Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 29s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Power Aware Tech Smart Systems For Precision Battery Health Management

Ms. K.Mohanappriya[1], Dr. N.Keerthivasan[2], Dr. M.Vaidhehi[3], Mrs. C.Muthuselvi[4], S.Roselin Mary[5]

[1]Assistant Professor,Department of Computer Science and Engineering,Saranathan College of Engineering Tirchirappalli, India.

Email ID: mohanapriya-cse@saranathan.ac.in

[2]Assistant Professor,Department of Mechanical Engineering, J J College of Engineering and Technology, Tirchirappalli India

Email ID: keerthinavaneethan@gmail.com

[3]Assistant Professor,Department of Computer Technologies, SRM University,Faculty of Engineering Technology, Kattankulathu Chennai India.

Email ID: vaidhehm@srmist.edu.in

[4]Assistant Professor, Department of Artificial Intelligence and Data Science, K.Ramkrishnan College of Engineering, Tirchirappalli, India.

Email ID: muthuselvi.ai@krce.ac.in

[5]Professor, Department of Computer Science and Engineering, KCG College of Technology Chennai,Tamilnadu, India.

Email ID: jesuroselin@gmail.com

.Cite this paper as: Ni Xiuqin, (2025) Power Aware Tech Smart Systems For Precision Battery Health Management. *Journal of Neonatal Surgery*, 14 (29s), 623-632.

ABSTRACT

This research presents a Battery Monitoring and Notification System that leverages machine learning to predict the Remaining Useful Life (RUL) of vehicle batteries. The system analyzes key indicators such as charging cycles, voltage, temperature, discharge patterns, and battery retention, in combination with real-time sensor inputs and historical performance data. A Decision Tree Regressor is employed to deliver accurate RUL predictions, with model performance evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score. The system features automated email notifications for critical battery conditions, including low voltage, overcharging, and performance degradation. Continuous voltage monitoring enables early detection of potential issues, supporting proactive maintenance. Data is efficiently collected and stored in an SQLite database, while a user-friendly Flask-based dashboard offers visual insights into battery trends, charging history, and predictive analytics. This integrated approach enhances maintenance planning, reduces operational downtime, and improves safety by enabling timely, data-driven decision-making.

Keywords: *Remaining Useful Life; SQLite; Machine Learning; Decision Tree Regressor; Mean Absolute Error; Mean Squared Error; R^2 Score.*

1. INTRODUCTION

Car batteries are critical to powering new cars, engine starting, and powering electrical systems. Battery health monitoring prevents failure and expensive repairs. Tools such as voltage and load testers test battery condition, while machine learning improves battery life and performance

forecasts[1]. The rising need for electrification, driven by environmental issues, has made advanced battery technologies more prominent. Lithium-ion batteries dominate EVs, but Lithium-sulfur and Sodium-ion batteries are gaining traction. High-performance Battery Management Systems (BMS) optimize performance by monitoring important parameters like State of Charge (SOC), State of Health (SOH), and Remaining Useful Life (RUL). ML-based innovation in battery materials, pack structure, and management systems improves safety, efficiency, and lifespan. Accurate RUL predictions also facilitate second-life use for EV batteries, i.e., grid storage[2].

1.1 Battery Parameters

The area of battery management involves a set of parameters critical to guaranteeing optimal performance, safety, and battery life of battery-powered systems. These parameters are voltage levels, charging and discharging cycles, cycle index, and health predictions. All these metrics pose serious challenges in real-time monitoring, precise prediction, and preventive action. Prediction of battery health is significant because it reflects the overall performance, energy capacity, and operational safety of a device. Detection of abnormalities in these parameters at an early stage is significant because it can prevent degradation, extend battery life, and prevent system failure[3]. The Table 1 shows parameters that offer key information about battery behaviour under varying operating conditions. Incorporating predictive models into battery management systems enables efficient anomaly detection and proactive maintenance practices. Improvements in battery diagnostics, real-time monitoring, and artificial intelligence have facilitated the creation of accurate forecasting instruments for battery performance and health. Table 1 presents some battery parameters and their definitions[4]

