IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 May 2024, accepted 5 June 2024, date of publication 12 June 2024, date of current version 20 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3413075

RESEARCH ARTICLE

Explainable Data-Driven Digital Twins for Predicting Battery States in Electric Vehicles

JUDITH NKECHINYERE NJOKU[1], (Member, IEEE),
COSMAS IFEANYI NWAKANMA[2], (Member, IEEE),
AND DONG-SEONG KIM[1], (Senior Member, IEEE)
[1]IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do 39177, South Korea
[2]ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do 39177, South Korea

Corresponding author: Dong-Seong Kim (dskim@kumoh.ac.kr)

This work was supported in part by the Innovative Human Resource Development for Local Intellectualization Program through the Institute for Information and Communications Technology Planning grant funded by Korean Government [Ministry of Science and Information and Communication Technology (ICT)] under Grant IITP-2024-2020-0-01612, (50%); and in part by the Priority Research Centers Program through the National Research Foundation, South Korea, funded by the Ministry of Education, Science and Technology under Grant 2018R1A6A1A03024003, (50%).

ABSTRACT Advancements in battery management systems (BMS) involve using digital twins to optimize battery performance in electric vehicles. The state of charge and health estimations are essential for battery efficiency and longevity. Digital twins allow for precise predictions of the state of charge and state of health by simulating battery behavior under different conditions. Using artificial intelligence (AI) in digital twins improves predictive capabilities, as demonstrated through studies employing deep neural networks (DNN) and long short-term memory networks (LSTM). However, incorporating AI presents challenges due to the opaque nature of the models, necessitating the need for explainable artificial intelligence (XAI) and trustworthy digital twin models. This study pioneered XAI methods such as SHapley Additive exPlanations, Local Interpretable Model-agnostic Explanations, and linear regression-based surrogate models to explain the predictions of DNNs and LSTMs in digital twin-supported BMSs. The results reveal that the DNN and LSTM digital twin models are more reliable for state-of-health and state-of-charge estimation due to higher R^2 scores, lower mean residuals, and better XAI results.

INDEX TERMS Battery management systems, digital twins, artificial intelligence, XAI, explainable artificial intelligence, machine learning.

I. INTRODUCTION

The energy storage and management landscape is evolving considerably, with battery management systems (BMS) at the forefront of this change [1], [2]. These systems are crucial in managing batteries' electrical and thermal performance and essential for applications in electric vehicles, renewable energy storage, and portable electronic devices [3]. The primary function of a BMS is to guarantee the secure and efficient operation of the battery pack, including monitoring its state of charge (SoC) and state of health (SoH), both of which are critical parameters that determine the efficiency and lifespan of the battery [4]. As the complexity and demands of these systems increase, the need for more advanced management tools becomes apparent.

Digital twins (DTs) have emerged as a powerful tool in various fields, and their applications in BMS are particularly noteworthy [5], [6]. A DT is a virtual representation of a physical entity that accurately reflects its characteristics and performance. In the context of BMS, a DT can simulate and analyze the behavior of a battery under different conditions and usage patterns, enabling more precise predictions of its SoC and SoH [7]. The enhanced predictive capability of DTs is not only a technical improvement but also

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Li.

essential for optimizing the operational efficiency of batteries, extending their lifespan, and ensuring safety, especially in high-stakes applications such as electric vehicles and grid storage systems [8], [9]. Recent studies have demonstrated the effectiveness of data-driven DTs in BMS, employing advanced artificial intelligence (AI) and machine learning (ML) models. In Fonso et al [10], neural networks were explored for developing a battery DT for SoC and SoH estimation. Similarly, Schmitt et al. [11] employed recurrent autoencoder battery DTs for SoH estimation. These studies highlight the role of AI in developing optimal DTs for BMS.

However, the integration of AI and ML in DT for BMS, while enhancing predictive accuracy, introduces the challenge of the black-box nature of these models. The need for explainable artificial intelligence (XAI) becomes crucial to address this issue [12]. XAI aims to make the outcomes of AI/ML models transparent and understandable to human users in BMS. XAI can explain how and why specific predictions about SoC and SoH are made, enabling operators to trust and effectively manage these complex systems [13]. Explainability is not just a matter of user convenience but a critical requirement for diagnosing system issues, complying with safety regulations, and facilitating continuous system improvement based on actionable insights [14]. Therefore, integrating XAI techniques into BMS DTs represents a significant step forward in combining the predictive power of AI/ML models with the clarity and transparency necessary for real-world applications.

Limited research has been conducted on applying XAI in BMS. One study examined XAI approaches for learning Li-Ion batteries [13], focusing on the battery rather than state estimation. Another study developed XAI models for SoH estimation in Li-ion batteries using deep neural networks (DNNs), marking a significant milestone in this research domain [12]. Furthermore, there has been work on explainable SoC estimation for Li-ion battery systems using 1D convolutional neural networks (CNN) [15]. Another study employed a hybrid CNN and recurrent unit models for SoC estimation while exploring its explainability [16]. All these studies demonstrate the need for explainable models for state estimation in BMS. However, XAI methods have yet to be explored for DT development for BMS. Moreover, these studies only focus on single-state estimation; however, a realistic DT should be able to predict as many battery states as possible [17].

This paper seeks to enhance the current understanding of XAI by specifically examining the effectiveness of Shapley Additive exPlanations (SHAP), Local Interpretable Model-agnostic Explanations (LIME), and linear regression-based surrogate models in clarifying the predictions of AI models, particularly DNNs and long-short-term memory (LSTM) networks utilized for SoH and SoC estimation in DTs supporting BMS—most studies implementing data-driven DTs for BMS utilized models based on DNNs or LSTM. To the best of our knowledge, this constitutes the first investigation into applying these diverse XAI methodologies in the context of state estimation for DT-supported BMS. The study aims to bridge this gap in the literature, providing valuable insights for improving the efficiency and reliability of energy storage systems. The findings possess the potential to significantly contribute to sustainable development and technological advancement in an increasingly energy-conscious global environment. The proposed approach emphasizes the significance of transparency and interpretability in AI-driven systems, particularly in critical applications such as energy storage and management.

II. BACKGROUND AND RELATED WORKS

A. SOC AND SOH ESTIMATION METHODS

The SoC and SoH are essential factors in managing battery systems, as they significantly impact the efficiency, safety, and longevity of batteries used in diverse applications. This subsection offers a comprehensive background on the methodologies employed in determining these estimations.

1) SOC ESTIMATION

SoC is a critical indicator in battery management, representing the remaining energy in a battery as a percentage of its maximum capacity [30]. It serves as a fuel gauge, directly measuring the available power. Precise SoC estimation is essential for optimal battery usage and health, impacting charging strategies, utilization, and longevity. Neural networks and the various variants model the complex, non-linear relationships of battery behavior [18]. Several types of neural networks have been applied for SoC estimation, including DNNs [1], LSTMs [19], and CNNs [21], amongst others. They learn from historical data to predict SoC with improved accuracy, addressing the drawbacks of conventional methods [30]. In [18] and [20], LSTM was employed and proved robust for SoC estimation. Another study in [22] explored support vector regression (SVR) for the same task of SoC estimation. There has been a growing interest in combining various methods for improved accuracy in recent years. For instance, hybrid models incorporating machine learning with electrochemical modeling are being explored to leverage the strengths of both data-driven and physics-based approaches. Another trend is using big data and cloud computing to enhance SoC estimation. With the advent of the Internet of Things (IoT), real-time battery data can be processed in the cloud, allowing for more sophisticated analysis and prediction models.

2) SOH ESTIMATION

The SoH is a vital metric for evaluating the overall condition of a battery, providing information on its capacity and power capabilities relative to its original state [1]. This metric is crucial for making informed decisions regarding battery maintenance, replacement, and management, particularly when reliability and longevity are paramount. Accurate SoH estimation is critical for forecasting the battery's life expectancy. It is essential for optimizing its use in various

TABLE 1. Summary of AI Models employed for SoC estimation, SoH estimation, and battery digital twins.

Task	Models
SoC Estimation	DNN [18], LSTM [18]–[20], CNN [21], SVR [22]
SoH Estimation	GPR [23], SVM [24]–[26], DNN [23], LSTM [26], FNN [26], RBF [26], RF [23], [27]
Data-Driven DTs for BMS	LSTM [28], LR [29], XGBoost [17], Reccurrent Auto Encoder [11], DNN [10]

applications, including electric vehicles and large-scale energy storage systems [3]. In recent years, there has been a growing trend towards utilizing data-driven approaches to enhance the accuracy of SoH estimation in batteries. One such approach that has gained prominence is a pipeline, which employs a combination of feature engineering, feature selection, and data augmentation, utilizing methods such as random forest (RF), DNN ensemble, Bayesian ridge regression, and Gaussian process regression [23]. This approach is particularly noteworthy for its ability to quantify uncertainty in SoH estimation, making it a valuable tool in battery health monitoring. Another notable method is a support vector machine (SVM)-based approach designed explicitly for lithium-ion batteries in electric vehicles, providing reliable SoH estimations in dynamic operating environments [24], [25], [26]. LSTMs, feedforward neural networks (FNN), Radial basis functions (RBF), and hamming networks were also employed in [26]. In [31], artificial neural networks were utilized to determine the capacity fade estimates in Li-ion batteries. Additionally, ensemble learning techniques and RF regression have proven to be effective in achieving real-time and robust SoH estimation, highlighting the potential of these ML models in this field [27].

Overall, these advancements represent a significant step forward in battery health monitoring. They leverage the power of data analytics and machine learning to provide deeper, more accurate insights into battery health.

B. DIGITAL TWINS IN BMS

The utilization of DTs for BMS has been the subject of numerous studies, many of which incorporate cloud platforms and AI or ML algorithms. However, there is also a significant focus on ML-exclusive approaches [32]. These studies typically involve ML algorithms learning to model battery behavior based on historical data, highlighting the versatility and efficacy of ML in battery management.

For instance, one study employed an LSTM model to understand the relationship between battery voltage and SoC, showing promise for ML-based DT in dynamic operating environments [28]. Another research project developed a Linear Regression (LR) model to map the relationship between various states of the battery and its voltage, demonstrating the utility of ML in creating accurate and reliable battery DTs [29].

Further, a novel approach combined LSTM with an Extended Kalman Filter in a hybrid model, enhancing the capability to accurately relate battery parameters to their state [19]. Additionally, a study utilized an Extreme Gradient Boost (XGBoost) model, a powerful ML technique, for developing a DT model. This model shows the potential of ML in providing sophisticated predictions of battery states based on both historical and real-time data [17]. These examples highlight the growing trend of leveraging ML for battery DTs, emphasizing its role in advancing state prediction and the overall management of battery systems. However, these studies have yet to consider the ML models' trustworthiness in developing their DTs. Table 1 summarizes the surveyed AI models employed for SoC estimation, SoH estimation, and battery DTs. This summary shows two models that have been prevalent across the three applications: SoC, SoH estimation, and data-driven DTs for BMS. These models include the LSTM and DNN models. This motivated the use of these models in the experiments conducted in this study.

C. OVERVIEW OF EXPLAINABLE AI (XAI)

XAI has become a significant area of study within AI and ML in response to the growing demand for transparency and comprehensibility in AI systems. The rise of XAI is directly tied to the increasing use of AI in critical domains such as healthcare, finance, and autonomous systems. In these fields, it is essential to understand the reasoning behind AI decisions and the decisions themselves [33]. The early stages of AI were characterized by rule-based systems that were inherently interpretable. However, the shift towards more sophisticated models, particularly DNNs, for improved accuracy resulted in the emergence of the "black-box" phenomenon, where the reasoning behind AI decisions is not easily discernible [34].

Comprehending the intricacies of XAI necessitates distinguishing between interpretability and explainability. According to [35], interpretability refers to the capacity to infer cause and effect, while explainability pertains to providing justifications for decisions. Although distinct, both concepts are essential to the essence of XAI, addressing different aspects of AI transparency. A plethora of methods have been developed to promote explainability in AI systems. Model-agnostic techniques, such as LIME [34] and SHAP [36], offer insights into AI behavior, irrespective of model architecture. On the other hand, model-specific methods are tailored for particular model types, like neural networks, and include tools such as attention mechanisms.

The field of XAI also encompasses local and global dimensions of explainability. Local explainability focuses on explaining individual predictions [34], providing detailed insights into specific cases. On the other hand, global explainability seeks to understand a model's overall behavior and decision patterns [33], offering a broader perspective on AI functionality. The choice between local and global approaches depends upon stakeholders' specific needs and the context of the AI application. In practical applications, XAI has proven to be a vital component across diverse sectors. For instance, XAI is instrumental in explaining predictive models utilized in patient diagnosis, enhancing confidence among medical professionals [37]. In the financial sector, XAI ensures transparency and regulatory compliance in activities like credit scoring and fraud detection [38]. Additionally, in the burgeoning field of autonomous vehicles, XAI is essential for interpreting the decisions made by AI systems, a key factor for ensuring safety and adherence to regulatory standards [39].

D. XAI IN BMS AND DT

The integration of XAI within BMS has yet to gain significant attention but is a crucial element for enhancing the dependability and performance of battery operations. Incorporating XAI in BMS aims to increase the clarity and comprehensibility of intricate AI and ML models for tasks including SoC and SoH estimation. Here are some reasons why XAI is needed in BMSs [13].

