Journal name:
Volume xx, Issue xx, Date
ISSN: xxxx-xxxx, DOI: x.x.x.

Data-Driven Optimization of Recyclable Battery Cells for Electrified Systems Using Machine Learning Regression Models

Ahmed Hereiz[1], Phelopater Ramsis[2], Youssef M. Gunaidi[3], Mennatallah A. Gomaa[4], Doha A. Gomaa[5], Ali Shoma[1], Basem M. Badr[6*]
[1]Mechatonics Department, Faculty of Engineering, Masoura, University, Mansoura, Egypt
[2]Renewable Energy Engineering Department, Faculty of Engineering, Zewail City of Science and Technology, Cairo, Egypt
[3]Communications and Electronics Engineering Department, Misr Higher Institute of Engineering and Technology, Egypt
[4]Renewable Energy Engineering Department, Faculty of Engineering, Indian Institute of Technology, Abu Dhabi, UAE
[5]Artificial Intelligence Department, Faculty of Engineering, Masoura, University, Mansoura, Egypt
[6*]CTO at Lotus Power, Vancouver, BC, Egypt
Email: [1]ahmed.hany.fathi@gmail.com, [2]phelopater.ramsis1@gmail.com, [3]youssefjunaidi7@gmail.com, [4]mennaayman062@gmail.com, [5]doha.ayman2005@gmail.com, [6*]bbadr.digital@gmail.com
*Corresponding Author

Abstract—This work presents a data-driven framework for optimizing and evaluating recyclable battery cells for electrified systems using machine learning (ML). A custom battery-tester system is designed to measure critical parameters that are voltage, current, capacity, internal resistance, and temperature. These measurements are then used to train multiple supervised ML models to estimate the battery's State of Charge (SoC), and estimate its State of Health (SoH). This work explores the prediction of the Remaining Useful Life (RUL) of battery cells using publicly available datasets. Several regression algorithms are evaluated, where the Random Forest model achieved the best performance with the lowest Mean Absolute Error (MAE) and strong generalization across test sets, showing effectiveness in forecasting battery lifespan. In addition to RUL, this study compares several regression algorithms for the dataset with and without temperature input. Results showed that Random Forest and Gradient Boosting achieved the lowest MAE, confirming their robustness and accuracy for SoC estimation under variable thermal conditions. A hybrid numerical approach is used to estimate the SoH, which combines capacity fade and resistance growth provided an effective unsupervised degradation analysis. This ML study helps in forecasting the performance of the recyclable battery cells that are configured for the battery package.

I. INTRODUCTION

Battery systems are fundamental components in modern systems, powering a wide range of devices, such as, electrified systems, electric mobility (e-mobility) systems, portable devices, and other utilities. Traditional batteries show significant environmental challenges, including resource consumption, recycling, and toxic waste generation, whereas using recyclable battery cells shows limitations with respect to the performance. The target and goal of using recyclable batteries are to solve these challenges by enabling the extraction and re-use of valuable materials, which conserves natural resources and minimizes waste, contributing to a greener and more sustainable tomorrow in the real-world [1-2]. In recent years, Artificial Intelligence (AI) and Machine Learning (ML) tools have been increasingly applied to enhance battery system development, offering advantages in modeling complex, nonlinear battery behavior, optimizing parameters, and predicting features such as State of Charge (SoC) and State of Health (SoH) to forecast battery patterns through using data-driven machine learning approaches [3-4]. The battery systems contain a battery package and electronic circuits (mainly including BMS (battery management system) and battery charging), where the battery package consists of modules that contain battery cells. These battery cells and modules are configured and wired in different configurations to provide the required power [1-3]. The integration of AI into the lifecycle of recyclable batteries represents a promising advancement, which can optimize various stages, from the design and manufacturing of batteries with improved recyclability to predictive maintenance and health management during operation. Furthermore, AI-driven models facilitate the identification and sorting of battery waste, improving recycling efficiency and ensuring higher recovery rates of critical materials. ML algorithms also play a pivotal role in forecasting degradation patterns and enabling second-life applications of batteries, thereby maximizing their lifespan before recycling [2-4].

Deploying AI and ML techniques makes battery systems smarter as the use of resources, cut costs, and help speed up the shift to more sustainable battery technologies [4]. In this paper, many benefits of recyclable batteries are explored, the obstacles that still prevent them from becoming widely adopted, and how AI-driven solutions are changing the way of the battery design, use, and recycling of battery cells. Ultimately, the goal is to highlight how these techniques can build a more resilient and environmentally friendly energy future. In this work, a data-driven approach is proposed to optimize and evaluate the use of recyclable battery cells. Using the potential of ML models to estimate SoC, using measurable parameters such as voltage, internal resistance, and temperature [5-6]. These ML and AI tools are utilized to evaluate and estimate the performance of the recyclable battery cells that are configured to provide power to e-mobility systems. A battery tester circuit is designed and developed to measure parameters of the battery cells, which performs controlled discharge and estimates the voltage, current, internal resistance, temperature, capacity accumulated, and the SoC. Regression models are designed and developed to predict SoC and SoH from those measured parameters, where the predicted SoC by the regression model

Received: xxxxx
Revised: xxxxx
Published: xxxx

is compared with the estimated SoC of the battery tester circuit [7]. Different data sets and types are used for the ML model to validate the performances and estimation of the recyclable battery cells, as described in Section V &VI.

This paper is structured as follows. Section I explains the benefits of deploying AI and ML techniques for recyclable battery system. Section II illustrates the literature search of the AI and ML tools for the battery systems. Section III describes the methodology of the battery tester circuit development and the models of the AI and ML tools. Section IV illustrates ML and AI models for the battery cells. Section V explains the system analysis and result of the developed ML and AI models. Finally, the conclusion is drawn in Section VI.

II. LITERATURE SURVEY

The ML techniques have emerged as a transformative tool in battery research, particularly for analyzing chemical and material properties. Various ML software tools are extensively utilized in both academia and industry to advance the understanding and development of battery materials. This survey examines prominent ML software tools, their primary functions, and their advantages and disadvantages with respect the usage, as described.

TensorFlow, developed by Google, is a versatile open-source software library for ML. It is widely used for both research and production across various fields, including battery material analysis. TensorFlow's primary functions include dataflow and differentiable programming, making it highly flexible and scalable. However, its steep learning curve and resource-intensive nature can be challenging for beginners [8]. PyTorch, an open-source ML framework from Facebook’s AI Research lab, is popular in academic research due to its simplicity and ease of use, especially for deep learning applications. PyTorch's dynamic computation graph is intuitive for researchers, and it integrates well with Python, which is widely used in scientific computing. While PyTorch has strong community support, it is slightly less mature in deployment capabilities compared to TensorFlow [9]. Scikit-learn, another widely used open-source ML library for Python, is known for its simplicity and efficiency. It is well-suited for basic ML algorithms and tasks in battery material analysis. Scikit-learn is extensive collection of ML algorithms and ease of use make it ideal for prototyping and smaller projects. However, it is not optimized for deep learning and has limited scalability for large-scale applications [10].

