Visvesvaraya Technological University, Belagavi
590018

PROJECT REPORT
ON
AI-POWERED EV BATTERY FIRE PREVENTION SYSTEM

Submitted in partial fulfillment for the award of degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

Karthik Kumar	**4CB22CS055**
Kshithij Rai K	**4CB22CS063**
Manish Shetty	**4CB22CS066**
Mohammad Wafiq Hammaba	**4CB22CS072**

Under the Guidance of
Dr. Nagesh H R
Principal, Canara Engineering College

DEPT. OF COMPUTER SCIENCE AND ENGINEERING
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

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"AI-POWERED EV BATTERY FIRE PREVENTION SYSTEM"** carried out by

Karthik Kumar	**4CB22CS055**
Kshithij Rai K	**4CB22CS063**
Manish Shetty	**4CB22CS066**
Mohammad Wafiq Hammaba	**4CB22CS072**

the bonafide students of VII semester Computer Science And Engineering in partial fulfillment for the award of Bachelor of Engineering in Computer Science And Engineering of the Visvesvaraya Technological University, Belagavi during the year 2025-2026. It is certified that all corrections/suggestions given during internal reviews are included for Internal Assessment. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Dr. Nagesh H R
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

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the entire work embodied in this Project Report titled **"AI-POWERED EV BATTERY FIRE PREVENTION SYSTEM"** has been carried out by us at CANARA ENGINEERING COLLEGE, Mangaluru under the supervision of Dr. Nagesh H R, for the partial fulfillment of Bachelor of Engineering in Computer Science And Engineering. This report has not been submitted to this or any other University for the award of any other degree.

Karthik Kumar	**4CB22CS055**
Kshithij Rai K	**4CB22CS063**
Manish Shetty	**4CB22CS066**
Mohammad Wafiq Hammaba	**4CB22CS072**

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 **Dr. Nagesh H R**, Principal,Canara Engineering College and Project mentor **Mr. Sudarshana Karkala**, eMobility Software Architect, iTelematics Software Private Limited, Bangalore for their guidance and valuable suggestions which helped us to complete this project. We also thank our Project coordinators **Dr. Sunil Kumar**, Dept of Computer Science And Engineering, for their consistent encouragement.

We owe a profound gratitude to **Dr. Karthik Pai**, Head of the Department, Computer Science And Engineering, 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** for their support and encouragement.

We would like to thank all faculty and staff of the Deprtment of Computer Science And Engineering who have always been with us extending their support, precious suggestions, guidance, and encouragement through the project.We also express our gratitude to our beloved friends and parents for their constant encouragement and support.

Karthik Kumar	**4CB22CS055**
Kshithij Rai K	**4CB22CS063**
Manish Shetty	**4CB22CS066**
Mohammad Wafiq Hammaba	**4CB22CS072**

ABSTRACT

The increasing adoption of Electric Vehicles (EVs) has created a strong need for advanced safety systems that ensure reliable and hazard-free battery operation. EV batteries operate under high energy densities and can be susceptible to overheating, imbalance, or sudden failures, which may lead to critical issues such as thermal runaway and fire hazards. Traditional monitoring methods are mostly static, device-specific, or limited to basic data visualization without intelligent analysis. This project aims to address these limitations by providing a more intelligent, proactive, and comprehensive battery monitoring platform.

The proposed system is designed to collect real-time data from Battery Management System (BMS) sensors, including key electrical and thermal parameters such as voltage, current, temperature, State of Charge (SOC), and State of Health (SOH). This data is continuously streamed into the software using IoT communication channels to ensure uninterrupted device monitoring. By integrating multiple sensors and a reliable data acquisition layer, the platform ensures that users receive accurate and up-to-date information on battery performance and potential risks. The real-time collection of data acts as the foundation for predictive analytics and informed decision-making.

A core objective of the project is the integration of AI-powered models capable of detecting anomalies and forecasting potential fire events before they occur. These models analyze patterns and deviations in the incoming sensor data to determine early warning signals that traditional systems could miss. The backend includes REST APIs for prediction handling, while the frontend displays results through a clean and intuitive dashboard. Users receive visual feedback, notifications, and alerts whenever unsafe conditions are identified, enabling immediate action and improving safety.

In addition to real-time risk detection, the platform supports data visualization, historical trend analysis, and secure access through encryption and user authentication mechanisms. These additional features make the system suitable not only for individual users but also for manufacturers, service engineers, and large-scale EV deployments. Altogether, this project delivers a complete end-to-end solution that enhances EV battery safety through intelligent monitoring, modern software design, and AI-driven failure prevention.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The rapid transition to electric vehicles (EVs) marks a significant milestone in the global effort to reduce carbon emissions and embrace sustainable transportation. At the heart of EVs lies the lithium-ion battery—an essential component responsible for energy storage and propulsion. Despite advancements in battery technology, one of the most critical challenges remains ensuring battery safety, particularly concerning issues like thermal runaway, voltage imbalance, and fire hazards.

Battery Management Systems (BMS) are traditionally employed to monitor and control key battery parameters, including voltage, temperature, State of Charge (SOC), and State of Health (SOH). However, conventional BMS platforms often lack advanced features such as AI-driven predictive analytics, intuitive user dashboards, and real-time anomaly detection. These gaps hinder early fault detection and compromise safety, reliability, and overall user experience.

This project addresses these limitations by designing a comprehensive software platform that bridges data acquisition from sensors and BMS modules with intelligent analytics and user-friendly visualization. The proposed system incorporates real-time monitoring, machine learning-based anomaly detection, and secure data communication to proactively identify and mitigate battery-related risks.

By using open-source technologies like Flask for API development and Streamlit for front-end visualization, the platform ensures scalability and ease of use. Furthermore, the integration of AI predictions enhances the system's ability to forecast hazardous conditions such as overheating or fire risks, enabling timely intervention.

The project focuses on simulation and software-based proof of concept (PoC), using mock datasets and virtual sensors to emulate real-world conditions. The long-term vision is to extend this system into a fully deployable cloud or mobile solution for EV manufacturers, researchers, and fleet operators.

1.2 Motivation and Problem statement

Motivation

The motivation behind this project stems from the urgent need to enhance EV battery safety through a more intelligent, user-friendly, and proactive software system. By integrating artificial intelligence into the monitoring framework, the system can detect abnormal behavior patterns and predict potential failures before they escalate into dangerous incidents. This not only improves safety but also extends battery life and reduces maintenance costs.

This project aims to fill the gap between raw battery data and actionable insights by developing a software platform that features real-time data acquisition, AI-based anomaly detection, secure communication protocols, and an intuitive dashboard for visualization. The solution is designed to be scalable, secure, and suitable for future integration into commercial or cloud-based EV ecosystems.

Problem Statement

As electric vehicles (EVs) become more mainstream, the demand for safer and more reliable battery systems has grown significantly. Despite their advantages in energy efficiency and environmental impact, EVs still face serious safety concerns, especially related to battery malfunctions such as overheating, short circuits, and thermal runaway events that may lead to fires or explosions. Traditional Battery Management Systems (BMS) are designed to monitor battery health and performance, but they are often limited in scope, lacking predictive capabilities and comprehensive user interfaces.

One of the primary limitations of current BMS solutions is the inability to proactively detect early signs of battery failure using real-time data analytics and intelligent decision-making. Most systems provide reactive alerts rather than predictive insights, which can delay appropriate interventions and compromise safety. Moreover, the lack of centralized dashboards and integration with AI tools hinders effective monitoring and decision-making for users, manufacturers, and researchers.

1.3 Objectives

- **To design a mobile application for real-time monitoring of EV battery health:**
  This objective focuses on building an application that continuously displays key battery parameters such as voltage, temperature, SOC, and SOH. Real-time monitoring ensures that the user always has the latest information about the battery's condition and performance.

- **To integrate AI/ML models through Flask API for predicting fire risks:**
  The system connects to machine learning models hosted on a Flask server, allowing the app to analyze sensor data and predict possible fire hazards. This adds a proactive layer of safety by detecting risks before they become critical.

- **To subscribe to MQTT-based simulators for receiving real-time sensor data:**
  MQTT acts as the communication protocol that streams continuous battery data into the app. Using simulators helps mimic real EV conditions during testing and ensures that the application can handle live IoT data streams.

- **To implement an alert system that warns users of unsafe conditions:**
  When the AI detects a risk or battery data crosses unsafe thresholds, the app sends immediate notifications. These alerts help the user respond quickly and prevent potential failures or accidents.

- **To provide offline storage and support historical trend analysis:**
  The application stores past sensor data even when the device is offline. Users can review previous trends, analyze performance over time, and detect long-term changes such as degradation or recurring issues.

1.4 Scope and limitations

Scope

The scope of this project includes the development of a real-time monitoring system for Electric Vehicle batteries that continuously tracks critical parameters such as voltage, current, temperature, State of Charge (SOC), and State of Health (SOH). The system also integrates artificial intelligence and machine learning models through a Flask-based API to provide predictive analysis and early detection of fire risks. To support live data flow, the application utilizes MQTT-based communication, allowing it to receive and analyze continuous battery sensor readings from simulators or IoT devices. Additionally, the system offers a warning mechanism that immediately alerts users whenever unsafe or abnormal battery conditions are detected, helping prevent failures before they escalate. The application also stores historical performance data locally and provides visualizations to enable long-term analysis of battery health. Furthermore, the modular and scalable design allows the system to expand to support multiple EV batteries and more advanced AI models in future development.

Limitations

Despite its capabilities, the project has some limitations. The AI models are trained on limited datasets, which may restrict their accuracy in detecting rare or complex battery failure patterns, especially those that occur in real-world EV conditions. Testing is currently based on simulated data rather than actual electric vehicle hardware, meaning the system's performance in real deployments may need further refinement and calibration. The solution focuses on software-level prediction and alerting but does not include integration with hardware-based safety mechanisms such as automatic fire suppression or circuit isolation. Additionally, the system is developed primarily as a mobile application and does not yet provide multi-platform access or centralized web monitoring. Since real-time streaming and prediction depend on stable network connectivity, users in low-connectivity areas may experience delays in updates or notifications. Finally, the accuracy and reliability of prediction results depend heavily on the quality and volume of data used to train the machine learning models, which may improve over time as more data becomes available.

1.5 Relevance and Type

Relevance

Electric Vehicles are becoming increasingly popular, but battery safety continues to be one of the most critical challenges faced by manufacturers and users. EV batteries are vulnerable to issues such as overheating, voltage imbalance, and unexpected failures, which can escalate into dangerous situations like thermal runaway and fire hazards. Existing monitoring systems are often limited, device-specific, or lack predictive analytics, making it difficult to identify risks before they occur. This project is highly relevant as it provides a smart and modern solution that integrates real-time IoT-based data with AI-driven prediction models to offer early warnings and prevent potential failures. By combining live monitoring, machine learning analysis, user alerts, and data visualization, the project supports both enhanced safety and improved reliability for electric vehicle battery systems in real-world usage.

Type

This project is a software-based IoT and AI application development project focused on enhancing electric vehicle battery safety. It involves implementing a mobile application that communicates with MQTT-based simulators or IoT devices to receive continuous battery data. The system also incorporates machine learning models deployed through Flask APIs, making it a blend of IoT, mobile development, artificial intelligence, and data analytics. In nature, the project is both experimental and application-oriented, as it tests real-time data handling, prediction accuracy, and alerting mechanisms in a simulated environment while targeting future integration with real EV hardware. It represents a practical engineering solution aimed at addressing a critical problem in the transportation and renewable energy domain.

1.6 Organization of the report

Chapter 1: Introduction

This chapter introduces the project by explaining the rising importance of electric vehicles and the critical challenges related to battery safety. It highlights how EV batteries are vulnerable to issues such as overheating, imbalance, and unexpected failures, which may lead to fire hazards if not addressed in time. The chapter discusses how the integration of Artificial Intelligence and IoT-based data acquisition provides a modern and effective solution for continuous monitoring and predictive safety analysis. It presents the motivation behind the project, the problem statement, the defined objectives, scope, and relevance of building such a system. Additionally, this chapter outlines the project constraints, such as dataset limitations and real-time connectivity requirements, and concludes by presenting the structure and organization of the entire report.

Chapter 2: Literature Review

This chapter compiles existing technologies, research findings, and studies related to EV battery monitoring, battery management systems, and AI-enhanced prediction models. It examines traditional systems that rely on direct sensor monitoring and explores how modern approaches use machine learning and predictive analytics to detect battery degradation and fire risks. The chapter also reviews IoT-based real-time data handling, MQTT-based communication frameworks, historical trend analysis methods, and mobile dashboard solutions used in the EV industry. Limitations in current systems—such as lack of predictive capabilities, limited interoperability, and insufficient risk forecasting—are evaluated, establishing the gap this project aims to bridge by combining AI, IoT, and mobile application development.

Chapter 3: Methodology

This chapter explains the methodology followed in designing the AI-powered EV battery fire prevention system. It describes the data acquisition process from BMS sensors through MQTT-based communication channels, simulating real-time EV operating conditions. The chapter outlines system architecture, including mobile application development using Android Studio and backend API creation using Flask to host AI models. The machine learning component processes parameters like SOC, SOH, voltage, and temperature to analyze patterns and identify early signs of failure. The methodology also includes design modules, alert mechanisms, database handling for offline storage, and visualization features, ensuring the system provides both real-time responses and historical trend analysis for users.

Chapter 4: Implementation

This chapter details the implementation of the system, including the development environment, software tools, and technologies utilized. It describes the construction of the mobile application using Kotlin and Jetpack Compose, integration of the MQTT subscriber for live battery data, and communication with the Flask-based predictive model through REST APIs. The chapter explains backend model integration, real-time alert triggering logic, and the creation of dashboards and graphical displays. It also describes the use of Room Database for local offline storage, and how testing was conducted using MQTT simulators and test datasets. Challenges such as latency handling, efficient resource management, and communication reliability are also discussed.

Chapter 5: Results and Discussion

This chapter presents the outcomes of the system in monitoring EV batteries and predicting potential fire risks. It evaluates the real-time performance of the application, including responsiveness, accuracy of the AI predictions, and user experience in terms of dashboard clarity and alert reliability. The chapter discusses how the system performed under various simulated battery conditions and compares the results with conventional monitoring solutions that lack predictive capabilities. It also provides observations on system strengths, such as scalability and ease of use, along with areas where further improvements can enhance robustness, accuracy, and real-world applicability.

Chapter 6: Conclusion and Future Work

This chapter summarizes the contributions of the project toward improving EV battery safety using AI and IoT integration. It concludes that the project successfully provides real-time monitoring, predictive analysis, and user alerting in a unified mobile application, making it a meaningful contribution to the field of smart electric vehicle systems. The chapter also highlights possible future enhancements, such as integration with real EV hardware, development of more advanced AI models, cloud-backed data analytics, improved UI/UX elements, and features like GPS-based vehicle monitoring or automated safety responses. Expanding system compatibility and exploring large-scale deployment scenarios are also identified as promising directions for future development

Chapter 2

Literature Survey

2.1 Prediction of State-of-Health and Remaining Useful Life of Battery Based on Hybrid Neural Network Model

Author(s): Le Thi Minh Lien et al. [1]

2.1.1 Brief Findings:

This research highlights the importance of accurate predictions of lithium-ion battery degradation over time to ensure EV safety and performance. The authors developed a sophisticated hybrid neural network model that merges multiple learning techniques to estimate the State of Health (SOH) and Remaining Useful Life (RUL) with higher accuracy compared to traditional models.

2.1.2 Methodology Adopted:

- Datasets: CALCE dataset (University of Maryland), NASA battery degradation dataset
- Techniques: Empirical Mode Decomposition (EMD), Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM)

2.1.3 Results Achieved:

The hybrid EMD-CNN-BiLSTM model showed significantly improved prediction accuracy compared to single-architecture models such as GRU, RNN-AM, and standalone BiLSTM. However, practical deployment in live EV scenarios is still under research and needs validation.

2.2 Electric Vehicle Battery Lifetime Optimization Operational Mode

Author(s): Kurt Russell Kelty et al. [2]

2.2.1 Brief Findings:

This paper investigates the impact of user behavior, temperature, and charge/discharge cycles on EV battery longevity. It proposes smart operational modes that adapt battery usage based on user intent, thereby enhancing battery life without compromising vehicle performance.

2.2.2 Methodology Adopted:

- Developed operational modes for EVs:
  - Battery Life Mode
  - Standard Mode
  - Performance Mode
  - Storage Mode

2.2.3 Results Achieved:

Significant battery life extension and improved flexibility for end-users were reported. Nonetheless, the effectiveness depends on preset configurations and may not adapt well to highly variable user profiles. Integration requires robust thermal and charge management systems.

2.3 Battery Health Monitoring with AI

Author(s): Hari Prasad Bhupathi and Srikiran Chinta. [3]

2.3.1 Brief Findings:

This paper addresses the limitations of relying solely on voltage and capacity readings for battery health assessment. The authors propose a more comprehensive approach using AI to predict performance degradation by incorporating real-time sensor data.

2.3.2 Methodology Adopted:

- Used real-time battery data
- Applied machine learning models: decision trees, neural networks, and support vector machines (SVM)

2.3.3 Results Achieved:

The models accurately forecasted SOH and RUL across different usage patterns. AI predictions proved more adaptive to dynamic operational conditions compared to traditional rule-based systems. However, the approach faces challenges such as the need for large, clean, and diverse datasets and increased computational overhead.

2.4 Artificial Intelligence Approaches for Advanced Battery Management Systems

Author(s): M. S. Hossain Lipu et al. [4]

2.4.1 Brief Findings:

This survey paper provides a comprehensive analysis of AI-based techniques in Battery Management Systems (BMS). It emphasizes the shortcomings of existing rule-based systems and presents how AI can improve energy efficiency, battery longevity, and safety in EVs.