1) Enhancing trust and interpretability: A key objective of XAI within BMS is to improve the interpretability and reliability of AI models for users, engineers, and stakeholders [12]. This is especially important when comprehending the rationale behind a model's projections, which is just as vital as those in safety-critical applications in electric vehicles or large-scale energy storage systems.
2) Enhancing decision-making: Expanding on the previous point, by elucidating the decision-making processes of AI models, XAI enables operators to comprehend the weighting of various inputs, such as battery temperature, voltage, and usage history, in predicting SoC or SoH [15]. This understanding empowers operators to make better-informed decisions regarding battery usage, maintenance, and management.
3) Facilitating compliance and debugging: In industries where regulatory compliance is essential, XAI ensures that AI-based predictions adhere to industry standards [16]. It also assists in debugging and enhancing AI models by pinpointing and rectifying biases or inaccuracies in their predictions.
4) Integration with digital twins: XAI can be integrated with DTs in BMS, enhancing the transparency and justifiability of DT simulations and predictions. This integration is vital for real-time monitoring and predictive maintenance, as understanding the basis of predictions leads to more effective interventions and optimizations. According to [40], one core component of DTs is the *update module*, which requires sophisticated and explainable AI models to update the DT with the most accurate predictions consistently. Another study explored XAI for DTs, showing the viability and need of XAI for developing DTs and cyber-physical systems [41].

In conclusion, XAI is an indispensable tool in BMS, providing clarity and confidence in AI-driven predictions and analyses.

III. METHODOLOGY

This section discusses the methodology employed in the study. Fig. 1 illustrates the framework of the proposed XAI-based battery DT.

A. METHODOLOGIES USED FOR STATE ESTIMATION

1) DATASET DESCRIPTION

In this study, we employed the NASA battery dataset [42]. This dataset is particularly well-suited for developing predictive models for battery SoC and SoH, which are crucial components in creating DTs for BMSs. The dataset was sourced from the Prognostics Center of Excellence at NASA Ames Research Center, and it comprises a range of experimental datasets, each representing a distinct battery subjected to various charge, discharge, and impedance cycles. The dataset includes vital measurements from lithium-ion batteries, such as voltage, current, temperature, and charge/discharge cycles. These are essential for analyzing and predicting battery behavior concerning SoC and SoH. We selected all batteries from the dataset to train the predictive models. The use of all the batteries was motivated by the need for a DT model that provides predictive analytics on new battery systems under various conditions. This approach helped validate the robustness and accuracy of the proposed DT models in real-world scenarios. Fig. 2 represents the variations in the temperature, current, and voltage data of all four batteries in the dataset using violin plots. These variations are crucial to ensuring a robust DT model.

2) DATA PRE-PROCESSING

In this study, we implemented a series of data preprocessing and feature engineering measures to guarantee the reliability of the data obtained from the NASA battery dataset [42]. Initially, we conducted an extensive cleaning process, where any missing or irrelevant data points were identified and handled to ensure the consistency and dependability of the dataset. As battery performance characteristics are time-sensitive, it was essential to align data points with their corresponding time stamps. This alignment facilitated the examination of temporal patterns and trends in battery behavior. Since the data did not include SoH and SoC estimates, we conducted feature engineering to derive these estimates. Given that V_{max} is the Maximum voltage measured for the battery, V_{min} is the Minimum voltage measured for the battery, and V represents the Current-voltage measurement of

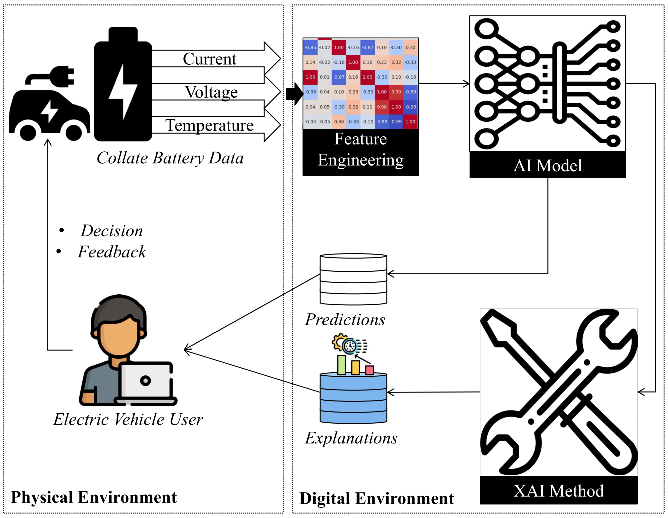
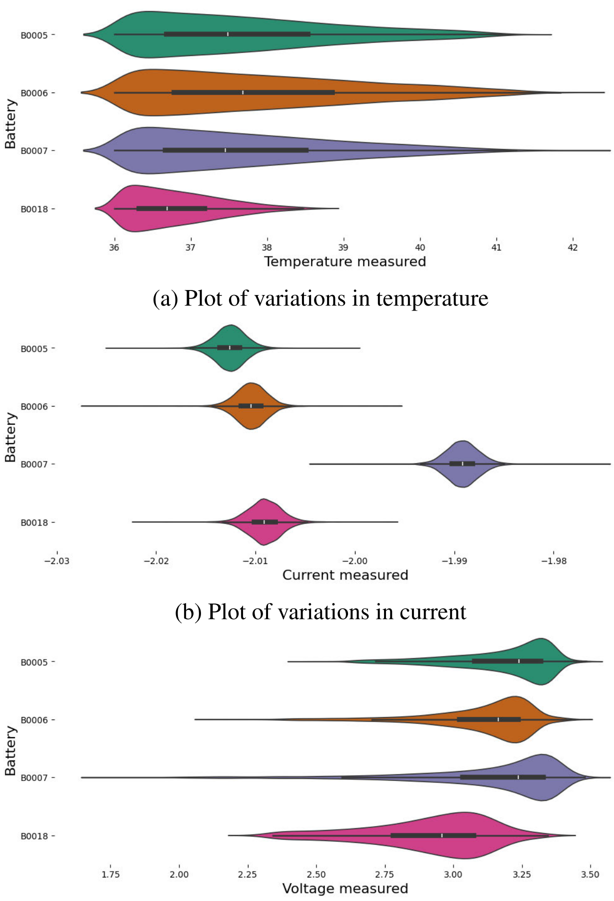


FIGURE 1. An illustration of the explainable battery digital twins framework.



(a) Plot of variations in temperature

(b) Plot of variations in current

(c) Plot of variations in voltage

FIGURE 2. Violin plots visualizing the temperature, current, and voltage of all batteries in the NASA dataset.

the battery, The formula for the estimated SoC is:

$$SoC = \left(\frac{V - V_{min}}{V_{max} - V_{min}}\right) \times 100. \tag{1}$$

Similarly, given that C is the Current capacity measurement of the battery, and C_i is the Initial capacity observed for the battery, the Capacity Fade, C_f is calculated as:

$$C_f = C_i - C. \tag{2}$$

Then, the formula for the estimated SoH is:

$$\text{SoH} = \left(\frac{C}{C_i}\right) \times 100. \tag{3}$$

Other features generated include:

i. **Discharge time**: The maximum time value within each charging cycle.
ii. **Average discharge current**: The mean of the current measured during each discharge cycle.
iii. **Resistance increase**: The ratio of *capacity fade* to the maximum capacity.
iv. **Temperature variations**: The standard deviation of the temperature measurements for each battery.

We conducted a correlation analysis to identify and select features that demonstrated the strongest correlation with the SoC and SoH. This analysis helped in reducing the feature space, thereby enhancing the efficiency of the models. These features were used to develop the ML models before conducting XAI. The entire system design is illustrated in Fig. 3.

B. MODELS

Two models that seemed prevalent in the development of model-based DT from the literature review were selected for use in developing the ML-based DTs for SoC and SoH estimation. These models include (1) DNN and (2) LSTM.

1) DEEP NEURAL NETWORK

Our study employed a DNN model designed using the Keras Sequential API to estimate SoC and SoH. The

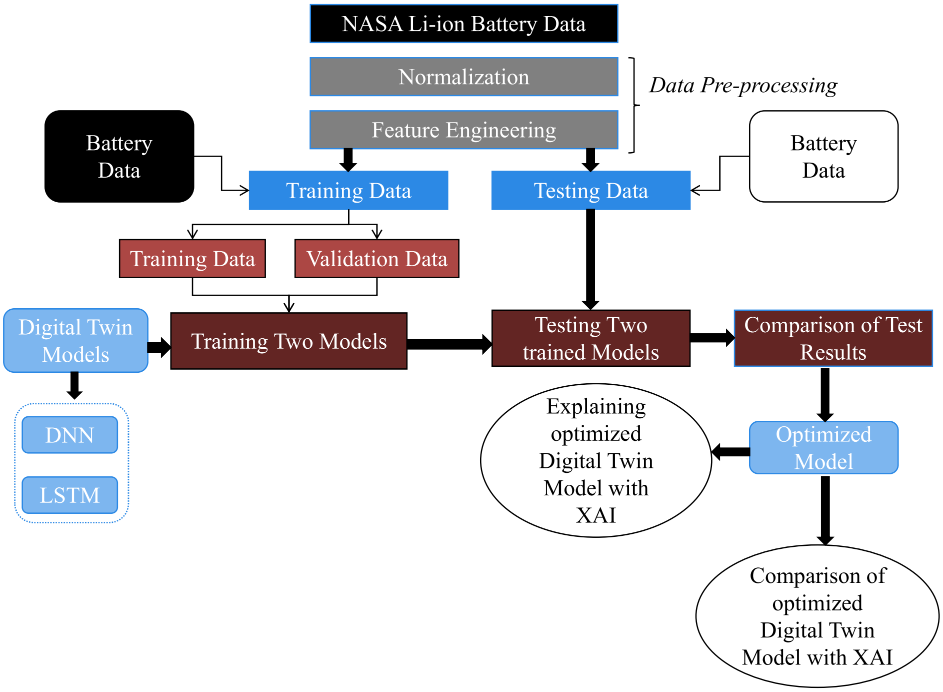


FIGURE 3. Overall design of the study for XAI and digital twin-based state estimation of electric vehicles.

model's architecture consists of three layers: the first two are dense layers with 64 neurons each, employing the rectified linear unit (ReLU) activation functions to capture non-linear relationships in the data. The final layer is a single-neuron output layer tailored for regression tasks, indicative of its application in continuous variable prediction such as SoC and SoH estimations.

2) LONG SHORT-TERM MEMORY NETWORK

In this study, we developed an LSTM model for the estimation of SoC and SoH in BMSs. The model architecture, constructed using the Keras Sequential API, comprises the following layers:

i. LSTM Layer: The first layer is an LSTM layer with 64 units, which is particularly efficient in processing time-series data, capturing temporal dependencies and patterns crucial for understanding battery behavior. The activation function for this layer is ReLU, and it is designed to process input data with a specified number of time steps and a single feature per time step.

ii. Dense Layer: Following the LSTM layer, there is a dense layer with 64 neurons, also utilizing the ReLU activation function. This layer serves to further process and interpret the features extracted by the LSTM layer.

iii. Output Layer: The final layer in the model is a single-neuron dense layer. This structure is typical for regression models, where the output is a continuous value, aligning with the requirements for SoC and SoH estimation.

This model, with its LSTM-based architecture, is particularly well-suited for the sequential and time-sensitive nature of battery data, providing a robust tool for accurate state estimation in the context of battery health management [43].

The DNN and LSTM models were compiled using the Adam optimizer, which is renowned for its adaptive learning rate capabilities, making it highly effective for deep learning models. The loss function for both models is configured to be the Mean Squared Error (MSE), represented by the formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{4}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

Additionally, the models employ the Mean Absolute Error (MAE) as an auxiliary metric to evaluate performance. MAE is expressed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{5}$$

In these formulas, $|y_i - \hat{y}_i|$ denotes the absolute difference between the predicted and the actual values, providing a straightforward interpretation of the average error magnitude across all predictions.

C. EXPLAINABLE ARTIFICIAL INTELLIGENCE METHODS

This study employed three main XAI approaches including (1) SHAP, (2) LIME, and (3) a Surrogate approach.

1) SHAP APPROACH

SHAP provides a unified framework for interpreting the predictions of ML models [44]. In this study, we employed SHAP to analyze and interpret the complex neural network models developed for SoC and SoH estimations.

SHAP values were utilized to quantify and visualize the contribution of individual features to the model's predictions, thereby enhancing our understanding of which features played pivotal roles in predicting the battery's state. The process and implications of SHAP as discussed in studies [36] and [44] are described as follows:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_x(S \cup \{i\}) - f_x(S)]. \quad (6)$$

where ϕ_i represents the SHAP value for feature i, F is the set of all features, S is a subset of features excluding i, $f_x(S \cup \{i\})$ and $f_x(S)$ are the model outputs with and without the feature i respectively, and $|S|$ denotes the number of features in subset S. This formula calculates the average marginal contribution of feature i across all possible combinations of features.

Fig. 4 illustrates how the SHAP approach was applied for the DNN and LSTM for the task of explaining the SoC and SoH predictions.

SHAP for DNN: The implementation of SHAP for DNN involved the following steps:

i. A subset of the training data (X_{train}) was selected using SHAP's sampling method to create a background dataset for the SHAP explainer.

ii. A SHAP Kernel Explainer was created using the model's predict function and the background dataset.

iii. SHAP values were computed for a sample of the test data (X_{test}).

iv. These SHAP values were then visualized using SHAP's summary plot function to illustrate the feature contributions.

SHAP for LSTM: The implementation of SHAP for LSTM involved the following steps:

i. Due to the LSTM layer in the model, a custom prediction wrapper function was created. This function reshaped the input data to a 3D format suitable for LSTM, performed prediction, and reshaped predictions to a 2D format for compatibility with SHAP.

ii. A subset of the training data (X_{train}) was selected using SHAP's sampling method to create a background dataset for the SHAP explainer. This background was reshaped to align with the LSTM input requirements.

iii. A SHAP Kernel Explainer was instantiated using the custom LSTM prediction function and the reshaped background dataset.

iv. SHAP values were computed for a sample of the test data (X_{test}), which was also reshaped to match the LSTM's input format.

v. To enhance interpretability, feature names were adjusted to reflect their respective time steps in the LSTM model.

vi. SHAP's summary plot was employed to visualize and analyze the contribution of individual features to the predictions, using the modified feature names.