Fault diagnosis is also essential to ensure the safety and efficiency of these systems. This survey explores various ML techniques and tools developed for these purposes. In fault diagnosis for battery systems, various AI/ML techniques are used to detect and identify faults, thereby preventing failures and ensuring safety. While support vector machines (SVM), artificial neural networks (ANN), and decision trees are widely used, they are just a few of the many techniques available. SVMs are particularly effective in high-dimensional spaces and robust to overfitting; although, they can be computationally intensive. On the other hand, ANN technique can capture complex relationships and are adaptable to various problems, but they require large amounts of data and can be prone to overfitting. Decision trees technique is a simple to understand and interpret, requiring little data preprocessing, while they may overfit and be less effective for more complex problems. Beyond these methods, other approaches such as ensemble methods (e.g., random forests, gradient boosting), clustering techniques (e.g., k-means, hierarchical clustering), and deep learning (e.g., convolutional neural networks, recurrent neural networks) have increasingly applied to fault diagnosis due to their ability to handle complex data patterns. Each technique has its advantages and limitations, and the choice of method often depends on the specific characteristics of the battery system and the available data [11]. Implementation of short-term memory (LSTM) technique is used in e-mobility systems to diagnose faults by focusing on electromechanical conversion chains. The technique used data like voltage, current, and speed from a simulated models for the fault diagnosis, which helped the system to identify fault conditions and refine the fault diagnosis before it happens in real-world scenarios. The output of LSTM helped in differentiate between normal and faulty conditions. The LSTM has many advantages like high accuracy, and, it works well for sequence prediction tasks which is good for analyzing time-series data from vehicle sensor. Moreover, it can be predicting new fault patterns due to learning from historical data. The negative side of the LSTM is that it’s so complex specially in real-time systems, where the competitional resources are low, it needs to be trained with large data set, and it can be easily distributed with any noisy or inaccurate input which leads to poor fault detection [12]. Deep learning architectures like convolutional neural network (CNN) and LSTM were used to analyze time-series data and handle large amount of real-time data of battery system. Those approaches helped in prediction of failures by some complex patterns, which increased diagnostic accuracy and they could handle noise, incomplete, or inconsistent data. Those models also could predict future failures; however, implementing those models needed competitional resources which could be expensive and took a lot of energy [13]. Zhang, *et al*., showed the start with future engineering, in which raw data from powertrain were used to get relevant features using some methods like Fourier transform, Wavelet transform, and Empirical Mode Decomposition. Then, use AI algorithms like SVM, neural network (NN), and deep learning approaches like CNN and recurrent neural networks (RNN) techniques helped in identifying faults from the patterns that are estimated/concluded from the raw data. Those models could be trained with time to be suitable for adaptation with new types of faults which also could help in predicting faults. However, it was too complex to be implemented, where it depended on the data, and it was difficult to understand how the fault is determined, which may concern some stakeholders about transparency [14]. An advanced method was developed to determine the faults of lithium-ion battery systems of e-mobility, which combined discrete wavelet transform (DWT) and general regression neural networks (GRNN). The main aim of the DWT was to remove noise while processing voltage data that is collected from the battery. GRNN used some features like variance, covariance, and voltage differences between battery cells to handle the nonlinear data and high fault tolerance. This method increased accuracy and speed of the process, which was very beneficial in safety of e-mobility; however, it had disadvantages like high complexity, very sensitive in tuning of parameters [15].

Accurate estimation of SoC and SoH is critical for the reliable operation and longevity of battery systems. SoC estimation involves predicting the remaining charge in a battery, and a variety of techniques are employed for this purpose. Traditional methods such as Kalman filters and particle filters are widely used. Kalman filters are efficient for real-time applications and provide good estimation accuracy, although they assume system linearity and may require extensive tuning. Particle filters handle nonlinearities better than Kalman filters and are adaptable to various models, but they are computationally expensive and require more processing power [16]. In addition to these methods, various AI and ML techniques are increasingly being applied to enhance SoC estimation. Artificial neural networks (ANNs), for instance, could model complex nonlinear relationships in battery behavior; although they require large datasets for accurate predictions. SVMs techniques were also used to classify battery states with high accuracy in high-dimensional spaces, however they could be computationally intensive. More recently, deep learning approaches such as RNN and LSTM networks have shown promise for handling time-series data, making them particularly useful for dynamic battery environments. These AI techniques provided more flexibility and adaptability compared to traditional methods, especially when dealing with large and complex datasets [17]. Li *et al*., introduced a novel hybrid Kalman filter that enhanced the accuracy of SoC estimation by incorporating temperature compensation and adaptive noise covariance. Unlike traditional Kalman filters, which operated under the assumption of constant noise, this hybrid method dynamically adjusted to environmental changes. This flexibility results in more reliable SoC predictions, especially in varying operational conditions, thereby addressing a key limitation of conventional techniques [18]. In contrast, Pisani *et al*., evaluated simpler ML methods, such as decision trees and random forests, to predict SoC based on historical data and battery characteristics. They highlighted that ANN and deep learning were often favored for their predictive capabilities, simpler models could achieve competitive accuracy with significantly lower computational demands. This made them particularly suitable for resource-constrained environments where efficiency was crucial [19]. Chung *et al*., focused on LSTM networks, emphasizing their ability to capture temporal dependencies in battery behavior. This study built on the discussions around LSTM networks, showing that they were not only outperform traditional methods but also surpass other ML approaches, particularly in long-term predictions. They illustrated a diverse landscape of techniques for improving SoC estimation, from advanced filtering methods to machine learning and deep learning strategies, each with its strengths and applications [20]. SoH estimation assesses the overall health and remaining lifespan of a battery. Techniques like equivalent circuit models (ECM), data-driven models, and hybrid models are used. ECMs provides a good balance between accuracy and complexity and are widely used in the industry, although they require parameter identification and may not capture all degradation mechanisms [21]. Data-driven models leverage large amounts of data and can capture complex degradation patterns, but they require extensive data collection and can be computationally intensive [22]. Chen, *et al*., presented a robust approach for estimating the SoH of lithium-ion batteries through an ECM approach. The authors developed a method to identify model parameters, which allowed the ECM to effectively reflect the battery's health status. Their findings demonstrated that the ECM could reliably estimate remaining capacity and overall battery health across multiple charge-discharge cycles, striking an effective balance between model complexity and accuracy [21]. In contrast, Su delved into various ML techniques for SoH estimation. He gathered the dataset from battery testing to develope predictive models that could capture complex degradation patterns. He emphasized the critical role of feature selection and model training, noting that while data-driven models necessitated comprehensive data collection. The models exceled at capturing nonlinear degradation behaviors compared to traditional methods [23]. Huang, *et al*., discussed the integration of equivalent circuit models and data-driven approaches, showcasing the benefits of hybrid models in enhancing estimation accuracy and robustness. The authors analyzed different hybrid techniques, highlighting how these models effectively leveraged the strengths of both methodologies. This comprehensive understanding of battery health did not only improve estimation but also extended lifespan predictions for batteries in real-world applications, illustrating a promising direction for future research and practical implementations [24].