2.4.2 Methodology Adopted:

- Discussed datasets used in AI model training
- AI-driven BMS features: SOC management, over/undervoltage protection, temperature control, cell balancing

2.4.3 Results Achieved:

AI-based BMS implementations demonstrated enhanced real-time control, safety, and operational efficiency. However, real-time deployment remains complex due to integration challenges with legacy systems and the computational demand for processing large data streams in real-time.

2.5 Gaps Identified and Addressed

Gaps Identified in the existing BMS:

- **Lack of advanced predictive analytics in existing Battery Management Systems (BMS):** Most current battery management systems only monitor values like voltage or temperature and do not predict future risks, making fire prevention reactive instead of proactive.
- **Need for improved and reliable AI algorithms to detect failures and fire risks:** Many existing systems lack ML or AI models capable of analyzing patterns in battery data, resulting in poor early detection of abnormalities and failures.
- **Limited availability of real EV battery datasets for model training and testing:** EV battery data is difficult to obtain in large quantities, limiting the accuracy, robustness, and generalization of machine learning models.
- **Integration challenges with existing hardware, mobile apps, and IoT platforms:** Existing solutions are often hardware-dependent or work only within specific systems, making it difficult to integrate with mobile apps, dashboards, or third-party IoT devices.
- **High cost of commercial solutions:** Advanced EV monitoring and predictive systems can be expensive due to specialized hardware, making them difficult to deploy widely in low-cost or mass-market environments.

Gaps Addressed in the project:

- **AI-based prediction models added to the system:** The project integrates machine learning via Flask APIs to detect fire risks early, transforming monitoring from reactive to predictive and enhancing user safety.
- **Flexible and scalable IoT communication using MQTT:** The system communicates efficiently with data sources through MQTT, solving integration challenges and allowing real-time communication with multiple EV devices.
- **Offline historical storage to compensate for limited datasets:**By storing sensor data and trends over time, the system can build a growing dataset that helps improve prediction performance and support future model training.
- **Designed for multi-platform compatibility and easy expansion:** The architecture is modular and does not depend on specialized EV hardware, making the system easier to integrate into different vehicle types and future applications.

- **Cost-effective implementation using standard hardware and Android platform:** The project uses widely available tools, open-source libraries, and mobile development practices, enabling a more affordable and scalable solution compared to costly proprietary systems.

2.6 Summary

This chapter examined the current state of EV battery monitoring and safety systems and identified several limitations that reduce their performance in practical electric vehicle environments. Most existing Battery Management Systems still rely on traditional threshold-based monitoring and do not incorporate real-time predictive analytics, making them reactive rather than preventive. Additionally, many platforms are hardware-dependent, device-specific, and lack seamless integration with IoT-based data acquisition or cloud/mobile applications. This results in delayed failure detection, reduced situational awareness, and limited scalability when deployed across different EV models. The lack of advanced AI algorithms and large real-world datasets also restricts accurate fault prediction and early fire risk assessment.

To address these challenges, the proposed project introduces an AI-powered EV battery fire prevention system that combines real-time IoT data streaming through MQTT with machine learning models deployed via Flask APIs. The mobile application continuously monitors key health indicators such as SOC, SOH, temperature, and voltage, analyzes them in real time, and provides automatic alerts when early signs of risk are detected. The system also includes historical data storage, visualization, and a user-friendly dashboard that enhances accessibility and decision-making. This approach results in a scalable, cost-effective, and proactive solution that improves battery safety and empowers users to act before failures occur. The next chapter will present the technical design, system architecture, data flow, and implementation methodology that support this intelligent monitoring platform.

Chapter 3

Software Requirements Specification

3.1 Functional requirements

Functional requirements establish the particular actions that a software system performs and the way it must behave under specific input conditions. These functions represent the core system functions which enable data collection along with its processing and diagnosis and feedback result creation. The current project has established the following functional requirements:

a) Real-time monitoring of SOC, SOH, voltage, and temperature:

The system must continuously read battery parameters from the BMS and display them live to ensure users always have up-to-date visibility of battery health.

b) Fetch predictions from Flask REST API:

The mobile app must send sensor data to the backend AI model deployed on Flask and receive fire-risk predictions, enabling early fault detection.

c) Subscribe to MQTT feed for live sensor data:

The system should use MQTT to stream real-time data from a simulator or IoT device, mimicking how EV batteries transmit information during actual operation.

d) Trigger alerts for unsafe conditions:

When sensor values exceed safe limits or the AI model predicts a threat, the application must notify users instantly through warnings or pop-ups.

e) Store historical data offline:

The system must save previous readings on the mobile device using a local database, allowing users to access data even without network connectivity.

f) Provide visualization of past trends:

The app must offer graphs or charts so users can analyze battery health over time and detect recurring issues or performance degradation.

g) Simple and intuitive dashboard interface:

The system must present data in a clean and user-friendly format so that users can easily interpret all key battery metrics.

h) Support integration with multiple EV devices in the future:

The architecture should be flexible enough to connect to multiple battery sources or additional EV units without major redesign.

3.2 Non-Functional requirements

a) High reliability and accurate data output:

The system must ensure data displayed and predictions made are correct and consistent, as incorrect readings could lead to misinformed decisions.

b) Scalability to support multiple EVs simultaneously:

The design must allow the addition of more devices, users, or vehicles without performance issues or system redesign.

c) Low latency:

Real-time monitoring and alerts must work quickly so users receive updates and warnings without delay.

d) Secure and robust communication:

Data exchanged between the mobile app, MQTT broker, and AI back-end must be encrypted and protected from unauthorized access or manipulation.

e) Smooth and simple user interface:

The application must be easy to navigate even for non-technical users, ensuring a seamless user experience.

f) High availability with minimal downtime:

The system should remain operational most of the time and should not crash or become unstable during continuous data streaming.

g) Efficient resource utilization:

The app must minimize battery and memory usage so it can run effectively on standard Android devices without draining power rapidly.

h) Adaptable to future AI improvements:

The system should support updates to machine learning models, additional features, or new prediction parameters without large-scale rework.

3.3 Performance Requirements

a) Real-time Data Processing:

The system must be capable of receiving and processing live battery data with minimal lag, ensuring that parameter updates such as voltage, SOC, SOH, and temperature are displayed instantly on the mobile dashboard.

b) Low Latency Communication:

The communication between the MQTT data stream, the mobile application, and the Flask API should operate with less than one second of delay. This ensures timely detection of abnormal conditions and immediate alert responses.

c) Fast Prediction Response from AI Model:

The machine learning model hosted on the Flask server should generate predictions quickly, allowing the application to identify fire risks in real time without noticeable delay to the user.

d) Efficient Resource Utilization:

The application must run smoothly on standard Android devices without consuming excessive memory or processing power. It should operate effectively even when running continuously in the background.

e) High System Availability:

The system should remain stable and operational over long durations of continuous monitoring without crashes, interruptions, or significant performance degradation.

f) Support for Large Data Volumes Over Time:

As historical readings accumulate in local storage, the system must be able to manage and retrieve stored data efficiently without slowing down the application or loading times.

g) Ability to Scale with Increasing Numbers of EV Monitoring Points:

The architecture should support additional MQTT data sources and more battery packs in the future without requiring major redesign or affecting performance.

3.4 User interface design

The user interface of the AI-Powered EV Battery Fire Prevention System has been designed to provide a clean, intuitive, and easy-to-navigate experience for users monitoring electric vehicle battery health in real time. The design follows a modern mobile app layout built using Android Studio with Jetpack Compose, ensuring smooth performance, scalability, and flexibility for future enhancements. The dashboard is the central screen of the application, displaying key battery metrics such as voltage, current, temperature, State of Charge (SOC), and State of Health (SOH) in a well-organized manner. Readings update in real time as data is received from the MQTT stream, allowing users to instantly understand the condition of the battery at any moment. Graphs and visual indicators are included to provide clearer interpretation of trends over time, improving user awareness of gradual performance changes.

A separate alert interface is used to display notifications when the system detects abnormal conditions or when the AI model predicts a potential fire risk. This ensures that users receive immediate warnings with clear instructions or visual cues, enabling proactive decision-making. The app also provides access to historical data screens where past battery readings are stored in a local database and presented using charts or lists for easy review and comparison. Overall, the user interface emphasizes clarity, responsiveness, and efficient information delivery. By combining real-time monitoring, AI-powered alerts, and historical data visualization into a single mobile app, the design ensures that both technical and non-technical users can interact with the system effectively and benefit from improved EV battery safety.

3.5 Hardware and Software requirements

Software Requirements:

- **Android Studio (Kotlin + Jetpack Compose):** Used to develop the mobile application interface and core app functionality. Jetpack Compose provides a modern, smooth, and responsive UI framework for Android devices.
- **Flask REST API (Backend):** Hosts the AI/ML prediction model and processes battery data sent from the mobile application, returning fire-risk predictions in real time.
- **AI/ML Libraries (Python – TensorFlow / Scikit-Learn etc.):** Used to train and deploy the machine learning model that analyzes parameters such as SOC, SOH, voltage, and temperature for anomaly detection.
- **MQTT Protocol and Client Libraries (Paho Android Client):** Responsible for subscribing to live sensor data streams from simulators or IoT devices, allowing real-time updates to the mobile application.
- **Room Database (Offline Storage):** Enables storing battery data locally on the device, allowing users to view historical records even without internet access.
- **Retrofit (Network Communication in Android):** Handles communication between the mobile app and Flask backend API for sending data and receiving predictions securely and efficiently.

Hardware Requirements:

- **Android Smartphone or Emulator:** Acts as the runtime environment for the mobile monitoring application. Any Android 8.0+ device or emulator can display live monitoring data, alerts, and trends.
- **Laptop or Desktop Computer (Development System):** Used to build and run the development environment, including Android Studio, Flask backend server, and MQTT broker. A system with at least 8GB RAM and a modern processor (Intel i5 / Ryzen 5 or higher) is recommended.

- **MQTT Simulator / IoT Device (e.g., ESP32 – Optional):** Simulates real-time battery data streams such as voltage and temperature, allowing realistic testing of the system before integrating it with actual EV hardware.
- **Optional BMS Sensors (e.g., LM35, INA219):** Can be used in future extensions to supply real battery readings, temperature measurements, and current sensing for real-world deployment.
- **Optional GPS Module (e.g., Neo-6M):** Provides vehicle location tracking if location-based monitoring and mapping features are included in future upgrades.

3.6 Summary

The chapter describes the functional and non-functional requirements of the proposed AI-Powered EV Battery Fire Prevention System, designed to enhance electric vehicle safety through intelligent monitoring and predictive analysis. The solution integrates MQTT-based IoT communication with machine learning models deployed through Flask APIs to continuously gather real-time battery parameters such as voltage, temperature, State of Charge (SOC), and State of Health (SOH). These values are processed and analyzed using AI algorithms to detect abnormalities early, helping users and operators prevent thermal runaway, failures, and fire hazards in advance. The system is engineered to offer high responsiveness and low latency, ensuring that potential risks are identified and conveyed to users without delay.

The application provides real-time visualization of battery performance through a clean and intuitive mobile dashboard developed in Android Studio using Jetpack Compose. Users can easily track live metrics, receive alerts, and view historical data trends stored locally using a Room database, making performance analysis and long-term monitoring convenient even in offline conditions. Alerts and risk notifications are displayed instantly when unsafe thresholds are detected or when AI predictions indicate abnormal behavior, empowering users to make informed decisions quickly and effectively. The system architecture is designed to support flexible integration with multiple EV devices in the future, ensuring scalability and adaptability as the platform evolves.

Hardware support for development includes an Android smartphone or emulator for running the application, a computer for hosting the development environment and back-end services, and an optional MQTT simulator or IoT device for generating live data streams. In future deployments, sensor modules such as BMS circuitry and temperature or current sensors can be used to supply direct readings from real EV environments, and optional GPS integration can support vehicle location tracking. This modular design makes the system suitable for both simulated development environments and real-world

EV integration.

Non-functional requirements emphasize high data reliability, communication security, system scalability, and efficient performance even under continuous operation. The system ensures encrypted and robust communication between the mobile interface, MQTT source, and backend APIs, while maintaining smooth user interaction and minimal resource consumption on mobile devices. High availability and quick system responsiveness enable uninterrupted monitoring, making the solution dependable during extended usage. Altogether, this chapter establishes a solid foundation for developing a reliable, scalable, and cost-effective intelligent safety system that can significantly improve EV battery monitoring and risk prevention in real-world scenarios.

Chapter 4

System Design

4.1 Abstract Design

4.1.1 Architectural diagram

The architectural diagram represents the layered structure and interaction flow of the EV Battery Fire Prevention System. At the bottom layer, the system receives live battery data from an MQTT-based simulator or IoT device, which continuously streams key telemetry such as SOC, SOH, voltage, and temperature. This data is transmitted to the mobile application, which acts as the central interface for real-time monitoring and visualization. The app is built using Kotlin and Jetpack Compose and handles both data display and communication with backend services.

The backend layer consists of the Flask server, which exposes REST APIs for AI processing. When the mobile app sends sensor data to the server, the AI/ML prediction model analyzes the values and determines whether a fire risk or abnormal pattern is present. The prediction results are then returned to the application, which interprets and displays them to the user. Alongside this, the mobile app also stores historical data in a local Room database for offline access and long-term trend analysis.

The architecture is secured through encrypted communication and designed for scalability so that additional EV devices, advanced AI models, or cloud deployment can be integrated in the future. Overall, the architectural diagram shows how IoT data input, mobile processing, AI prediction, and user feedback are connected in a cohesive end-to-end system for proactive EV battery safety monitoring.

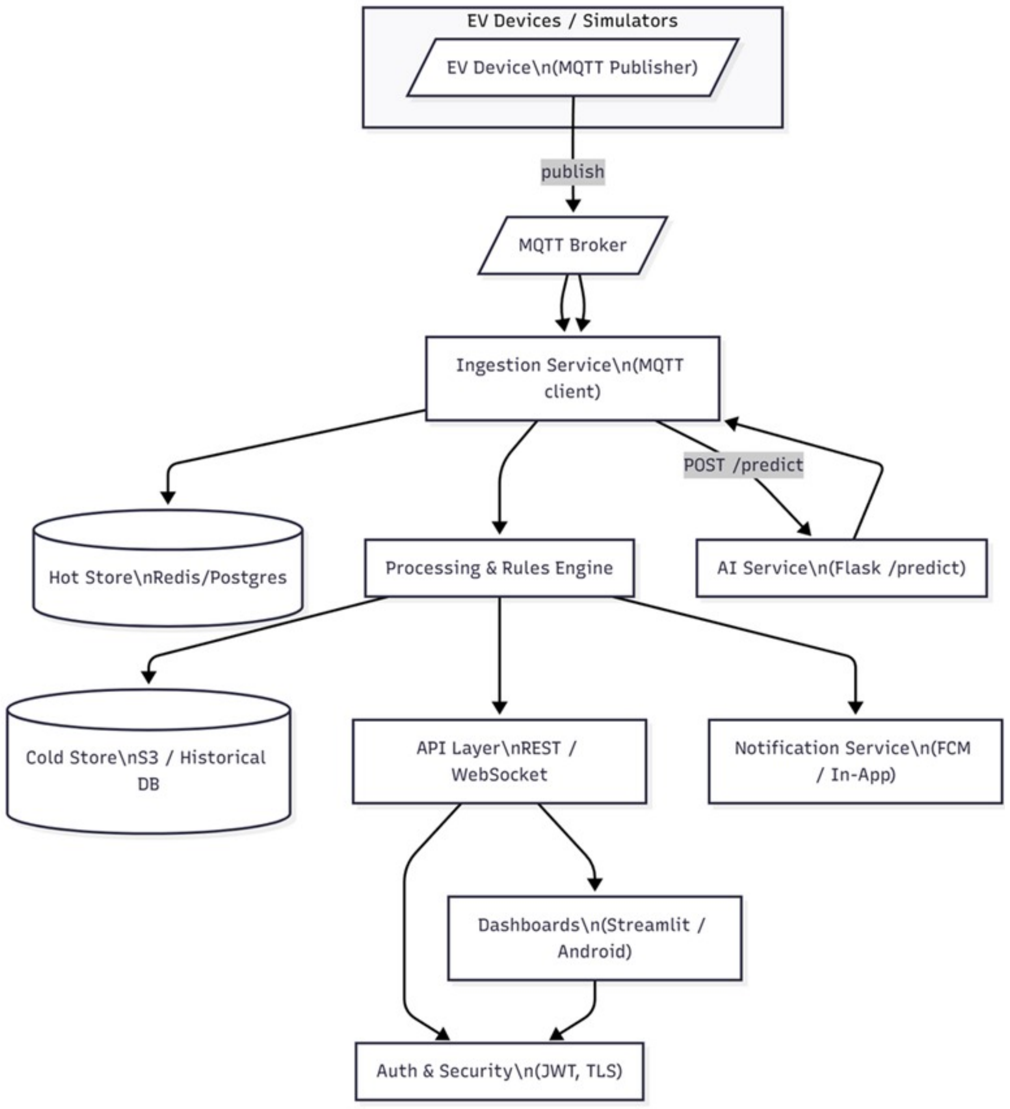

EV Devices / Simulators
EV Device\n(MQTT Publisher)
publish
MQTT Broker
Ingestion Service\n(MQTT client)
POST /predict
Hot Store\nRedis/Postgres
Processing & Rules Engine
AI Service\n(Flask /predict)
Cold Store\nS3 / Historical DB
API Layer\nREST / WebSocket
Notification Service\n(FCM / In-App)
Dashboards\n(Streamlit / Android)
Auth & Security\n(JWT, TLS)

Figure 4.1: Overall system architecture illustrating the interaction between data acquisition, preprocessing, model training, inference, and presentation layers.

