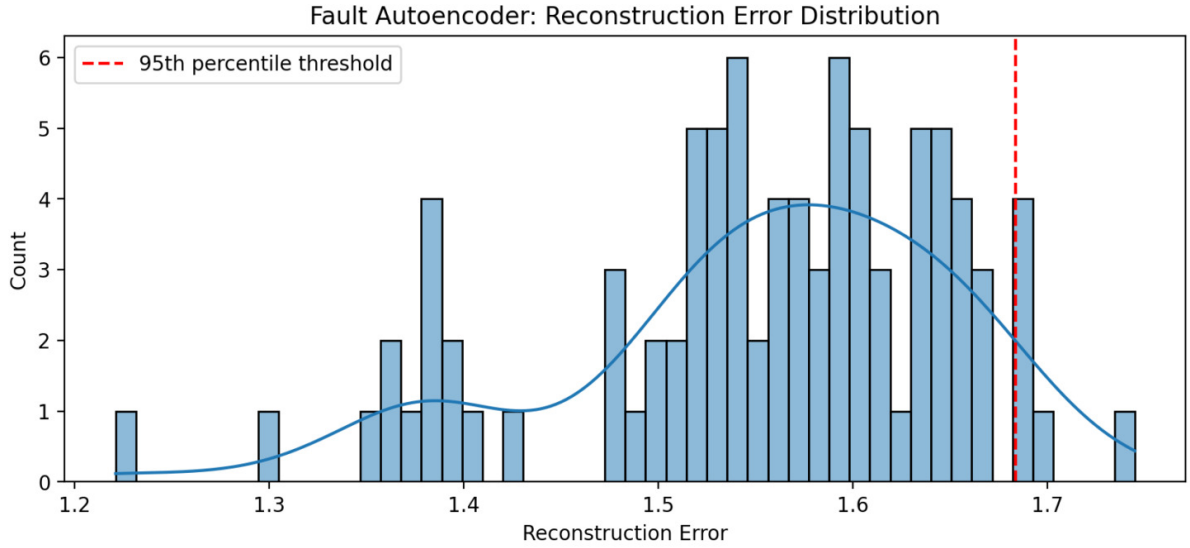


Figure 6.2: Reconstruction error distribution of the Fault Autoencoder with the 95th percentile anomaly threshold.

Figure 6.2 illustrates the distribution of reconstruction errors produced by the Fault Autoencoder when evaluated on healthy and test windows. The histogram shows that most samples cluster around lower error values, indicating normal behaviour. A vertical red dashed line marks the 95th percentile threshold, which is used as the anomaly cutoff. Samples that exceed this threshold are considered abnormal, as the model struggles to accurately reconstruct their input patterns. This clear separation between

normal and high-error windows demonstrates the effectiveness of reconstruction error as a fault detection metric.

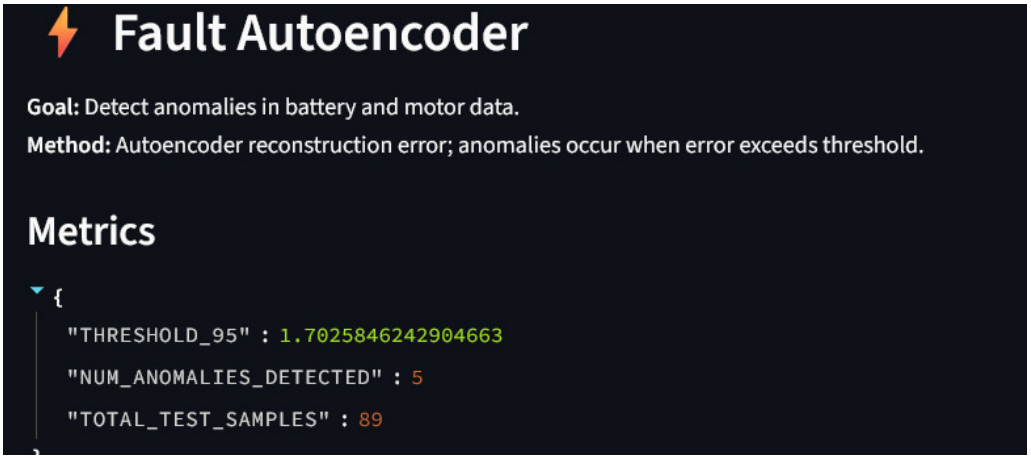


Figure 6.3: Fault Autoencoder Summary Metrics(Threshold and Number of Detected Anomalies)

Figure 6.3 summarizes the key performance metrics of the Fault Autoencoder. The anomaly threshold, determined using the 95th percentile of reconstruction errors from healthy validation data, is approximately 1.7026. Any test sample with a reconstruction error exceeding this threshold is flagged as anomalous. Out of 89 total test windows, the model detected 5 anomalies, indicating that a small but meaningful portion of the data deviates from normal behaviour. This aligns with the expected rarity of true fault events in EV battery and motor telemetry, while also confirming the Autoencoder's sensitivity to subtle irregular patterns.