The choice of ML software tools for chemical and material analysis in batteries depends on the specific requirements of the research or industry project. TensorFlow and PyTorch are preferred for deep learning applications due to their flexibility and scalability, while Scikit-learn is ideal for simpler, more traditional ML tasks. The choice of techniques for fault diagnosis, SoC, and SoH estimations depends on the specific application and the available data. Combining multiple techniques often provides better accuracy and reliability. The following sections describe the proposed battery tester circuit that is designed and developed to gather the parameters of the recyclable battery cells. These collected datasets are used for ML and AI tools to estimate SoC and SoH, and forecast the behavior of these recyclable battery cells.

III. System Design of Battery Tester and ML Models

This study purpose-built battery-tester circuit with a supervised ML pipeline to estimate the SoC and infer SoH. The methodological approach is mainly three parts. In the first approach, firmware (FW) and hardware (HW) are brought in, along with electrical and thermal parameters of controlled discharge and the external verification process through a commercial tester. In the second scheme, the measurement datasets are pre-processed for learning and modeling, the learning process feature definition, pre-processing, training, and validation are schematized. In the third approach, model selection findings and a two-stage evaluation approach are presented, with an ablation quantifying the temperature input feature contribution. This work adds two contributions that are equipment and analytics scheme. On the instrumentation side, the battery tester circuit performs SoC estimation, cross-validates it against open-circuit voltage and internal resistance, and stores a detailed, time-synchronized feature set that minimizes characterization time without loss of precision. On the analytics side, comparative benchmarking for SoC prediction with different regressors with and without temperature is conducted followed by testing on an unseen dataset to assess generalization. A capacity-retention and impedance-growth surrogate-based field SoH formulation is also offered. Hardware and analytics work together to form an end-to-end process from embedded measurement to deployable models.

A battery tester circuit is used to measure the parameters (voltage, internal resistance, temperature, and SoC) of the battery cells to train these parameters and build the ML models. The used battery tester circuit is based on the previous battery tester circuit with upgrades, which was developed and had limitations [25]. The upgrades include improving the HW and FW to eliminate the limitations of the previous battery tester circuit. The previous battery tester measured the voltage and current over time, where a constant discharge current was achieved through a resistor of the battery cell under test. Initially, the battery voltage started from V_{max} as it was fully charged, after which the discharge process continued, the voltage continued to drop until it reached the minimum voltage (cutoff voltage that is 2.7 V), and the SoC was estimated based on CC (column counting) method from the constant discharging current [16]. This approach takes a long time for characterizing the battery cells and the CC SoC estimation process relies on obtaining the full current profile during the battery cell's discharge, where the SoC process was happening till the cutoff voltage was achieved. The proposed battery tester circuit is programmed to estimate the SoC from the current discharge regardless of voltage value set by software or hardware, which saves both time and battery charge. The SoC value depends on the open-circuit voltage (V_{OC}) and internal resistance [26], so the updated battery tester circuit estimates the SoC using the CC topology from the discharge current, the FW validates and verifies the estimated SoC with the internal resistance and the V_{OC} through a lookup table (array), as shown in the flowchart (Fig. 1). The schematic of the upgraded battery tester circuit is illustrated in Fig. 2, which comprises an Arduino Nano microcontroller (MCU), amplifier, LCD, temperature sensor, and a load resistor. The MCU, (Arduino nano) is chosen for its simplicity, small size, and extensive library support, and the amplifier (IC: LM358) is used to drive the switch (transistor Q_1) at a constant current. The load resistor (R_1, 100 Ω, 10 W) is used to ensure safe power dissipation during testing, whereas three 100 Ω resistors are connected in parallel to limit the current, which guarantees the constant amount of current discharge, even if the applied input voltage varies. The LCD provides easy monitoring of the battery tester readings, which communicates with the MCU through I^2C topology.

The proposed battery tester circuit is upgraded from the previous battery tester circuit with respect to the HW & FW, which are displaying, monitoring temperature, current selection control, enhanced alerts and test flow, and improving capacity estimation. Multi-pages display with instant switching (voltage, capacity, resistance, temperature, and SoC), timer always visible for better monitoring. Display state loops through four screens using UP/DOWN buttons to avoid "hanging" and improves user experience. Added a temperature sensor, displays real-time battery temperature, improving safety during heavy discharge and preventing overheating. Nine distinct current levels (0–620 mA), controlled on-the-fly via UP/DOWN buttons, PWM increment/decrement, and immediate display update. More flexible discharge testing and capacity adaptation for different battery sizes. Dual-tone audio feedback for start/stop, clear visual "Test Complete" and final readings, with additional confirmation step before starting (press-and-hold UP button), which prevents accidental test initiation. Capacity is calculated with time broken down into hours/minutes/seconds, uses unsigned long math for overflow protection, and updates every second for accuracy.