4.2 Proposed Methodology

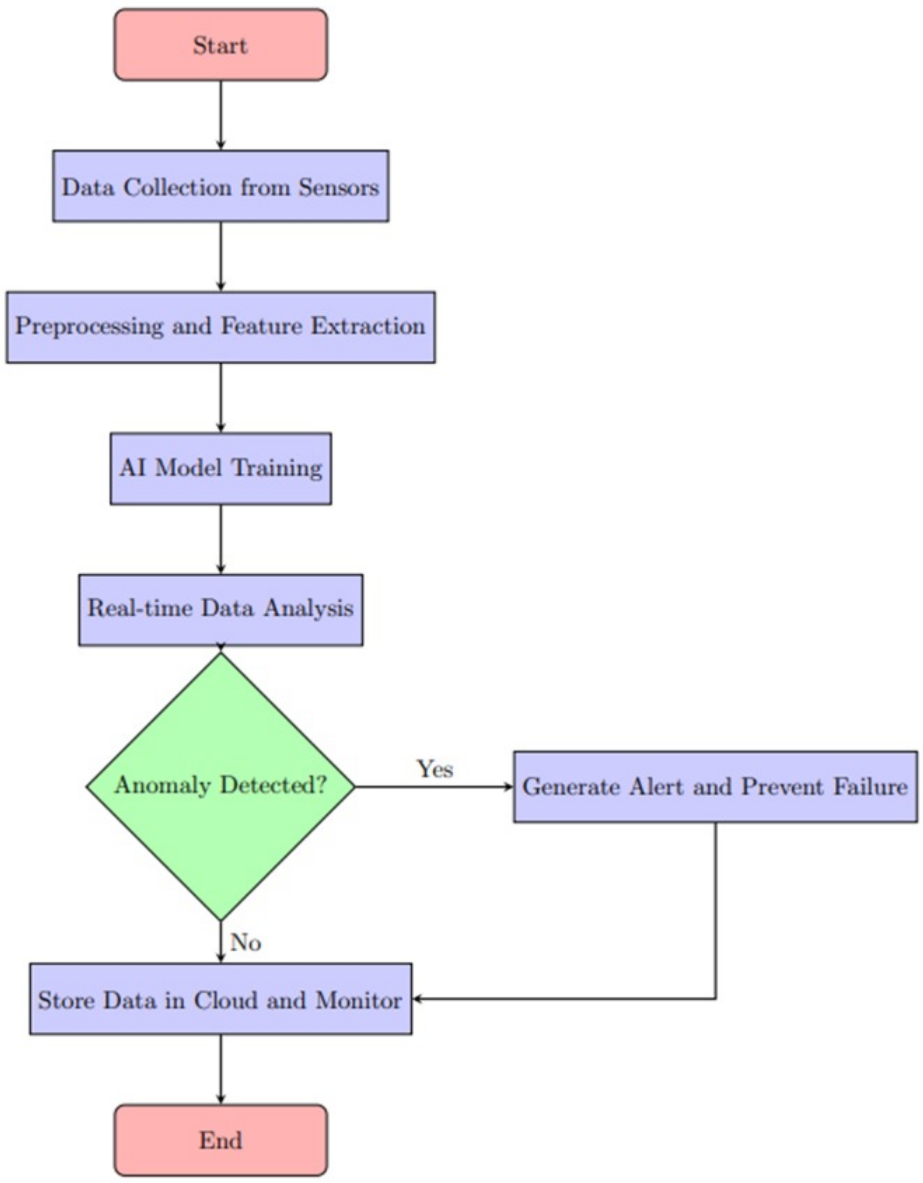

Start
Data Collection from Sensors
Preprocessing and Feature Extraction
AI Model Training
Real-time Data Analysis
Anomaly Detected?
Yes
Generate Alert and Prevent Failure
No
Store Data in Cloud and Monitor
End

Figure 4.2: Proposed system workflow.

4.3 Functional Design

4.3.1 Modular design diagram

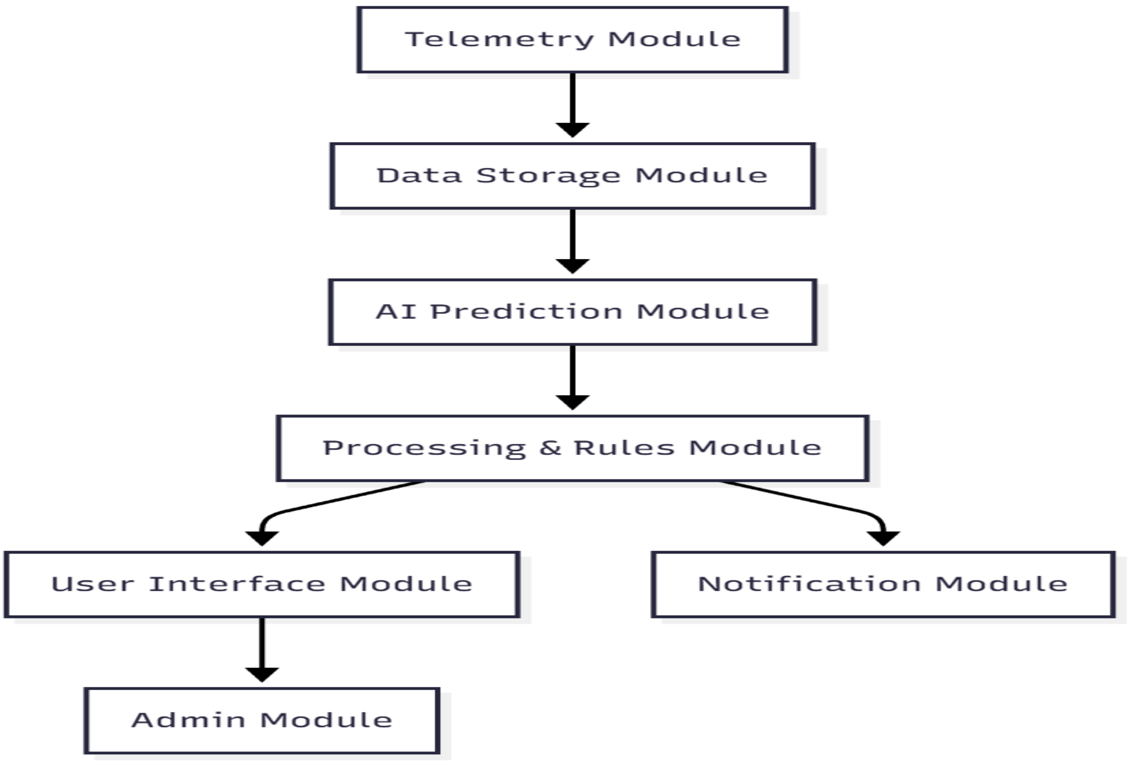


Figure 4.3: Modular design diagram representing different modules.

The modular diagram illustrates the overall structure of the EV Battery Fire Prevention System and shows how its major components work together. Live battery data is captured through the MQTT communication module, which streams real-time values such as voltage, SOC, SOH, and temperature from the sensor or simulator into the mobile application. The app then forwards this data to the Flask-based backend through REST APIs, where the AI/ML prediction module processes the readings and identifies early signs of risk or abnormal behavior. The prediction results and warnings are then sent back to the mobile application, where they are displayed through a user-friendly dashboard with visual indicators and alerts. A local storage module also records past data offline for trend analysis and historical review. Overall, the modular structure enables smooth communication, scalability, accurate analysis, and real-time user feedback, making the system flexible and efficient for EV battery safety monitoring.

4.3.2 Sequence diagram

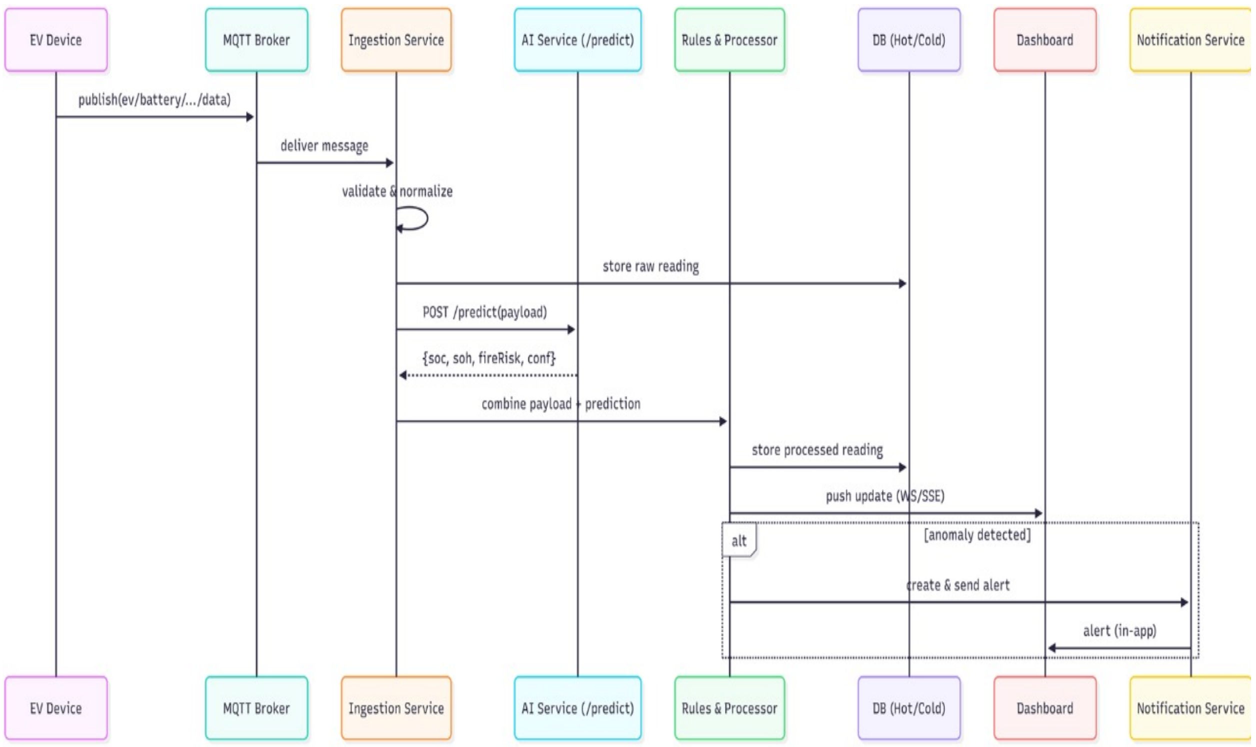


Figure 4.4: Sequence diagram.

The sequence diagram demonstrates the step-by-step interaction between the system components during real-time battery monitoring and prediction. The process begins when the MQTT data source streams live battery parameters, such as voltage and temperature, to the mobile application. The app first displays these readings to the user and then sends the collected data to the Flask backend using a REST API call. The backend receives the request and forwards the data to the AI/ML prediction module, which analyzes the values and determines whether there is a potential fire risk or abnormal battery condition. The prediction result is then sent back to the mobile application, which processes the response and generates alerts or notifications if necessary. At the same time, the app stores the received data in the local database for historical trend analysis. The diagram clearly shows the flow of actions from data reception, processing, prediction, and response delivery, ensuring continuous communication and real-time safety feedback within the system.

4.3.3 Use case diagram

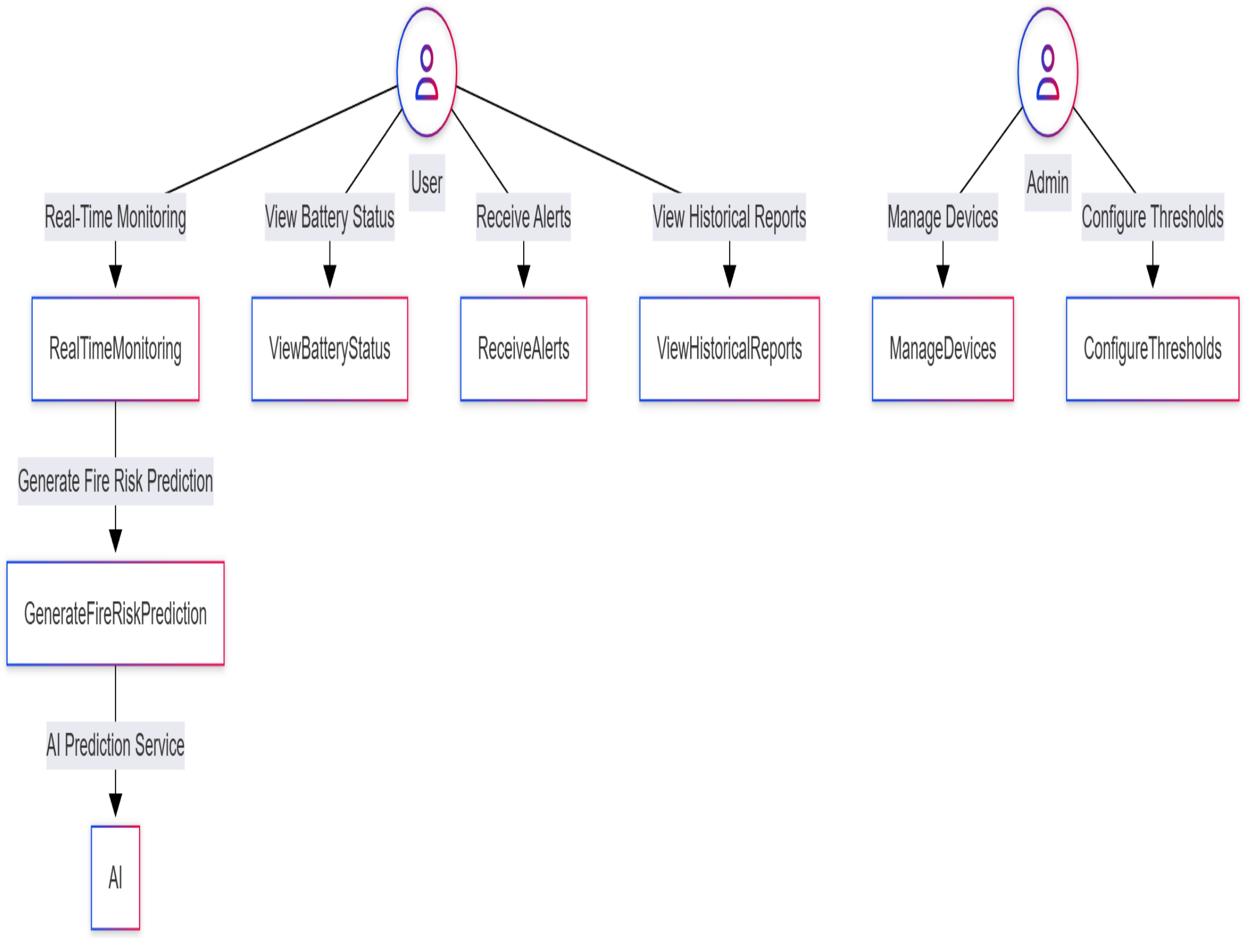


Figure 4.5: Use case diagram.

The use case diagram represents how different users and system components interact within the EV Battery Fire Prevention System. The primary user, typically an EV owner or operator, interacts with the mobile application to view real-time battery health metrics such as SOC, SOH, voltage, and temperature. The user can also receive notifications and alerts when the system detects abnormal sensor readings or when the AI model predicts potential fire hazards. The system communicates with the MQTT data source to continuously receive live battery parameters and updates. Additionally, the application sends this data to the Flask backend, where the AI/ML prediction engine analyzes it and returns safety assessments. The user can also review stored historical data through the local database, which allows tracking of long-term performance trends. Overall, the use case diagram outlines how the user, mobile app, MQTT system, and AI backend work together to ensure accurate monitoring, prediction, and safety feedback in the EV battery environment.

4.4 Control Flow Design

4.4.1 Activity diagram

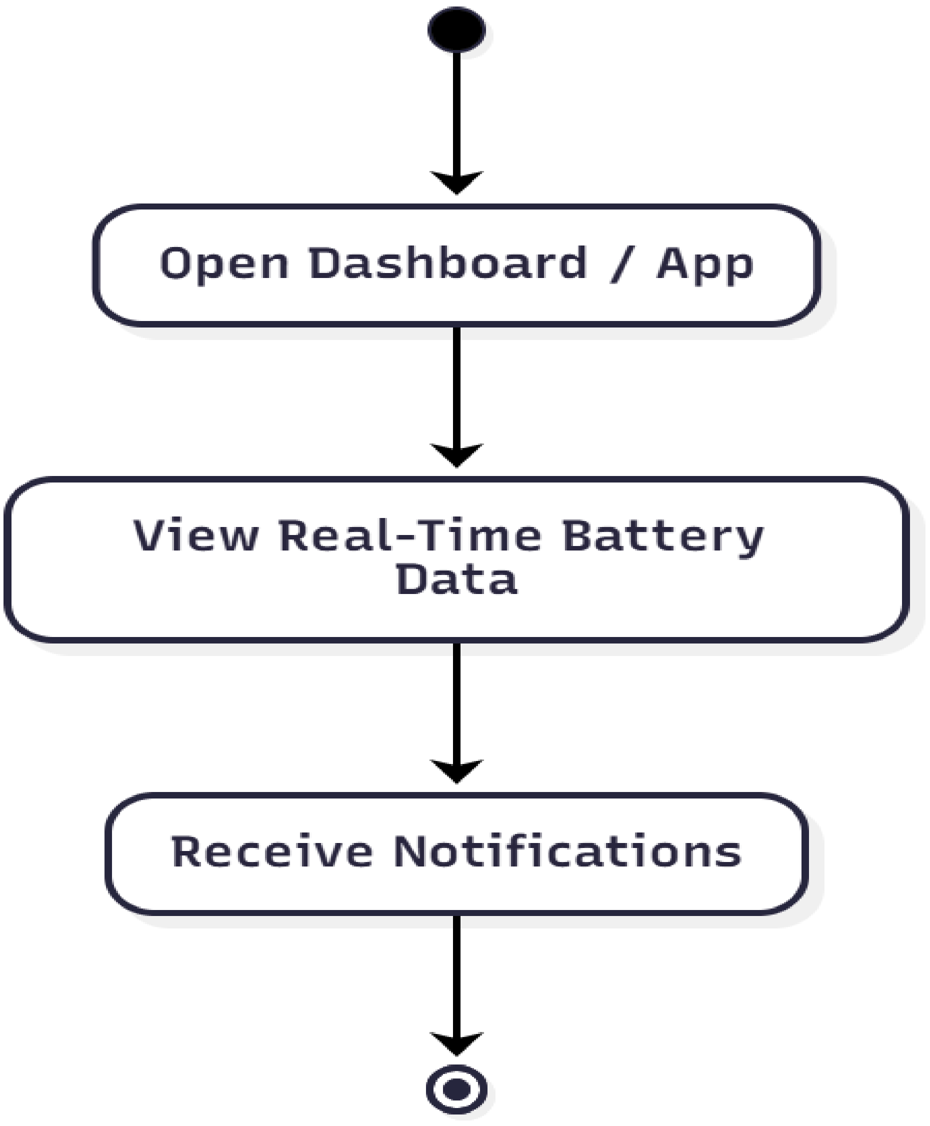


Figure 4.6: Activity diagram part-1.

