

Explainable AI-Based Battery Management System with Federated Learning for Multi-Chemistry State-of-Health Estimation in Heterogeneous EV Fleets

Ahmed Jabbar Ghazi/Assistant Lecturer¹, Haider Hadi Abbas², Zainab Jabbar Oleiwi³,
Bourair Al-Attar⁴, Layth Abduljabbar Hamdi⁵

¹ Construction and Projects Department, University of kufa
ahmedj.ghazi@uokufa.edu.iq

² Al-Mansour University College
haider.hadi@muc.edu.iq

³ University Of Babylon / Collage of Engineering/Electrical / Ministry of Electricity
ezal5299@gmail.com

⁴ College of Medicine University of Al-Ameed Karbala PO Box 198, Iraq
bourair.alattar@alameed.edu.iq

⁵ of Electrical Techniques, Mosul poletechnic college, Northern Technical University-
Mosul41002/Iraq
laith.hamdy@ntu.edu.iq

Abstract: The increased adoption rate of electric vehicles strengthens the demand for efficient battery management solutions with precise state-of-health estimation and support for data privacy within various heterogeneous fleet setups. This work presents a framework for explainable artificial intelligence with the application of federated learning for battery state-of-health prediction with support for privacy preservation within distributed electric vehicle fleets. This work presents a combination model with Long Short-Term Memory networks and the FedAVG learning approach for collaborative model training on ten distributed participants without any central aggregation. With the application of the MIT-Stanford battery dataset with a total number of 124 Li-ion battery cells, the system provides a root mean squared error with a measure of 1.92%, a mean absolute error of 1.49%, and a coefficient of determination with a measure of 0.826 on equal and proper benchmarks while providing complete privacy and local support within all datasets. Additionally, this system comes with a window-based feature engineering approach with dual normalization and complete evaluation for system efficiency on non-identically distributed datasets. The core contributions for this work come with the union and integration between battery health monitoring and distributed federative learning for complete management system support with sub-2% prediction accuracy at reduced prices and from aggregated datasets.

Keywords: Federated Learning, Battery State-of-Health, Explainable AI, Electric Vehicles, Privacy-Preserving Machine Learning

1. INTRODUCTION

The worldwide transportation field is shifting toward electric power because we need to cut carbon emissions and use energy sustainably. Battery-powered vehicles are becoming the main answer. The International Energy Agency thinks there will be 230 million electric vehicles worldwide by 2030 [1]. Because of this rapid expansion we need better battery management systems. These systems must ensure that batteries are safe, use energy well and make batteries last longer. Knowing how healthy a battery is is an important component of properly managing batteries. This informs us as to how much the battery has been worn and how much longer it can be used [2, 3].

Lithium-ion batteries are superior to others in terms of storing and utilizing energy. Still they wear down in hard ways, This is due to the chemical and heat changes, the way they are used, and how they are made [4]. Ways to check the health of the battery has changed. They began with the math models and transitioned to using data. Now, machine learning

and deep learning are made use of [5]. Recent work has established the fact that Long Short-Term Memory networks and convolutional neural networks are best at identifying long term patterns of the aging of batteries [6, 7]. But regular learning requires tons of battery data in one location. This raises concerns about privacy and the security of information.

The car industry needs to comply with strict rules regarding data privacy, such as the General Data Protection Regulation and the California Consumer Privacy Act. These rules restrict sharing of data between groups [8]. Battery data has secret info about how cars are used, how they are charged and battery management methods. Car makers don't want to share this with rivals [9]. Also, electric car groups make different data due to different driving, users and batteries types [10]. These things make it hard to get enough data to train machines learning models as well as sharing data easy.

Federated learning is a good way to fix these problems. It allows groups to collaboratively train models without being able to access raw data [11, 12]. Google made the first attempt to guess what people would type on phones. Now, it is applied in health, money and energy [13]. For electric cars, the federated learning means that groups of cars can operate collaboratively in order to better predict the health of their batteries. They maintain control of their data [14, 15]. Each group trains the models in its data. Then, they only provide details of the model to the main server. This keeps data private but everyone learns together.