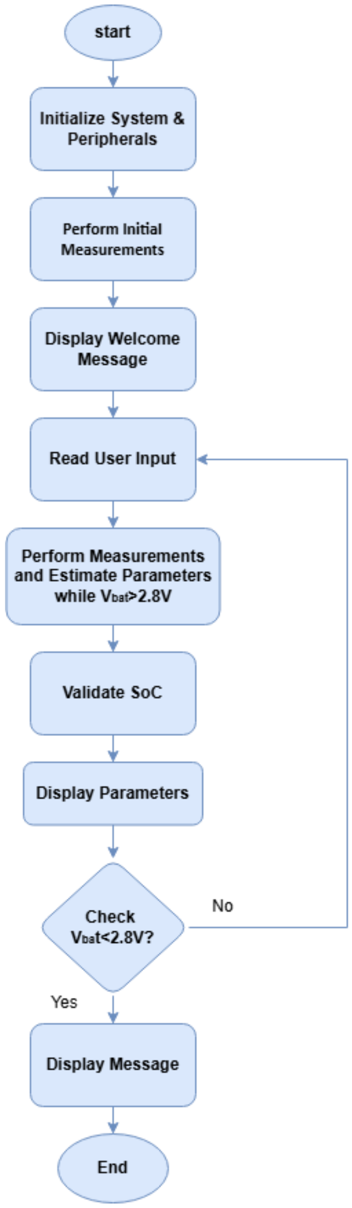

Fig. 1. Flowchart of upgraded battery tester

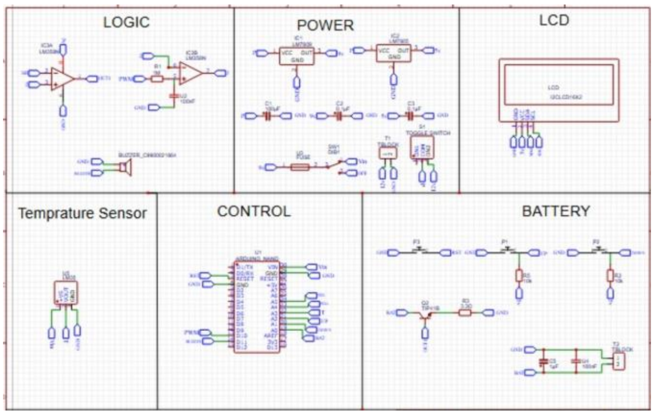

Fig. 2. Schematic of upgraded battery tester

Different data sets are collected from recyclable battery cells, each containing measurements collected under two categories: good and bad recyclable battery cells. The datasets cover the entire cycle, from a fully charged state to an empty state, with capturing variations in the SoC across different operating conditions. All of those are conducted to predict SoC using ML models. Five input parameters of the battery cells are considered as potential predictors of estimating SoC and SoH, which are voltage, current, capacity, internal resistance, and temperature. Two experiments are conducted to test the effective of temperature on prediction accuracy, where the collected datasets are with and without temperature data. The SoC and SoH of the recyclable battery cells are estimated using a combination of numerical and ML methods. SoC is estimated with a supervised regression model trained on time-series data collected from the battery operating under various test conditions. The model employed a feature vector of voltage, current, capacity, internal resistance, and in some cases, temperature to judge the influence of thermal effect on prediction accuracy. The data-driven enabled accurate mapping from electrical parameters to corresponding SoC values. For SoH estimation, a quantitative, unsupervised approach is applied, as there are no cycle-wise health labels in the dataset. Two measures of degradation—capacity fade and growth of internal resistance—are extracted to define the prevailing aging mechanisms. Both measurements are normalized and combined into a unified SoH measure to quantify both electrochemical and resistive degradation effects. Finally, a smoothing filter is applied to the estimated resulting SoH trend to denoise and emphasize the long-term degradation trend of the cell.

A different set of ML regression models is used to find the most effective model that is used for the SoC. The proposed regression models are Random Forest Regressor, Gradient Boosting Regressor, AdaBoost Regressor, Linear Regression, Ridge Regression, Lasso Regression, Support Vector Regressor (SVR), K-Neighbors Regressor, and Decision Tree Regressor. These models are evaluated based on the MAE (mean absolute error) value, where the best model shows the MAE of the specific recyclable battery cell either good or bad ones. Two verification processes are used to evaluate the model as follows. In the first verification step, datasets are divided into 80% training data and 20% test data, to make sure that each model is trained on the most of the data while keeping some for testing. In the second verification step, the chosen model (lowest MAE from the first verification) is tested using a new independent dataset which is not previously used in training or testing. The main purpose of this second verification is to make sure about the predicted data twice under different battery conditions.

Data mining plays a crucial role in extracting useful insights from large datasets, particularly when it comes to analyze battery performance. In this project, the goal is to build and train a model that would allow us to predict various battery parameters, such as its RUL and using real-world data collected from a battery tester. This section covers the process, the datasets used, the models applied, and the anticipated future steps. The first dataset is collected using a battery tester that tracks voltage, capacity, and resistance during the battery's discharge cycle over a specific period. This data is critical because it provides valuable insights into how the battery's performance degraded over time, which could be used to predict and estimate future performance. The developed model is trained on an openly available dataset from Kaggle before using the collected data from battery tester circuit to validate the performance. This dataset includes information about the RUL of various batteries, which is a key metric in battery health prediction, where the Kaggle data [27] helps in experimenting using different machine learning models and refine the approach before working with the more specific dataset from the battery tester. RUL refers to the time or number of cycles a battery is expected to continue functioning effectively before it can no longer meet performance requirements. Estimating RUL is critical in applications, where battery reliability is important, such as electrified systems and renewable energy storage. By predicting the RUL, battery usage can be optimized and avoid unexpected failures, ultimately reducing costs and improving safety. Tracking the RUL during battery testing affirms forecasting the nearing the end of battery's useful life, allowing for proactive management. It also serves as a vital metric to compare the degradation rate of different battery cells and train models for predictive maintenance [28]. To estimate the RUL of the battery cells, several performance-related features from the dataset are used. The input parameters include: Cycle_Index, Discharge Time (s), Decrement 3.6–3.4V (s), Max. Voltage Dischar. (V), Min. Voltage Charg. (V), Time at 4.15V (s), Time constant current (s), Charging time (s), and the corresponding RUL value as the target variable.

Each of these parameters reflects a different aspect of the battery's charge–discharge behavior. For example, six different machine learning models are trained to predict RUL from these input parameters: Random Forest, Decision Tree, SVM, K-Nearest Neighbors (KNN), Gradient Boosting, and AdaBoost. The Random Forest model is an ensemble of multiple decision trees that combines their predictions for better accuracy and generalization. It reduces overfitting and handles complex relationships between features effectively. This model gives the best results for RUL prediction. The Decision Tree model is a simple tree-based model that splits data based on feature thresholds to make predictions. It is easy to interpret and fast to train but can overfit on small datasets if not pruned properly.

The SVM uses hyperplanes to separate data points and find the optimal decision boundary. Although powerful for classification, SVM can be slower and less accurate on larger or noisy regression datasets like ours. The KNN predicts values based on the closest training samples in feature space. It is intuitive and non-parametric but becomes computationally expensive with larger datasets. The Gradient Boosting builds models sequentially, where each new model corrects the errors of the previous one. It provides strong predictive power but requires careful tuning to avoid overfitting. The AdaBoost model is another boosting method that focuses on misclassified samples by assigning them higher weights. While it can improve weak learners, it is sensitive to noisy data and outliers, which affected its RUL prediction accuracy [29].

The ML models are used for estimating SoC from the collected data of the battery tester circuit to validate the estimated SoC (based on CC) from the battery tester. The collected data are good and bad recyclable battery cells, whereas the ML models forecast the performance of these battery cells. These ML models are Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, SVR, and KNN. Linear Regression model assumes a linear relationship between the input features (x) and the target (y). It fits a line (or hyperplane) that minimizes the sum of squared residuals, which is defined as [30]:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \quad (1)$$

$$\hat{\theta} = (X^T X)^{-1} X^T y \quad (2)$$

where, θ represents the model parameters (weights and intercept). The Ridge Regression adds an L_2 penalty term to the loss function to reduce overfitting by shrinking coefficients. The L_2 penalty refers to the squared magnitude of all coefficients which is defined as [29]:

$$min(\theta)\, ||y - X\theta||^2{}_2 + \lambda||\theta||^2{}_2 \quad (3)$$

where, λ controls the strength of regularization. The lasso regression adds an L_1 penalty, which can shrink some coefficients to zero, performing feature selection, which is defined as [29]:

$$min(\theta)\, ||y - X\theta||^2{}_1 + \lambda||\theta||_1 \quad (4)$$

The decision tree model splits the dataset recursively into subsets that minimize prediction error such as mean squared error (MSE) that is defined as [31]:

$$MSE_{Split} = \sum_{i\epsilon left}(y_i - \hat{y}_{left})^2 + \sum_{i\epsilon right}(y_i - \hat{y}_{right})^2 \quad (5)$$

where, y_i is the actual target value, $\bar{y}_{left}$ and $\bar{y}_{right}$ are the mean target values of the left and right child nodes, respectively. The random forest averages predictions from multiple decision trees trained on different data subsets to improve robustness and reduce variance, which is defined as [31]:

$$\hat{y} = (1/N) \sum_{i=1}^{N} T_i(x) \quad (6)$$

where, N is the total number of trees in the forest. The gradient boosting builds trees sequentially, whereas each new tree corrects the errors of the previous model using gradient descent on the loss function, which is defined as [32]:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (7)$$

where, m is the iteration index, η is the learning rate controlling step size, and $h_m(x)$ is the weak learner (typically a small regression tree) trained to fit the residuals of the previous model. The AdaBoost model combines multiple weak learners by reweighting samples so that the next model focuses more on previously misclassified instances, which is defined as [33]:

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x) \quad (8)$$

$$\alpha_m = log((1 - e_m)/e_m) \quad (9)$$

where, α_m is the model weight proportional to its accuracy. The SVR model fits a function within an ε-insensitive tube and minimizes error outside it, balancing model flatness and tolerance for deviations, which is defined as [34]:

$$min\_\{w, b\}\, (1/2).\, ||w||^2 + C \sum_{i=1}^{m}(\xi_i + \xi_i{}^*) \quad (10)$$

subject to:

$$y_i - (w^T x_i + b) \le \varepsilon + \xi_i$$

$$(w^T x_i + b) - y_i \le \varepsilon + \xi_i{}^*$$

$$\xi_i, \xi_i{}^* \ge 0 \quad (11)$$

where, w is the weight vector, b is the bias term, ξ_i and $\xi_i{}^*$ are slack variables for errors outside the ε margin, and C is a regularization constant controlling the trade-off between flatness and tolerance to error. The KNN model predicts the target value by averaging the outputs of the k nearest training samples based on a distance metric like Euclidean distance, which is defined as [35]:

$$\hat{y}(x) = (1/k) \sum_{i\epsilon left} y_i \quad (12)$$

ML equations are used for estimating the SoC with and without temperature data, where the feature victor is initialized for time step as [30]:

$$X(t) = [V(t), I(t), cap(t), R_{int}(t), T(t)] \quad (13)$$

where, V is the terminal voltage, I is current (A), R_{int} is internal resistance (mΩ), C is accumulated capacity (mAh or Ah), and T is temperature (°C) — optional feature. The equation is used with and without the temperature data. The trained regressor $f(x)$ estimates SoC as [30]:

$$f(x_t) = SoC\,(t) \quad (14)$$

The model $f(x)$ is learned from data (not a fixed physics formula). Given the battery's instantaneous, directly measurable signals (x), it outputs an estimate SoC. The SoH is estimated through a purely numerical, unsupervised approach. Since the measured data lacked cycle-wise SoH labels, a physics-based numerical estimation is implemented. Two degradation indicators are considered, which are capacity fade and resistance growth. The SoH decreases as the available charge capacity diminishes relative to the rated nominal capacity. The SoH decreases as the internal resistance increases due to aging mechanisms and loss of active material. The instantaneous SoH is therefore defined as the normalized combination of these two independent degradation factors. The SoH based on capacity scheme is defined as [30]:

$$SoH_{cap}(t) = \frac{Q(t)}{Q_{rated}} \quad (15)$$

where, $Q(t)$ is the measured accumulated capacity at time t and Q_{rated} is the nominal rated capacity of a new cell. The SoH based on resistance scheme is defined as [30]:

$$SoH_{res}(t) = \frac{R(0)}{R(t)} \quad (16)$$

where, $R(0)$ is the internal resistance measured at the beginning of life (first data point) and $R(t)$ is the instantaneous internal resistance. To improve robustness against noise or temporary fluctuations, both indicators are fused equally [30]:

$$SoH_{num}(t) = \frac{1}{2}\left(SoH_{cap}(t) + SoH_{res}(t) \right) \quad (17)$$

The resulting series is smoothed with a 5-point moving average filter to obtain a continuous health trend.

IV. *System Analysis and Discussion*

The battery tester circuit of one battery holder is built, where the LCD displays the measured parameters that are voltage, internal resistance, SoC, temperature, and battery capacity, as shown in Fig. 3. These measured parameters data are saved in log files using the UART protocol, which are used for ML. The user navigates through the menus of the battery tester through tactile switches, as described in Section III. The measurements of the battery tester circuit are verified and validated using known battery cells and an external battery tester, where the measured parameters are compared with the parameters of name plate of the battery cells. A Liitokala battery tester is used to validate the FW of the proposed battery tester circuit, where the battery capacity and internal resistance are compared. Recyclable battery cells (with and without name plates) are used to measure their parameters (capacity and internal resistance) and compared with the Liitokala battery tester. The maximum error percentage of the capacity and internal resistance are 0.65 and 5.2 %, respectively, which affirm the accurate measurements of the battery circuit.

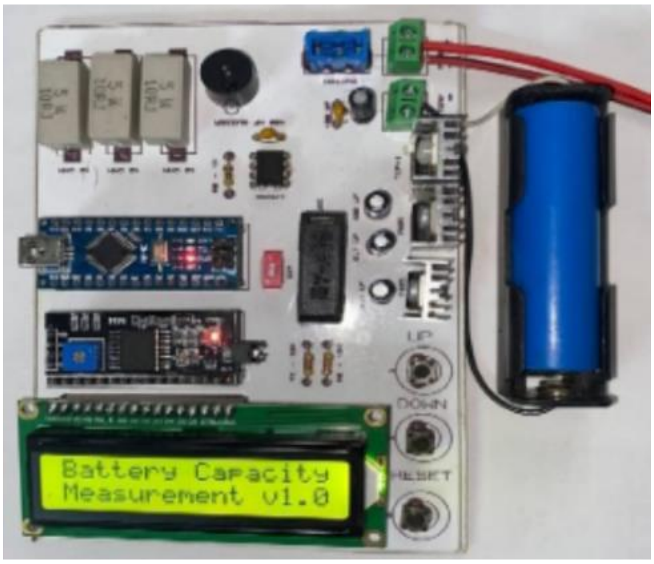

Fig. 3. Proposed battery tester circuit

The battery tester circuit is used to measure the parameters of good and bad recyclable battery cells, which are voltage, current, internal resistance, SoC based on CC, and temperature. These datasets of the good and bad recyclable battery cells are trained using proposed ML model to develop the model and evaluate the performance of the ML models. The ML models are trained and evaluated of the dataset with and without the temperature data. Each model is evaluated using MAE that measures the average difference between predicted and actual RUL values — lower MAE means better accuracy. The time period of the train and test dataset is reported for each model. Table I lists the MAE and time of these models, where the random forest and decision tree show the lowest MAE and less time value, respectively. Overall, the Random Forest model achieves the lowest MAE with a good balance between accuracy and computation time, making it the most suitable candidate for future model refinement once the real battery tester data becomes available. Figure 4 illustrates the scatter plot, histogram error, and residual plot, where scatter with regression line plot compares the actual values (on the x-axis) with the predicted values (on the y-axis) from the Random Forest model. The closer the points are to the diagonal line, the more accurate the predictions, as illustrated in Figure 4(a). A perfect model would have all points lying on this line. The histogram of errors plot shows the distribution of prediction errors (difference between predicted and actual values). A well-performing model typically has a narrow, centered distribution with most errors close to zero, as shown here. The residual plot displays the residuals (errors) against the predicted values, as shown in Figure 4(c). Ideally, the residuals should be randomly scattered around the horizontal line at zero, indicating no clear pattern and unbiased predictions. Patterns or trends could indicate issues with the model. These results reflect the initial experiments using the Kaggle dataset, and further tuning is done for the real dataset from the battery tester.