Table 1 Battery Parameters

S.NO	Battery Parameters	Definition
1	Cycle_Index	The number of charge/discharge cycles that the battery has gone through. Each cycle involves charging the battery to full capacity and then discharging it until it reaches a certain minimum voltage.
2	Discharge Time (s)	The time it takes for the battery to discharge from its fully charged state to its minimum voltage level, typically measured in seconds
3	Decrement 3.6-3.4V (s)	The time it takes for the battery voltage to decrease from 3.6V to 3.4V during discharge, typically measured in seconds
4	Max. Voltage Discharge(V)	The maximum voltage that the battery can output during discharge, typically measured in volts
5	Min. Voltage Charge(V)	The minimum voltage that the battery needs to be charged to, typically measured in volts
6	Time at 4.15V (s)	The time it takes for the battery to reach a voltage of 4.15V during charging, typically measured in seconds
7	Time constant current (s)	The time it takes for the battery to reach a constant current during charging, typically measured in seconds
8	Charging time (s)	The time it takes for the battery to be charged from its minimum voltage level to its fully charged state, typically measured in seconds

2. PROPOSED SYSTEMATIC PLAN

Robust prediction of battery health and longevity is an increasingly important requirement, particularly in relation to electric cars and smart grids[5]. Overcoming the shortcomings of conventional battery monitoring techniques necessitates a structured framework that combines data acquisition, forecasting modeling, and user engagement. Machine learning techniques, especially Decision Trees (DT), provide efficient solutions for such predictive problems by considering different battery-related features like voltage, charge cycles, and usage patterns. For this project, a web-based battery life prediction system is implemented using real- world data to improve the reliability and accuracy of battery performance estimation. The system involves a systematic methodology including preprocessing, feature selection, model training, and real-time prediction. The front-end interface is achieved through HTML and CSS, and the backend is addressed with Python and Flask to facilitate effective user input and predictive output interaction. These intelligent systems facilitate proactive battery care, timely replacement, and operation optimization in electric vehicle management and industrial contexts.

A. Methodology

The methodology contains various steps like Data Collection, Preprocessing, Feature and Target Extraction, Model Training, Model Evaluation, and User Interface development.

B. Systematic Plan

a. Data Collection

The dataset contains data taken from several vehicle batteries, recording crucial operating parameters over different charge/discharge cycles. The database is filled with more than 15,000 entries, with each entry representing a single charge-discharge reading. The vital features covered were the cycle number, discharge time total, battery health prediction, peak and bottom voltage readings while operating, and the total charging time. This organized data is the basis for the analysis of battery behavior, predictive model training, and building a stable web-based battery health monitoring system.[11]

b. Data Preprocessing

The gathered battery dataset is preprocessed to maintain its quality and consistency before model training. Abnormalities like non-numeric characters are eliminated through regular expressions, and the values are transformed into proper numerical formats for precise computation. This process is necessary to address possible data noise, provide uniformity to all features, and preserve the integrity of battery health parameters during predictive analysis.

c. Feature and Target Extraction

In order to construct the predictive model, the dataset is split in a way that it is separated into input features and the target variable. The input features include necessary battery parameters like voltage, cycle index, and charge/discharge times. They are the independent variables with which the model will be trained. The target variable, indicating the health status of the battery or performance category, is the dependent variable. This organized separation permits the machine learning model to learn significant patterns from the battery properties and make correct predictions at inference.

d. Model Training

A Decision Tree model is utilized for model training because it is able to capture nonlinear relationships and offer explainable decision-making rules. The algorithm learns from the past battery data by extracting significant patterns across parameters like voltage levels, cycle numbers, and charge/discharge times. With training, the model can make fairly accurate predictions of battery health or performance based on new or unknown data.

e. Model Evaluation

To evaluate the reliability and performance of the battery life prediction model, a testing dataset is employed. The assessment is conducted by applying standard regression measures, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Coefficient of Determination (R^2 Score). The measures in total provide information on the accuracy of the model to predict the remaining useful life (RUL) of the battery. In addition, a feature correlation heatmap is used to plot the correlation between various battery parameters. Also, a scatter plot of actual and predicted values of RUL is shown to validate the prediction quality and find any deviations, hence proving the model's validity and interpretability.

f. User Interface

The system provides a simple, easy-to-use platform for real-time EV battery health monitoring and charging assistance. It has an interactive dashboard with predictive analytics, voltage analysis, and cycle trends, as well as a location-based Charging Station Finder utilizing Leaflet maps and Google Maps directions. The system is intended for technical and non-technical users alike, and it gives simple, easy-to-understand visuals, color-coded notifications, and easy-to-view insights for wiser EV maintenance and travel.

3. PROPOSED WORK: DECISION TREE REGRESSOR

This paper focuses on machine learning-based predictive models, specifically the Decision Tree Regressor, which are instrumental in estimating the Remaining Useful Life (RUL) of vehicle batteries by analyzing real-time sensor data and historical performance metrics..