Furthermore, the mean absolute SHAP values for each feature are calculated using the formula:

$$\bar{S} = \frac{1}{n} \sum_{i=1}^{n} |S_i|, \quad (7)$$

where $\bar{S}$ represents the Mean Absolute SHAP Values, S_i denotes the SHAP value for the i^{th} feature, and n is the number of features.

The overall decision impact ratio, representing the average impact of the features, was then computed as:

$$\text{Overall Decision Impact Ratio (ODIR)} = \frac{1}{n} \sum_{i=1}^{n} \bar{S}_i \quad (8)$$

where $\bar{S}_i$ is the mean absolute SHAP value of the i^{th} feature.

In addition, the confidence scores for the model's predictions are computed as follows:

$$C = \max(p_k), \quad (9)$$

where C represents the Confidence Scores and p_k is the predicted probability for each class k.

The average confidence score across all predictions was calculated using:

$$\text{Average Confidence Score} = \frac{1}{m} \sum_{j=1}^{m} C_j, \quad (10)$$

where C_j is the confidence score for the j^{th} prediction and m is the total number of predictions.

2) LIME APPROACH

In this study, LIME was utilized to explain individual predictions made by the models developed for SoC and SoH estimation [45]. LIME's capability to provide model-agnostic explanations is particularly beneficial for detailed insights into specific decisions of the model [46]. As discussed by studies [45] and [46], LIME functions by approximating the complex model locally with an interpretable one, as follows:

1) **Approximation process**:

$$\hat{f}(x) \approx g(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i, \quad (11)$$

where $\hat{f}(x)$ is the prediction of the complex model at instance x, $g(x)$ is the interpretable model (like a linear model), β_0 is the intercept, β_i are the coefficients, and x_i are the feature values of the instance.

2) **Feature perturbation and weighting**: LIME generates new samples by perturbing the instance's features and weighing these samples based on their proximity to the original instance. This helps in understanding how changes in feature values alter the model's predictions.

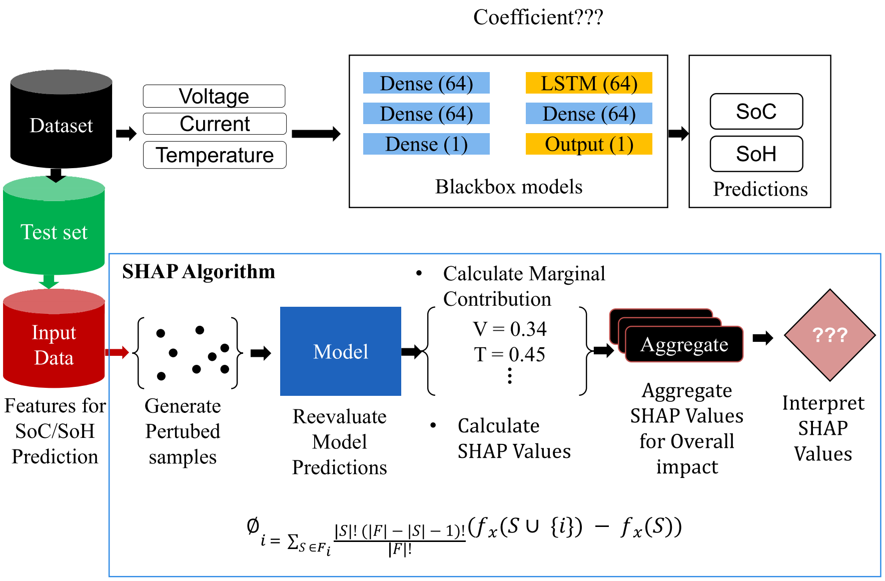

Coefficient???
Dataset
Voltage
Current
Temperature
Dense (64)
Dense (64)
Dense (1)
LSTM (64)
Dense (64)
Output (1)
Blackbox models
SoC
SoH
Predictions
Test set
Input Data
Features for SoC/SoH Prediction
SHAP Algorithm
Generate Pertubed samples
Model
Reevaluate Model Predictions
• Calculate Marginal Contribution
V = 0.34
T = 0.45
• Calculate SHAP Values
Aggregate
Aggregate SHAP Values for Overall impact
???
Interpret SHAP Values
$\emptyset_i = \sum_{S \in F_i} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!}(f_x(S \cup \{i\}) - f_x(S))$

FIGURE 4. An illustration of how SHAP works for SoC and SoH predictions.

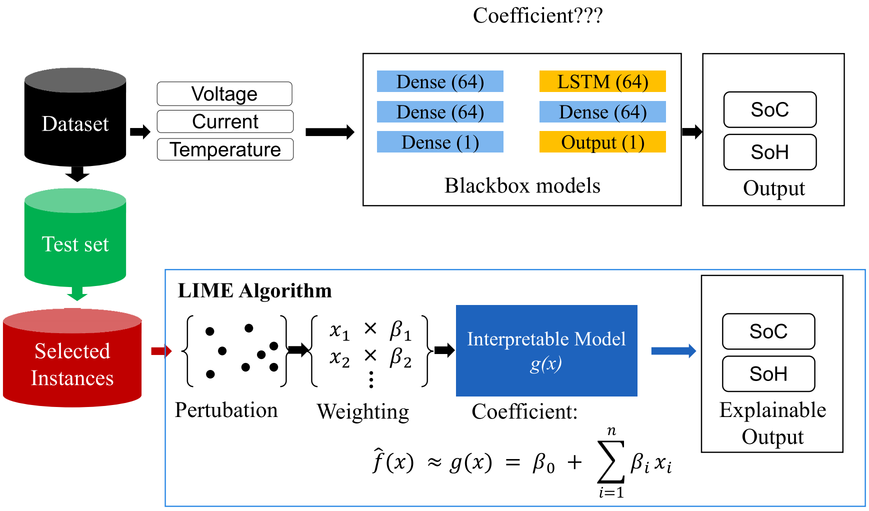


FIGURE 5. An illustration of how LIME works for SoC and SoH predictions.

3) **Interpretable model training**: LIME trains the interpretable model $g(x)$ on these weighted samples to learn the local behavior of the complex model.
4) **Feature importance extraction**:

$$\text{Feature Importance} = \beta_i \quad \text{for each feature } i \qquad (12)$$

The coefficients β_i from the interpretable model provide insights into the contribution of each feature towards the prediction for the specific instance.

Fig. 5 illustrates how the LIME approach was applied for the DNN and LSTM to explain the SoC and SoH predictions.

LIME for DNN: The implementation of LIME for DNN involved the following steps:

i. A LIME Tabular Explainer for regression was configured using the training dataset X_{train}, including the feature names and class names indicating a regression output.
ii. An individual instance from the test set, X_{test}, was selected to demonstrate LIME's ability to offer localized explanations.
iii. The prediction function, *predict_fn*, was defined to accept a 2D numpy array and return the DNN model's predictions.
iv. LIME explanations were generated for the selected instance using *predict_fn*, highlighting the influence of each feature on the model's prediction for that specific data point.

v. The explanation was visualized using a bar plot to display the impact of each feature, represented by their respective weights, on the model's prediction.

LIME for LSTM: The implementation of LIME for LSTM involved the following steps:

i. A specialized prediction function, *lstm_predict_fn*, was defined to reshape the input data into a suitable 3D format for the LSTM. This function aligns the input data with the LSTM layer's requirements and then obtains predictions from the LSTM model.
ii. Feature names were expanded to reflect the time steps in the LSTM model, ensuring proper alignment with the LSTM's temporal structure. This expansion is critical for accurate feature representation in the LIME analysis.
iii. A LIME Tabular Explainer, tailored for regression, was configured using the reshaped training data. The explainer incorporated expanded feature names and specified 'output' as an indicator of regression analysis.
iv. An instance from the test dataset, denoted as X_{test}, was selected for the explanation. This instance was appropriately reshaped to conform to the LSTM's input format.
v. The LIME explanation for the reshaped instance was generated using the *lstm_predict_fn*. This process provided insights into the influence of each feature on the model's prediction for the selected instance.
vi. The LIME explanation was visualized through a bar plot, illustrating the relationship between feature weights and their impact on the model's prediction. This visualization played a key role in interpreting the model's behavior for the chosen instance.
vii. The LIME explanation was visualized through a bar plot, illustrating the relationship between feature weights and their impact on the model's prediction. This visualization played a key role in interpreting the model's behavior for the chosen instance.
viii. The sum of absolute feature importances, derived from the LIME explanation, was calculated to quantify the overall decision impact. This measure reflects the cumulative impact of features on the model's decision-making process.

We calculated the confidence score and decision impact ratio as follows:

$$R^2_{\text{conf}} = \text{LIME Explanation's R-squared,} \tag{13}$$

$$\text{Decision Impact Ratio (DIR)} = \sum_{i=1}^{n} |w_i|, \tag{14}$$

where R^2_{conf} represents the confidence score derived from the R-squared value of the LIME explanation, and w_i denotes the weight of the i^{th} feature from the LIME explanation. The DIR is computed as the sum of the absolute values of the feature weights provided by the LIME explanation.

3) SURROGATE APPROACH

Surrogate models in XAI are simpler, more interpretable models used to approximate and explain the behavior of more complex, often opaque ML models [47]. The primary goal of these models is to gain insights into how the complex model makes its decisions, which is crucial in applications where understanding and trust in the model's predictions are important. Surrogate models function by learning to mimic the predictions of the complex model. They utilize the same input features but are inherently designed to be much simpler and more interpretable. Common surrogate model types include linear regression models or decision trees. The functioning of a surrogate model can be represented by the following equation [47]:

$$y_{\text{sur}} = f_{\text{sur}}(x_1, x_2, \ldots, x_n) \tag{15}$$

where y_{sur} represents the output of the surrogate model, f_{sur} is the interpretable function (e.g., linear regression), and $x_1, x_2, \ldots, x_n$ are the input features. For a linear regression surrogate model, this equation can be detailed as:

$$y_{\text{sur}} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{16}$$

Here, $\beta_0, \beta_1, \ldots, \beta_n$ represent the coefficients (weights) learned by the surrogate model, indicating the contribution of each feature towards the prediction. Once trained, the surrogate model's predictions and internal structures, such as feature weights β_i or decision paths in a decision tree, can be analyzed to infer the operation of the complex model. Fig. 6 illustrates how the surrogate approach was applied to the DNN and LSTM for the task of explaining the SoC and SoH predictions.

Surrogate for DNN and LSTM: The implementation of the surrogate approach for DNN and LSTM models involved the following steps:

i. Predictions were made on the training set, X_{train}, and test set, X_{test}, using the models.
ii. A linear regression model is chosen as the surrogate model due to its simplicity and interpretability.
iii. The surrogate model is trained using the input features from the training set and the predictions from the models as the target.
iv. After training, the surrogate model is interpreted by examining the coefficients, which indicate the importance of each feature in the prediction.
v. The surrogate model is evaluated using metrics like MSE and R^2 on the test set.
vi. The coefficients from the surrogate model are visualized in a bar plot, providing a clear representation of each feature's contribution to the model's predictions.

The surrogate model is evaluated using metrics like MSE and R^2 on the test set. R^2 is particularly important as it represents the proportion of variance in the model's predictions that is explained by the surrogate model, serving as a confidence score. The mean residual can be calculated to measure the average difference between the models' predictions and the surrogate model's predictions. The DIR can be quantified as

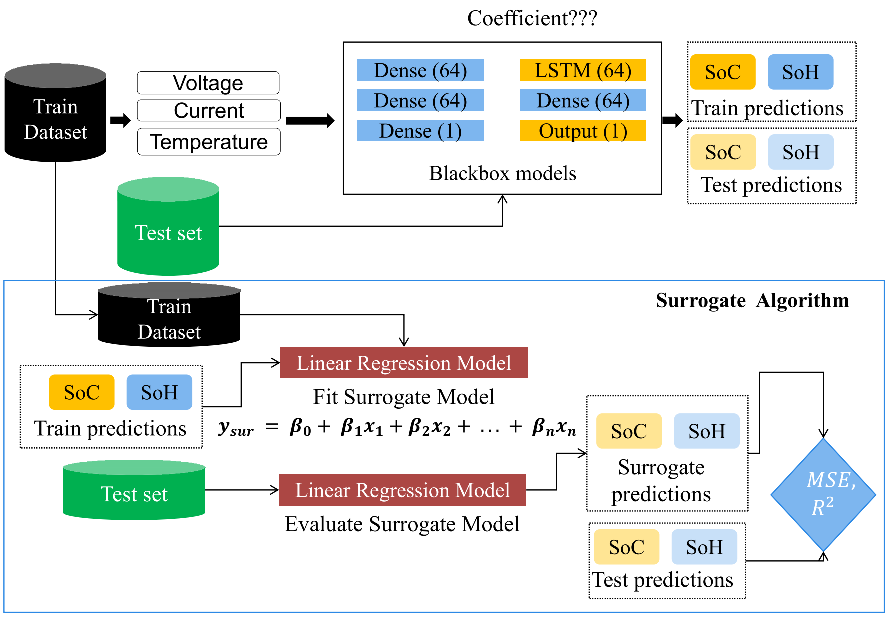


FIGURE 6. An illustration of how the Surrogate approach works for SoC and SoH predictions.

the sum of the absolute values of the coefficients, indicating the overall impact of the features.

MSE: MSE measures the average of the squares of the errors or deviations; that is, the difference between the original model's predictions and the surrogate model's predictions. It is calculated as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{17}$$

where y_i represents the actual predictions from the original model, $\hat{y}_i$ represents the predictions from the surrogate model, and n is the number of observations in the test set.