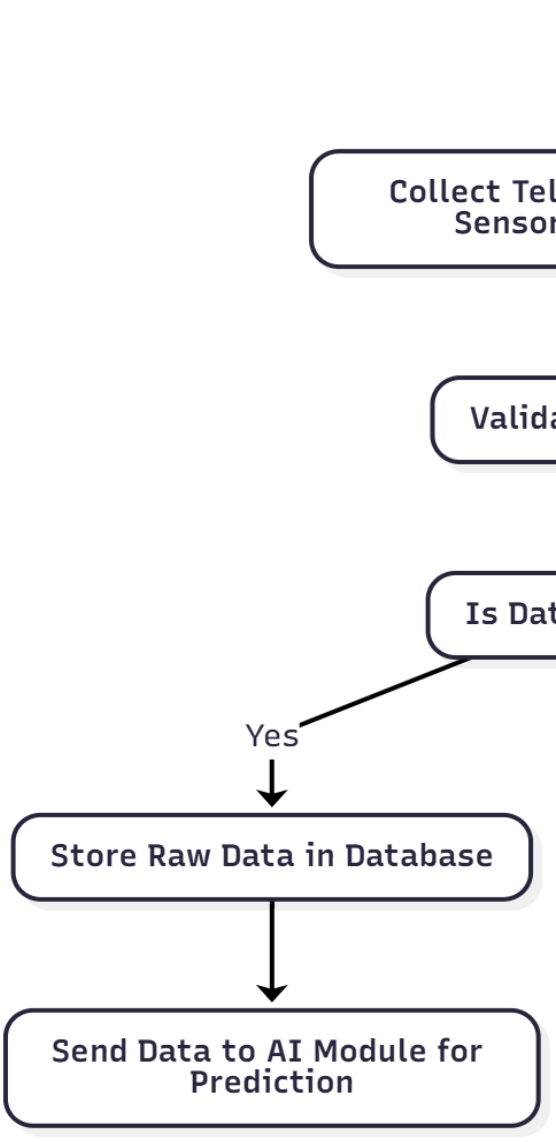

Collect Telemetry from Sensors / MQTT
Validate Data
Is Data Valid?
Yes
No
Store Raw Data in Database
Reject / Log Invalid Data
Send Data to AI Module for Prediction

Figure 4.7: Activity diagram part-2.

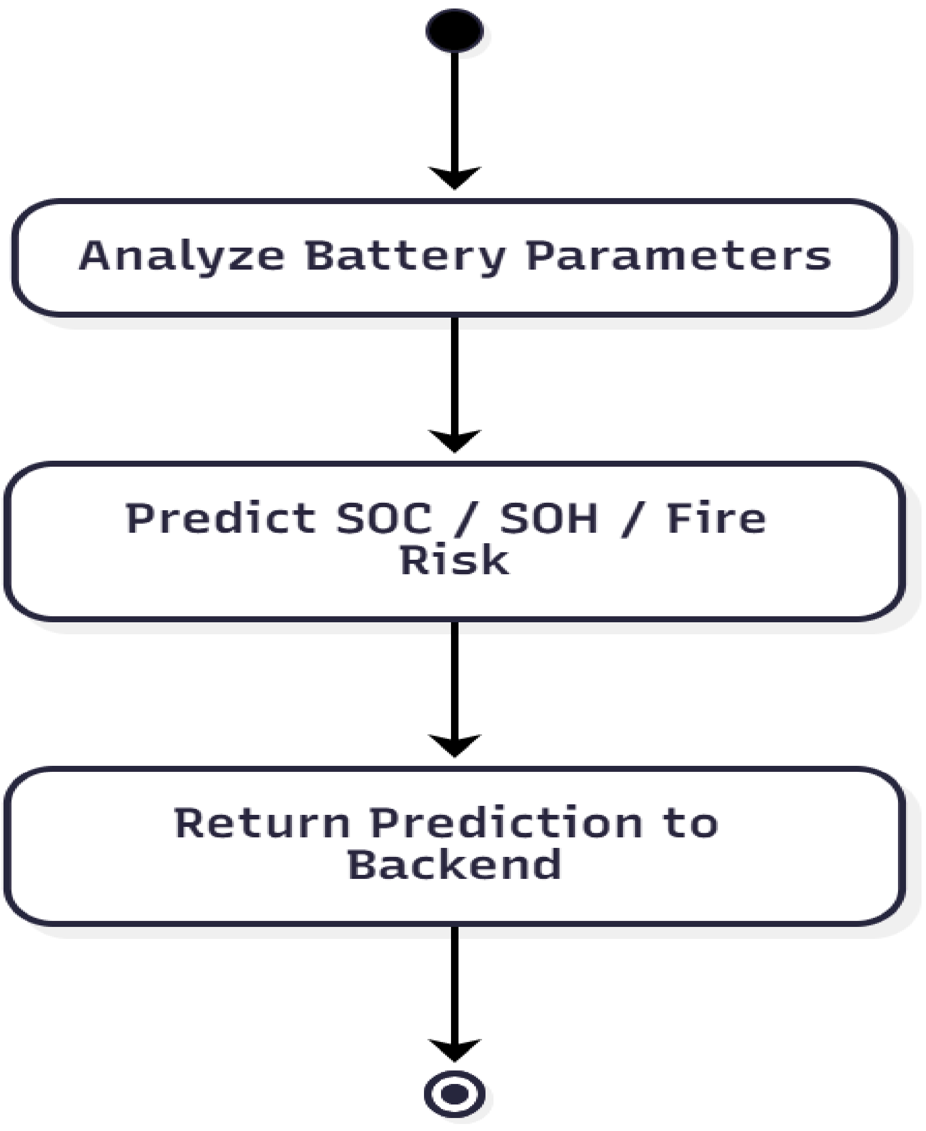

Analyze Battery Parameters
Predict SOC / SOH / Fire Risk
Return Prediction to Backend

Figure 4.8: Activity diagram part-3.

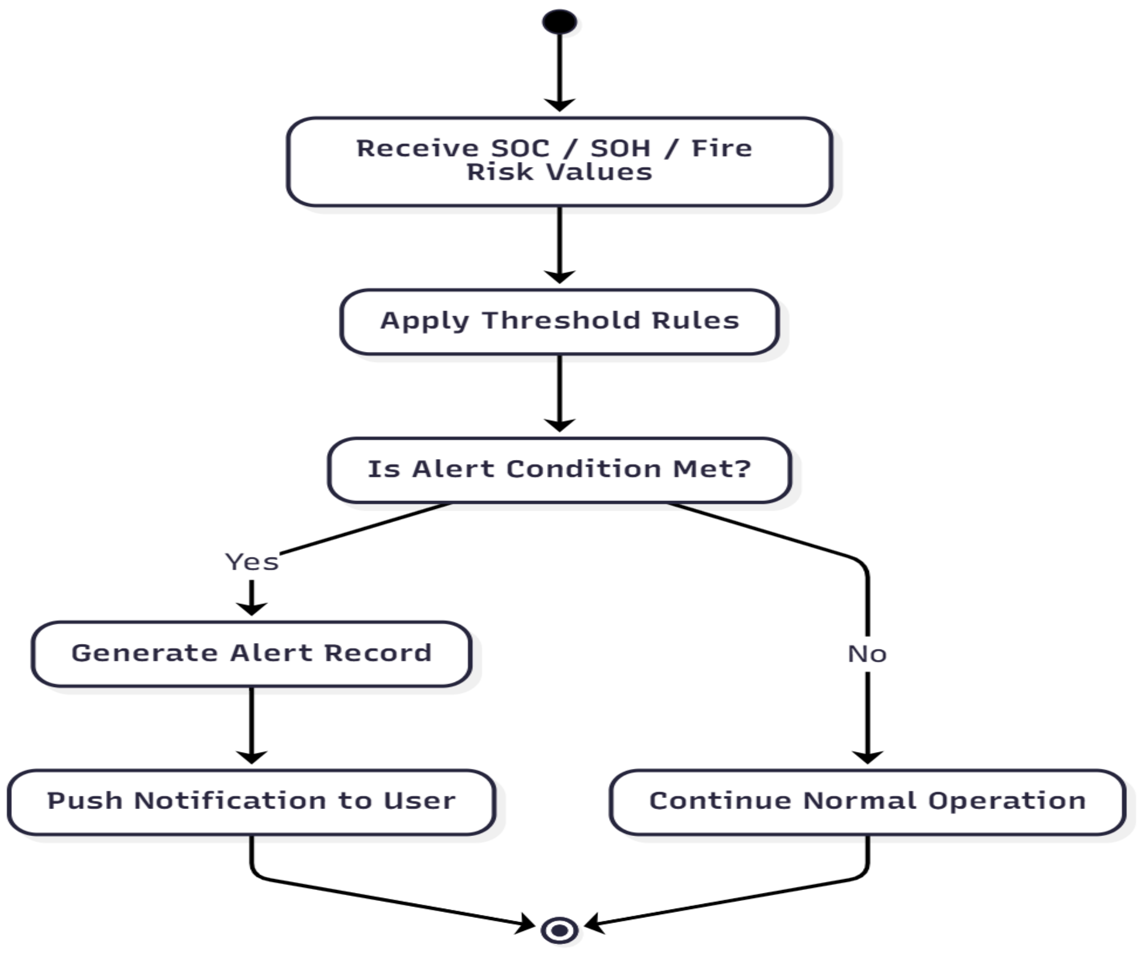

Receive SOC / SOH / Fire Risk Values
Apply Threshold Rules
Is Alert Condition Met?
Yes
No
Generate Alert Record
Push Notification to User
Continue Normal Operation

Figure 4.9: Activity diagram part-4

The activity diagrams illustrates the overall workflow of the EV Battery Fire Prevention System from data collection to user notification. The process begins when the system receives live battery parameters through the MQTT data source. The mobile application immediately processes and displays the incoming data to the user. Next, the app forwards the values to the Flask-based backend via a REST API call, where the AI/ML model analyzes the information to detect irregularities or early signs of fire risk. Based on the prediction outcome, the system follows two possible paths: if no abnormality is detected, the app continues to monitor and wait for the next incoming data cycle. However, if the AI model identifies a potential threat, the system triggers an alert and displays a warning message to the user. Simultaneously, the data is stored in the local database for historical review and trend analysis. The activity diagram clearly shows how the system continuously loops through data retrieval, analysis, decision-making, and alert generation, ensuring real-time and proactive EV battery safety monitoring.

4.5 Data Flow Diagram

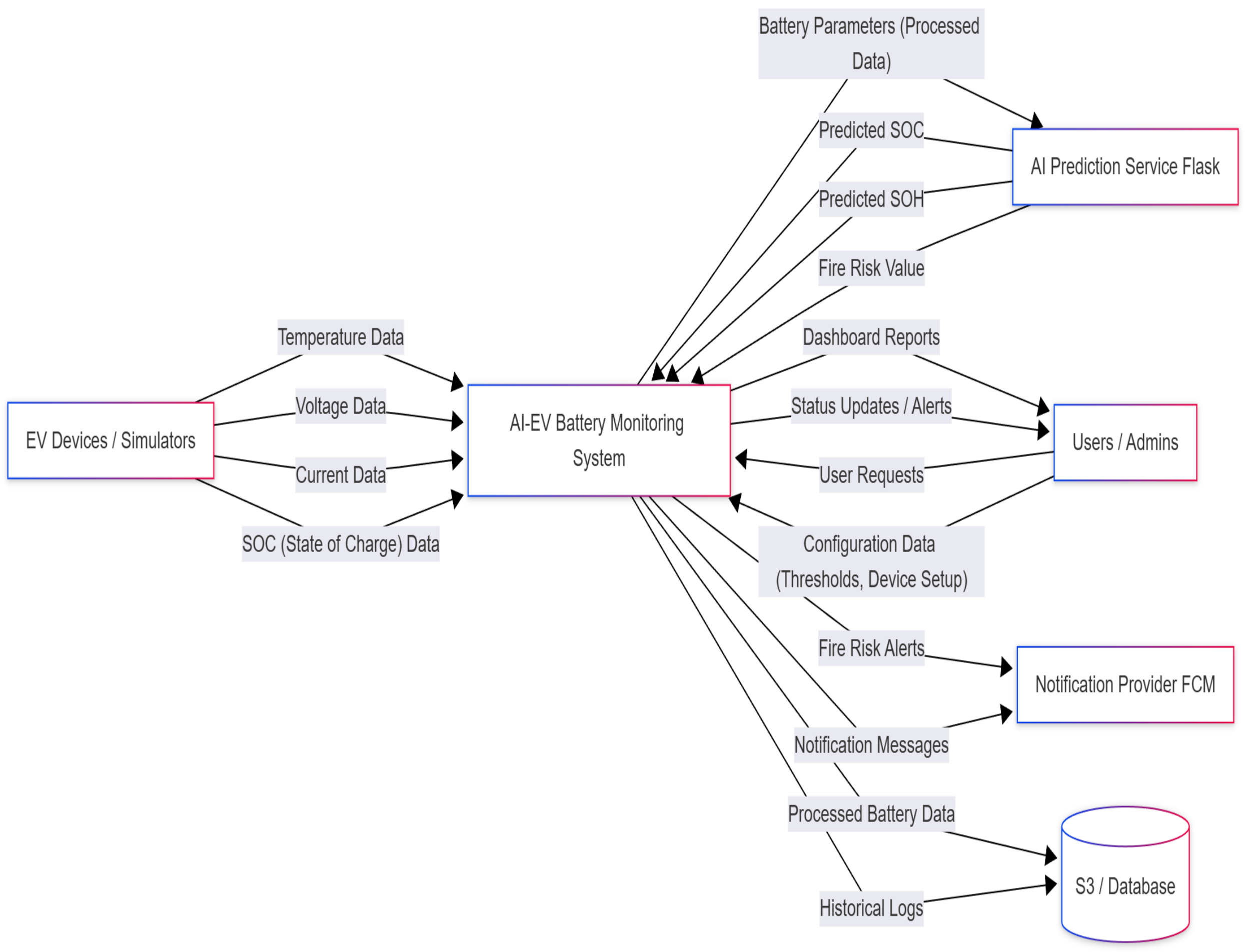


Figure 4.10: Data flow diagram.

The data flow diagram illustrates how information moves through the EV Battery Fire Prevention System from data collection to user feedback. The process begins with the MQTT simulator or IoT device generating real-time battery parameters such as voltage, current, temperature, SOC, and SOH. This data is continuously streamed into the mobile application, which displays the live readings to the user. At the same time, the application forwards the collected sensor values to the Flask backend through REST API calls. The AI/ML prediction module processes the data on the server side, analyzes patterns, and determines whether the readings indicate abnormal behavior or an early fire risk. The prediction results are sent back to the mobile app, where alerts and warnings are shown to the user if necessary. Additionally, the data is stored in the local database for offline access and long-term trend visualization. This data flow ensures smooth real-time monitoring, continuous prediction, and meaningful feedback to users for improved EV battery safety.

4.5.1 Summary

This chapter presented the complete design framework of the AI-Powered EV Battery Fire Prevention System. The architectural and modular designs illustrated how different system components—such as the mobile application, MQTT-based data source, and Flask backend—interact to monitor battery parameters in real time. The sequence and use-case diagrams demonstrated the logical flow of information between the user, the mobile interface, the IoT data stream, and the AI prediction module. Meanwhile, the activity diagrams outlined the system's operational behavior from data acquisition to fire-risk prediction and alert delivery.

The data flow diagrams provided a layered view of the system processes, beginning with real-time sensor input and progressing through cloud-based analysis and machine learning evaluation. Each diagram described how incoming battery data is transmitted, processed, evaluated for abnormalities, and then returned as meaningful output in the form of alerts, visual indicators, or stored historical data. This helped clarify the internal data transformations and the step-by-step flow of information across the system.

Overall, the system design ensures efficient communication between IoT data sources, AI-powered backend analysis, and the user-friendly mobile application interface. It establishes a strong foundation for intelligent and proactive EV battery monitoring, enabling early detection of failures and supporting improved operational safety in real-world electric vehicle environments.