Decision Tree Regressor

A decision tree is a tree structure that resembles a flowchart in which an internal node signifies a feature (or attribute), a branch signifies a decision rule, and every leaf node signifies the outcome. The highest node in a decision tree is referred to as the root node. It learns to divide based on the attribute value. It recursively divides the tree, known as recursive partitioning. It's a flowchart-like form that assists you in decision-making. It's a visualization, such as a flowchart diagram, that can readily simulate human-level thought. That is why decision trees are simple to interpret and understand. Decision Tree is a white box category of ML algorithms. It has internal decision-making logic, which is not present in the black box category of algorithms, like Neural Networks. Its training time is less than the neural network algorithm. The time complexity of decision trees is a function of the number of records and the number of attributes in the data provided. A decision tree is a non-parametric or distribution-free technique, which does not rely upon assumptions of probability distributions. Decision trees are able to tackle high-dimensional data with a good level of accuracy. To construct a decision tree, the algorithm initially chooses the optimal attribute to split the dataset based on Attribute Selection Measures (ASM) such as Information Gain, Gain Ratio, or Gini Index. This attribute then becomes the decision node, and the dataset gets partitioned into smaller subsets based on its values. The recursion is then carried out for every subset. Building the tree stops when one of the stopping points is reached: all instances of a subset belong to the same class label, there are no more attributes available to split upon, or there are no further instances to handle.

Entropy

It is a measure of the uncertainty of data. For Classification Machine Learning, Entropy is a measure of the diversification of the labels. A low Entropy suggests that the data labels are highly uniform. A high Entropy indicates the labels are in disarray

$$\boldsymbol{H = -(\sum p_i \log_2 p_i)} \qquad \textbf{(1)}$$

Equation (1) measures how hard we guess the label of the randomly taken sample from the dataset, where p_i ˙ is the proportion of class i in the dataset.

Information Gain

Equation (2) depicts the difference in entropy is referred to as Information Gain, and is an indicator of the amount of information contributed by a feature to the target variable. Entropy_parent denotes the parent node entropy, and Entropy_children indicates the average child nodes' entropies following this variable

$$\textbf{Information Gain} = \textbf{Entropy}_{\textbf{parent}} - \textbf{Entropy}_{\textbf{child}} \qquad \textbf{(2)}$$

The Information Gain of a split is the initial Entropy minus the weighted average of the sub-entropies, where the weights are the fraction of data samples being relocated to the sub-datasets

$$\mathbf{IG_{split} = H - \left(\sum \frac{|Dj|}{|D|} * H_j\right)} \qquad \textbf{(3)}$$

In Equation (3), D is the original data. D_j is the j-th sub-dataset after splitting.|D| and |Dj| are the sizes of the samples that belong to the original dataset and the sub-dataset, respectively. H_j is the Entropy of the j-th sub-dataset.

Gain Ratio

Equation (4) tries to reduce the bias of Information Gain towards highly branched predictors by adding a normalizing factor known as the Intrinsic Information

$$\textbf{Gain Ratio} = \frac{\textit{Information Gain}}{\textit{Intrinsic Information}} \qquad \textbf{(4)}$$

Intrinsic Information is the entropy of sub-dataset ratios. That is, how difficult for us to estimate in which branch a randomly chosen sample is placed is depicted by Equation (5)

$$\mathbf{I\,I = -(\sum \frac{|Dj|}{|D|} * \log_2 \frac{|Dj|}{|D|})} \qquad \textbf{(5)}$$

Gini Index

Equation (6) is another method for splitting a decision tree. The Gini Index, or Impurity, estimates the probability that a randomly selected instance would be misclassified

$$\mathbf{Gini\ Index = 1 - \left(\sum p_i^2\right)} \qquad \textbf{(6)}$$

4. MODULES DESCRIPTIOM

Remaining Useful Time Prediction

This module in real-time receiving input from users via a web-based interface within an interactive dashboard. Users within this interface are asked to input different key measures surrounding battery use, including how many charge-discharge cycles (`cycle_index`), the discharging and charging time, and voltage behavior data like maximum and minimum voltage readings. Also, users input information regarding the duration of the battery spent at critical voltage levels (e.g., the 4.15V overcharge state) and the times spent in particular charging stages, such as constant current and constant voltage modes. Once the data is input, it gets sent to a backend Flask application, which processes the input and sends it to a pre-trained machine learning algorithm. This algorithm, trained on past battery performance datasets, reads the input parameters to forecast the current health of the battery and estimate its Remaining Useful Time (RUT). This smooth flow of user input, smart backend processing, and real-time prediction makes proactive monitoring and effective battery health management possible.