The results are given per feature setting and dataset (with/without temperature data), which are chosen model, most significant hyperparameters, test MAE (initial verification), independent-set MAE (follow-up verification), and ΔMAE to quantify the contribution of temperature. This presentation gives an open, reproducible platform for selecting the functional SoC estimators. For the initial verification, the data are split into 80% of the data for training and 20% for testing for the verification. Tables II and III represent the MSE, MAE, and R_2 score of the good battery cells with and without temperature data, respectively. The lowest MSE and MAE are achieved of the good battery cells of using Gradient Boost and Random Forest model with and without temperature data, respectively. Same performance and pattern are observed of the ML models of the bad battery cells, as listed Tables IV and V. The analysis and data show that the ML model of the dataset including the temperature shows higher R_2 score and lower MSE and MAE than the ML model of the dataset without temperature data. The MSE and MAE values of the bad recyclable battery cells are higher than the good recyclable battery cells, because of the low life time of the bad battery and fast degrading over time. These results illustrate the importance of the temperature data, especially for the bad recyclable battery cell, as the temperature increases while discharging of the bad recyclable battery cells.

TABLE I
MAE AND TIME OF ML MODELS FOR RUL

Model	**MAE**	**Time (sec)**
Random Forest	2.674390	0.044941
Decision Tree	2.883344	0.002311
SVM	17.03846	0.929667
KNN	3.636907	0.011726
Gradient Boosting	5.728628	0.004883
AdaBoost	11.35746	0.012682

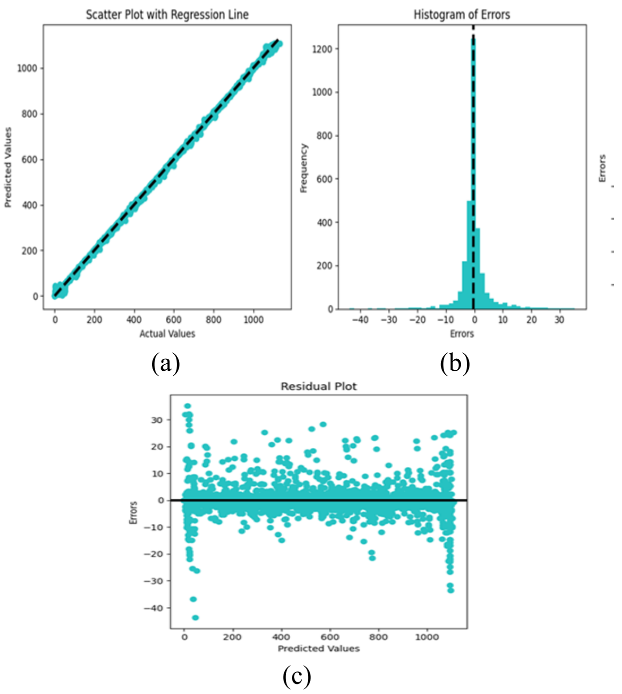

Fig. 4. RUL results of (a) scatter plot, (b) histogram error, and (c) residual plot

The developed ML models are validated using an independent dataset containing operational measurements from different battery cells, where voltage, current, resistance, temperature, and capacity readings are collected over a 2-minute discharge period. This external dataset is not used during model training, ensuring that the validation process truly evaluates the generalization capability of the proposed approach. The second stage of validation compares the SoC estimated by the machine learning models with the measured SoC obtained directly from the battery tester, which serves as the ground-truth benchmark. This evaluation reflects the generalization ability of the trained models when tested on unseen operational data. The MAE of the SoC of the good battery cells with and without temperature are 0.3032 and 0.3598; although, the MAE of the bad battery cells are 3.5564 and 3.614, respectively. The performance comparison demonstrates that the SoC estimation from the ML-based approach is well aligned with the measured SoC for healthy batteries, while the estimation accuracy decreases for degraded bad battery cells due to their nonlinear behavior while discharging.

TABLE II
PERFORMANCE OF ML MODELS OF GOOD BATTERY CELLS WITH TEMPERATURE

Model	**MSE**	**MAE**	**R_2 Score**
Random Forest	0.135260	0.286751	0.999807
Gradient Boosting	0.130932	0.282711	0.999814
AdaBoost	10.772254	2.860413	0.984659
Linear Regression	26.831218	3.952419	0.961788
Ridge Regression	26.828516	3.952860	0.961792
Lasso Regression	27.171612	4.151354	0.961303
Support Vector Regressor	0.681579	0.467522	0.999024
K-Neighbors Regressor	0.176062	0.316200	0.999749
Decision Tree	0.135626	0.287002	0.999807

TABLE III
PERFORMANCE OF ML MODELS OF GOOD BATTERY CELLS WITHOUT TEMPERATURE

Model	**MSE**	**MAE**	**R^2 Score**
Random Forest	0.128990	0.279590	0.999816
Gradient Boosting	0.130807	0.282408	0.999814
AdaBoost	10.772254	2.860413	0.984659
Linear Regression	26.850925	3.947539	0.961760
Ridge Regression	26.848138	3.947998	0.961764
Lasso Regression	27.171612	4.151354	0.961303
Support Vector Regressor	0.169031	0.318137	0.999759
K-Neighbors Regressor	0.153348	0.298671	0.999782
Decision Tree	0.128952	0.279512	0.999816

The SoH of the battery cells is estimated numerically using both capacity and internal resistance indicators extracted from the diagnostic dataset. The initial internal resistance value at the beginning of operation is used as the reference resistance R_0. The first SoH indicator reflects the reduction in effective capacity and is calculated as the ratio between the measured capacity and the rated capacity of the cell, the second indicator accounts for degradation-induced resistance growth and is computed as the ratio of the reference resistance to the current resistance.