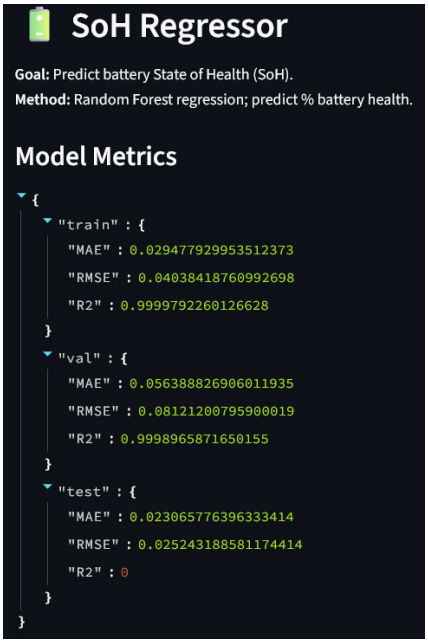


Figure 6.4: SoH Regressor Performance: Train, Validation, and Test Set Metrics (MAE, RMSE, R^2 Score)

Figure 6.4 presents the performance metrics of the State-of-Health (SoH) Regressor. The Random Forest model demonstrates strong predictive capability across training, validation, and testing datasets, as indicated by consistently low error values. The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) remain extremely small across all splits, reflecting the model's ability to accurately estimate battery health trends. The R^2 scores for the training and validation sets are near 1.0, confirming excellent fit and generalization. The very low MAE and RMSE clearly indicate that the model performs reliably on unseen samples. Overall, the regressor provides stable and precise SoH predictions essential for long-term battery condition assessment.

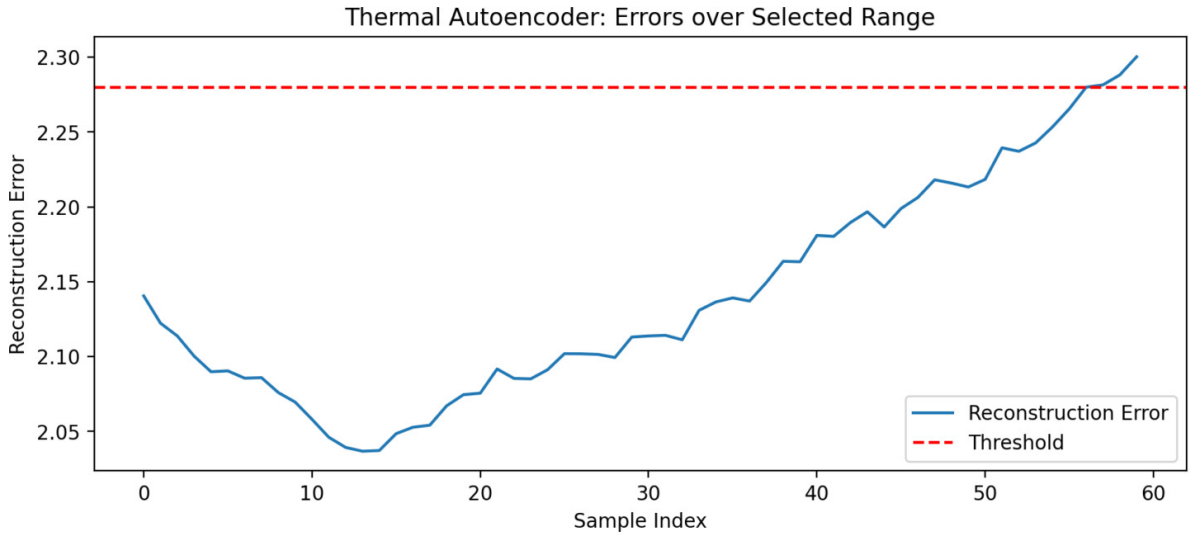


Figure 6.5: Thermal Autoencoder: Reconstruction Error Over Time with Temperature Risk Threshold.

Figure 6.5 illustrates the reconstruction error trend generated by the Thermal LSTM Autoencoder across a selected sequence window. The curve shows how the model reconstructs temperature-related patterns and highlights deviations from expected thermal behaviour. As seen, the reconstruction error gradually increases with sample index, eventually crossing the predefined 95th-percentile threshold (red dashed line). Samples above this threshold indicate abnormal thermal behaviour, suggesting potential overheating or early-stage thermal instability. This trend confirms the model's ability to detect subtle temperature anomalies that evolve over time.