Dynamic Battery Status

This module is the brain of an internet-based Battery Health Monitoring System, predicting the Remaining Useful Time (RUT) of vehicle batteries from user-provided parameters like charge cycles, voltage values, and charging durations. The data is processed by a Flask backend and evaluated by a pre-trained ML model. Depending on the forecasted RUT, the system classifies battery status into modes like 'Full Performance', 'Balanced', 'Power Saving',' Critical', or 'Sleep Mode'. These statuses generate alerts and are also graphically represented on the dashboard through color-coded table rows so that the battery may be rapidly evaluated and managed proactively.

Automated Email Alerts

This module is designed to constantly observe the voltage health of vehicle batteries and automatically notify users by email when important thresholds are exceeded. Acting as a digital safety and maintenance assistant, it guarantees that any case of over-voltage or under-voltage is instantly reported to the vehicle owner or fleet manager, averting possible battery damage or system failure. Every email alert is carefully crafted with rich HTML formatting to present clearly, with urgency and user interest. Alerts contain dynamic information like battery metrics, safety advice, and direct links to nearby charging or service points. Readability is boosted with visually pleasing GIFs and icons, while customized subject lines such as " High Voltage Alert for TN01AB1234" increase relevance and urgency

Real Time Voice Alerts

This module introduces an audible warning system to the Battery Health Monitoring configuration through the Web Speech API, where users are provided with live voice warnings for critical battery incidents. It tracks live data and sends voice warnings for Low battery health, instructing users to proceed towards the nearest charging point. Voltage surpassing 4.6V, warning of possible overheating and prompting quick evacuation. By using Speech Synthesis Utterance, the system provides audible, clear notifications, enhancing safety, accessibility, and awareness, particularly to users who might not be glancing at the screen.

Navigation System For Nearest EV Charging Stations

This module enables users to easily locate the nearest electric vehicle (EV) charging stations by utilizing an interactive and basic map. When the page loads, it tries to determine the user's location. If location access is allowed, the map zooms in on the user and shows the five nearest charging stations. The stations are plotted on the map, and there is a button that allows users to easily open directions in Google Maps. The system also displays a list of these stations close by at the bottom of the map, along with the distance to each station. If the location is inaccessible, the app remains active by showing all charging points within the location. With its basic design and friendly interface, this module is most helpful during crisis situations like battery warnings, hence being a quick addition to any EV support system.

F. Dashboard for Real-Time Battery Health Insights B.Model Selection and Hyperparameter Tuning

A variety of deep learning models, such as DenseNet-121, This module is a web-based Battery Health Dashboard to track and visualize the condition of electric vehicle batteries. Plotly.js provides real-time information through interactive charts, such as battery health forecasts, voltage analysis, charging vs discharging time, and behavior trends. A color-coded table indicates battery performance modes (e.g., Full Performance, Power Saving, Critical). With live data refresh, intuitive visuals, and predictive diagnostics, it facilitates effective decision-making for fleet operators and EV monitoring systems