R^2: The R^2 statistic represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It serves as a measure of how well the surrogate model predictions approximate the original model's predictions, acting as a confidence score.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{18}$$

Here, $\bar{y}$ is the mean of the actual predictions from the original model.

Mean Residual: The mean residual quantifies the average difference between the predictions of the original model and the surrogate model.

$$\text{Mean Residual} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{19}$$

DIR: DIR can be quantified as the sum of the absolute values of the coefficients in the surrogate model, indicating the overall impact of the features.

TABLE 2. Experimental settings for DNN and LSTM models.

Parameter	DNN	LSTM
Number of Epochs	100	100
Batch Size	32	32
Activation Function	ReLU	ReLU
Optimizer	Adam	Adam
Loss Function	MSE	MSE
Number of Layers	3 Dense	3 (1 LSTM, 2 Dense)
Number of Neurons/Layer	64, 64, 1	64 (LSTM units)
Data-split	70:20:10	70:20:10

$$\text{DIR} = \sum_{i=1}^{m}|\beta_i| \tag{20}$$

where β_i are the coefficients of the surrogate model and m is the number of features.

These evaluations provide a comprehensive understanding of how well the surrogate model captures the behavior of the original model, thereby assessing its effectiveness as a tool for interpretability in complex AI systems.

D. IMPLEMENTATION OVERVIEW

For our experiments, we leveraged the high-performance GPU cores provided by Google Colab. Specifically, we utilized the NVIDIA Tesla K80/T4 GPUs, which significantly accelerated the computational process. These GPUs offer a substantial increase in speed over traditional CPU cores, making them ideal for the computationally intensive tasks involved in training neural networks. Table 2 highlights the experimental settings used during implementation.

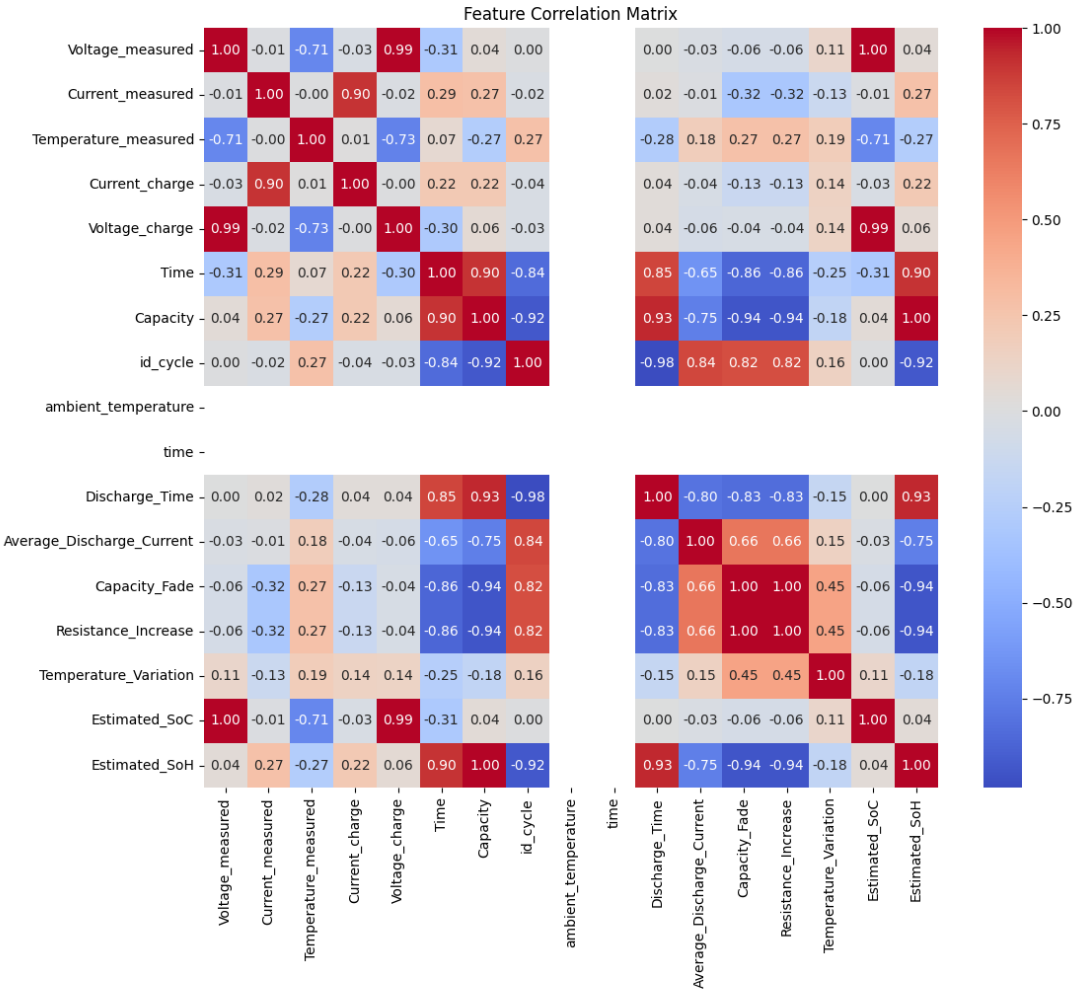


FIGURE 7. Feature correlation for SoC and SoH.

IV. EVALUATION RESULTS AND DISCUSSION

A. RESULTS FOR FEATURE SELECTION

The correlation analysis focusing on the estimated SoC and SoH revealed several significant relationships. The feature correlation matrix, as illustrated in Fig. 7, indicates the following key associations:

i. The estimated SoC exhibits a perfectly positive correlation with *Voltage_measured* (correlation coefficient $=$ 1.00), suggesting that the voltage measurement is a reliable indicator of SoC.

ii. There is a high positive correlation between the estimated SoC and *Voltage_charged* (correlation coefficient $=$ 0.99), implying that as the voltage charge increases, the SoC also increases.

iii. A notably high negative correlation exists between the estimated SoC and *Temperature_measured* (correlation coefficient $= -0.71$), implying that as the temperature increases, the estimated SoC tends to decrease.

iv. The estimated SoH shows a strong positive correlation with *Capacity* (correlation coefficient $=$ 1.00), which is intuitive as the battery's health is directly tied to its ability to hold a charge.

v. Conversely, there is a strong negative correlation between estimated SoH and *Capacity_Fade* (correlation coefficient $= -0.94$), highlighting capacity fade as a critical factor in the degradation of battery health.

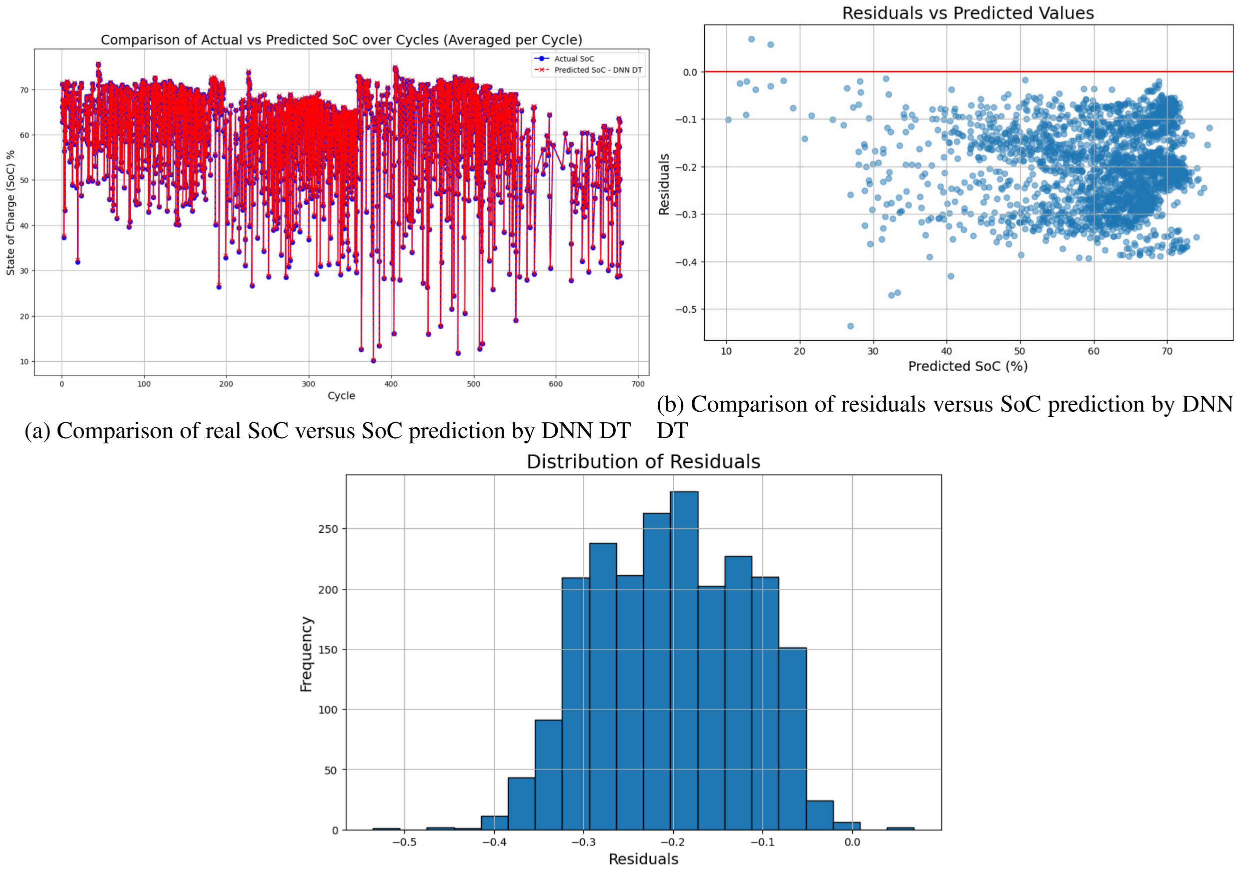
As a result, the following features were selected for SoC estimation: *voltage measured*, *voltage charge*, and *temperature measured*. The features selected for SoH included: *capacity*, *discharge time*, *time*, *average discharge current*, *resistance increase*, and *capacity fade*.

B. RESULTS FOR SOC ESTIMATION

The results for SoC estimation will be discussed in line with the following: (1) Model results; (2) SHAP XAI results; (3) LIME XAI results; and (4) Surrogate XAI results.

1) SOC ESTIMATION RESULTS FOR DNN DT

i. Model Results: The prediction results of the DNN-based DT for SoC estimation are illustrated in Fig. 8a. Both the actual and predicted SoC values exhibit similar trends across the cycles. This suggests that the DNN DT model can capture the overall pattern of SoC changes throughout the battery's life cycle. There are noticeable



(a) Comparison of real SoC versus SoC prediction by DNN DT

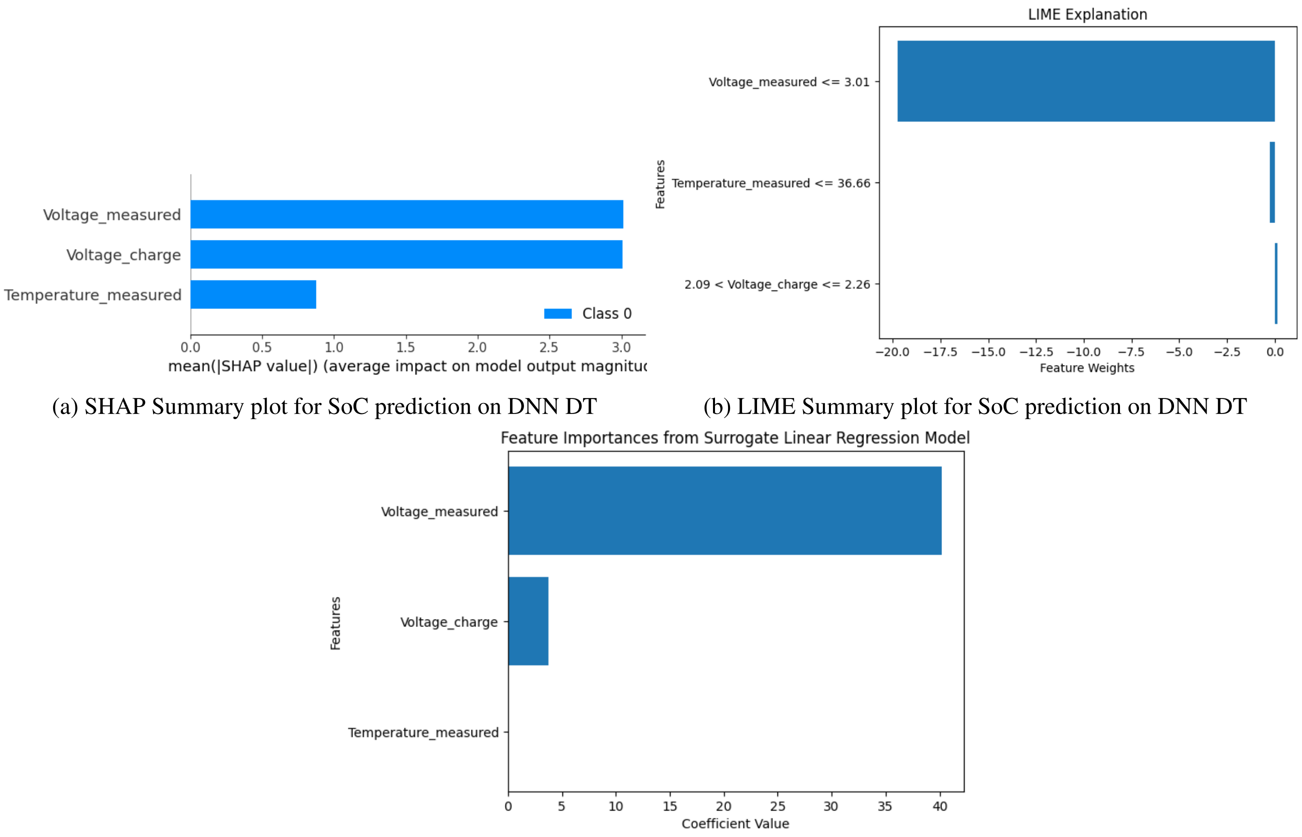
(b) Comparison of residuals versus SoC prediction by DNN DT