TABLE IV
PERFORMANCE OF ML MODELS OF BAD BATTERY CELLS WITH TEMPERATURE

Model	MSE	MAE	R_2 Score
Random Forest	0.480580	0.537255	0.999118
Gradient Boosting	0.464488	0.536658	0.999148
AdaBoost	1.907825	1.117717	0.996499
Linear Regression	88.97285	7.200109	0.836711
Ridge Regression	88.96625	7.201977	0.836723
Lasso Regression	91.54232	7.540931	0.831995
Support Vector Regressor	3.737671	1.121381	0.993140
K-Neighbors Regressor	0.765967	0.631741	0.998594
Decision Tree	0.483797	0.538982	0.999112

TABLE V
PERFORMANCE OF ML MODELS OF BAD BATTERY CELLS WITHOUT TEMPERATURE

Model	MSE	MAE	R_2 Score
Random Forest	0.448353	0.526973	0.999177
Gradient Boosting	0.449926	0.530315	0.999174
AdaBoost	1.907525	1.117717	0.996499
Linear Regression	110.286862	8.172627	0.797594
Ridge Regression	110.284860	8.175130	0.797597
Lasso Regression	111.078613	8.448170	0.796141
Support Vector Regressor	1.435098	0.668823	0.997366
K-Neighbors Regressor	0.569687	0.571926	0.998954
Decision Tree	0.448867	0.526996	0.999176

The backend system is implemented using FastAPI, a high-performance web framework designed for building and serving machine learning applications. The system exposes an endpoint (predict) that receives sensor and diagnostic data from battery cells, including voltage, current, resistance, temperature, and capacity. Upon receiving the input, the API dynamically loads the appropriate ML model (.joblib file) depending on the selected mode—either for full-to-empty or full-to-low discharge evaluation. The model then predicts the SoC using a trained regression algorithm. In addition to SoC estimation, the backend computes and displays the SOH based on two key parameters: the ratio of the measured to rated capacity and the ratio between reference internal resistance (R_0) and the current measured resistance. These two indicators are averaged to form a combined SoH value, which is further bounded between 0 and 1 to ensure valid health representation. The entire prediction pipeline is encapsulated within a RESTful interface, allowing seamless integration with the frontend visualization system and external monitoring devices. The modular structure enables model updates and the inclusion of temperature-dependent behavior through a toggle (use_temp), providing flexible adaptability for different operational conditions. Figure 5 illustrates the GUI (graphic user interface) of the backend code that displays the SoC and SoH value, where the battery cell is selected, either good or bad battery cell.

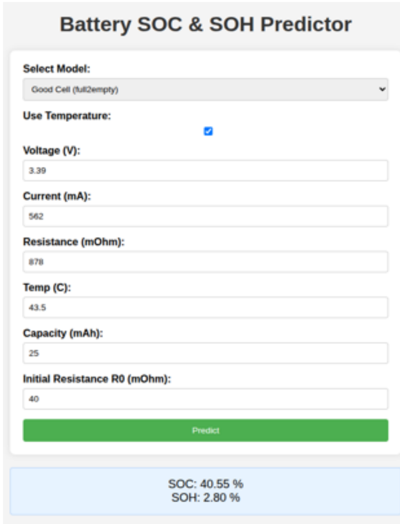

Fig. 5. GUI of the battery cell parameters

V. CONCLUSION AND FUTURE WORKS

This research paper presented the study of using ML for estimation and analysis of recyclable/recoverable battery cells. The customized battery tester circuit was designed and developed to measure the parameters of good and bad recyclable battery cells. Supervised ML models were conducted for the collected parameters of the bad and good battery cells, the SoC and the SoH were estimated. The validation with and without temperature demonstrated the impact of temperature conditions on estimation accuracy. For RUL, Random Forests model was the best with the lowest MAE. The Random Forest and Gradient Boosting Model were the best for the SoC prediction with and without temperature, respectively. The analysis and results showed that capacity loss and internal resistance growth parameters were the main two factors in the SoH estimation, where it provided a strong unsupervised battery degradation trend analysis. This entire integration of ML into the constructed test hardware is an excellent and scalable model for smart battery characterization that is deemed hugely valuable for electrified and sustainable recycling applications.

A smart battery tester system with integration of IoT (internet of things) can be developed, it will allow automatic evaluation, data logging, remote access, cloud based analytic, and real-time tracking of battery performance. Also, another future focus would be on implementing more advanced ML like LSTM networks to capture more time-dependent behaviors under different conditions which will help in storing historic data so it will forecast in detecting faults before even it happens and make the system efficient. Moreover, it will expand the dataset to include more temperature ranges, charge–discharge cycles, and different cell chemistries will also help improve model results.

REFERENCES

[1] S. R. Abdelmalek, M. A. F. B. Retig, F. S. Ansari, M. A. Gomaa, A. Alaaeldin, S. S. Kaddah, and B. M. Badr, "Design and Development of Semi-Steering and IoT Systems for Electric Mobility Systems," *6th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Giza, Egypt, Oct. 19-21. 2024. DOI: 10.1109/NILES63360.2024.10753180.

[2] J. Li, S. He, and Q. Yang, "A Comprehensive Review of Second-Life Batteries Toward Sustainable Mechanisms: Potential, Challenges, and Future Prospects," IEEE Transactions on Transportation Electrification, vol. 9, no. 4, pp. 4824 – 4845, Dec., 2023. DOI:10.1109/TTE.2022.3220411.

[3] D. N. T. How, M. A. Hannan, M. S. H. Lipu, and P. J. Ker, "State of Charge Estimation for Lithium-Ion Batteries Using Model-Based and Data-Driven Methods: A Review," IEEE Access, vol. 7, pp. 136116-136136, Sept. 2019. DOI: 10.1109/ACCESS.2019.2942213.

[4] C. Vidal, P. Malysz, P. Kollmeyer, and A. Emadi, "Machine Learning Applied to Electrified Vehicle Battery State of Charge and State of Health Estimation: State-of-the-Art," IEEE Access, vol. 8, pp. 52796–52814, March 2020. DOI: 10.1109/ACCESS.2020.2980961.

[5] Z. He, Z. Yang, X. Cui, and E. Li, "A Method of State-of-Charge Estimation for EV Power Lithium-Ion Battery Using a Novel Adaptive Extended Kalman Filter," IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14618 - 14630, Dec. 2020. DOI: 10.1109/TVT.2020.3032201.

[6] A. S. Alneklawy, A. M. Elbeshbeshy, A. Gomaa, Y. M. El-Hawary, H. A. Ahmed, B. M. Badr, "Design and Development of Low-cost Active Battery Balancing Technique for Electric Motorcycles," Journal of Engineering Research and Reports, vol. 27, no. 7, pp. 188-203, July 2022.

[7] L. Chen, X. Bao, A. M. Lopes, C. Xu, X. Wu, H. Kong, S. Ge, and J. Huang, "State of Health Estimation of Lithium-ion Batteries Based on Equivalent Circuit Model and Data-driven Method," *Journal of Energy Storage*, vol. 73, Part D, Dec., 2023. DOI: doi.org/10.1016/j.est.2023.109195.

[8] M. Abadi, et al., "TensorFlow: A System for Large-scale Machine Learning," *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*. Nov. 2–4, 2016, Savannah, GA, USA.

[9] A. Paszke, et al., "PyTorch: An imperative Style, High-performance Deep Learning Library," *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, no. 721, pp. 8026 – 8037, December 2019. DOI: https://doi.org/10.48550/arXiv.1912.01703.

[10] F. Pedregosa, et al., "Scikit-learn: Machine Learning in Python," The Journal of Machine Learning Research, vol. 12, pp. 2825 - 2830, Nov. 2011.

[11] K. Liu, S. Zhao, Y. Wang, K. Li, J. Wang, Y. Sun, Q. Wu, and Q. Peng, "Advanced Fault Diagnosis in Batteries: Insights into Fault Mechanisms, Sensor Fusion, and Artificial Intelligence," Advances in Applied Energy, vol. 20, Dec., 2025.