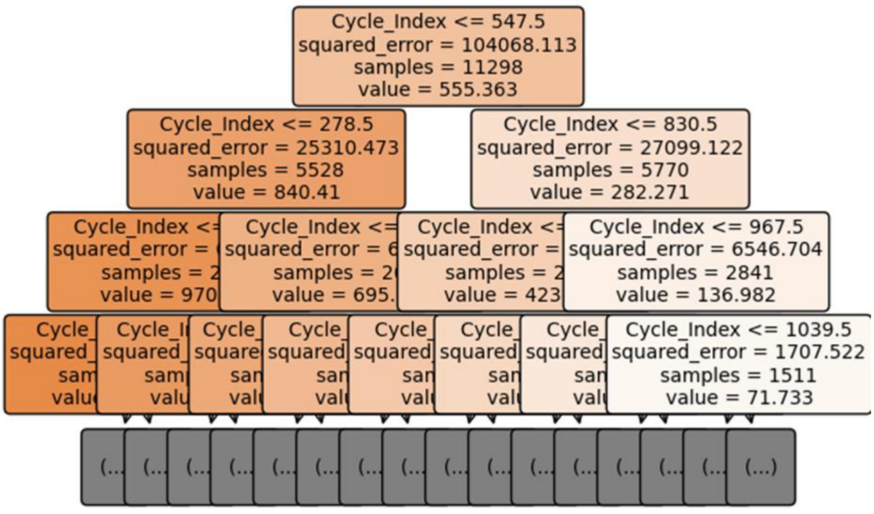

Fig 1 . Decision Tree

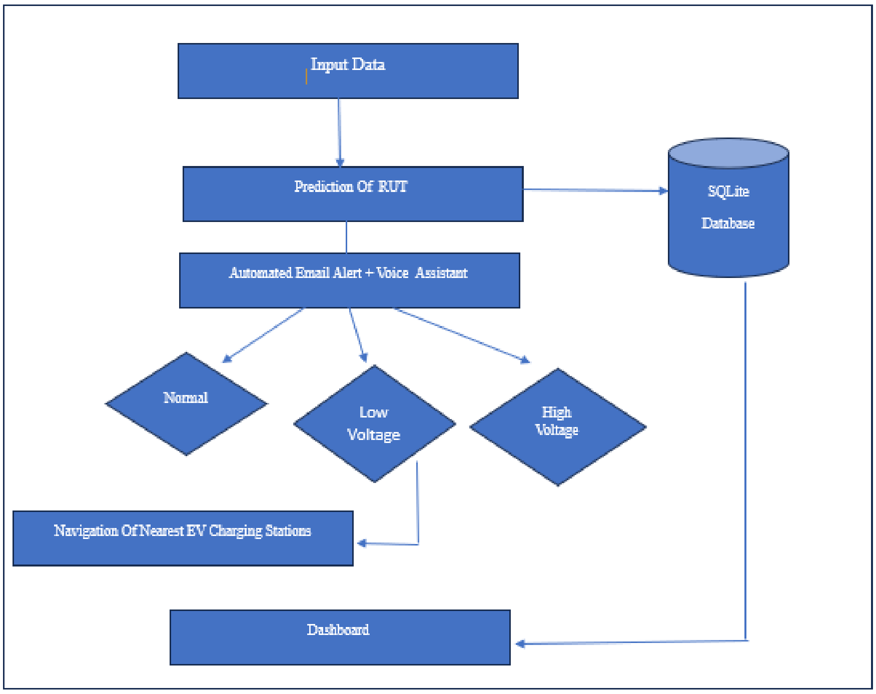

Fig 2 Battery Status Classification

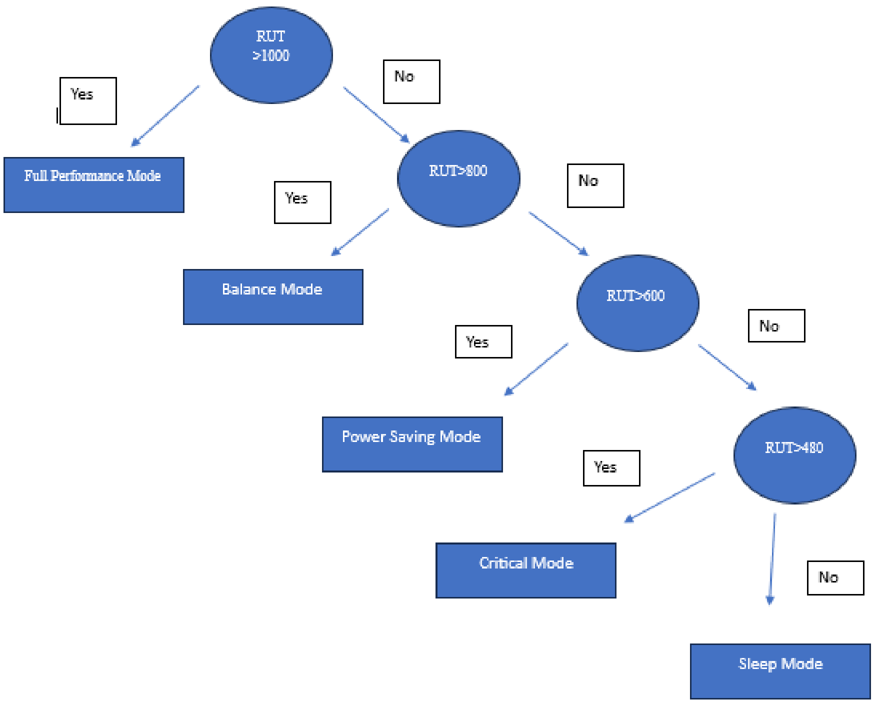

Fig 3 Proposed Approach Structure

5. RESULT AND DISCUSSION

The Decision Tree Regressor model shows excellent performance and thus qualifies as a good contender for real-time use, such as in a battery monitoring system. With a training time of 0.0661 seconds and a testing time of 0.0011 seconds, the model proves to be both lightweight and efficient, making it perfect for use in time-constrained situations. For model precision, the Mean Absolute Error (MAE) is 2.52, indicating that the model's predictions deviate from actual values on average by approximately 2.5 units.