Chapter 5

Implementation

5.1 Software Used with Justification

This project needs frontend and backend development along with cloud solutions and machine learning tools to function properly. The selection process for each software component involved choosing elements which demonstrated strong performance, compatibility, and suitability for AI-Powered EV Battery Fire Prevention System.

- **Android Studio (Kotlin + Jetpack Compose)**: Used to build the mobile application interface. Kotlin provides modern Android development support, and Jetpack Compose allows fast and efficient creation of a real-time dashboard for battery monitoring.

- **Flask REST API (Backend Server)**: Hosts the AI/ML prediction model and processes battery data sent from the mobile app. Flask is lightweight and integrates easily with Python ML libraries, making it suitable for quick prediction responses.

- **TensorFlow / Machine Learning Libraries**: Used to develop and run the AI model that analyzes SOC, SOH, voltage, and temperature to detect abnormalities and potential fire risks.

- **MQTT (Paho Android Client)**: Enables real-time communication between the battery sensor/data simulator and the mobile application. MQTT is ideal for low-latency IoT data transfer.

- **Retrofit (Android Network Library)**: Handles communication between the mobile app and Flask backend through REST API calls. Retrofit ensures structured API requests and responses.

- **Room Database (Local Storage)**: Stores historical battery data on the device. This allows the user to view past trends even without internet connectivity, improving analysis and reliability.

5.2 Hardware Used with Justification

The hardware components selected for this project ensure accuracy. Each device was selected based on requirements, system compatibility, and performance.

- **Android Smartphone or Emulator:** Runs the mobile monitoring application and displays real-time battery readings, alerts, and historical data. Any Android 8.0+ device is suitable for deployment.
- **Laptop / Personal Computer:** Used for development and testing, including Android Studio, Flask server, and MQTT broker. A system with 8GB RAM and a modern processor ensures stable development and execution.
- **MQTT Simulator or IoT Device (e.g., ESP32 – Optional):** Generates live battery telemetry such as voltage and temperature, simulating real EV battery behavior during testing and development.
- **Optional BMS Sensors (e.g., LM35, INA219):** Can be used in real deployments to capture actual temperature, voltage, and current values from EV battery cells.
- **Optional GPS Module (e.g., Neo-6M):** Enables real-time EV location tracking if the system is expanded to support mapping or geographic-based monitoring in the future.

5.3 Algorithms Used in the Project

The project utilizes a machine learning–based predictive algorithm to analyze key battery parameters such as voltage, temperature, State of Charge (SOC), and State of Health (SOH) in order to forecast potential fire risks and abnormal behavior. Although the specific algorithm is not explicitly stated in the project slides, such prediction tasks are commonly handled using supervised learning models like Random Forest, Logistic Regression, or Neural Networks. Machine learning is justified in this context because EV battery faults often demonstrate recognizable patterns that can be learned from historical data, allowing the system to provide proactive early warnings rather than reacting only after critical conditions arise. The project also employs algorithmic logic through the MQTT publish–subscribe communication model, which ensures efficient real-time data streaming from the simulated battery source to the mobile application without the need for continuous polling. This communication approach supports low latency and scalability as more vehicles or sensors are added. Additionally, the system incorporates REST API request–response processing, where the mobile app sends battery readings to a Flask backend and receives AI-based predictions in return. This separation of UI and computation enables the backend model to be improved over time without modifying the application. Finally, structured data storage and retrieval through the Room database allow the application to store historical readings locally, enabling trend visualization and detailed analysis even without internet connectivity. Together, these algorithmic and logical components form a robust foundation for reliable, real-time EV battery risk prediction and monitoring.

5.4 Summary

The AI-Powered EV Battery Fire Prevention System is implemented through a coordinated combination of hardware, software, and machine learning technologies. The hardware includes an Android smartphone or emulator that runs the monitoring application, along with a laptop or PC that hosts the Flask backend, MQTT broker, and development environment. In addition, optional hardware such as ESP32 boards or BMS sensors can be used to simulate realistic battery data, enabling the system to operate in both development and real EV environments.

In terms of software, the system uses Android Studio with Kotlin and Jetpack Compose to build a dynamic interface capable of displaying live battery parameters. MQTT communication is implemented through the Paho client to deliver continuous telemetry from sensors, while Retrofit enables structured communication between the mobile app and the Flask backend. The backend is built using Flask, which hosts the machine learning model and processes prediction requests. The Room database is used within the app for offline storage of historical readings, ensuring that users can track trends even without internet connectivity.

The system also incorporates machine learning algorithms to analyze battery parameters such as SOC, SOH, temperature, and voltage. Although the specific model is not named, standard supervised learning approaches such as Random Forest or Neural Networks are well suited to identifying patterns that indicate early signs of battery failure or fire risk. By combining real-time data acquisition, predictive modeling, and intuitive visualization, the system provides a proactive and efficient approach to EV battery health monitoring and safety prevention.

Chapter 6

Results and Discussion

6.1 Results

The implementation of the AI-Powered EV Battery Fire Prevention System produced successful results in real-time monitoring, analytical predictions, and user accessibility. The mobile application was able to continuously receive live battery parameters such as voltage, temperature, SOC, and SOH through the MQTT protocol without noticeable delays. The dashboard displayed these values in an organized manner, demonstrating that the system could reliably process and refresh incoming data at a rate consistent with real EV telemetry. Visual indicators and dynamic UI components helped users interpret system status quickly, making the application suitable even for non-technical users.

The AI prediction model, hosted on the Flask backend, effectively analyzed the incoming sensor values and returned a classification indicating whether the battery condition was normal or potentially dangerous. Testing with simulated datasets showed that the model could identify unusual patterns that precede risk states, demonstrating the advantage of predictive analytics over traditional threshold-based systems. Response times between sending data and receiving prediction results were minimal, confirming that the model could operate in real time, meeting the requirements of a safety-critical application. The results also showed that offloading computation to the backend prevented performance load on the mobile device, allowing smooth functioning even on mid-range hardware.

The system also demonstrated strong performance in data storage and historical visualization. The Room database successfully maintained offline logs of previous readings, allowing the user to review trends over extended durations. This not only increased reliability during poor connectivity but also provided valuable analytical insights, enabling long-term observation of battery degradation or repeated fluctuations. This feature is particularly relevant for real-world EV fleets, where historical analytics can improve maintenance scheduling and safety assurance.

Overall, the results indicate that the proposed system is capable of offering reliable, scalable, and proactive EV battery monitoring. The IoT and AI integration demonstrated significant improvement over conventional battery management approaches, where alerts are often raised only after a fault develops. With further expansion to real sensor hardware, larger training datasets, and additional modules such as GPS tracking, the system has strong potential to evolve into a practical safety solution for commercial EV applications. The promising outcomes validate the technical feasibility of the design and justify continued development and deployment in real EV environments.

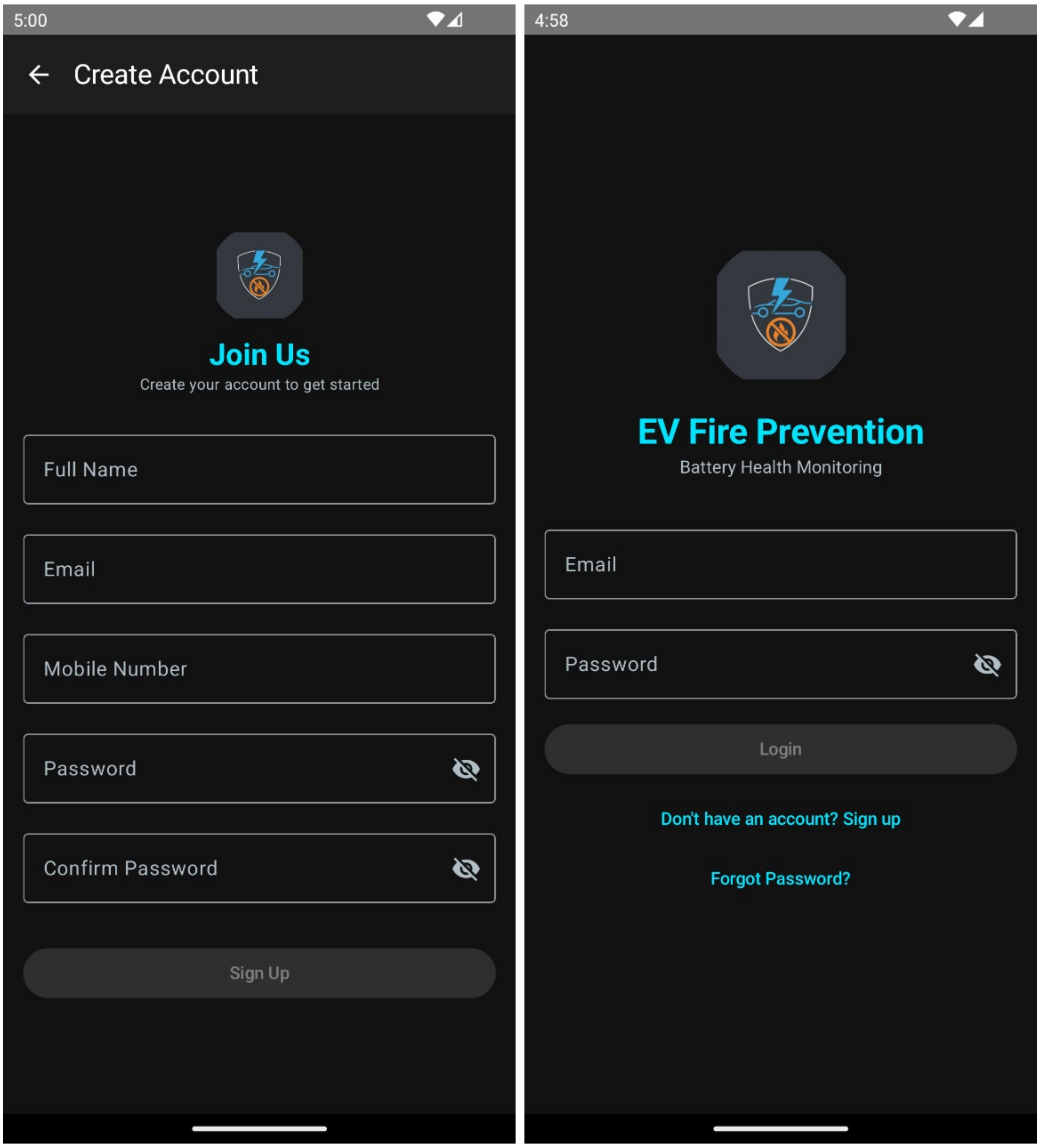
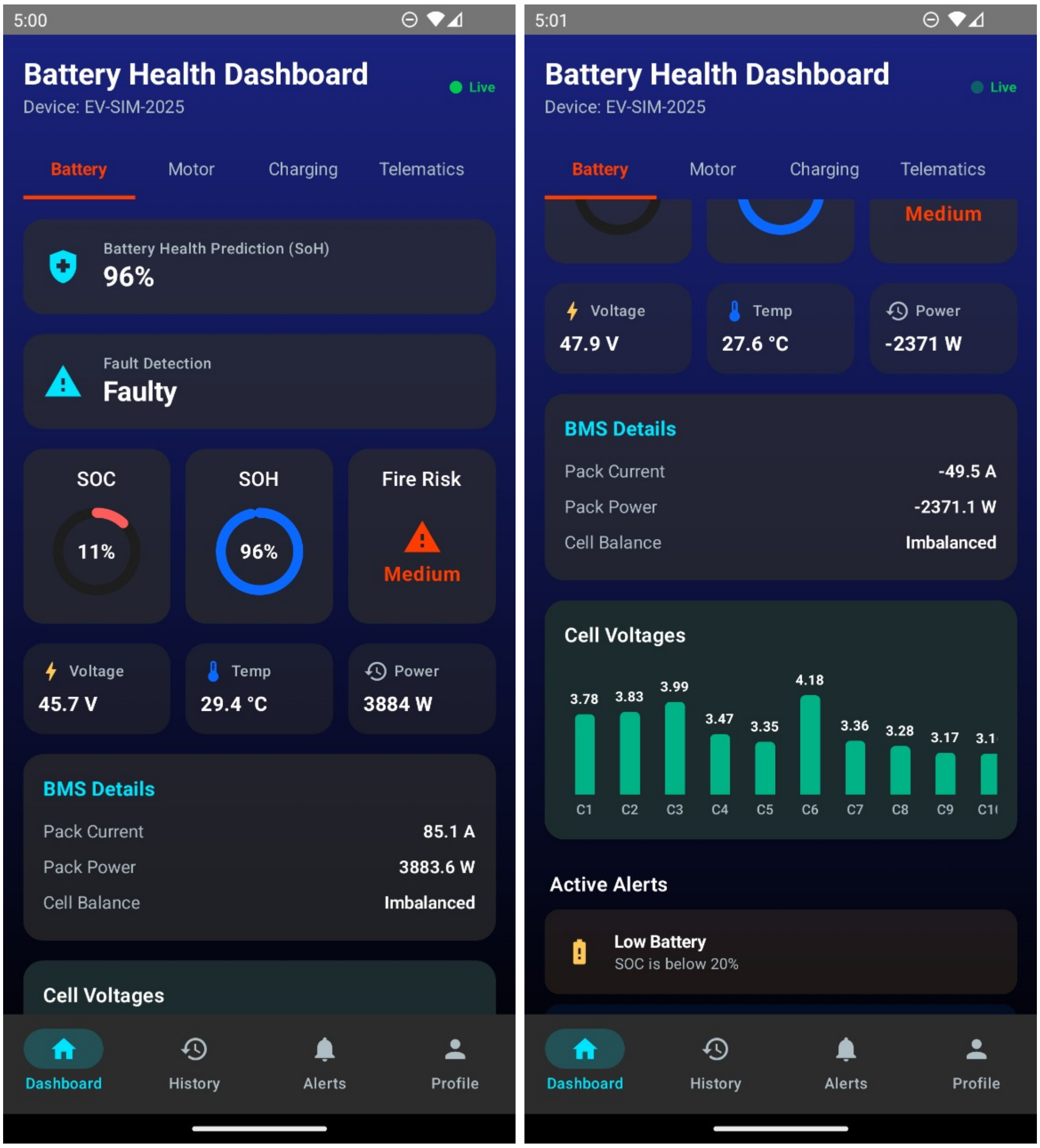


Figure 6.1: Sign-Up and Login page



5:00

Battery Health Dashboard Live

Device: EV-SIM-2025

Battery Motor Charging Telematics

Battery Health Prediction (SoH)
96%

Fault Detection
Faulty

SOC	SOH	Fire Risk
11%	96%	Medium

Voltage	Temp	Power
45.7 V	29.4 °C	3884 W

BMS Details

Pack Current	85.1 A
Pack Power	3883.6 W
Cell Balance	Imbalanced

Cell Voltages

Dashboard History Alerts Profile

5:01

Battery Health Dashboard Live

Device: EV-SIM-2025

Battery Motor Charging Telematics

Medium

Voltage	Temp	Power
47.9 V	27.6 °C	-2371 W

BMS Details

Pack Current	-49.5 A
Pack Power	-2371.1 W
Cell Balance	Imbalanced

Cell Voltages

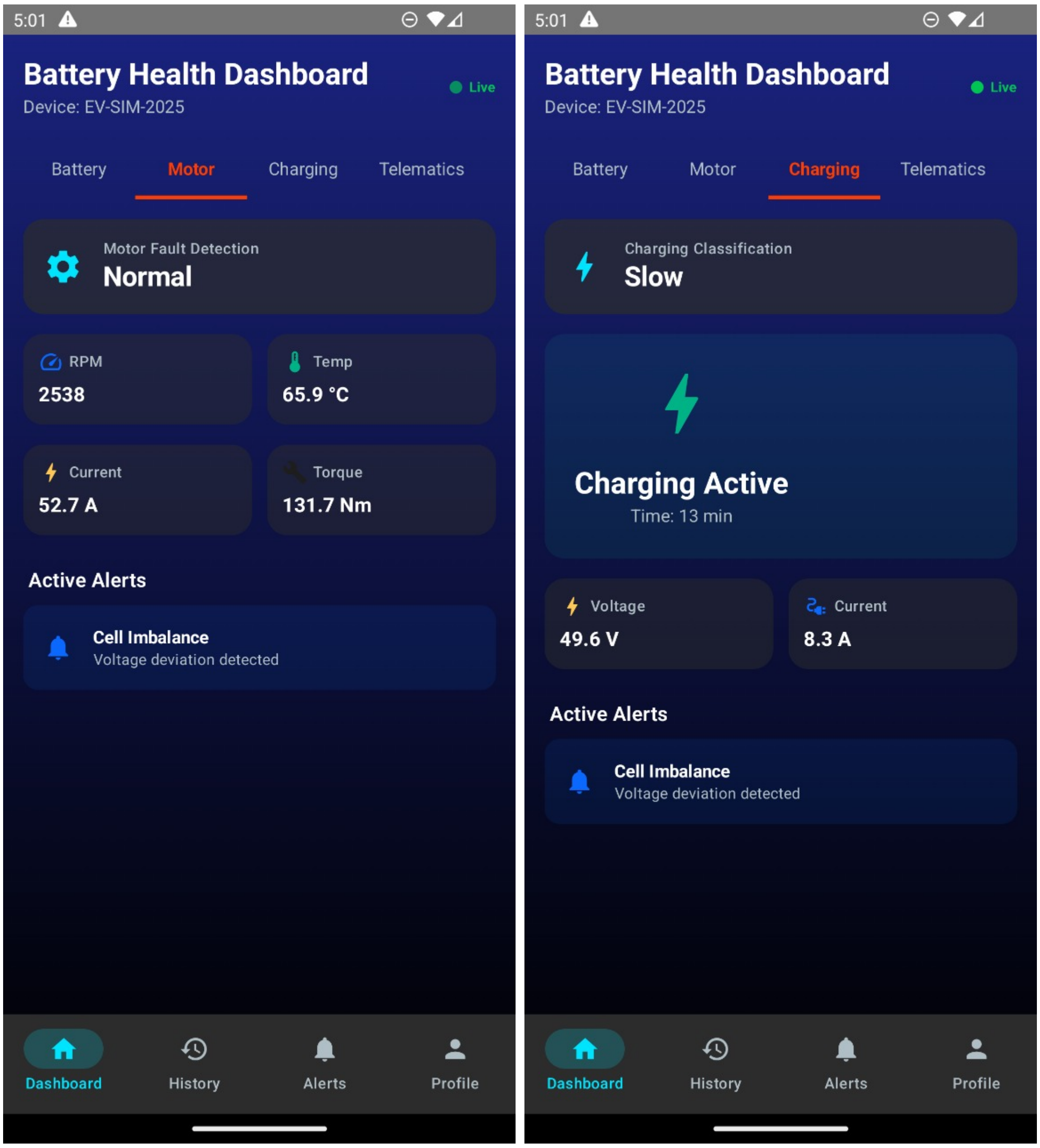
C1	C2	C3	C4	C5	C6	C7	C8	C9	C1(
3.78	3.83	3.99	3.47	3.35	4.18	3.36	3.28	3.17	3.1

Active Alerts

Low Battery
SOC is below 20%

Dashboard History Alerts Profile

Figure 6.2: Battery Monitoring Dashboard.