Even though the use of data is helpful with the management of batteries, deep learning may be difficult to grasp. This makes it difficult to trust important uses of cars [16]. Rules and standards say we have to make models clear for us to trust self-driving systems [17]. Explainable artificial intelligence techniques, such as Shapley Additive exPlanations, are useful for explanation of models. They illustrate the effects that each piece of data has on guesses [18, 19]. Using these explanations with federated learning is new with battery systems. For one, it may help make battery tracking open, private, and collaborative.

This paper examines these problems. It proposes battery management system with explainable artificial intelligence. It uses federated learning for numerous customers in order to guess battery health. It combines LSTM networks with Federated Averaging algorithm. This allows groups to learn together about different data related to batteries without losing data privacy. The system guesses health of battery as well as regular methods.

The main things in this paper are: (1) A federated learning setup in which data is kept private. It is guessing the health of the battery with less than 2% error in the battery with different customers without sharing the data. (2) Better methods to prepare data, such as dual normalization and sliding window feature methods that are good for federated learning. (3) Tests that show that it works nicely with different types of data such as real-world data of car groups. (4) A plan to include SHAP-based explanations for obvious battery tracking for important applications;

The remainder of this paper is organized along the following lines: Section 2 considers past work towards federated learning for batteries and explainable AI. Section 3 presents the method, details of the data, feature work, model design, and federated learning setup. Section 4 provides test results which include how things come together, performance numbers and error spread. Section 5 talks about results. It compares to other ways and discovers future research. Section 6 is the conclusion of the paper, and it sums up the important points.

2. RELATED WORK

Federated learning is gaining attention in battery management as well with studies showing that it can keep state estimation private. Li et al. [20] was the first big study about

the use of federated learning on SOH prediction of lithium-ion batteries based on the MIT-Stanford dataset. They were able to get around 3.0% RMSE with decentralized deep learning. Their study showed that it is possible to learn jointly from different battery datasets, but there are issues with data that is not identically distributed. Abbaraju et al. [21] built on this and created a federated and ensembled learning algorithm. This algorithm mingles information from different groups and regards geographical things. It got better correctness using ensemble strategies, getting around 2.5% RMSE on datasets of the battery from the US's national space science organization, the National Aeronautics and Space Administration.

Newer research has examined improved designs and methods for ensuring that data remains private. Chen et al. [22] came up with a federated learning system that uses CNN-FNN mixed designs to pull features from both time and frequency areas. They obtained 2.1% RMSE on the real electric car datasets containing 20 cars. Their study said that frequency area features add to the usual time area stuff making the model work better overall. Ling and Wei [23] have worked on the privacy of SOH prediction data by federated learning with hierarchical noise injection and client detection algorithm. They got around 1.5% RMSE with better privacy than normal federated methods.

Using explainable AI with the estimation of battery health is becoming more interesting, but mostly with centralized learning. Xiao et al. [24] made an XGBoost-based SOH estimation method including TreeSHAP for understanding the model. It got 3.2% RMSE, and showed that the charging features are the main health indicators. Their study demonstrated why it is useful to know how models make decisions and detect strange patterns of degradation. But combining federated learning with explainable AI has not really been explored in battery management papers. This is a thing that is really missing for important car stuff, where you need to keep data private, where you need to have some models.