Table 2 : Result Metrics

Evaluation Metric	**Value**
Mean Absolute Error	2.51805629314923
Mean Squared Error	35.21640998406798
R^2 Score	0.9996600986398253
5-Fold Cross-Validation R^2 Scores	[0.99971658 0.99976054 0.99974207 0.99973464 0.99976152]
Training Time	0.0661 seconds
Testing Time	0.0011 seconds

Mean Squared Error (MSE) is 35.22, which is very low and suggests that the model is not making huge mistakes very often. More importantly, the R^2 measure is 0.99966, so 99.966% of the variation in the target variable is accounted for by the model. This is an excellent result and suggests a very accurate fit.

Figure 4 easily shows the excellent performance of your Decision Tree Regressor model. The blue dots for the predicted values plot very close to the red dashed line, the line of perfect prediction correlation (predicted RUT equals actual RUT). The very close correlation throughout the whole range of values shows that your model is always correct, with predictions mirroring actual values for low up to high RUT.

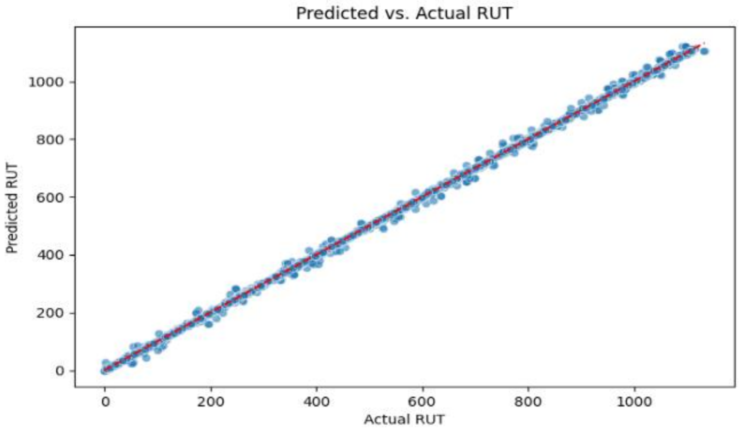

Fig 4 Predicted Vs Actual RUT

Additionally, no discernible systematic error exists within the plot. The points' distribution has no evident skewness, like systematic overestimation or underestimation in specific areas. That is indicative of the model's ability to remain stable and consistent across all ranges of the data. The tight grouping of points along the ideal prediction line also shows that the model has good precision and low variability in its predictions. Coupled with your test metrics—an R^2 value of 0.99966, mean absolute error of 2.52, and mean squared error of 35.22. The plot visually verifies that the model is well-fitted and highly precise.[13]

5-Fold Cross Validation

The 5-Fold Cross-Validation R^2 values are all very close to 1 (approximately 0.99975), showing that the model is explaining almost all of the variance in the target variable very accurately. The narrow box and clustered points indicate

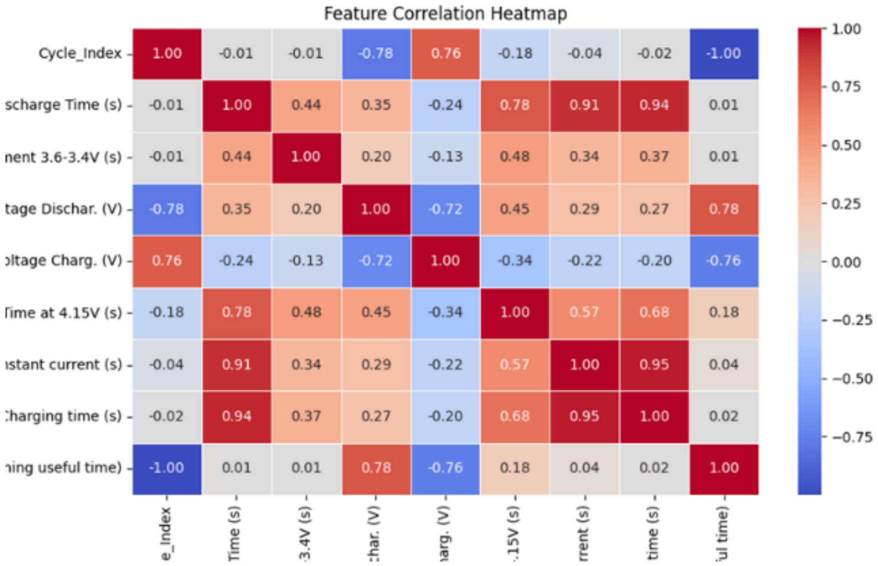

Fig 5 Feature Correlation Heatmap

low variance and similar performance for all folds, reinforcing the robustness of the model. Also, the fact that there are no outliers in the boxplot implies stable and consistent predictions with different data splits.