(c) Distribution of residuals for SoC prediction DNN DT

FIGURE 8. Visualizations of SoC Predictions and Residuals using DNN DT.

fluctuations in SoC values across the cycles, which are common in battery data due to various factors like charging and discharging rates, temperature, and aging effects. The DNN DT model appears to closely follow the actual SoC data points, with the red crosses frequently overlapping with the blue dots. This indicates a high level of accuracy in the model's predictions. The MSE and MAE results of 0.047774 and 0.201637 were obtained. Residuals are the differences between the actual SoC values and the predicted SoC values by the model. A residual of zero indicates a perfect prediction, while positive or negative residuals indicate overestimations and underestimations, respectively. Fig. 8b compares the residuals with the predictions by the DNN DT model. The residuals appear to be fairly evenly distributed around the zero line, which suggests that there isn't a systematic error in the predictions. There is no clear pattern or trend in the residuals as the predicted SoC increases, implying that the model's errors are not dependent on the magnitude of the predicted value, which is a good sign of model reliability. Fig. 8c is a histogram of the distribution of residuals for the SoC predictions. The distribution of residuals is centered around zero, which indicates that the model does not have a consistent bias toward over- or under-predicting. The shape of the histogram resembles a normal distribution, suggesting that the errors are randomly distributed, which is a characteristic of a well-fitted model.

ii. SHAP XAI results: The SHAP summary plot in Fig. 9a, shows the average impact of three features on the output of the DNN model for SoC prediction. The length of the bars represents the mean absolute SHAP value for each feature, which quantifies the average impact of the features on the model's output magnitude. The *Voltage_measured* feature has the highest mean absolute SHAP value, indicating it has the most significant impact on the DNN model's SoC predictions. The average impact is above 3.0, suggesting that changes in the measured voltage are strongly associated with changes in the SoC. The second most influential feature is the *Voltage_charge*, with a mean absolute SHAP value of around 3.0. This suggests that the voltage during charging is also an important predictor of SoC, albeit less so than the measured voltage. The *Temperature_measured* during operation has the least



(a) SHAP Summary plot for SoC prediction on DNN DT

(b) LIME Summary plot for SoC prediction on DNN DT

(c) Surrogate Summary plot for SoC prediction on DNN DT

FIGURE 9. **Interpretability plots for SoC prediction on DNN DT.**

impact among the three features, but it still contributes noticeably to the model's predictions, with a mean absolute SHAP value of just under 1.0. This indicates that temperature plays a role in the SoC prediction, but its influence is smaller than the voltages. These SHAP values help us understand which features the DNN model relies on most when estimating the SoC and how much each feature drives the prediction. This kind of interpretability is crucial for validating the model's behavior and ensuring that it makes decisions based on sensible and understandable factors, especially in critical applications like BMSs.

iii. LIME XAI results: The LIME result displayed in Fig. 9b shows the feature weights for three features: *Voltage_charge*, *Voltage_measured*, and *Temperature_measured*. These weights represent the contribution of each feature to a specific prediction made by the model. The feature *Voltage_measured* has the most significant impact, with a negative weight of approximately -17.5, indicating a strong influence on the model's prediction for the particular instance being explained. The condition *Voltage_measured* $\leq$ 3.01 highlights the specific threshold below which this feature notably affects the model's output.

In contrast, *Temperature_measured* weights -20, making it the feature with the highest weight, contradicting the initial statement. This suggests that temperatures below 36.66°C are crucial in influencing the model's decision. The feature *Voltage_charge* exhibits a slightly positive influence with its weight close to 2.5, contained within the range 2.09 < *Voltage_charge* $\leq$ 2.26. This indicates that values within this specific range moderately impact the model's output.

Overall, the weights indicate that both *Temperature_measured* and *Voltage_measured* significantly drive the prediction, albeit in a negative direction, while *Voltage_charge* has a lesser, yet positive impact. This analysis helps in understanding how different conditions of these features affect the model's predictions.

iv. Surrogate XAI results: Fig. 9c illustrates the feature importance derived from the surrogate linear regression model used to approximate the DNN-DT model for the SoC prediction. The surrogate model coefficients indicate the strength and direction of the relationship between each feature and the predicted SoC. The *Voltage_measured* feature has the highest positive coefficient value, suggesting that it strongly influences the SoC prediction. The larger the value of *Voltage_measured*, the higher the SoC is predicted to be by the surrogate model. The coefficient value for *Voltage_charge* is positive but less than that of *Voltage_measured*, indicating that it also contributes positively to the SoC prediction. Still, its impact is smaller compared to *Voltage_measured*.

2) SOC ESTIMATION RESULTS FOR LSTM DT

i. Model results: The prediction results of the LSTM-based DT for SoC estimation are illustrated in Fig. 10a. The LSTM-DT model predictions (indicated by red crosses) track closely with the actual SoC values (shown by blue dots). This close following suggests that the LSTM model is effectively capturing the temporal dynamics of the SoC over the cycles. There is a high degree of overlap between the predicted and actual SoC values, indicating a strong predictive accuracy of the LSTM-DT model. However, there are instances where the model overestimates or underestimates the SoC, as seen by the red crosses deviating from the blue dots. The SoC values show variability across cycles, which the LSTM-DT seems to capture well. This variability is typical in battery datasets and reflects the real-world conditions of battery usage.

   Figs. 10b and 10c represent the residuals and residuals distribution. The plot in Fig. 10b indicates that the majority of residuals are close to zero, suggesting good model accuracy, with no obvious patterns of errors as the predicted SoC changes. The distribution of the residuals is centered around zero and appears normally distributed, indicating that the model's errors are symmetrically distributed around the true values and there's no systematic bias in the predictions. The model is generally accurate, with most errors being small, but there are some cases of larger errors, as shown by the tails of the distribution.

ii. SHAP XAI results: The SHAP summary plot for the LSTM DT-model in Fig. 11a displays the mean absolute SHAP values for three features at the first timestep of a sequence used for prediction. The *Voltage_measured* feature has the highest mean absolute SHAP value, indicating it has the most significant impact on the LSTM model's SoC predictions at the first timestep. The second feature, *Voltage_charge* at the first timestep, also positively influences the SoC prediction, but its impact is less than that of *Voltage_measured*. The *Temperature_measured* at the first timestep has the smallest mean absolute SHAP value among the three, suggesting it has the least impact on the SoC prediction at this particular timestep.

iii. LIME XAI results: The LIME XAI result in Fig. 11b for the LSTM-DT model displays feature weights for three specific features at the first timestep of the input sequence. The *Voltage_measured* feature has the largest positive weight, indicating it has the strongest positive influence on the model's prediction at the first timestep. The value range provided (3.24 to 3.32) suggests that within this interval, the feature contributes significantly to the increase in the predicted SoC. The feature *Voltage_charge* at the first timestep has a moderately positive weight. Its contribution is positive but not as strong as *Voltage_measured*. The value range (2.28 to 2.36) indicates the interval where this feature's impact is most notable. Interestingly, the *Temperature_measured* at the first timestep shows a negative weight, which is quite substantial compared to the other features. This indicates that within the specified range (36.67 to 37.48), higher temperatures are associated with a decrease in the predicted SoC.

iv. Surrogate XAI results: The surrogate XAI result for the LSTM-DT model is illustrated in Fig. 11c. This result also shows the feature importance derived from the surrogate linear regression model. Similar to the DNN-DT model, the *Voltage_measured* feature has the highest positive coefficient value, suggesting that it strongly influences the SoC prediction. The feature *Voltage_charge* at the first timestep has a moderate positive coefficient value, indicating its positive but lesser impact on SoC predictions compared to *Voltage_measured*. Its importance suggests it is a relevant but not dominant factor in the LSTM DT model's decision-making process at this timestep. The *Temperature_measured* at the first timestep shows a negative coefficient value, which is substantially lower in magnitude compared to the other two. This implies that an increase in temperature is associated with a decrease in the predicted SoC, suggesting an inverse relationship.

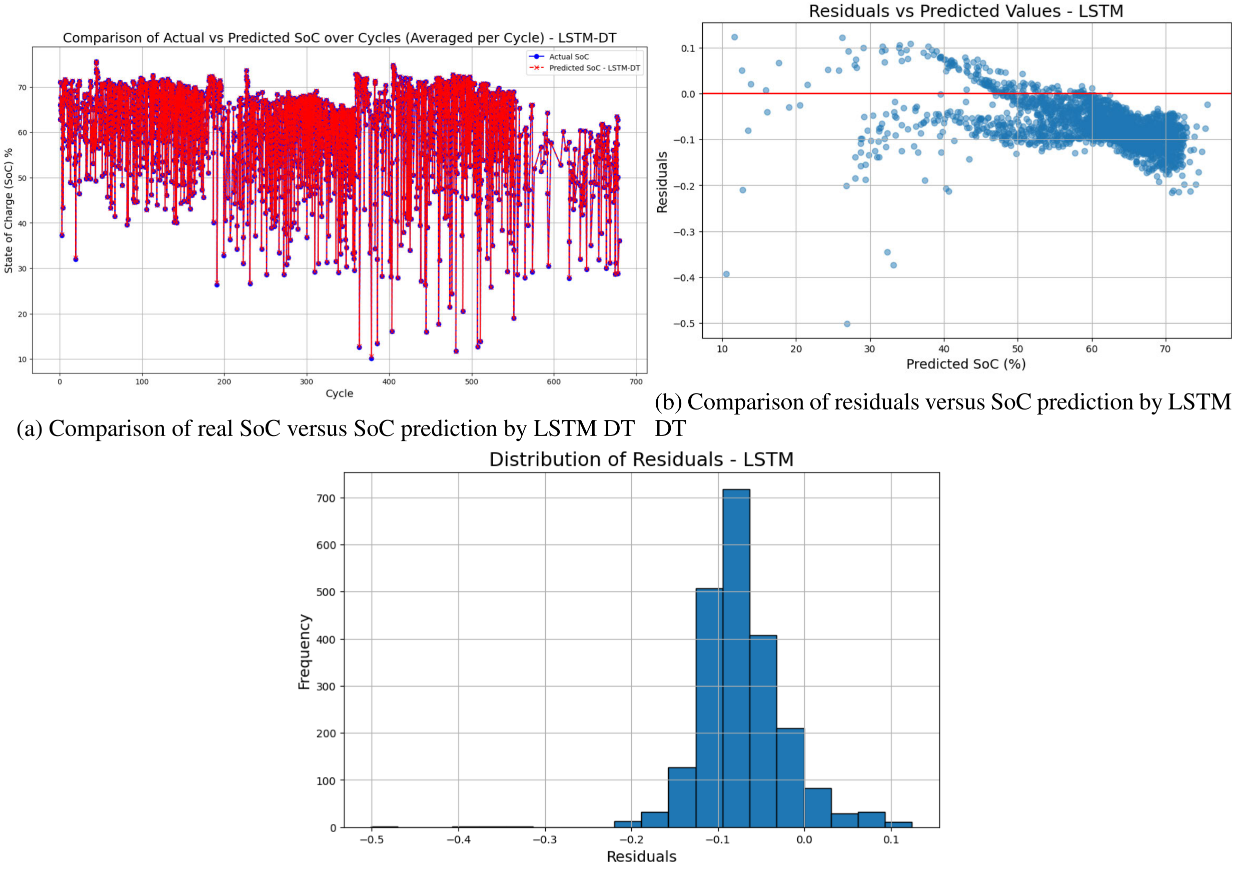
C. RESULTS FOR SOH ESTIMATION

The results for SoH estimation will be discussed in line with the following: (1) Model Results; (2) SHAP XAI results; (3) LIME XAI results; and (4) Surrogate XAI results.

1) SOH ESTIMATION RESULTS FOR DNN DT

i. Model results: The prediction results of the DNN-based DT for SoH estimation are presented in Fig. 12a. The predicted SoH values closely follow the trend of the actual SoH values, suggesting that the DNN-DT model effectively captures the degradation pattern of the battery over time. The actual and predicted SoH show a downward trend, indicating the gradual decline in battery health with each cycle, a typical behavior due to battery aging. There are noticeable peaks and troughs in the SoH values, which the DNN-DT model appears to replicate with reasonable accuracy. These could correspond to battery recovery phenomena or measurement errors in the actual data.