[12] T. Kambiz, H. Kaplan, and M. Jamshidi, "A Fault Diagnosis Design Based on Deep Learning Approach for Electric Vehicle Applications," *Energies*, vol. 14, no. 20, 2021. DOI: 10.3390/en14206599.

[13] J. Zhao, X. Qu, Y. Wu, M. Fowler, and A. F. Burke, "Artificial intelligence-driven real-world battery diagnostics," *Energy and AI*, vol. 18, Dec., 2024. DOI: 10.1016/j.egyai.2024.100419.

[14] X. Zhang, Y. Hu, J. Deng, H. Xu and H. Wen, "Feature Engineering and Artificial Intelligence-Supported Approaches Used for Electric Powertrain Fault Diagnosis: A Review," IEEE Access, vol. 10, pp. 29069-29088, 2022, DOI: 10.1109/ACCESS.2022.3157820.

[15] L. Yao, Y. Xiao, X. Gong, J. Hou, and X. Chen, "A Novel Intelligent Method for Fault Diagnosis of Electric Vehicle Battery System Based on Wavelet Neural Network," *Journal of Power Sources*, vol. 453, March 2020. DOI: 10.1016/j.jpowsour.2020.227870.

[16] A. M. Elbeshbeshy, O. Matar, A. Shawky, A. Alaaeldin, H. S. Faiod, M. A. Gomaa, H. A. Ahmed, A. G. Shaheen, S. S. Kaddah, and B. M. Badr, "Design and Development of a Low-cost Battery System for Electric Motorcycles," in *Proc. 2023 Int. Middle East Power Syst. Conf. (MEPCON)*, 2023. DOI: 10.1109/MEPCON58725.2023.10462347.

[17] L. Ma and T. Zhang, "Deep Learning-based Battery State of Charge Estimation: Enhancing Estimation Performance with Unlabelled Training Samples," Journal of Energy Chemistry, vol. 80, no. 6, pp. 48-57, May 2023.

[18] C. Li and G.-W. Kim, "Improved State-of-charge Estimation of Lithium-ion Battery for Electric Vehicles Using Parameter Estimation and Multi-innovation Adaptive Robust Unscented Kalman Filter," Energies, vol. 17, no. 1, Jan., 2024. DOI: https://doi.org/10.3390/en17010272

[19] M. A. P. Orta, D. G. Elvira, and H. V. Blavi, "Review of State-of-Charge Estimation Methods for Electric Vehicle Applications," World Electric Vehicle Journal, vol. 16, no. 2, Feb. 2025. DOI: https://doi.org/10.3390/en16020087

[20] D.-W. Chung, J.-H. Ko, and K.-Y. Yoon, "State-of-Charge Estimation of Lithium-Ion Batteries Using LSTM Deep Learning Method," Journal of Electrical Engineering & Technology, vol. 17, pp. 1931–1945, 2022. DOI: 10.1007/s42835-021-00954-8.

[21] L. Chen, X. Bao, A. M. Lopes, C. Xu, X. Wu, H. Kong, S. Ge, and J. Huang, "State of health estimation of lithium-ion batteries based on equivalent circuit model and data-driven method," Journal of Energy Storage, vol. 73, Part D, Dec., 2023. DOI: https://doi.org/10.1016/j.est.2023.109195.

[22] C. Zhao, S. Hashemi, P. B. Andersen, and C. Træholt, "Data-driven state of health modeling of battery energy storage systems providing grid services," in *Proc. 11th Int. Conf. Power, Energy and Electrical Engineering (CPEEE)*, pp. 43–49, Shiga, Japan, Feb. 26-28, 2021. DOI: 10.1109/CPEEE51686.2021.9383356.

[23] L. Su, Y. Xu, and Z. Dong, "State-of-Health Estimation of Lithium-Ion Batteries: A Comprehensive Review," Energy Conversion and Economics, vol. 5, no. 4, pp. 224-242, Aug., 2024.

[24] P. Huang, P. Gu, Y. Kang, Y. Zhang, B. Duan, and C. Zhang, "The state of health estimation of lithium-ion batteries based on data-driven and model fusion method," Journal of Cleaner Production, vol. 366, Sept. 2022. DOI: https://doi.org/10.1016/j.jclepro.2022.132742.

[25] O. Matar, A. S. Alneklawy, Y. M. El-Hawary, A. M. Elbeshbeshy, A. Shoman, A. R. Alagmy, A. Mashaly, A. G. Saheen, S. S. Kaddah, and B. M. Badr, "Enhance the Design of Low-cost Fast Charging Battery Systems for Electric Mobility Systems," *Mansoura Engineering Journal*, vol. 50, no. 3, 2025. DOI:10.58491/2735-4202.3255.

[26] Y. Ethman, M. Elbastawesy, M. Emad, I. Younes, B. I. Al-Hosseny, O. Matar, F. Ahmed, S. S. Kaddah, and B. M. Badr, "Design Low-Cost Battery Management System for Low Power Applications of Photovoltaic Systems," *IEEE Conference on Power Electronics and Renewable Energy (CPERE)*, Feb. 19-21, Luxor, Egypt. DOI: 10.1109/CPERE56564.2023.10119604.

[27] I. Viñuales, "Battery Remaining Useful Life (RUL)" (dataset), Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/ignaciovinuales/battery-remaining-useful-life-ru.

[28] "Remaining Useful Life (RUL) | Partner ESI," Partner ESI Resources, 2025. [Online]. Available: https://www.partneresi.com/resources/glossary/remaining-useful-life-rul/.

[29] J. N. C. Sekhar, B. Domathoti, and E. D. R. Santibañez Gonzalez, "Prediction of Battery Remaining Useful Life Using Machine Learning Algorithms," Sustainability, vol. 15, no. 21, Oct. 2023. DOI: 10.3390/su152115283.

[30] J. VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, 2nd ed. O'Reilly Media, 2023.

[31] J. S. Kushwah, "Comparative Study of Regressor and Classifier with Decision Tree," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 3, pp. 1234–1242, 2022. DOI: https://doi.org/10.1016/j.jksuci.2021.07.657.

[32] J. Mushava, "Flexible loss functions for binary classification in gradient boosting decision trees," Applied Soft Computing, vol. 128, 2024. DOI: https://doi.org/10.1016/j.asoc.2023.109474

[33] E. Sevinç, "An Empowered AdaBoost Algorithm Implementation: A COVID-19 Dataset Study," Computers & Industrial Engineering, vol. 165, Jan. 2022. DOI: 10.1016/j.cie.2021.107912.

[34] A. Malandii and S. Uryasev, “Support Vector Regression: Risk Quadrangle Framework,” *ArXiv preprint arXiv:2212.09178*, 2022. [Online]. Available: https://arxiv.org/abs/2212.09178.

[35] P. Srisuradetchai and K. Suksrikran, "Random Kernel K-nearest Neighbors Regression," Front. Big Data, vol. 7, 2024. DOI: 10.3389/fdata.2024.1402384.