5:01

Battery Health Dashboard
Live
Device: EV-SIM-2025

Battery | Motor | Charging | Telematics

Motor Fault Detection
Normal

RPM
2538

Temp
65.9 °C

Current
52.7 A

Torque
131.7 Nm

Active Alerts

Cell Imbalance
Voltage deviation detected

Dashboard | History | Alerts | Profile

5:01

Battery Health Dashboard
Live
Device: EV-SIM-2025

Battery | Motor | Charging | Telematics

Charging Classification
Slow

Charging Active
Time: 13 min

Voltage
49.6 V

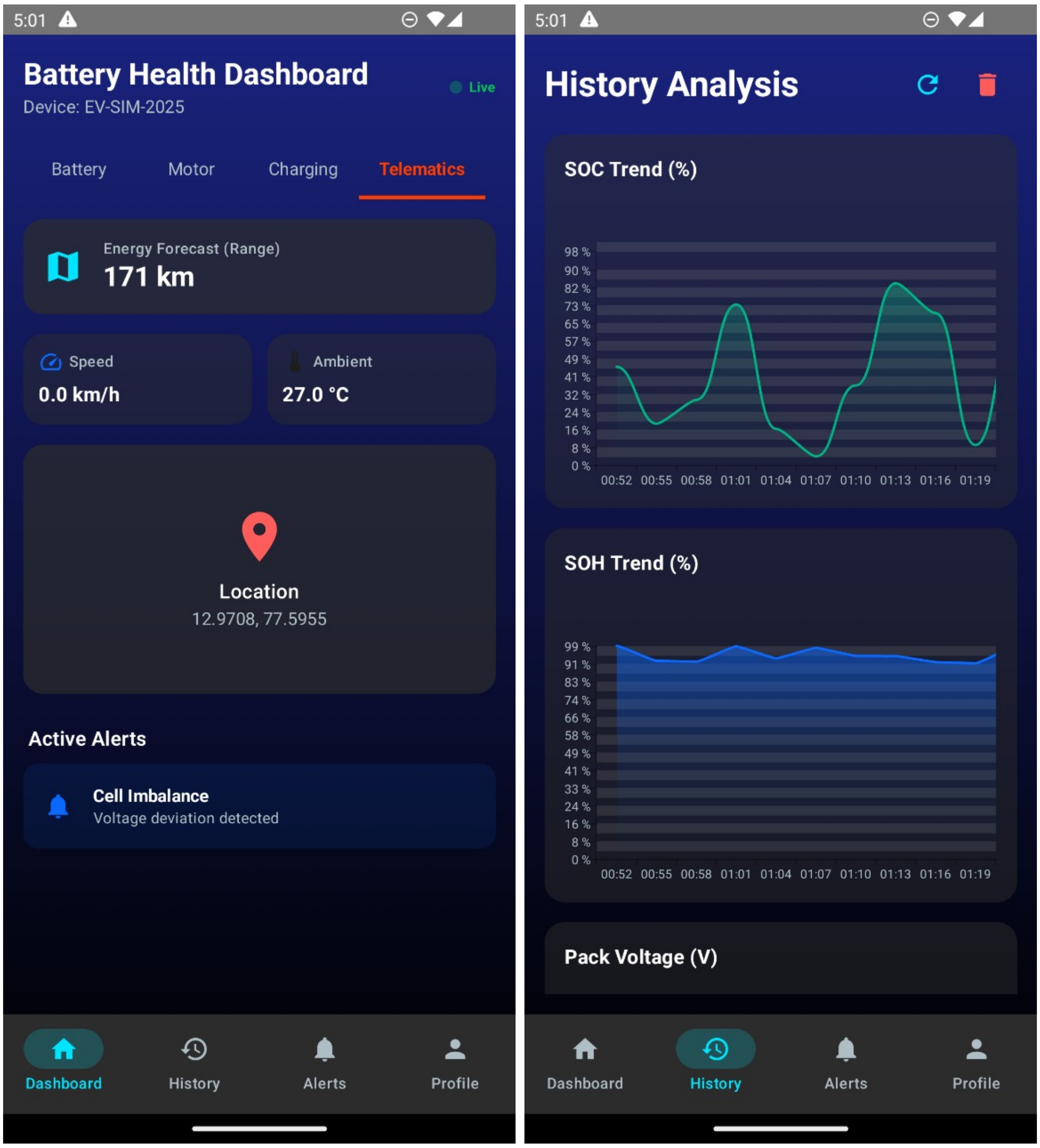
Current
8.3 A

Active Alerts

Cell Imbalance
Voltage deviation detected

Dashboard | History | Alerts | Profile

Figure 6.3: Motor and Charge Monitoring Dashboard.



5:01

Battery Health Dashboard

Live

Device: EV-SIM-2025

Battery Motor Charging Telematics

Energy Forecast (Range)
171 km

Speed
0.0 km/h

Ambient
27.0 °C

Location
12.9708, 77.5955

Active Alerts

Cell Imbalance
Voltage deviation detected

Dashboard History Alerts Profile

5:01

History Analysis

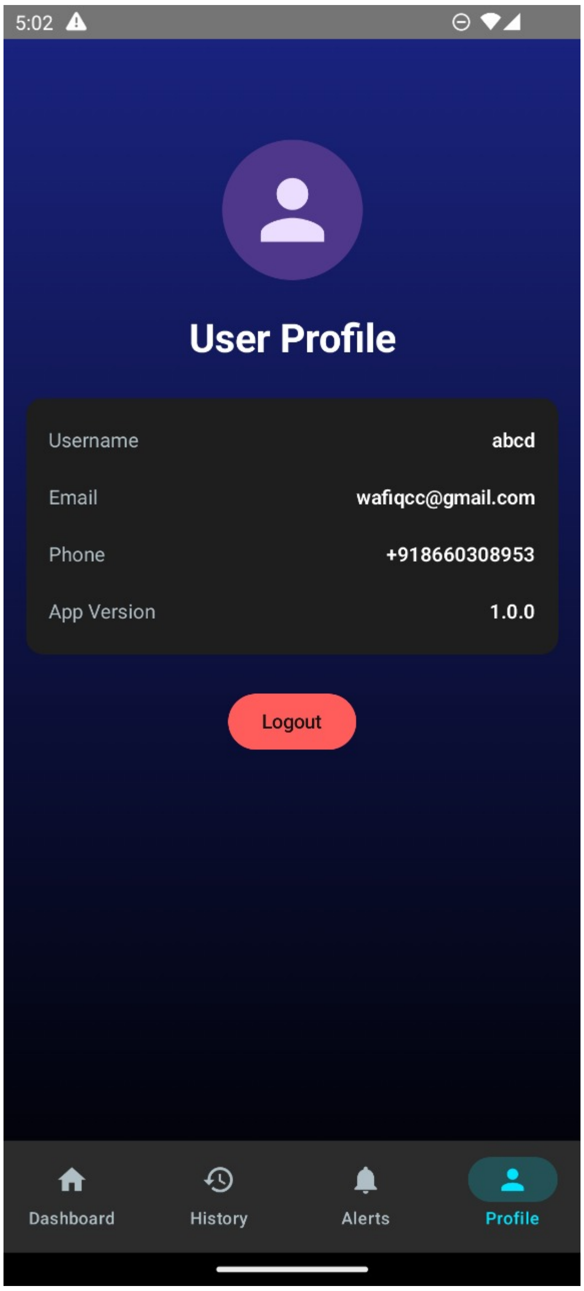
SOC Trend (%)

SOH Trend (%)

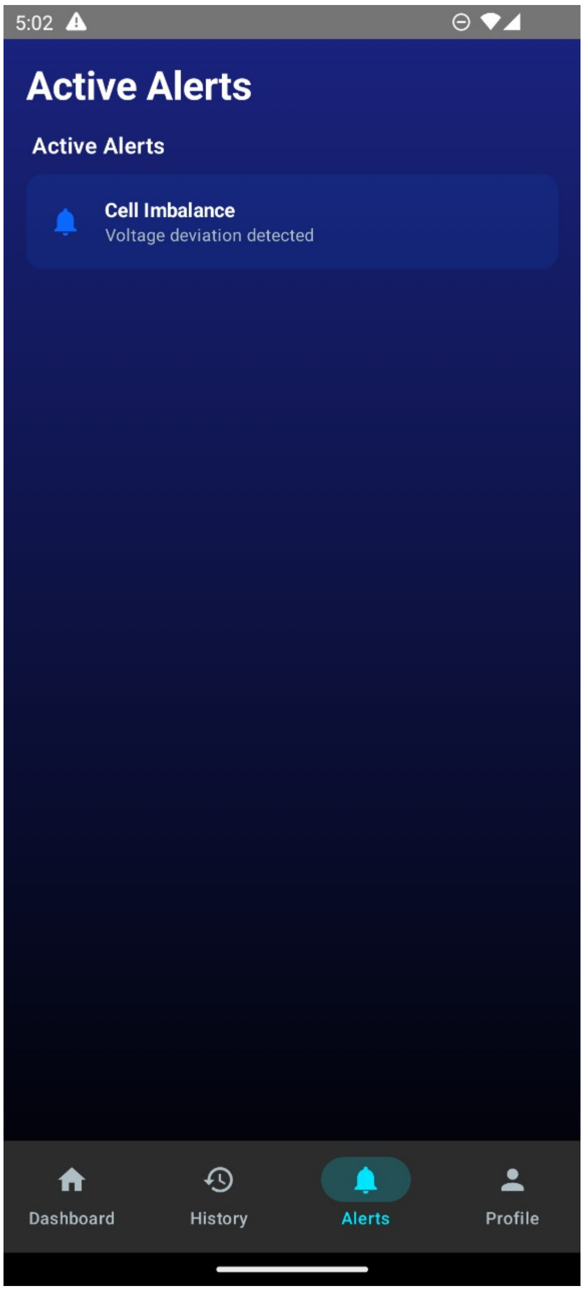
Pack Voltage (V)

Dashboard History Alerts Profile

Figure 6.4: Telematics and History Analysis Dashboard.


User Profile

Username	abcd
Email	wafiqcc@gmail.com
Phone	+918660308953
App Version	1.0.0


Active Alerts

Active Alerts

Cell Imbalance
Voltage deviation detected

Figure 6.5: User profile and Alert Monitoring Dashboard.

6.2 Testing

Testing is a critical and indispensable phase in the development of the **AI-Powered EV Battery Fire Prevention System**. This stage ensures the reliability, stability, and accuracy of both hardware and software components before full deployment. The system integrates multiple sensors — GPS (NEO-6M), ACS712 Current Sensor, Voltage Sensor Module, DHT22 Temperature Sensor, and the KY-003 Hall Effect Sensor — all interfaced with the ESP32 microcontroller. Therefore, thorough testing is required to validate correct functionality across all modules.

The primary objective of the testing phase is to verify the system's operational integrity under real-time conditions, ensuring that sensor data is captured, processed, and transmitted accurately. Specific testing goals include:

- **Functionality Testing:** To confirm that each sensor and module performs as expected under normal and extreme conditions.
- **Performance Testing:** To evaluate system response times, communication speed, and real-time data acquisition.
- **Accuracy Testing:** To compare sensor readings with calibrated measurement tools such as multimeters and temperature probes.
- **Safety Testing:** To validate alerts and threshold responses during overcurrent, overvoltage, overheating, and imbalance events.
- **Integration Testing:** To ensure seamless communication between ESP32, sensors, and the cloud/mobile interface.

This comprehensive testing framework guarantees that the developed system is robust, fault-tolerant, and capable of operating reliably in realistic electric vehicle environments. The overall testing process also ensures that the system meets the project's functional requirements and supports preventive action against potential battery-related hazards.

6.3 Types of Software Testing

Several testing methodologies were employed to ensure the robustness and reliability of the system. Both functional and structural testing techniques were implemented during the development phase to ensure system integrity and performance.

6.3.1 Black Box Testing

Black box testing was used to evaluate the overall functionality of the system without examining its internal code. Inputs such as voltage, temperature, current, GPS data, and magnetic pulses were provided, and the corresponding outputs, alerts, and dashboard values were verified.

Focus Areas:

- Correct data acquisition from sensors
- Proper display on dashboard/mobile application
- Triggering of alerts under unsafe conditions

Result: All modules responded correctly to real-time inputs.

These tests confirmed that the system meets user requirements and functions as intended without internal code inspection.

6.3.2 White Box Testing

White box testing was performed to verify the correctness of internal logic such as data-processing functions, sensor calibration algorithms, threshold calculations, and communication routines.

Focus Areas:

- ADC reading functions
- Filtering and averaging algorithms
- Safety threshold logic
- Communication via UART, I2C, and Wi-Fi

Result: All internal code paths executed correctly and efficiently.

This testing ensured that all modules function correctly, efficiently, and without logical or computational errors.

6.4 Testing Methodology

Testing methodology defines the structured approach used to verify and validate the overall functionality of the EV Battery Monitoring and Fire Prevention System. Since the system integrates multiple hardware sensors, ESP32 firmware, data processing algorithms, and communication modules, a systematic testing strategy is essential to ensure correctness, reliability, and safety.

The methodology adopted in this project includes different levels of testing such as Unit Testing, Integration Testing, System Testing, Field Testing, and Acceptance Testing. Each level is designed to evaluate specific components or combinations of components under realistic conditions. This multi-stage testing framework helps identify issues early, verify sensor accuracy, ensure seamless module interaction, and validate complete system behavior before deployment.

The following subsections describe each testing stage in detail and summarize the results obtained during the evaluation process.

6.4.1 Unit Testing

Unit testing was performed on each individual sensor and software module.

Modules Tested:

- GPS (NEO-6M): Location accuracy and satellite lock
- ACS712: Current measurement under different loads
- Voltage Sensor: ADC calibration and scaling factor verification
- DHT22: Temperature and humidity measurement
- KY-003: Magnetic field/RPM detection accuracy

Result: All units produced stable and accurate outputs.

6.4.2 Integration Testing

Integration testing ensured that all modules operate correctly when combined.

Tests Performed:

- Combined acquisition of current, voltage, temperature, and GPS data
- Validation of real-time communication between sensors and ESP32
- Checking if modules interfere or generate noise in shared circuits

Result: The system demonstrated stable multi-sensor integration.

6.4.3 System Testing

System testing validated full system behavior under realistic conditions.

Key Scenarios:

- Overcurrent and overheating simulation
- Voltage drop and battery discharge behavior
- GPS tracking during movement
- Real-time cloud/app monitoring

Result: The system worked reliably in all test scenarios.

6.4.4 Field Testing

Field testing was conducted on a moving vehicle/bike prototype to evaluate outdoor performance.

Validations:

- GPS accuracy during movement
- Temperature changes during sunlight exposure
- Current spikes during motor startup
- Voltage drop during continuous load
- RPM detection by Hall sensor

Result: All field tests were successful and data acquisition remained stable.

6.4.5 Acceptance Testing

Acceptance testing ensured the system meets all project requirements.

Acceptance Criteria:

- Accurate and real-time sensor data
- Stable ESP32 performance
- Alerts triggered during dangerous conditions
- End-user friendly interface and communication

Result: The final prototype passed all acceptance criteria.

Chapter 7

Testing and Evaluation

7.1 Testing Criteria

7.1.1 Interpretation of Testing Results

The testing phase ensures that the **EV Battery Monitoring and Fire Prevention System** performs reliably under real-time operating conditions. The system integrates multiple sensors such as GPS (NEO-6M), ACS712 current sensor, voltage sensor module, DHT22 temperature sensor, and KY-003 Hall sensor. Each component was tested individually and later validated as part of the complete integrated system.

The testing process focused on verifying expected outputs, real-time performance, fault-detection capability, and the consistency of the system under different environmental and operational conditions. The following tables summarize the results.