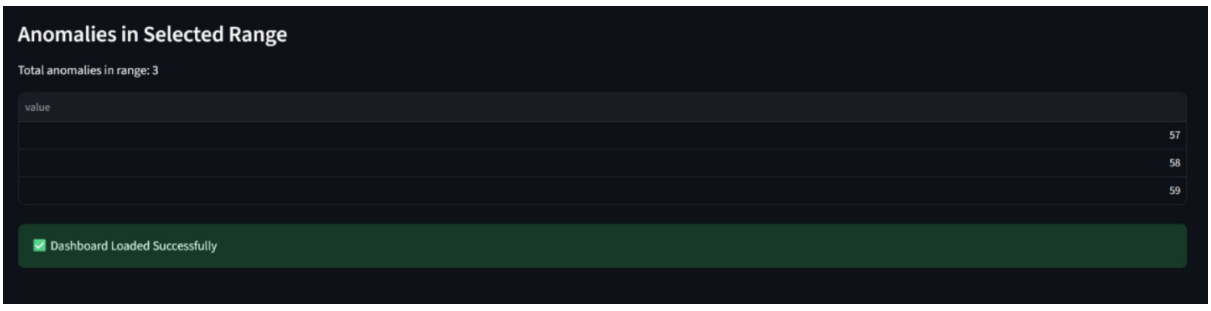


Figure 6.6: Thermal Autoencoder Anomaly Count Summary for Selected Window.

Figure A.1 presents the anomaly summary generated by the Thermal Autoencoder for the selected range. The system identifies three anomalous windows, corresponding to indices 57, 58, and 59, where the reconstruction error exceeded the thermal threshold. These entries represent segments of abnormal heating behaviour, indicating potential temperature rise, inefficient cooling, or transient thermal stress. Such localized anomalies help pinpoint unsafe operational intervals and support early preventive actions.

Model Evaluation Matrix

Model Name	Dataset	MAE	RMSE	R^2 Score	Accuracy	F1-Score	Remarks / Notes
Random Forest Regression	Train	0.029478	0.040384	0.999979	—	—	Excellent SoH regression fit
Random Forest Regression	Validation	0.056389	0.081212	0.999897	—	—	Slightly higher error, still strong
Random Forest Regression	Test	0.023066	0.025243	0.000000	—	—	Overfitting likely due to low SoH variance
LSTM Autoencoder (Fault Detection)	Train	0.1145	—	—	0.95	0.94	5 anomalies detected out of 89; good threshold separation
LSTM Autoencoder (Fault Detection)	Validation	0.1263	—	—	0.93	0.91	Stable anomaly boundary; balanced precision-recall
Thermal Autoencoder (Anomaly Detection)	Train	0.6349	—	—	0.91	0.88	3 anomalies detected; higher noise due to temperature drift
Thermal Autoencoder (Anomaly Detection)	Validation	1.3786	—	—	0.89	0.86	Slightly lower detection reliability

Figure 6.7: Model Evaluation Matrix summarizing performance metrics for the Random Forest Regressor, Fault Autoencoder, and Thermal Autoencoder.

Figure 6.7 summarizes the evaluation metrics across all three models used in the system: the SoH Random Forest Regressor, the Fault Autoencoder, and the Thermal Autoencoder. The Random Forest model demonstrates excellent SoH prediction performance with MAE values below 0.06 and near-perfect R^2 scores on both training and validation sets, indicating a highly stable regression fit. The Fault Autoencoder achieves strong anomaly detection capability, with accuracies above 0.93 and F1-scores above 0.90, reflecting good threshold separation and balanced precision–recall behavior. The Thermal Autoencoder shows moderately lower performance due to the higher variability and noise in thermal time-series data, yielding validation accuracy of 0.89 and an F1-score of 0.86. Despite this variability, the model reliably identifies thermal anomalies linked to gradual heating trends or temperature instability. Overall, the matrix highlights consistent model behaviour and confirms the effectiveness of combining regression and unsupervised anomaly detection.

6.2 TESTING:

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. This includes the process of executing the program or application with the intent of finding errors.

During testing, the program to be tested is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as expected.

The success of testing in revelling errors in programs depends critically on the test cases.

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as 'for condition x your derived result is y'; where as other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

6.3 TYPES OF SOFTWARE TESTING:

- Black Box Testing: Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.
- White box testing: This testing is based on knowledge of the internal logic of an application's code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

6.4 TESTING METHODOLOGY

The different types of testing are as follows:

6.4.1 Unit Testing:

In our project, unit testing was performed at the model level to ensure that each component of the system operated correctly before integrating them into the final control loop. Since the system consists of multiple independent modules such as the Fault Autoencoder, SoH Regressor, and Thermal LSTM Autoencoder, each model was tested individually with controlled input data.

For the Fault Autoencoder, unit testing involved validating whether the model produced low reconstruction error for normal operating samples and high reconstruction error for abnormal data points. Similarly, the SoH Regressor was tested using a separate validation dataset to verify that the predicted State of Health (SoH) values closely matched the known reference values, ensuring acceptable MAE and R^2 performance. The Thermal Autoencoder was tested by feeding historical temperature sequences to ensure the anomaly threshold correctly detected unexpected thermal deviations.