6. DISCUSSION

The Figure 6 displays the Pearson correlation coefficients between different features in the battery dataset, providing useful information about their inter-relationships. Features like Discharge Time, Time at 4.15V, Constant Current, and Charging Time have high positive correlations with one another and with the Remaining Useful Time (RUT), which are frequently above 0.9. This means that as these variables rise, RUT tends to rise with them, and so they are good predictors of overall battery condition. Conversely, the Cycle_Index has a perfect negative correlation with RUT (−1.00), as one would expect, since battery life tends to reduce as the number of charge-discharge cycles grows. Voltage features also have inverse correlations, though less so. With this in mind, the Cycle_Index is an outstanding feature for the Decision Tree model. The fact that it has a perfect negative correlation with RUL, is simple and easy to interpret, and gives a direct depiction of battery aging, makes it an effective and non-redundant predictor. For light, fast models such as the one used here, Cycle_Index yields a meaningful, clear split that improves model efficiency and interpretability. Voltage-based alerting is the best method of real-time battery monitoring, as it reflects the state of charge, health, and safety of the battery. Extreme values of voltage—either too high or too low—are obvious pointers to serious conditions such as overcharging, deep discharging, or cell failure, which necessitate immediate corrective action. By comparison, other attributes such as Cycle_Index (appropriate for aging prediction), Discharge Time (usage-sensitive), Charging Time (method-dependent), Current (naturally fluctuating), and Time at 4.15V (calculated) are less suitable for real-time notification due to variability, indirect application, or absence of predetermined thresholds. Voltage thus remains the most dependable and actionable threshold for triggering automatic email notification.

7. COMPARISON MODEL

Table 3 clearly shows the training time effectiveness of different machine learning models applied to battery life prediction. The Decision Tree Regressor is most effective with the shortest training time of only 0.0661 seconds, making it very fast and ideal for real-time usage. Conversely, complex ensemble models such as Random Forest, CatBoost, and Gradient Boosting take much more time, and deep learning models such as LSTM are the most time-consuming, up to 300 seconds. Though some intricate models might have slightly higher accuracy, the speed and ease of Decision Tree make it extremely practical for light and real-time systems such as battery monitoring and alerting, where decisions have to be made rapidly.[12]

Table 3 Comparison of Decision Tree Regressor

ML Model	Training Time
Decision Tree Regressor	0.0661
Extra Tree Regressor	0.7090

Random Forest Regressor	1.1350
Extreme Gradient Boosting	0.3550
CatBoost Regressor	2.0920
Gradient Boosting Regressor	0.3380
Linear Regression	0.5730
Light Gradient Boosting	0.1510
LSTM-Based Regressor	300.000
Support Vector Regressor	0.8000

8. LIMITATIONS

The limits of the system include a variety of technical and practical difficulties that impair its function in field implementation. One major constraint is the dependency on static geolocation information, which limits the system's ability to show real-time station availability, charging status, and dynamic pricing. Lack of real-time integration with provider networks constrains the accuracy and utility of navigation choices for end-users. Moreover, the system does not have intelligent filtering functions, like charger types, speed, or network operators-based filtering, hence influencing personalization and convenience for users. Relying on browser-based geolocation also introduces inconsistency in location accuracy and usability, particularly when dealing with permission denial or unsupported devices. Additionally, scalability is also an issue, as the system is geographically limited and cannot support further expansion or global interoperability. These constraints, while they present significant hurdles, also highlight the prospects of innovation with the use of real-time APIs, cloud-computing-compatible architecture, and user-focused capabilities that can reinvent such locators as adaptable and responsive solutions for next-gen electric mobility options.

Table 4 The List Of Abbreviations

ABBREVIATION	DEFINITION
BMS	Battery Management System
SOC	State of Charge
SOH	State of Health
RUT	Remaining Useful Time
ML	Machine Learning
DT	Decision Tree
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SQL	Structured Query Language
EV	Electric Vehicle
MAE	Mean Absolute Error
MSE	Mean Squared Error
R2	The Coefficient Of Determination
GIF	The Graphics Interchange Format
ASM	Attribute Selection Measures

9. CONCLUSION

In conclusion, machine learning is a valuable tool for vehicle battery performance prediction through the examination of varied data sources to detect patterns and accurately estimate battery health and residual life. Regression and decision tree algorithms, when trained on large battery performance datasets, can be used to improve prediction significantly. This has direct applications in stationary energy storage systems and electric cars, maximizing the use of the battery and avoiding sudden failures. The web application, built with a simple and intuitive interface developed using HTML, CSS, Python, and Flask, allows users to input parameters like charge amount, usage time, and charge history to receive forecasts and graphical analyses. In the future, the system can be made more efficient by integrating advanced machine learning techniques such as ensemble learning or deep learning, and real-time data collection via IoT sensors for dynamic and more accurate observation. Mobile optimization, interactive visualization, and multi-language support can also improve user usability and accessibility. Support for different types of batteries and implementation of a recommendation system for ideal charging and discharging behaviors will make it more versatile. Last but not least, deployment on cloud platforms will provide assurance for scalability and effortless integration with automotive and fleet management systems to set the stage for mass use across the industry.