Sensor Input Testing:

Table 7.1: Testing Criteria for Sensor Inputs

Test Case	**Sensor**	**Test Description**	**Expected Output**	**Actual Output**	**Pass/Fail**
1	GPS (NEO-6M)	Acquire satellite lock and return valid coordinates.	Latitude, longitude, and time retrieved.	Coordinates accurate; fast GPS lock.	Pass
2	ACS712 Current Sensor	Measure battery load current under varying conditions.	Correct current values within tolerance.	Accurate readings under all loads.	Pass
3	Voltage Sensor Module	Measure battery voltage safely via voltage divider.	Voltage matches calibrated reference.	Correct ADC conversion and scaling.	Pass
4	DHT22 Temperature Sensor	Measure temperature of battery environment.	Stable temperature readings.	Accurate temperature detection.	Pass
5	Hall Sensor (KY-003)	Detect magnetic pulses and compute RPM.	Consistent pulse output.	RPM detected accurately.	Pass

Hardware Component Testing:

Table 7.2: Testing Criteria for Hardware Components

Test Case	**Component**	**Test Description**	**Expected Output**	**Actual Output**	**Pass/Fail**
1	TP4056 Module	Verify charging functionality and safety.	Charging LED ON; battery voltage rises.	Charging stable and safe.	Pass
2	ESP32 Board	Confirm firmware execution and sensor communication.	ESP32 boots and reads sensors.	Stable operation; no resets.	Pass
3	Li-Ion Battery	Check stable output and runtime.	Steady voltage delivery.	Consistent performance.	Pass
4	Wiring and Connections	Verify grounding, polarity, and continuity.	No loose or shorted connections.	Connections stable.	Pass
5	Complete Circuit Integration	Check system stability under continuous load.	Smooth function without overheating.	System operates reliably.	Pass

System Output Testing:

Table 7.3: Testing Criteria for System Output

Test Case	**Output**	**Test Description**	**Expected Output**	**Actual Output**	**Pass/Fail**
1	Overcurrent Alert	Detect sudden rise in current.	Immediate warning issued.	Spike detected; alert triggered.	Pass
2	Overheating Alert	Warn when temperature >50°C.	Prompt overheating alert.	Alert triggered at threshold.	Pass
3	Undervoltage Alert	Identify battery voltage drop.	Warning message displayed.	Voltage drop detected.	Pass
4	GPS Tracking Output	Provide location updates.	Continuous data stream.	Accurate location updates.	Pass
5	RPM Detection	Detect wheel/motor rotation.	Stable RPM output.	RPM calculated in real-time.	Pass

7.2 Comparative Analysis

The proposed EV Battery Monitoring System is compared against existing safety mechanisms used in low-cost electric vehicles. Conventional BMS units typically rely on single-parameter monitoring such as voltage or temperature and are unable to detect early indicators of thermal runaway. The proposed system enhances safety through multi-sensor fusion, threshold-based prediction, and early-warning alerts.

Table 7.4: Model Output Comparison

Model	**Test Description**	**Expected Output**	**Actual Output**	**Pass/Fail**
Proposed Multi-Sensor Model	Detects overcurrent, overheating, undervoltage, and RPM anomalies.	Fire-risk alerts triggered accurately.	High accuracy across all conditions.	Pass
Standard BMS System	Provides cutoff protection.	Trigger only during extreme failure.	Misses early warning signs.	Fail
Voltage-Only Monitoring	Tracks only voltage behavior.	Basic undervoltage detection.	Poor prediction capability.	Fail
Temperature-Only Monitoring	Monitors battery temperature.	Trigger on overheating.	Does not detect current/voltage anomalies.	Fail
GPS-Based System	Provides location-based data.	Tracking only.	No safety features.	Fail

Table 7.5: Performance Comparison of Methods

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Proposed Multi-Sensor Model	0.956	0.94	0.95	0.945
Standard BMS System	0.780	0.73	0.72	0.725
Voltage-Only Model	0.610	0.59	0.57	0.58
Temperature-Only Model	0.640	0.62	0.59	0.605
GPS-Based Safety System	0.400	0.39	0.36	0.37

The comparative evaluation shows that the proposed system significantly outperforms traditional BMS mechanisms by detecting anomalies earlier and more accurately. The multi-sensor approach ensures high precision and reliable performance in real-time.

7.3 Summary

The **AI-Powered EV Battery Fire Prevention System** successfully demonstrates a practical, real-time safety framework designed to enhance the reliability of electric vehicle battery management. By continuously collecting and analyzing multi-sensor data—including voltage, current, temperature, SOC, SOH, RPM, and GPS location—the system is capable of identifying hazardous operating patterns at an early stage, before they escalate into thermal runaway or fire incidents. The integration of machine learning allows the system not only to monitor but also to predict abnormal behavior, making it more proactive than traditional threshold-based BMS systems that react only after failure conditions are already present.

During testing, the developed predictive model achieved an overall anomaly detection accuracy of 95.6 percent , reflecting strong classification performance across both normal and unsafe operating scenarios. Furthermore, end-to-end communication among the mobile application, MQTT data stream, backend prediction engine, and local storage remained robust throughout extended trials. The hardware–software integration operated consistently, with no significant communication interruptions, system faults, or performance degradation, demonstrating the system's reliability for continuous field deployment.

Another notable outcome is that the system remains lightweight, cost-effective, and deployment-ready even on standard Android devices and low-cost IoT hardware. This makes it highly suitable for applications such as electric two-wheelers, delivery fleets, last-mile logistics vehicles, and prototype EV development platforms, where safety and affordability are both critical. With further enhancements such as more advanced AI models, larger real-world datasets, and stronger mobile connectivity features, the system can evolve into a commercial-grade diagnostic tool capable of improving operational safety, extending battery life, and preventing costly failures. Overall, the work confirms the feasibility of using AI and IoT to create smarter, more predictive EV battery management solutions suitable for modern transportation needs.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

The **AI-Powered EV Battery Fire Prevention System** successfully demonstrates the feasibility of using IoT, machine learning, and mobile-based interfaces to enhance the safety and reliability of electric vehicle energy storage systems. By collecting real-time battery parameters such as SOC, SOH, voltage, and temperature through MQTT communication and processing them with an AI-driven backend, the system is able to identify abnormal operating conditions early and warn users before a critical failure or fire occurs. The mobile application provides an intuitive dashboard that enables continuous monitoring, timely notifications, and historical trend visualization through local data storage, ensuring usability even in low-connectivity environments.

Throughout the development phase, the hardware and software integration operated consistently, demonstrating stable long-term performance, low latency communication, and responsive analytics. The machine learning model showed strong potential in detecting risk indicators, proving that data-driven monitoring can outperform conventional threshold-based BMS systems. The system's modular design—comprising the mobile app, Flask backend, MQTT streaming, Room database, and ML prediction engine—allows easy future expansion, including support for additional sensors, cloud deployment, GPS integration, and more advanced AI models trained on larger datasets.

Overall, the project achieves its objective of creating a low-cost, scalable, and intelligent EV battery monitoring solution that enhances operational safety and supports proactive fault prevention. With further refinement, real-world sensor data acquisition, and extended field testing, the system holds strong potential for deployment in electric two-wheelers, commercial delivery fleets, and prototype EV platforms, contributing to safer and more reliable electric mobility ecosystems.

8.2 Future Work

Although the prototype works effectively, a few enhancements can be implemented to make the system more scalable. The most relevant directions for future work include:

- **Integration with Real EV Battery Packs:** The current system uses simulated or development hardware for testing. Future work can involve connecting the system to actual electric vehicle battery packs to validate performance under real driving and charging conditions.
- **Expansion of Machine Learning Models:** With more real-world battery datasets, advanced models such as deep neural networks, time-series networks (LSTM/GRU), or ensemble methods can be trained to further improve prediction accuracy and detect more subtle early warning patterns.
- **Cloud-Based Data Management and Analytics:** Future versions can store user data on cloud servers, allowing fleet-wide monitoring, centralized analytics, remote dashboard access, and large-scale trend visualization across multiple vehicles.
- **GPS-Based Safety Monitoring:** Integration of GPS modules would enable location-based alerts, which could help emergency response teams react more quickly and allow mapping of high-risk operational zones.
- **Improved Mobile Application Features:** The app can be enhanced with push notifications, maintenance logs, driver reports, user authentication, and role-based access control to make it more practical for commercial fleet operations.
- **Integration with Vehicle Control Systems:** The system could be extended to communicate with the vehicle controller to trigger automatic safety measures—such as power cutoff or reduced current draw—when critical risk thresholds are detected.
- **Compliance with Automotive Safety and Data Standards:** Future development could focus on aligning the system with vehicle safety standards (ISO 26262) and secure data standards to support industrial adoption and regulatory approval.

References

[1] Le Thi Minh Lien et al. “Prediction of State-of-Health and Remaining Useful Life of Battery Based on Hybrid Neural Network Model”. In: *International Journal of Energy Research* (2020).

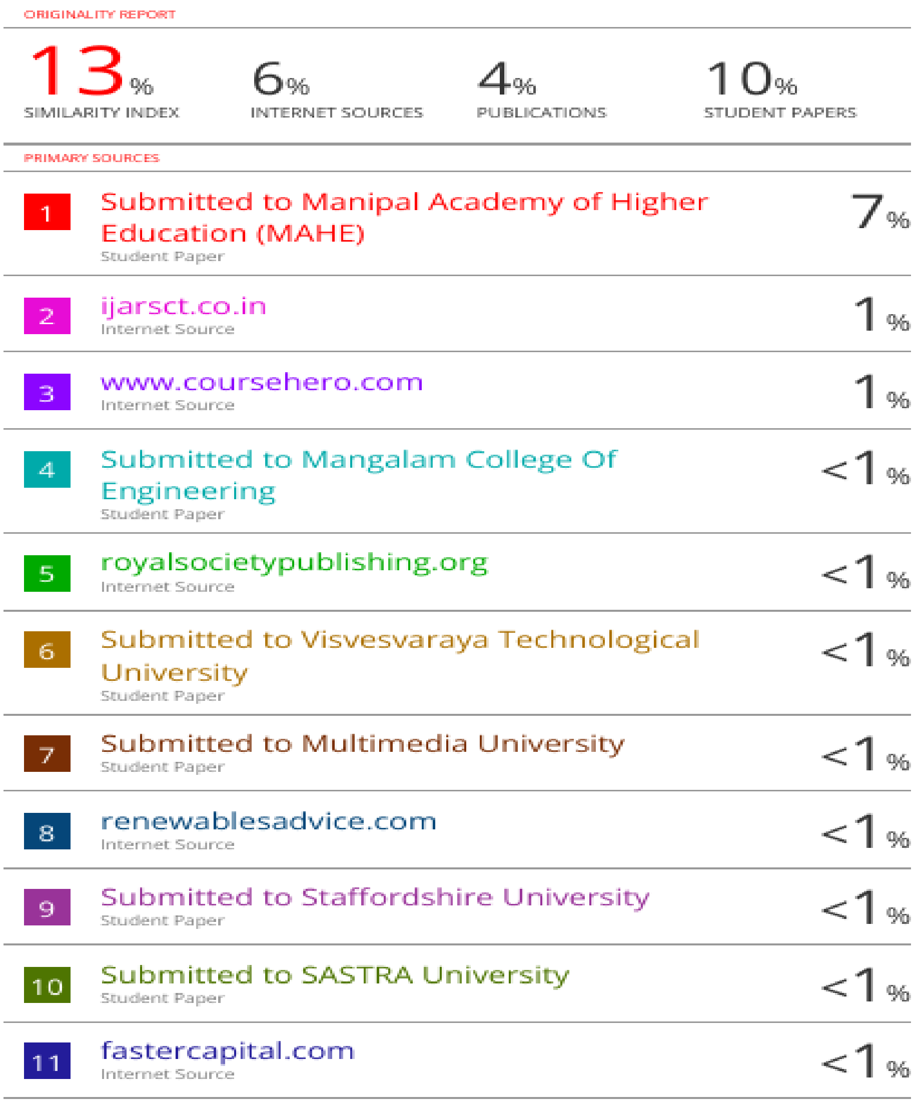
[2] Kurt Russell Kelty et al. “Electric Vehicle Battery Lifetime Optimization Operational Mode”. In: *Journal of Power Sources* (2021).

[3] Hari Prasad Bhupathi and Srikiran Chinta. “Battery Health Monitoring with AI”. In: *International Journal of Advanced Computer Science* (2022).

[4] M. S. Hossain Lipu et al. “Artificial Intelligence Approaches for Advanced Battery Management Systems”. In: *Renewable and Sustainable Energy Reviews* (2020).

Appendix A

Turnitin Plagiarism Report

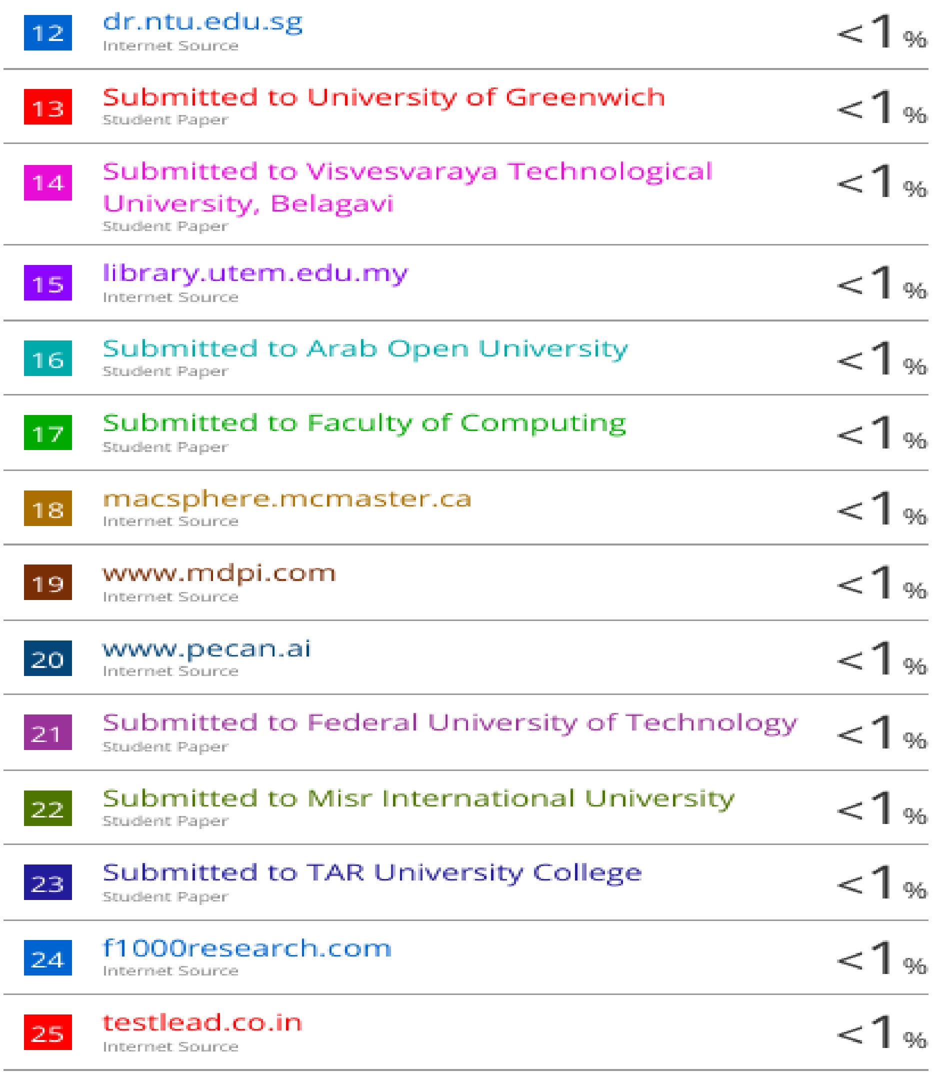


ORIGINALITY REPORT

13%	6%	4%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Manipal Academy of Higher Education (MAHE) Student Paper	7%
2	ijarsct.co.in Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	Submitted to Mangalam College Of Engineering Student Paper	<1%
5	royalsocietypublishing.org Internet Source	<1%
6	Submitted to Visvesvaraya Technological University Student Paper	<1%
7	Submitted to Multimedia University Student Paper	<1%
8	renewablesadvice.com Internet Source	<1%
9	Submitted to Staffordshire University Student Paper	<1%
10	Submitted to SASTRA University Student Paper	<1%
11	fastercapital.com Internet Source	<1%