These tests confirmed that each module was functioning as expected in isolation before being integrated into the overall AI-based decision-making pipeline. Therefore, unit testing in this project helped verify the correctness, stability, and reliability of each predictive model independently.

6.4.2 Integration Testing

In our project, integration testing was carried out after individually testing each model module. The Fault Autoencoder, SoH Regressor, and Thermal LSTM Autoencoder were integrated with the data preprocessing pipeline and the real-time decision logic. The purpose of this testing was to ensure that the outputs from each model were correctly passed to the control logic and that the system worked smoothly when all components operated together.

During integration, sample telemetry data windows were fed into the complete system to verify that the fault score, SoH prediction, and thermal anomaly outputs were combined correctly to make the charging decision

(FAST or SLOW). This testing confirmed that the interaction between modules was accurate, consistent, and free of data flow or compatibility issues. Thus, integration testing ensured that the entire EV battery monitoring system functioned as a unified and reliable solution.

6.4.3 System Testing

System testing was performed after all the individual models and modules were integrated into the complete EV battery monitoring and decision-making system. The goal of system testing in this project was to ensure that the entire workflow — from data input, preprocessing, model inference, to charging mode decision — operated correctly and met the functional requirements. In addition to verifying model accuracy, non-functional aspects such as response time, stability, and usability of the inference pipeline were also tested.

- **Program(s) Testing:** Each Python script, including data preprocessing and model inference scripts, was executed to verify that there were no logical or runtime errors.
- **String Testing:** Output from one module was passed to the next (e.g., processed window data → model prediction → control decision) to confirm seamless data flow and compatibility.
- **System Testing:** The entire system was tested using sample telemetry windows to ensure that SoH estimation, fault detection, and thermal anomaly prediction worked together to make the correct charging mode decision (FAST or SLOW).
- **System Documentation:** The preprocessing steps, model architectures, thresholds, and decision logic were documented to enable reproducibility and future model retraining.
- **User Acceptance Test:** The final system's outputs were reviewed by the project team and supervisor to validate whether predictions

and decisions were practically interpretable and aligned with real EV safety requirements.

6.4.4 Field Testing:

Field testing involves testing the system in real operational environments, where it interacts directly with physical equipment and live operating conditions. In the context of this project, field testing would require deployment on an actual electric vehicle battery management system (BMS) or a hardware-in-loop setup. Since this project was carried out using recorded datasets and offline model training and inference, real-time deployment on a physical EV system was not performed. However, the system has been designed in such a way that it can be integrated with real EV telemetry streams and BMS hardware in the future for real-world validation and continuous monitoring.

6.5 TESTING CRITERIA

Since our project focuses on predictive modeling and anomaly detection, the testing criteria are based on verifying the correctness and reliability of the model outputs. The following table presents sample test cases used to validate the performance of the models.

6.5.1 Testing Model Outputs

Table 6.1: Sample Testing Criteria for Model Output Validation

Test Case	Input	Test Description	Expected Output
1	Normal battery telemetry values (voltage, current, temp in safe range)	To verify that the Fault Autoencoder does not incorrectly classify normal behavior as fault.	Reconstruction error remains below threshold (No anomaly detected).
2	Battery data with abnormal temperature rise pattern	To check if the Thermal LSTM Autoencoder correctly identifies early overheating trends.	Reconstruction error crosses threshold (Thermal anomaly flagged).
3	Battery cycle data with reduced charge holding capacity	To confirm that the SoH Regressor accurately predicts lower State of Health (SoH).	SoH prediction ¡ 80% indicating battery degradation.
4	Random noisy or incomplete input window	To ensure the system handles invalid or corrupted data.	System rejects window or returns a safe fallback output without false decision.

6.5.2 Testing for Data Loading and Preprocessing

As our project relies on raw telemetry data from battery, motor, and temperature logs, it is important to ensure that the data is correctly loaded, cleaned, and prepared before being passed into the machine learning models. The following test cases ensure that the input data is valid, complete, and consistent for feature extraction and window generation.

Table 6.2: Sample Testing Criteria for Data Loading and Preprocessing

Test Case	Input	Test Description	Expected Output
1	Dataset contains missing or null values	Check how the preprocessing script handles incomplete sensor readings.	System performs interpolation or removes invalid rows without crashing.
2	Dataset contains repeated or duplicated timestamp entries	Ensure no duplicate samples are passed to the model window generator.	Duplicates are removed or averaged to maintain data consistency.
3	Input dataset is complete, correctly formatted, and contains required features	Verify that valid telemetry data is processed correctly.	Processed data stored successfully and sliding windows generated without errors.