   The plots in Figs. 12b and 12c show the residuals of SoH estimation using a DNN-DT model. In Fig. 12b, most residuals cluster around zero, indicating generally accurate predictions. There is no clear pattern that



(a) Comparison of real SoC versus SoC prediction by LSTM DT

(b) Comparison of residuals versus SoC prediction by LSTM DT

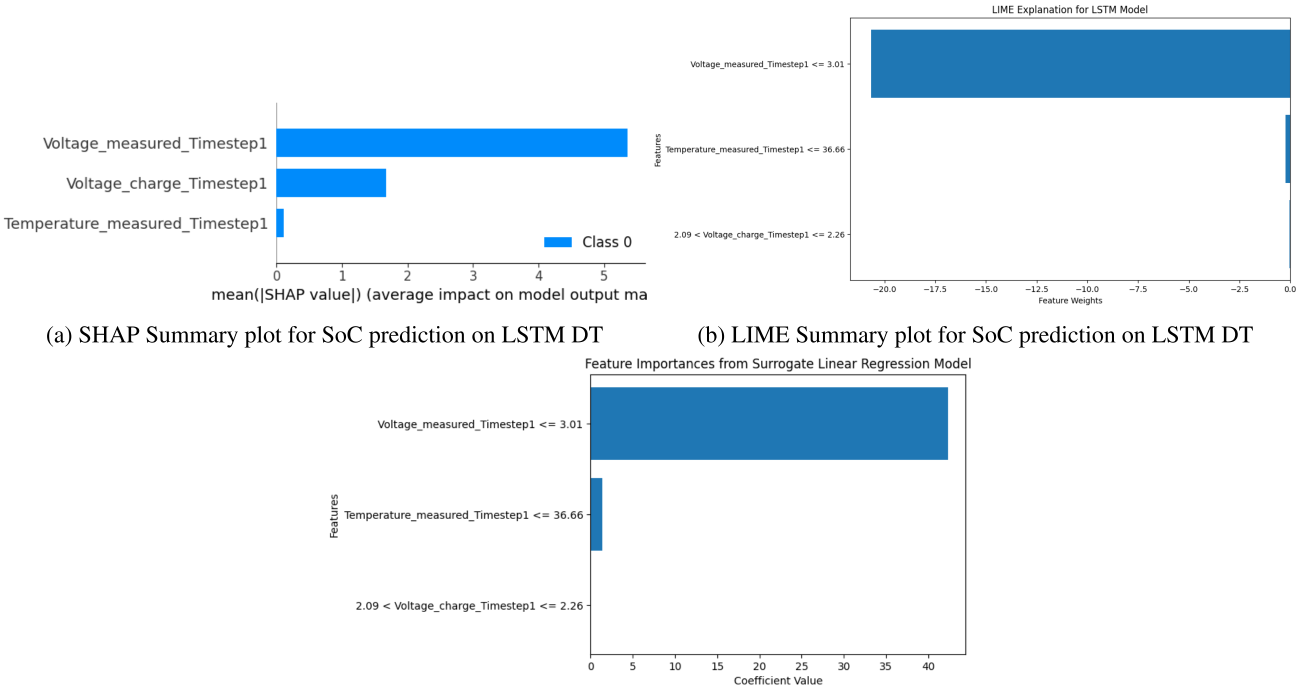
(c) Distribution of residuals for SoC prediction LSTM DT

FIGURE 10. Visualizations of SoC Predictions and Residuals using LSTM DT.

suggests systematic bias in the model, as the residuals are scattered above and below the zero line without any discernible trend. The histogram in Fig. 12c shows that the residuals are approximately normally distributed, centered around zero, which is indicative of a model making predictions with no consistent overestimation or underestimation. The spread of the residuals suggests that while many predictions are close to the actual values, there are still a fair number of predictions with larger errors, as evidenced by the distribution's tails.

ii. SHAP XAI results: The SHAP plot in Fig. 13a presents the mean absolute SHAP values for various features used in the DNN-DT for SoH prediction. These values represent the average impact of each feature on the model's output magnitude for SoH predictions. The *Discharge_Time* feature has the most substantial average impact on the model's output. This suggests that the time taken to discharge the battery is the most influential factor in predicting its health. The number of cycles (*id_cycle*) has the second-highest impact on SoH predictions, indicating that the wear and tear associated with the battery cycle count is a significant predictor of battery health. The *Time* feature also shows a considerable impact on the model's predictions, likely reflecting the total operating time of the battery. *Capacity_Fade* and *Resistance_Increase*: Both features, related to the battery's degradation over time, show a meaningful impact on SoH predictions, which aligns with expectations that battery health declines as capacity fades and internal resistance increases. The *Capacity* feature, which directly relates to the battery's ability to hold a charge, also has a noticeable impact on the SoH prediction, though not as pronounced as *Discharge_Time*. The *Average_Discharge_Current* feature has the least impact among those listed, suggesting that the average current during discharge, while relevant, is not as strong a predictor as the other features in the context of SoH estimation.

iii. LIME XAI results: The LIME result in Fig. 13b presents the feature weights, indicating how each one influences the prediction for a specific instance in the model. The feature *Discharge_Time* with a value greater than 3126.80 has the most substantial positive weight, significantly increasing the predicted SoH. Similarly, *Time* with values above 2803.52 and *id_cycle* values less



(a) SHAP Summary plot for SoC prediction on LSTM DT

(b) LIME Summary plot for SoC prediction on LSTM DT

(c) Surrogate Summary plot for SoC prediction on LSTM DT

FIGURE 11. **Interpretability plots for SoC prediction on LSTM DT.**

than or equal to 65 have considerable positive impacts on SoH predictions.

In contrast, the *Capacity* with a value greater than 1.65 has a positive, though less pronounced, impact on the predicted SoH. The features *Capacity_Fade* and *Resistance_Increase*, which are within specific ranges, show moderate negative influences on the SoH, with weights suggesting a decrease in SoH as these parameters increase within their respective ranges. The *Average_Discharge_Current*, especially when falling within a negative range, contributes negatively.

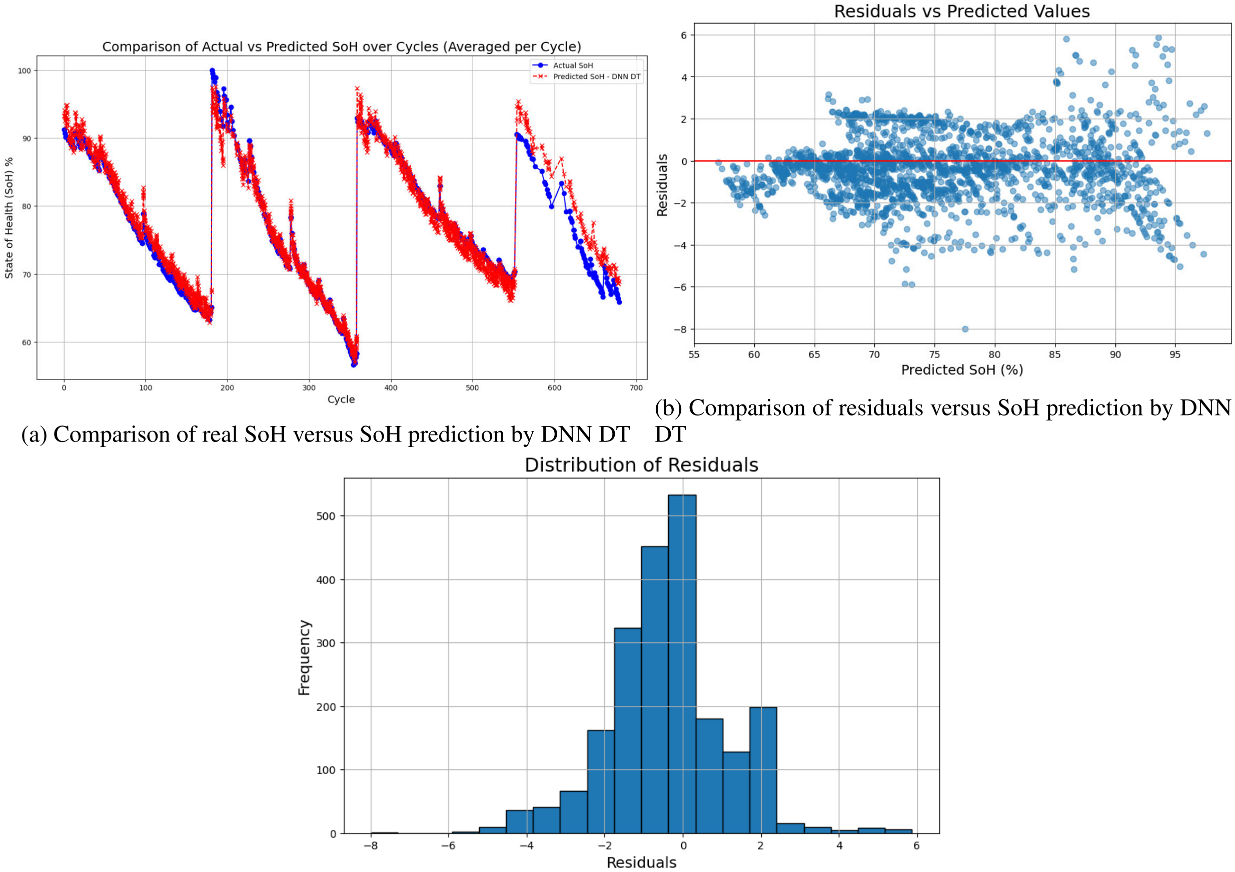
iv. Surrogate XAI results: The plot in Fig. 13c presents feature importances from the surrogate linear regression model approximating the DNN-DT for SoH prediction. The coefficients represent the strength and direction of each feature's relationship with the SoH. Contrary to the SHAP and LIME assessments, the feature *Average_Discharge_Current* has a substantial negative coefficient, indicating that increases in this variable are associated with lower SoH predictions by the surrogate model. Similarly, *id_cycle* also exhibits a negative coefficient, further suggesting that higher values in this feature are linked with decreased SoH predictions.

Conversely, the *Resistance_Increase* and *Capacity_Fade* features, which were previously understood to have negative impacts, are indeed confirmed as such with their negative coefficients, reinforcing their association with lower SoH predictions. Unexpectedly, the features *Capacity*, *Discharge_Time*, and *Time*, previously noted to have positive impacts, are negatively correlated with SoH predictions, as evidenced by their negative coefficients.

2) SOH ESTIMATION RESULTS FOR LSTM DT

i. Model Results: The prediction results of the LSTM-based DT for SoH estimation are presented in Fig. 14a. This plot depicts the actual versus predicted SoH values over various battery cycles. The solid blue line represents the actual SoH values of Battery B0005, while the red crosses indicate the SoH as predicted by the LSTM-DT model. The close overlap between the two indicates that the LSTM-DT model can capture the SoH trend across cycles effectively, although there are occasional discrepancies where the model's predictions deviate from the actual values.

The plots in Figs. 14b and 14c show the residuals of SoH estimation using the LSTM-DT model. Ideally, the residuals should be randomly distributed around zero (indicated by the red line), which would suggest that the model's errors are random in Fig. 14b. While many residuals cluster around zero, indicating accurate predictions, there's a spread across the spectrum of SoH values, which suggests some systematic errors in the predictions. Fig. 14c shows the distribution of the residuals. A normal distribution of residuals, centered around zero, would suggest a well-fitting model. Here, we see a bell-shaped distribution with a center around zero, which is a good sign. However, there is a tail



(a) Comparison of real SoH versus SoH prediction by DNN DT

(b) Comparison of residuals versus SoH prediction by DNN DT

(c) Distribution of residuals for SoH prediction DNN DT

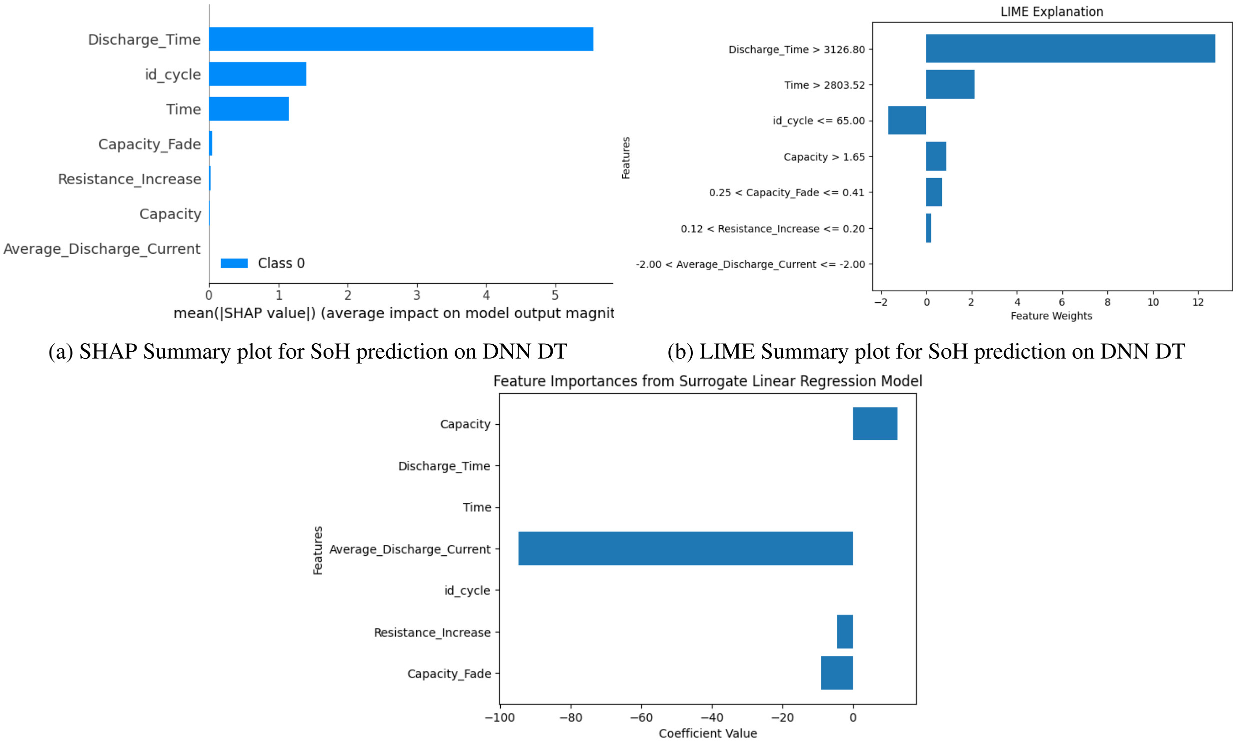
FIGURE 12. **Visualizations of SoH Predictions and Residuals using DNN DT.**

on both sides of the distribution, which points to some predictions with significant errors.

ii. SHAP XAI results: The SHAP plot in Fig. 15a reveals the importance of various features at a given timestep in the LSTM model's prediction of the SoH. It illustrates the mean absolute SHAP values, which are indicative of the average impact of each feature on the model's output. The *Discharge_Time* feature exhibits the highest mean absolute SHAP value, signifying its paramount importance in the SoH estimation at the initial timestep. *Time* and *id_cycle* also contribute significantly to the model's output, reflecting the battery's operational duration and usage frequency's relevance. Features related to battery degradation, such as *Capacity_Fade* and *Resistance_Increase*, are crucial in the model's prediction, indicating the model's sensitivity to changes in battery health over time. The *Average_Discharge_Current* holds the least impact on the SoH estimation, although it is still a factor considered by the model. These SHAP values facilitate a more transparent understanding of the predictive model, highlighting the influence of individual features on the estimated SoH.

iii. LIME XAI results: The LIME result in Fig. 15b offers insight into the LSTM model's predictions at Timestep 1. This analysis reveals significant implications of each feature's weight on the SoH prediction. Notably, the *Discharge_Time_Timestep1* with a value greater than 3126.80 shows a large positive weight, indicating a strong positive relationship with SoH predictions, contrary to previously suggested inverse relationships.