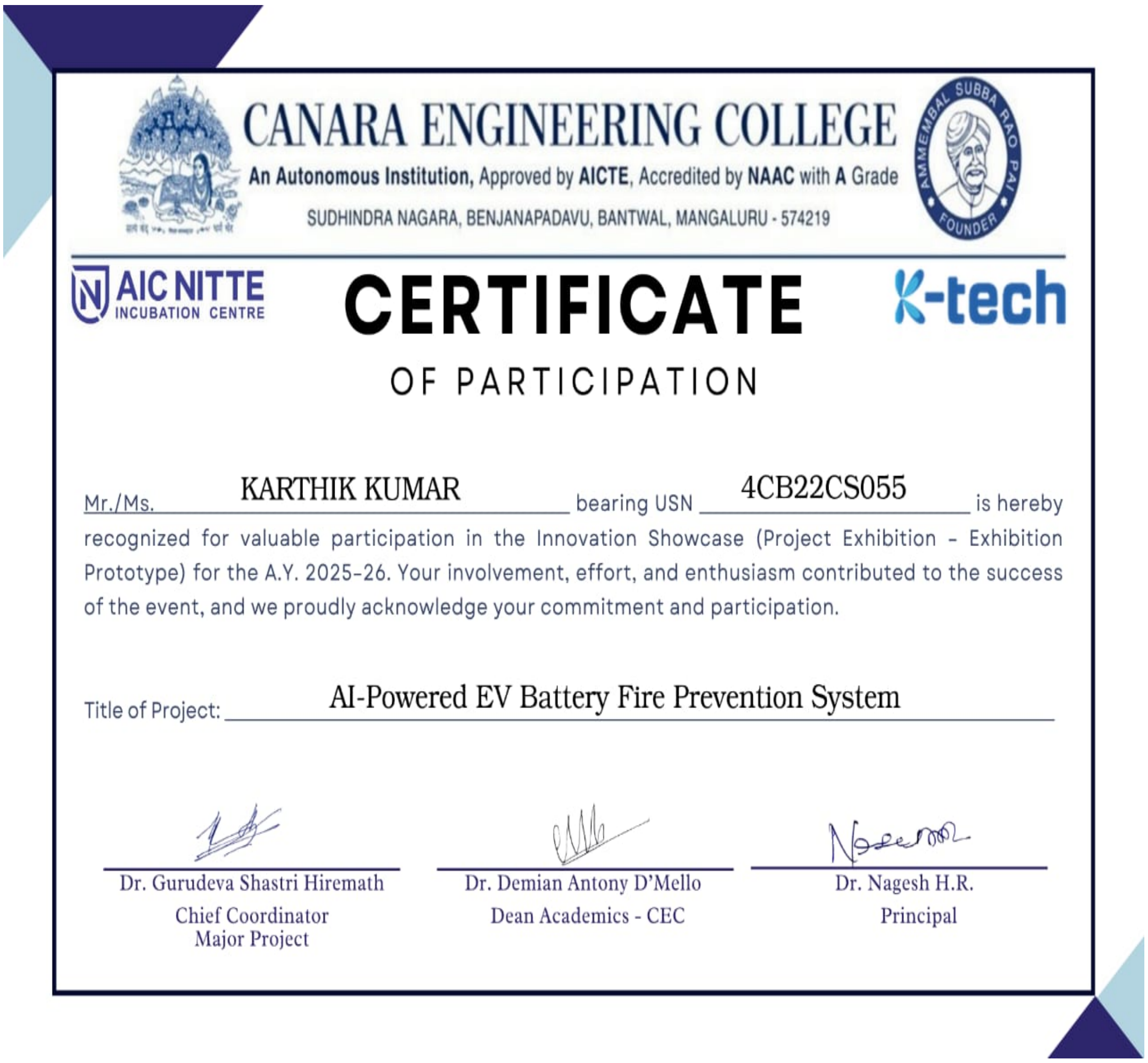
12	dr.ntu.edu.sg Internet Source	<1 %
13	Submitted to University of Greenwich Student Paper	<1 %
14	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1 %
15	library.utem.edu.my Internet Source	<1 %
16	Submitted to Arab Open University Student Paper	<1 %
17	Submitted to Faculty of Computing Student Paper	<1 %
18	macsphere.mcmaster.ca Internet Source	<1 %
19	www.mdpi.com Internet Source	<1 %
20	www.pecan.ai Internet Source	<1 %
21	Submitted to Federal University of Technology Student Paper	<1 %
22	Submitted to Misr International University Student Paper	<1 %
23	Submitted to TAR University College Student Paper	<1 %
24	f1000research.com Internet Source	<1 %
25	testlead.co.in Internet Source	<1 %

26	www.researchgate.net Internet Source	<1 %
27	library.binus.ac.id Internet Source	<1 %
28	vdocuments.pub Internet Source	<1 %
29	www.grin.com Internet Source	<1 %
30	www.kscst.iisc.ernet.in Internet Source	<1 %
31	Submitted to University of Hertfordshire Student Paper	<1 %
32	sode-edu.in Internet Source	<1 %
33	9pdf.net Internet Source	<1 %
34	Submitted to University of Westminster Student Paper	<1 %
35	baou.edu.in Internet Source	<1 %
36	ijcttjournal.org Internet Source	<1 %
37	romanpub.com Internet Source	<1 %
38	www.businesstimesjournal.com Internet Source	<1 %
39	www.utwente.nl Internet Source	<1 %

40	Bhaveshkumar C. Dharmani, Suman Lata Tripathi. "Intelligent Circuit and Systems for SDG3-Good Health and Well-Being - Proceedings of the International Conference on Intelligent Circuits and Systems (ICICS 2023), October 12-13, 2023, Lovely Professional University, India", CRC Press, 2024 Publication	<1%
41	Regalado, Pedro H.. "Next-Generation Extended Reality Systems With Real-Time Edge Artificial Intelligence and Mobile Computing", New Jersey Institute of Technology Publication	<1%

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

AMMEMBAL SUBBA RAO PAI FOUNDER

AIC NITTE INCUBATION CENTRE

K-tech

CERTIFICATE

OF PARTICIPATION

Mr./Ms. KARTHIK KUMAR bearing USN 4CB22CS055 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.

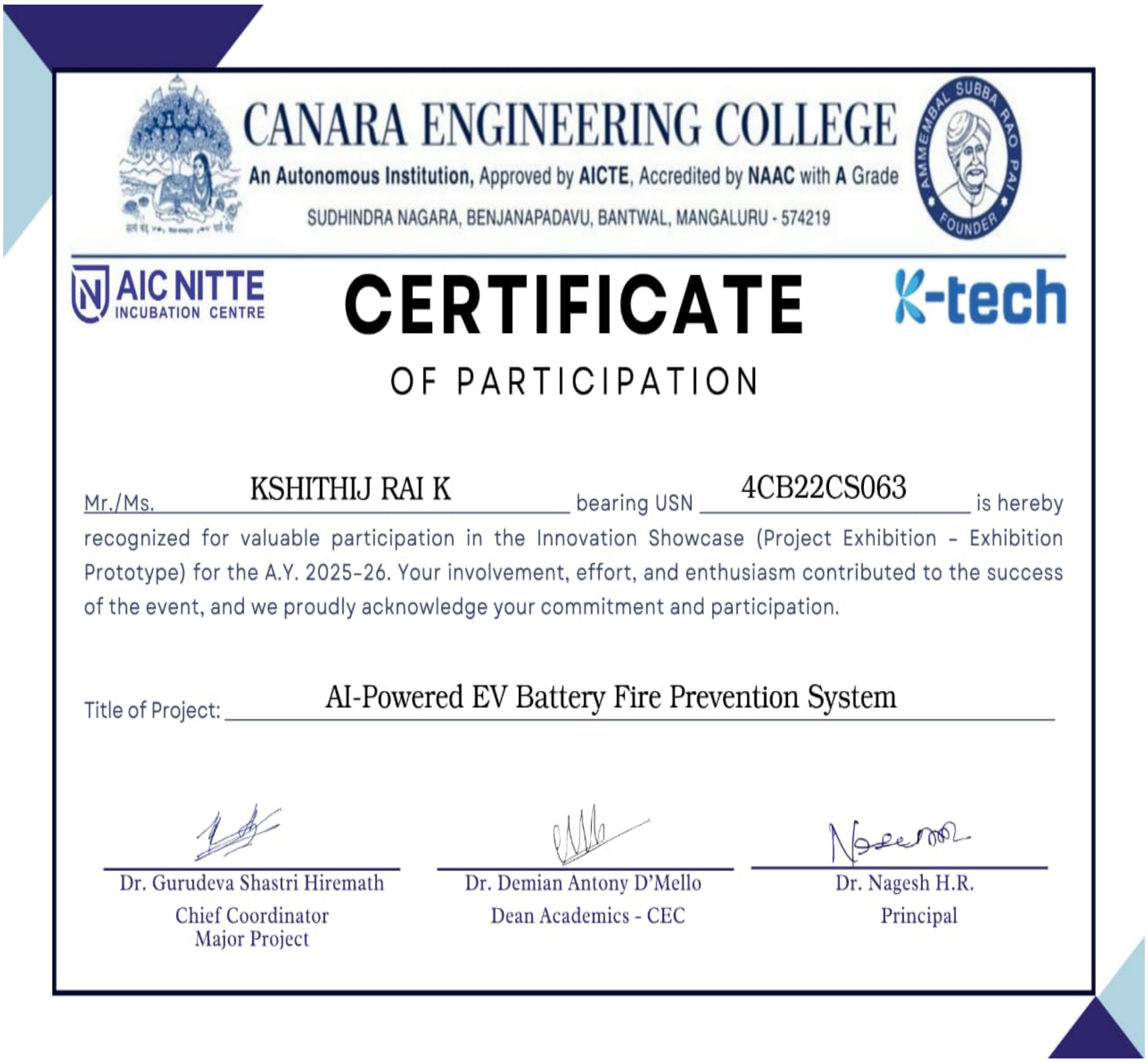
Title of Project: AI-Powered EV Battery Fire Prevention System

[illegible] [illegible] [illegible]

Dr. Gurudeva Shastri Hiremath
Chief Coordinator
Major Project

Dr. Demian Antony D'Mello
Dean Academics - CEC

Dr. Nagesh H.R.
Principal



CANARA ENGINEERING COLLEGE

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

SUDHINDRA NAGARA, BENJANAPADAVU, BANTWAL, MANGALURU - 574219

AMMEMBAL SUBBA RAO PAI FOUNDER

AIC NITTE INCUBATION CENTRE

K-tech

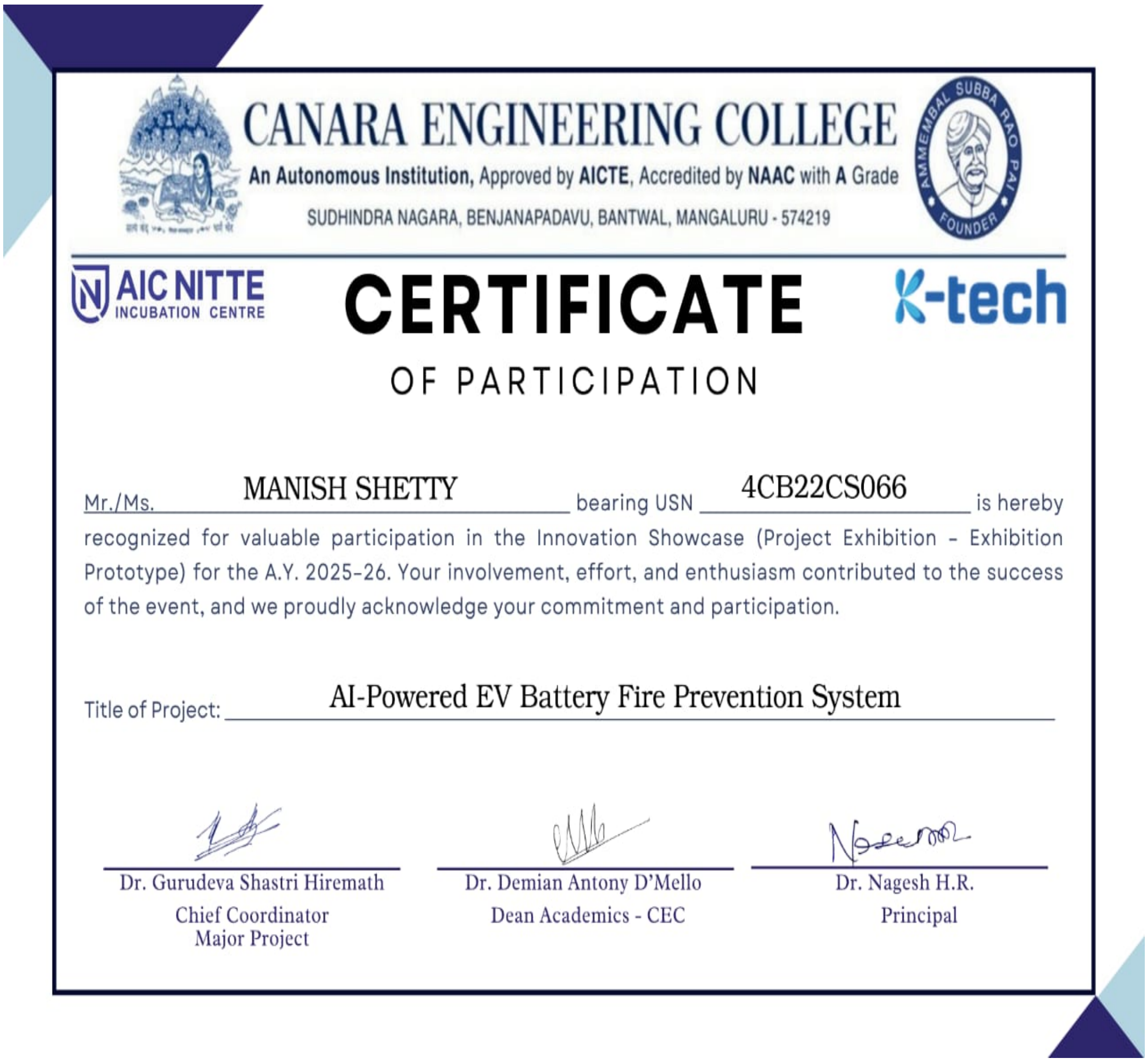
CERTIFICATE

OF PARTICIPATION

Mr./Ms. KSHITHIJ RAI K bearing USN 4CB22CS063 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 EV Battery Fire Prevention System

[illegible]	[illegible]	[illegible]
Dr. Gurudeva Shastri Hiremath	Dr. Demian Antony D'Mello	Dr. Nagesh H.R.
Chief Coordinator Major Project	Dean Academics - CEC	Principal



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

K-tech

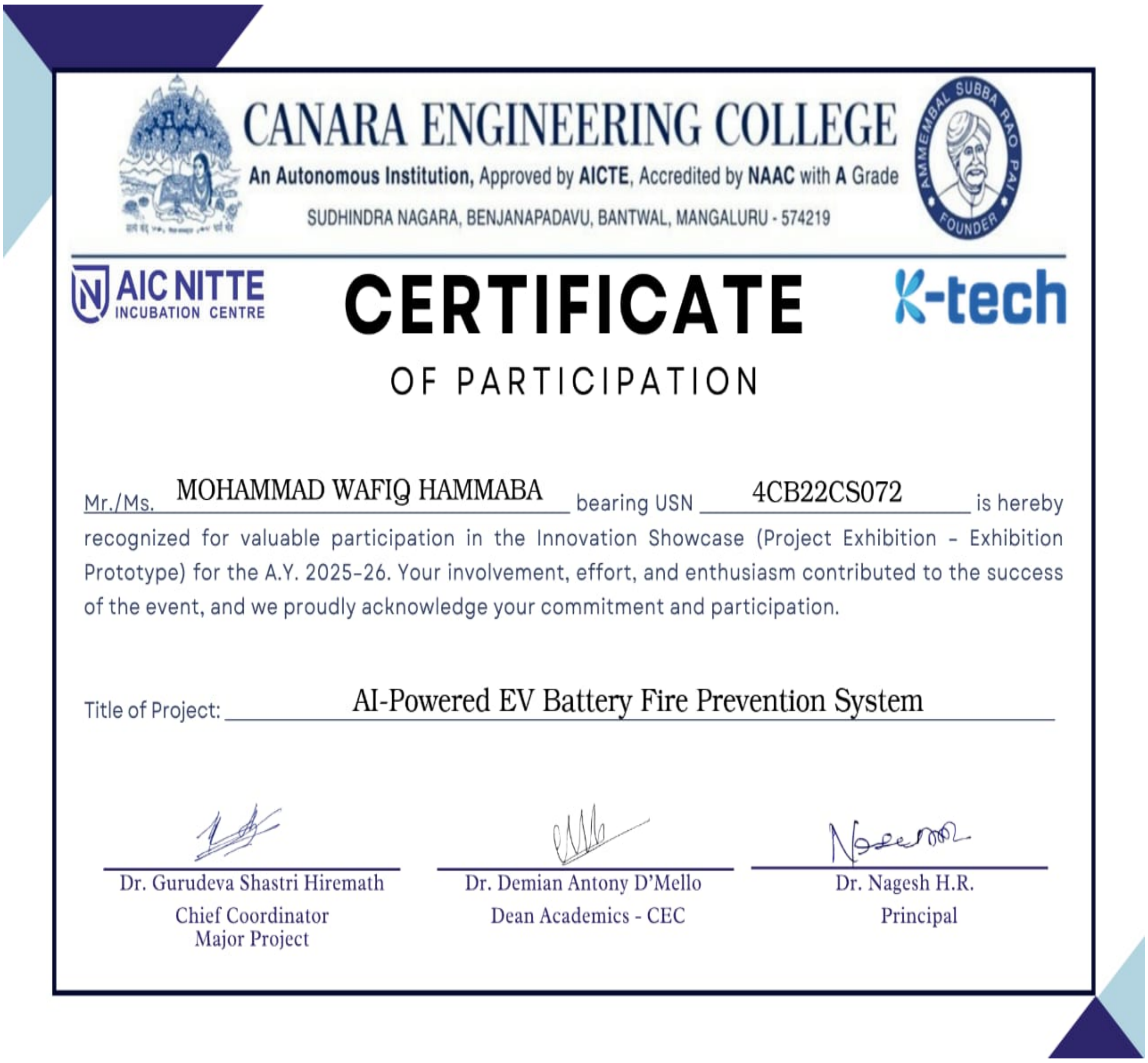
CERTIFICATE

OF PARTICIPATION

Mr./Ms. MANISH SHETTY bearing USN 4CB22CS066 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 EV Battery Fire Prevention System

[illegible]	[illegible]	[illegible]
Dr. Gurudeva Shastri Hiremath	Dr. Demian Antony D'Mello	Dr. Nagesh H.R.
Chief Coordinator Major Project	Dean Academics - CEC	Principal



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

K-tech

CERTIFICATE

OF PARTICIPATION

Mr./Ms. MOHAMMAD WAFIQ HAMMABA bearing USN 4CB22CS072 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 EV Battery Fire Prevention System

[illegible] [illegible] [illegible]

Dr. Gurudeva Shastri Hiremath
Chief Coordinator
Major Project

Dr. Demian Antony D'Mello
Dean Academics - CEC

Dr. Nagesh H.R.
Principal