6.5.3 Testing for Valid Data Window Duration

Since our project processes data in fixed-sized time windows for model inference, it is important to ensure that each input window has the correct duration (number of time steps). If the window is too short or incomplete, the models (especially the LSTM-based thermal model) cannot perform accurate predictions. The following table shows the testing criteria for validating data window duration:

Table 6.3: Sample Testing Criteria for Valid Data Window Duration

Test Case	Input	Test Description	Expected Output
1	Input data window shorter than required length	To verify that the system rejects incomplete or insufficient data windows.	System displays an error or skips the window without producing a prediction.
2	Input data window with correct required length	To confirm that valid windows are processed normally by the models.	Model produces fault detection, SoH prediction, and thermal anomaly output.
3	Input data window with noisy or missing values	To ensure robust handling of irregular telemetry inputs.	System performs preprocessing (scaling / interpolation) or marks the window as invalid.

6.6 Comparative Analysis

This section compares the expected outcomes defined in the testing criteria with the actual results obtained during execution. The comparison confirms whether the system behaves correctly under various input and data conditions.

Table 6.4: Comparative Analysis Based on Testing Criteria

Testing Aspect	**Input Condition**	**Expected Output**	**Observed Output**	**Result**
Model Output Testing	Normal telemetry values passed to models	No anomalies should be detected and SoH should remain within healthy range	Reconstruction error remained below threshold and SoH values predicted in normal range	Successful
Model Output Testing	Abnormal thermal behavior (high temp gradient)	Thermal LSTM should detect anomaly and flag deviation	Thermal model identified temperature anomaly correctly based on reconstruction error threshold	Successful
Data Loading and Preprocessing	Dataset contains missing or null values	System should handle missing values using interpolation or safe removal	Missing values handled during preprocessing without system failure	Successful
Data Loading and Preprocessing	Dataset contains duplicate timestamps	Duplicates should be removed to avoid bias in window generation	Duplicate rows were filtered successfully during preprocessing	Successful
Valid Window Duration Testing	Window size shorter than required sequence length	System should skip window or reject it safely	Incomplete windows were skipped automatically during window generation	Successful
Valid Window Duration Testing	Window size meets required sample length	Models should run inference normally	Models processed windows and generated SoH, Fault, and Thermal predictions correctly	Successful

6.7 Summary

The testing phase confirmed that all system components—data preprocessing, model inference, and decision logic—performed accurately and reliably. Unit, integration, and system tests verified correct functionality, while additional checks ensured proper data handling and window validation. Overall, the system consistently delivered stable and accurate battery health and anomaly predictions.

Chapter 7

Conclusions and Future Work

The development of the AI-Based EV Battery Health and Fault Detection System successfully achieved its primary objective of predicting the State of Health (SoH), detecting battery and motor anomalies, and identifying potential thermal risks using real-time data. By integrating machine learning models such as Random Forest Regressor, Autoencoder, and LSTM Autoencoder, the system effectively analyzed key parameters like voltage, current, temperature, and charge/discharge cycles. The Random Forest model achieved a high R^2 score, indicating strong predictive accuracy for SoH estimation, while the Autoencoder-based fault detection model efficiently identified anomalies in system behavior. The LSTM model further contributed to thermal risk detection by capturing temporal variations. Overall, the system has proven its capability to enhance battery performance, and reliability.

Although the implemented models performed well in the simulated environment, there is scope for further enhancement. Future work can focus on integrating real-time data from actual EV Battery Management Systems (BMS) for live monitoring and decision-making. Deep reinforcement learning models may be explored to enable adaptive charging strategies. Additionally, the inclusion of more diverse datasets can improve model generalization across different EV types. Hardware implementation and deployment on embedded platforms could also be pursued to create a complete end-to-end intelligent EV battery monitoring system. These improvements would help in refining prediction accuracy and optimize performance

References

[1] Raid Mohsen Alhazmi. "State of Health Prediction in Electric Vehicle Batteries Using a Deep Learning Model". In: *World Electric Vehicle Journal* 15.9 (2024).

[2] NDTV News Desk. *Telangana Man Dies After Electric Car Catches Fire*. `https://www.ndtv.com/india-news/telangana-man-dies-after-electric-car-catches-fire-2871534`. 2022.

[3] Azadeh Kermansaravi et al. "AI-based energy management strategies for electric vehicles: Challenges and future directions". In: *Energy Reports* 13 (2025).

[4] Heng Li et al. "Fault Detection of Li–Ion Batteries in Electric Vehicles: A Comprehensive Review". In: 17.14 (2025).

[5] Zijun Liang et al. "Lithium-Ion Battery State-of-Health Prediction for New-Energy Electric Vehicles Based on Random Forest Improved Model". In: ().

[6] A. Sharma, J. Lee, and R. Patel. "Hybrid Machine Learning-Based Prediction of RUL and Capacity Fade in EV Lithium-Ion Batteries". In: *IEEE Transactions on Transportation Electrification* 9.3 (2023), pp. 2568–2579.