The *Time_Timestep1* feature, when greater than 2803.52, also demonstrates a positive impact, although less significant than *Discharge_Time*, suggesting that longer operational times slightly enhance SoH predictions. Surprisingly, *id_cycle_Timestep1* with values less than or equal to 65, exhibits a minor positive effect, revising earlier interpretations of a negative correlation. Moreover, *Capacity_Timestep1* exceeding 1.65 and *Resistance_Increase_Timestep1* within the range of 0.12 to 0.20 display moderate negative impacts on SoH. These weights imply that increases in these values could deteriorate SoH. The *Capacity_Fade_Timestep1* between 0.25 and 0.41, and particularly *Average_Discharge_Current_Timestep1* within the range −2.00 to −2.00, present the most substantial negative weights, significantly driving down SoH predictions.



(a) SHAP Summary plot for SoH prediction on DNN DT

(b) LIME Summary plot for SoH prediction on DNN DT

(c) Surrogate Summary plot for SoH prediction on DNN DT

FIGURE 13. Interpretability plots for SoH prediction on DNN DT.

iv. Surrogate XAI results: The plot in Fig. 13c presents feature importances from the surrogate linear regression model approximating the LSTM-DT model behavior at Timestep 1. The *Discharge_Time_Timestep1* with values greater than 3126.80 shows a significant positive coefficient, indicating a strong and direct influence on SoH predictions. Similarly, *Time_Timestep1* with values above 2803.52 also displays a moderately positive coefficient, suggesting a beneficial impact on SoH.
Conversely, *id_cycle_Timestep1* with values less than or equal to 65 has a slight negative influence, contrary to the earlier interpretation of a positive effect. The features *Capacity_Timestep1*, with values greater than 1.65, and *Resistance_Increase_Timestep1*, within a specific range, both exhibit negative coefficients, which underscores their detrimental effect on SoH. *Capacity_Fade_Timestep1*, between specific values, also shows a negative relationship with SoH, confirming its adverse impact.
Notably, the *Average_Discharge_Current_Timestep1*, particularly when falling within a specific negative range, has a significant negative coefficient, highlighting that higher discharge currents substantially decrease SoH predictions.

3) SUMMARY OF RESULTS

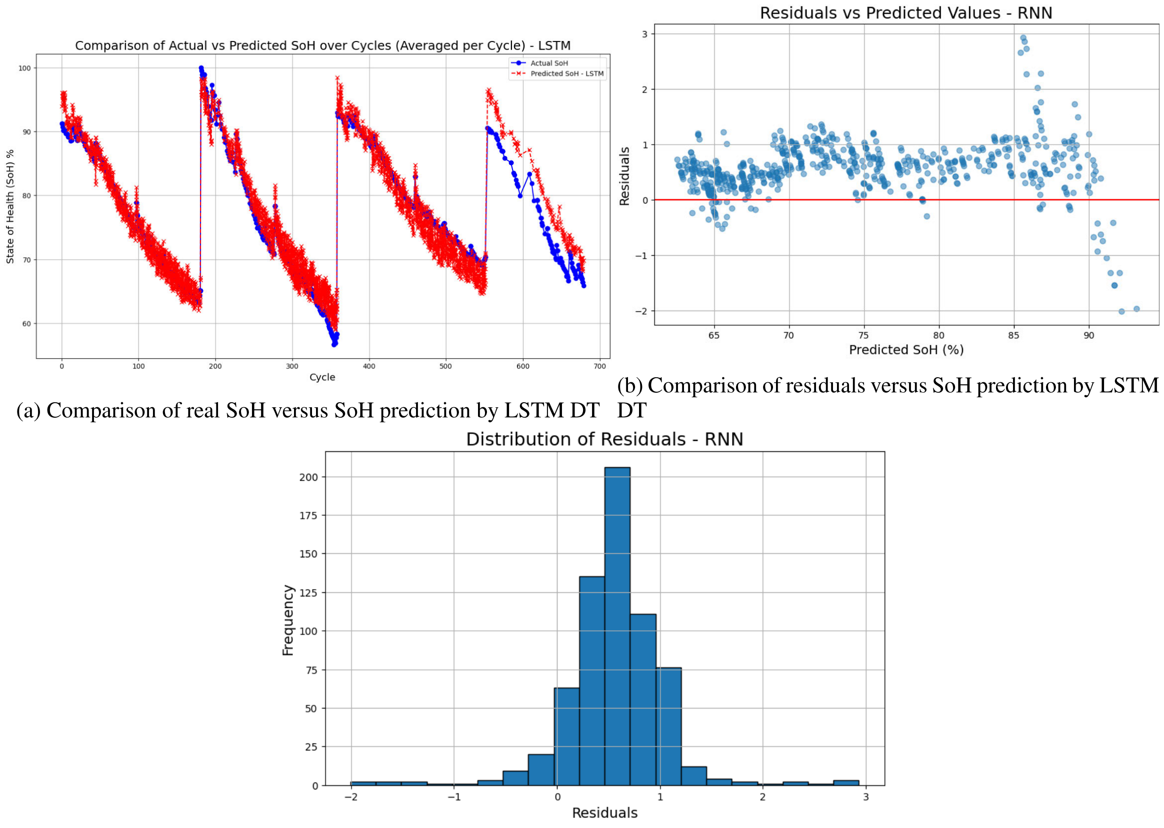
The results of all experiments were summarized in Table 3. The comparative analysis of the DT models and XAI approaches indicates that for the SoH estimation, the DNN model outperforms the LSTM in terms of both MSE and MAE. Additionally, the SHAP DI and SHAP Confidence (Conf) scores are higher for the DNN, suggesting more reliable and interpretable predictions. Conversely, for the SoC estimation, the LSTM model demonstrates superior performance with a lower MSE and a higher surrogate R^2, indicating a more accurate and consistent model.

TABLE 3. Model evaluation for the state of health and the state of charge.

Metric	State of Health (SoH)		State of Charge (SoC)	
	DNN	LSTM	DNN	LSTM
Test Results				
MSE	0.181413	0.546891	0.0477749	0.007654
MAE	0.320084	0.629549	0.201637	0.077932
XAI Results				
SHAP DIR	1.75788	1.460568	0.88617	0.881019
SHAP Conf	75.316	75.32802	60.6196	60.4904
LIME Conf	0.10761	0.687376	0.77030	0.76608
LIME DIR	7.2927	9.045096	20.1304	20.926914
Surrogate MSE	0.56473	0.0510	0.000001473	0.00092
Surrogate R^2 (Conf)	0.99311	0.99939	0.999999	0.9999990
Mean residuals	0.5903	0.17767	0.006797	0.020644
Surrogate DIR	120.66	170.507	44.037182	43.738813

The XAI approaches reveal that SHAP provides a higher DIR compared to LIME for both SoH and SoC estimations. However, LIME offers better local explanations, as indicated by the confidence scores. The surrogate model results suggest that the LSTM model may be more interpretable due to higher R^2 and lower mean residuals.

Based on these findings, it is recommended to use the DNN model for SoH estimation and the LSTM model for SoC estimation. For explainability, SHAP is suggested

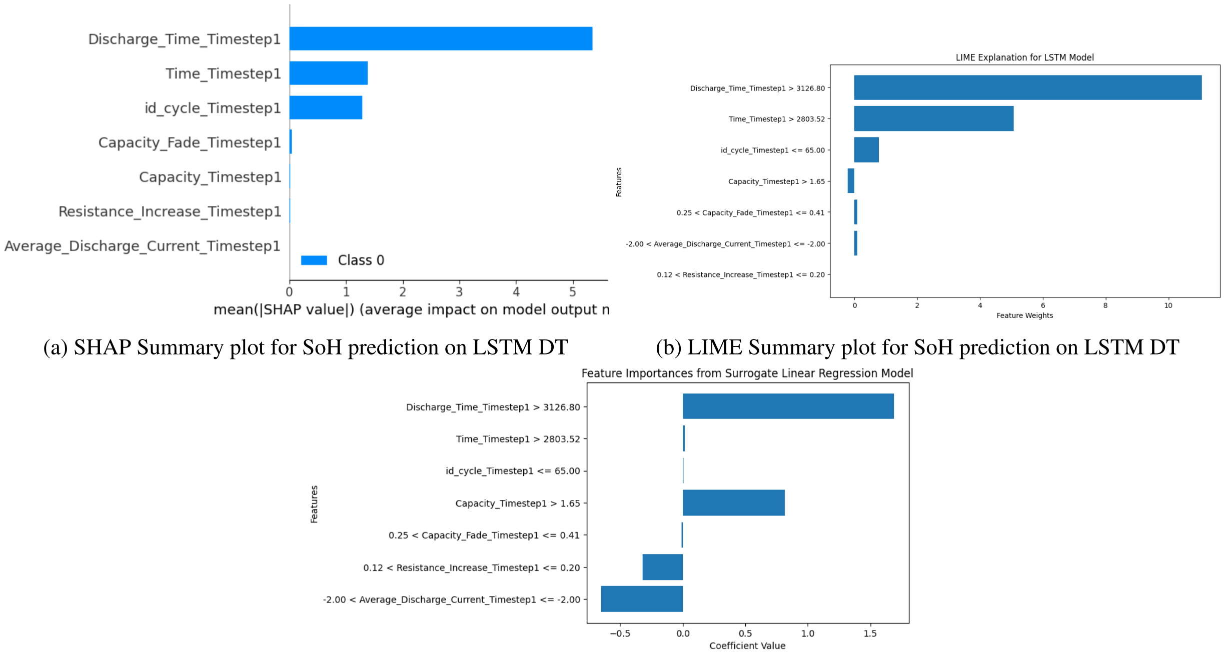


(a) Comparison of real SoH versus SoH prediction by LSTM DT

(b) Comparison of residuals versus SoH prediction by LSTM DT

(c) Distribution of residuals for SoH prediction LSTM DT

FIGURE 14. Visualizations of SoH Predictions and Residuals using LSTM DT.



(a) SHAP Summary plot for SoH prediction on LSTM DT

(b) LIME Summary plot for SoH prediction on LSTM DT

(c) Surrogate Summary plot for SoH prediction on LSTM DT

FIGURE 15. Interpretability plots for SoH prediction on LSTM DT.

for global understanding, while LIME is recommended for local explanations. The surrogate model can provide additional interpretability, especially for understanding the feature importance in the context of LSTM predictions.

V. CONCLUSION

This study highlights the importance of employing XAI in selecting appropriate models for deployment in DT-based BMS systems. In this study, SoC and SoH estimation were conducted using two DT-based models: DNNs and LSTMs. Three XAI approaches were employed to enhance the trustworthiness and explainability of these models. While the SHAP XAI approach focuses on global explanations, the LIME approach focuses on local explanations of the model predictions. We also employ a surrogate model, which can provide additional interpretability. The experiments and results demonstrate that applying XAI techniques significantly enhanced the interpretability of the LSTM model compared to the DNN model. The LSTM's ability to handle temporal sequences effectively makes it particularly suitable for the dynamic and complex datasets typically encountered in BMS applications, as evidenced by previous studies and the results in this study. While this study focused on the effectiveness of DNNs and LSTMs, we acknowledge the potential of other predictive models such as Gated Recurrent Units (GRUs), Elman RNNs, and Support Vector Regression (SVR). These models also hold promise for BMS applications and could be explored in future research to determine their comparative effectiveness and suitability. Incorporating such diverse models could broaden the understanding of model behavior under different BMS scenarios and enhance the robustness of the findings.

ABBREVIATIONS AND THEIR MEANINGS

AI	Artificial Intelligence
BMS	Battery Management Systems
CNN	Convolutional Neural Network
DIR	Decision Impact Ratio
DL	Deep Learning
DNN	Deep Neural Networks
DT	Digital Twins
FNN	Feedforward Neural Network
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
IoT	Internet of Things
LIME	Local Interpretable Model-agnostic Explanations
LR	Linear Regression
LSTM	Long-Short-Term Memory
MAE	Mean Absolute Error
ML	machine learning
MSE	Mean Squared Error
RBF	Radial Basis Function
RF	Random Forest
RNN	Recurrent Neural Networks
ReLu	Rectified Linear Unit
R^2	R-Squared
SoH	State of Health
SoC	State of Charge
SHAP	SHapley Additive exPlanations
SVR	Support Vector Regression
SVM	Spport Vector Machine
Conf	Confidence Score
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boost

REFERENCES

[1] S. Surya, V. Rao, and S. S. Williamson, "Comprehensive review on smart techniques for estimation of state of health for battery management system application," *Energies*, vol. 14, no. 15, p. 4617, Jul. 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/15/4617

[2] M. Yemini, A. Nedic, A. J. Goldsmith, and S. Gil, "Characterizing trust and resilience in distributed consensus for cyberphysical systems," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 71–91, Feb. 2022.