REFERENCES

[1] R. Xiong, Y. Pan, W. X. Shen, H. L. Li, and F. C. Sun, "Lithium-ion battery aging mechanisms and diagnosis method for automotive applications: recent advances and perspectives," Renewable and Sustainable Energy Reviews, vol. 131, pp. 110048, Oct. 2020.

[2] J. Tian, R. Xiong, W. Shen and F. Sun, "Electrode Aging Estimation and Open-Circuit Voltage Reconstruction for Lithium-Ion Batteries", Energy Storage Materials, vol. 37, pp. 283–295, May 2021.

[3] X. Liu, Y. Jin, S. Zeng, X. Chen, Y. Feng, S. Liu and H. Liu. "Online identification of power battery parameters for electric vehicles using a decoupling multiple forgetting factors recursive least squares method", CSEE Journal of Power and Energy Systems, vol. 6, no. 3, pp. 735–742, 2020.

[4] Q. S. Wang, B. B. Mao, S. I. Stoliarov, and J. H. Sun, "A review of lithium ion battery failure mechanisms and fire prevention strategies," Progress in Energy and Combustion Science, vol. 73, pp. 95–131, Jul. 2019.

[5] X. N. Feng, M. G. Ouyang, X. Liu, L. G. Lu, Y. Xia, and X. M. He, "Thermal runaway mechanism of lithium ion battery for electric vehicles: a review," Energy Storage Materials, vol. 10, pp. 246–267, Jan. 2018.

[6] Z. D. Zhang, X. D. Kong, Y. J. Zheng, L. Zhou, and X. Lai, "Real-time diagnosis of micro-short circuit for Li-ion batteries utilizing low-pass filters," Energy, vol. 166, pp. 1013–1024, Jan. 2019.

[7] W. K. Gao, Y. J. Zheng, M. G. Ouyang, J. Q. Li, X. Lai, and X. S. Hu, "Micro-short-circuit diagnosis for series-connected lithium-ion battery packs using mean-difference model," IEEE Transactions on Industrial Electronics, vol. 66, no. 3, pp. 2132–2142, Mar. 2019.

[8] M. B. Chen, F. F. Bai, S. L. Lin, W. J. Song, Y. Li, and Z. P. Feng, "Performance and safety protection of internal short circuit in lithiumion battery based on a multilayer electro-thermal coupling model," Applied Thermal Engineering, vol. 146, pp. 775–784, Jan. 2019.

[9] C. Kupper, S. Spitznagel, H. Doring, M. A. Danzer, C. Gutierrez, A. ¨ Kvasha, and W. G. Bessler, "Combined modeling and experimental study of the high-temperature behavior of a lithium-ion cell: Differential scanning calorimetry, accelerating rate calorimetry and external short circuit," Electrochimica Acta, vol. 306, pp. 209–219, May 2019.

[10] W. X. Wu, S, F. Wang, W. Wu, K. Chen, S. H. Hong, and Y. X. Lai, "A critical review of battery thermal performance and liquid based battery thermal management," Energy Conversion and Management, vol. 182, pp. 262–281, Feb. 2019.

[11] Palanisamy R, Gnana Kousalya C, Ramkumar R, Usha S, Thamizh Thentral TM, D Selvabharathi, Shanmugasundaram V,Analysing UPQC performance with dual NPC converters: Three-dimensional and two-dimensional space vector modulation, Results in Engineering,Volume25,2025,103611,ISSN25901230.

[12] Chandrasekar, L.B., Ramkumar, R., Kiruba, S. et al. The Influence of Bi Doping Concentration on Structural, Dielectric and Optical Properties of Bi-doped ZnO thin Films. Semiconductors 59, 61–69 (2025). https://doi.org/10.1134/S1063782624602127

[13] Geetha, A., Usha, S., Padmanabhan, J.B., Palanisamy, R., Alexander, A., Peter, G.,Ramkumar, R., Ganji, V.: Performance evaluation of coloured filters on PV panels in an outdoor environment. IET Renew. Power Gener. 1– 23 (2024). https://doi.org/10.1049/rpg2.13040