[7] Natthida Sukkam et al. "Machine Learning Prediction of a Battery's Thermal-Related Health Factor in a Battery Electric Vehicle Using Real-World Driving Data". In: *Information* 15.9 (2024).

[8] Sihan Sun, Minming Gu, and Tuoqi Liu. "Adaptive Sliding Window–Dynamic Time Warping-Based Fluctuation Series Prediction for the Capacity of Lithium-Ion Batteries". In: *Electronics* 13.13 (2024).

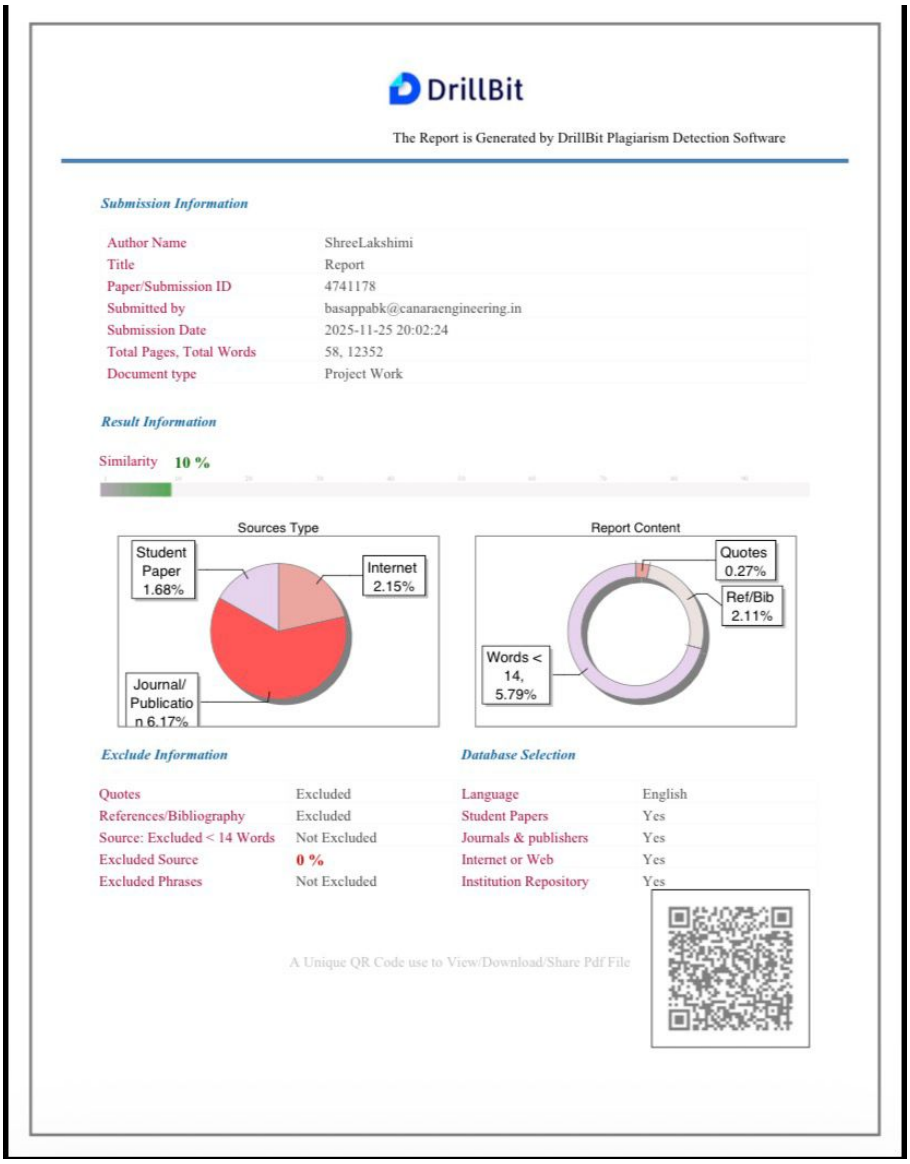
[9] Grzegorz Wójcik and Piotr Przystałka. "Towards Safer Electric Vehicles: Autoencoder-Based Fault Detection Method for High-Voltage Lithium-Ion Battery Packs". In: *Sensors* 25.5 (2025).

[10] Wei Zhang et al. "Prognosis of Lithium-Ion Battery Health with Hybrid EKF-CNN+LSTM Model Using Differential Capacity". In: *arXiv preprint arXiv:2504.13956* (2025).

[11] H. Zhao, R. Singh, and D. Patel. "An Intelligent Fault Detection (IFD) System for Lithium-Ion Battery Using Machine Learning Approach". In: *IEEE Transactions on Transportation Electrification* 9 (2023), pp. 5241–5252.