[3] G. Le Gall, N. Montavont, and G. Z. Papadopoulos, "IoT network management within the electric vehicle battery management system," *J. Signal Process. Syst.*, vol. 94, no. 1, pp. 27–44, Jan. 2022.

[4] H. Gabbar, A. Othman, and M. Abdussami, "Review of battery management systems (BMS) development and industrial standards," *Technologies*, vol. 9, no. 2, p. 28, Apr. 2021. [Online]. Available: https://www.mdpi.com/2227-7080/9/2/28

[5] Y. Zhang, F. Liang, S. Li, C. Zhang, S. Zhang, X. Liu, S. Zhao, S. Yang, Y. Xia, J. Lin, B. Guo, H. Cheng, M. Wang, M. Jiang, and D. Wang, "A review on battery thermal management and its digital improvement-based cyber hierarchy and interactional network," *Int. J. Energy Res.*, vol. 46, no. 9, pp. 11529–11555, Jul. 2022, doi: 10.1002/er.7957.

[6] J. N. Njoku, E. C. Nkoro, R. M. Medina, P. M. Custodio, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Metaverse and digital twin for BMS using MATLAB and unreal engine," in *Proc. Korean Inst. Commun. Inf. Sci. (KICS) Summer Conf.*, Jeju Island, South Korea, 2023, pp. 1824–1825. [Online]. Available: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11487907

[7] J. N. Njoku, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Model comparison and selection for battery digital twin development using PyBaMM," in *Proc. 33rd Joint Conf. Commun. Inf. (JCCI)*, Yeosu, South Korea, 2023, pp. 1–2.

[8] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Adv. Model. Simul. Eng. Sci.*, vol. 7, no. 1, pp. 1–13, Dec. 2020, doi: 10.1186/s40323-020-00147-4.

[9] X. Liu, L. Zhang, H. Yu, J. Wang, J. Li, K. Yang, Y. Zhao, H. Wang, B. Wu, N. P. Brandon, and S. Yang, "Bridging multiscale characterization technologies and digital modeling to evaluate lithium battery full lifecycle," *Adv. Energy Mater.*, vol. 12, no. 33, Sep. 2022, Art. no. 2200889, doi: 10.1002/aenm.202200889.

[10] R. Di Fonso, P. Bharadwaj, R. Teodorescu, and C. Cecati, "A battery digital twin based on neural network for testing SoC/SoH algorithms," in *Proc. IEEE 20th Int. Power Electron. Motion Control Conf. (PEMC)*, Sep. 2022, pp. 655–660.

[11] J. Schmitt, I. Horstkötter, and B. Bäker, "State-of-health estimation by virtual experiments using recurrent decoder–encoder based lithium-ion digital battery twins trained on unstructured battery data," *J. Energy Storage*, vol. 58, Feb. 2023, Art. no. 106335. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22023246

[12] F. Wang, Z. Zhao, Z. Zhai, Z. Shang, R. Yan, and X. Chen, "Explainability-driven model improvement for SOH estimation of lithium-ion battery," *Rel. Eng. Syst. Saf.*, vol. 232, Apr. 2023, Art. no. 109046.

[13] M. Faraji Niri, K. Aslansefat, S. Haghi, M. Hashemian, R. Daub, and J. Marco, "A review of the applications of explainable machine learning for lithium–ion batteries: From production to state and performance estimation," *Energies*, vol. 16, no. 17, p. 6360, Sep. 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/17/6360

[14] C. I. Nwakanma, L. A. C. Ahakonye, J. N. Njoku, J. C. Odirichukwu, S. A. Okolie, C. Uzondu, C. C. Ndubuisi Nweke, and D.-S. Kim, "Explainable artificial intelligence (XAI) for intrusion detection and mitigation in intelligent connected vehicles: A review," *Appl. Sci.*, vol. 13, no. 3, p. 1252, Jan. 2023.

[15] T. Heitzmann, A. Samet, T. Mesbahi, C. Soufi, I. Jorge, and R. Boné, "SocHAP: A new data driven explainable prediction of battery state of charge," in *Computational Science—ICCS 2023*, J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Eds. Cham, Switzerland: Springer, 2023, pp. 463–475.

[16] S. M. Shahriar, E. A. Bhuiyan, M. Nahiduzzaman, M. Ahsan, and J. Haider, "State of charge estimation for electric vehicle battery management systems using the hybrid recurrent learning approach with explainable artificial intelligence," *Energies*, vol. 15, no. 21, p. 8003, Oct. 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/21/8003

[17] S. Jafari and Y.-C. Byun, "Prediction of the battery state using the digital twin framework based on the battery management system," *IEEE Access*, vol. 10, pp. 124685–124696, 2022.

[18] M. S. H. Lipu, M. A. Hannan, A. Hussain, M. H. M. Saad, A. Ayob, and F. Blaabjerg, "State of charge estimation for lithium-ion battery using recurrent NARX neural network model based lighting search algorithm," *IEEE Access*, vol. 6, pp. 28150–28161, 2018.

[19] K. Zhao, Y. Liu, W. Ming, Y. Zhou, and J. Wu, "Digital twin-driven estimation of state of charge for Li-ion battery," in *Proc. IEEE 7th Int. Energy Conf. (ENERGYCON)*, May 2022, pp. 1–6.

[20] H. Chaoui and C. C. Ibe-Ekeocha, "State of charge and state of health estimation for lithium batteries using recurrent neural networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 8773–8783, Oct. 2017.

[21] W.-Y. Chang, "The state of charge estimating methods for battery: A review," *ISRN Appl. Math.*, vol. 2013, pp. 1–7, Jul. 2013, doi: 10.1155/2013/953792.

[22] V. D. Prasanna, "Estimation of state of charge of a lead acid battery using support vector regression," *Proc. Technol.*, vol. 21, pp. 264–270, Jan. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017315002546

[23] D. Roman, S. Saxena, V. Robu, M. Pecht, and D. Flynn, "Machine learning pipeline for battery state-of-health estimation," *Nature Mach. Intell.*, vol. 3, no. 5, pp. 447–456, Apr. 2021.

[24] V. Klass, M. Behm, and G. Lindbergh, "A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation," *J. Power Sources*, vol. 270, pp. 262–272, Dec. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775314011707

[25] X. Feng, C. Weng, X. He, X. Han, L. Lu, D. Ren, and M. Ouyang, "Online state-of-health estimation for Li-ion battery using partial charging segment based on support vector machine," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8583–8592, Sep. 2019.

[26] 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, 2020.

[27] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J. C.-W. Chan, P. van den Bossche, J. Van Mierlo, and N. Omar, "Random forest regression for online capacity estimation of lithium-ion batteries," *Appl. Energy*, vol. 232, pp. 197–210, Dec. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261918315010

[28] X. Qu, Y. Song, D. Liu, X. Cui, and Y. Peng, "Lithium-ion battery performance degradation evaluation in dynamic operating conditions based on a digital twin model," *Microelectron. Rel.*, vol. 114, Jan. 2020, Art. no. 113857. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026271420304765

[29] H. Li, M. Bin Kaleem, I.-J. Chiu, D. Gao, and J. Peng, "A digital twin model for the battery management systems of electric vehicles," in *Proc. IEEE 23rd Int. Conf. High Perform. Comput. Commun.; 7th Int. Conf. Data Sci. Syst.; 19th Int. Conf. Smart City; 7th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (HPCC/DSS/SmartCity/DependSys)*, Dec. 2021, pp. 1100–1107.

[30] M. Danko, J. Adamec, M. Taraba, and P. Drgona, "Overview of batteries state of charge estimation methods," *Transp. Res. Proc.*, vol. 40, pp. 186–192, Apr. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146519301905

[31] A. A. Hussein, "Capacity fade estimation in electric vehicle Li-ion batteries using artificial neural networks," *IEEE Trans. Ind. Appl.*, vol. 51, no. 3, pp. 2321–2330, May 2015.

[32] C. Semeraro, A. G. Olabi, H. Aljaghoub, A. H. Alami, M. A. Radi, M. Dassisti, and M. A. Abdelkareem, "Digital twin application in energy storage: Trends and challenges," *J. Energy Storage*, vol. 58, Feb. 2023, Art. no. 106347. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22023362

[33] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," 2018, *arXiv:1802.01933*.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.

[35] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370218305988

[36] A. Temenos, N. Temenos, M. Kaselimi, A. Doulamis, and N. Doulamis, "Interpretable deep learning framework for land use and land cover classification in remote sensing using SHAP," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[37] A. Holzinger, B. Malle, P. Kieseberg, P. M. Roth, H. Müller, R. Reihs, and K. Zatloukal, "Towards the augmented pathologist: Challenges of explainable-AI in digital pathology," 2017, *arXiv:1712.06657*.

[38] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[39] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*.

[40] K. Kobayashi and S. B. Alam, "Explainable, interpretable, and trustworthy AI for an intelligent digital twin: A case study on remaining useful life," *Eng. Appl. Artif. Intell.*, vol. 129, Mar. 2024, Art. no. 107620. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623018043

[41] S. Suhail, M. Iqbal, R. Hussain, and R. Jurdak, "ENIGMA: An explainable digital twin security solution for cyber–physical systems," *Comput. Ind.*, vol. 151, Oct. 2023, Art. no. 103961. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361523001112

[42] B. Saha and K. Goebel. (2007). *Battery Data Set*. NASA Ames Prognostics Data Repository, Moffett Field, CA, USA. [Online]. Available: http://ti.arc.nasa.gov/project/prognostic-data-repository

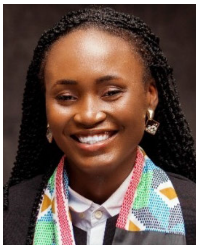
[43] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. D. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835219304838

[44] N. Kostopoulos, D. Kalogeras, D. Pantazatos, M. Grammatikou, and V. Maglaris, "SHAP interpretations of tree and neural network DNS classifiers for analyzing DGA family characteristics," *IEEE Access*, vol. 11, pp. 61144–61160, 2023.

[45] M. Zolanvari, Z. Yang, K. Khan, R. Jain, and N. Meskin, "TRUST XAI: Model-agnostic explanations for AI with a case study on IIoT security," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 2967–2978, Feb. 2023.

[46] A. Hosein Oveis, E. Giusti, S. Ghio, G. Meucci, and M. Martorella, "LIME-assisted automatic target recognition with SAR images: Toward incremental learning and explainability," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 9175–9192, 2023.

[47] E. Mariotti, A. Sivaprasad, and J. M. A. Moral, "Beyond prediction similarity: ShapGAP for evaluating faithful surrogate models in XAI," in *Explainable Artificial Intelligence*, L. Longo, Ed. Cham, Switzerland: Springer, 2023, pp. 160–173.



JUDITH NKECHINYERE NJOKU (Member, IEEE) was born in Ilorin, Kwara, Nigeria. She received the Bachelor of Engineering degree in petroleum engineering from the Federal University of Technology, Owerri, Nigeria, in 2014, and the master's degree in aeronautics, mechanical and electronics engineering from the Kumoh National Institute of Technology, Gumi, South Korea, in 2021. She was an Intern with Drillog Petro-Dynamics Ltd. (PDL), Nigeria, in 2011 and 2013. She also interned with the Ministry of Energy and Mineral Resources, Alausa, Ikeja, Lagos, Nigeria, in 2016. From 2017 to 2019, she was with Sterling Bank Nigeria PLC, Nigeria. She is currently a Doctoral Researcher with the Networked Systems Laboratory, Kumoh National Institute of Technology. Her research interests include data-driven intelligent transportation systems (DDITS), digital twin, intelligent energy management systems, and metaverse for the industry. She is a member of the IEEE Young Professionals, the IEEE Women in Engineering (WIE), the Society of Petroleum Engineers (SPE), the WomenTech Network, and Nigerian Society of Engineers Nigeria. She serves as a reviewer for the *IET Communications Journal*.

COSMAS IFEANYI NWAKANMA (Member, IEEE) received the National Diploma degree (Hons.) in electrical/electronics engineering from the Federal Polytechnic Nekede Owerri, Imo, Nigeria, in 1999, the B.Eng. degree in communication engineering, the M.Sc. degree in information technology, and the M.B.A. degree in project management technology from The Federal University of Technology, Owerri, Nigeria, in 2004, 2012, and 2016, respectively, and the Ph.D. degree in IT-convergence engineering from the Networked System Laboratory, Department of IT-Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea, in 2022. He was an Intern with Asea Brown Boveri (ABB), Nigeria, in 2003. From 2009 to 2019, he was a Lecturer and a Researcher with the Federal University of Technology, Owerri. Since 2022, he has been a Senior Research Fellow with the ICT Convergence Research Center (ICT-CRC), Kumoh National Institute of Technology. His research interests include the application of explainable artificial intelligence (XAI) and the Internet of Things (IoT) to smart factories, homes, farms, vehicles, and the digital twin (metaverse). He is a member of the Computer Professionals Registration Council of Nigeria and Nigeria Society of Engineers and registered by the Council for the Regulation of Engineering in Nigeria.

DONG-SEONG KIM (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003. From 1994 to 2003, he was a full-time Researcher with ERC-ACI, Seoul National University. From March 2003 to February 2005, he was a Postdoctoral Researcher with the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently the Director of the KIT Convergence Research Institute and ICT Convergence Research Center (ITRC and NRF Advanced Research Center Program) supported by Korean Government, Kumoh National Institute of Technology. His research interests include the real-time IoT and smart platforms, industrial wireless control networks, and networked embedded systems. He is a Senior Member of ACM.