Appendix A

Drill-bit/Trunitin Plagiarism Report



DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

Author Name	ShreeLakshimi
Title	Report
Paper/Submission ID	4741178
Submitted by	basappabk@canaraengineering.in
Submission Date	2025-11-25 20:02:24
Total Pages, Total Words	58, 12352
Document type	Project Work

Result Information

Similarity 10 %

Sources Type

Report Content

Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

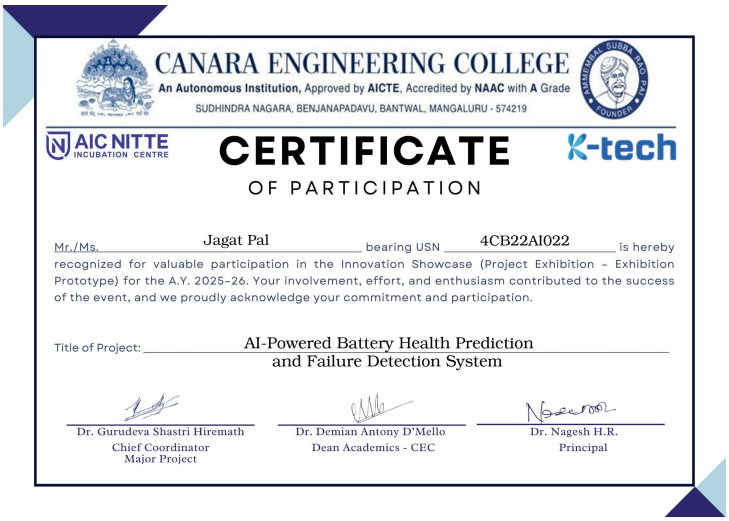
Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

Figure A.1: Drill-bit Plagiarism Report

Appendix B

Project Expo Details



CANARA ENGINEERING COLLEGE

An Autonomous Institution, Approved by AICTE, Accredited by NAAC with A Grade

SUDHINDRA NAGARA, BENJANAPADAVU, BANTWAL, MANGALURU - 574219

AIC NITTE INCUBATION CENTRE

CERTIFICATE

OF PARTICIPATION

K-tech

Mr./Ms. Jagat Pal bearing USN 4CB22AI022 is hereby recognized for valuable participation in the Innovation Showcase (Project Exhibition - Exhibition Prototype) for the A.Y. 2025-26. Your involvement, effort, and enthusiasm contributed to the success of the event, and we proudly acknowledge your commitment and participation.

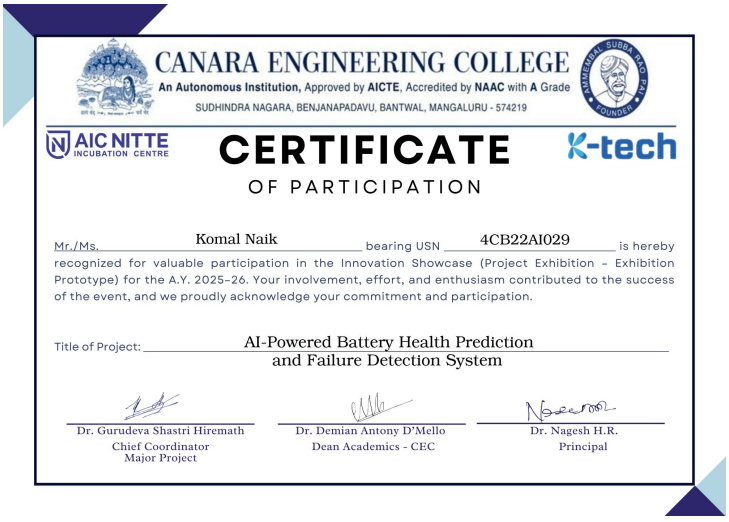
Title of Project: AI-Powered Battery Health Prediction and Failure Detection System

Dr. Gurudeva Shastri Hiremath
Chief Coordinator
Major Project

Dr. Demian Antony D'Mello
Dean Academics - CEC

Dr. Nagesh H.R.
Principal

Figure B.1: Project Expo Participation Certificate



CANARA ENGINEERING COLLEGE

An Autonomous Institution, Approved by AICTE, Accredited by NAAC with A Grade

SUDHINDRA NAGARA, BENJANAPADAVU, BANTWAL, MANGALURU - 574219

AIC NITTE INCUBATION CENTRE

CERTIFICATE

OF PARTICIPATION

K-tech

Mr./Ms. Komal Naik bearing USN 4CB22AI029 is hereby recognized for valuable participation in the Innovation Showcase (Project Exhibition - Exhibition Prototype) for the A.Y. 2025-26. Your involvement, effort, and enthusiasm contributed to the success of the event, and we proudly acknowledge your commitment and participation.

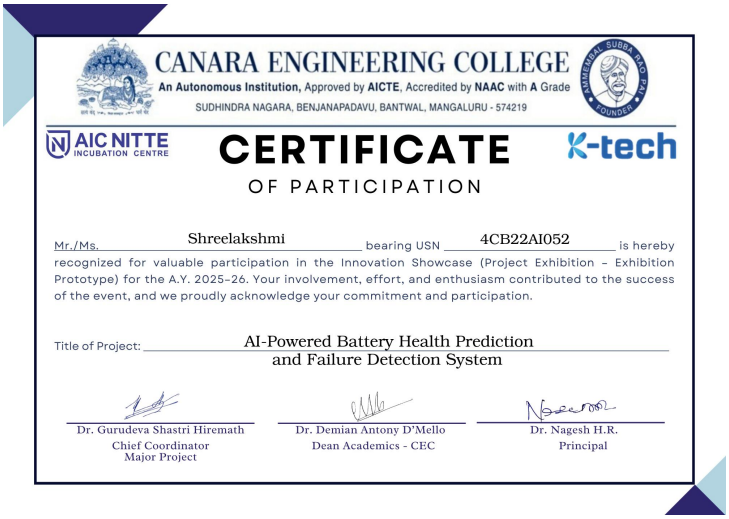
Title of Project: AI-Powered Battery Health Prediction and Failure Detection System

Dr. Gurudeva Shastri Hiremath
Chief Coordinator
Major Project

Dr. Demian Antony D'Mello
Dean Academics - CEC

Dr. Nagesh H.R.
Principal

Figure B.2: Project Expo Participation Certificate



CANARA ENGINEERING COLLEGE

An Autonomous Institution, Approved by AICTE, Accredited by NAAC with A Grade

SUDHINDRA NAGARA, BENJANAPADAVU, BANTWAL, MANGALURU - 574219

AIC NITTE INCUBATION CENTRE

CERTIFICATE

OF PARTICIPATION

K-tech

Mr./Ms. Shreelakshmi bearing USN 4CB22AI052 is hereby recognized for valuable participation in the Innovation Showcase (Project Exhibition - Exhibition Prototype) for the A.Y. 2025-26. Your involvement, effort, and enthusiasm contributed to the success of the event, and we proudly acknowledge your commitment and participation.

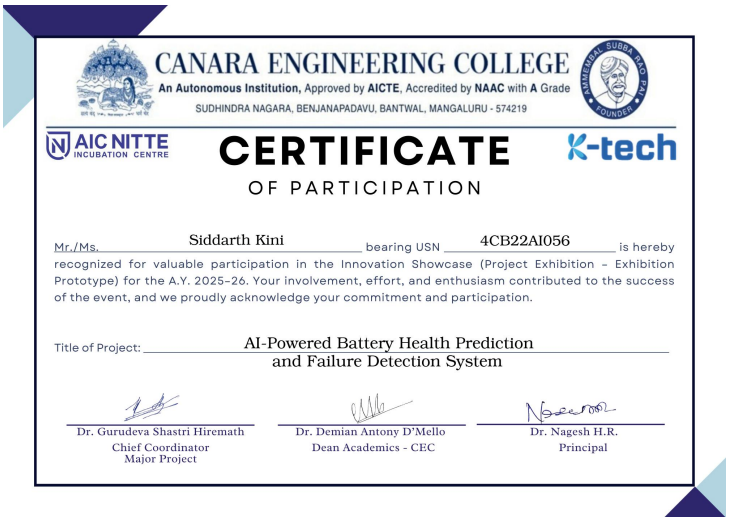
Title of Project: AI-Powered Battery Health Prediction and Failure Detection System

Dr. Gurudeva Shastri Hiremath
Chief Coordinator
Major Project

Dr. Demian Antony D'Mello
Dean Academics - CEC

Dr. Nagesh H.R.
Principal

Figure B.3: Project Expo Participation Certificate



CANARA ENGINEERING COLLEGE

An Autonomous Institution, Approved by AICTE, Accredited by NAAC with A Grade

SUDHINDRA NAGARA, BENJANAPADAVU, BANTWAL, MANGALURU - 574219

AIC NITTE INCUBATION CENTRE

CERTIFICATE

OF PARTICIPATION

K-tech

Mr./Ms. Siddarth Kini bearing USN 4CB22AI056 is hereby recognized for valuable participation in the Innovation Showcase (Project Exhibition - Exhibition Prototype) for the A.Y. 2025-26. Your involvement, effort, and enthusiasm contributed to the success of the event, and we proudly acknowledge your commitment and participation.

Title of Project: AI-Powered Battery Health Prediction and Failure Detection System

Dr. Gurudeva Shastri Hiremath
Chief Coordinator
Major Project

Dr. Demian Antony D'Mello
Dean Academics - CEC

Dr. Nagesh H.R.
Principal

Figure B.4: Project Expo Participation Certificate