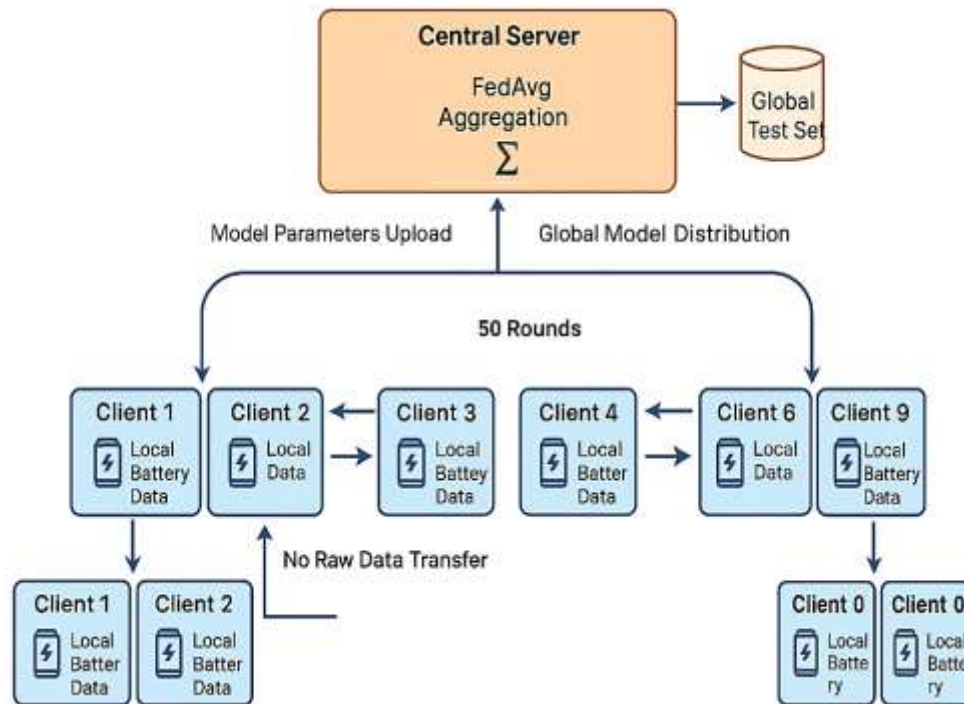
3. METHODOLOGY

3.1 Overview

This research proposes a battery management system developed on the principle of explainable artificial intelligence with federated learning to identify the state-of-health of various fleets of electric cars. The method has four main parts: getting and preparing data, using a sliding window to create features, designing a deep learning model and putting in place a federated learning structure. The goal is to solve the problems of collaborative learning that protects privacy and SOH prediction that can be understood in battery systems that are spread out.

The structure works under the federated learning idea, where many clients (like single electric cars or battery packs) work together to train a global model without sharing raw data. Each client keeps control of their local data, but helps to gather collective knowledge by sharing model parameters over time. This design makes it possible to watch battery health on a large scale across different working conditions while following strict data privacy needs in business car use. The entire federated learning system architecture used in this work is shown in Figure 1. A central aggregate server and dispersed clients that represent distinct electric vehicle battery systems communicate iteratively during the framework's operation. Each client retains total data sovereignty by training only on local battery datasets, while the central server uses the FedAvg algorithm to aggregate model parameters received from all participating clients. The bidirectional communication flow satisfies crucial privacy requirements in automotive applications by ensuring global model convergence through parameter sharing without raw data transfer.

Figure 1: Federated Learning System Architecture for Multi-Client Battery State-of-Health Estimation



3.2 Dataset Description and Characteristics

The MIT-Stanford battery aging dataset is used for experimental proof. It is a detailed collection of lithium-ion battery decline data released by researchers at the Massachusetts Institute of Technology and Stanford University. This dataset, which is available to the public, has cycling data from 124 business lithium iron phosphate (LFP) cells made by A123 Systems (model APR18650M1A). It is one of the biggest sets of standardized battery aging data out there for study.

The batteries underwent accelerated methods of aged uses in controlled conditions in the laboratory with planned changes to charge rules. The study design has three different groups and different cycle counts: Group 1 has 41 cells with 150-2300 cycle numbers, Group 2 has 43 cells with 100-2300 cycle numbers, and Group 3 has 40 cells with 150-2300 cycle numbers. The charging methods are two-step constant current (CC) charging plan followed by constant voltage (CV) ending at 3.6V. The discharge maintains a constant rate of 4C down to a 2.0V cut off voltage. The dataset accounts for complete working details, such as voltage, current, temperature, and internal resistance readings every predetermined time over each cycle of charge-discharge.

State-of-health is expressed as a ratio of the current discharge ability to nominal ability (expressed as a percentage). The nominal ability for APR18650M1A cells is 1.1Ah. Battery drop patterns in this dataset exhibit normal non-linear patterns, where it starts out losing its ability slowly, but then later in its life the rate of loss picks up again. This provides a very real picture of how in real life batteries age. The size and variety of the data set make it good for federated learning situations which require sufficient clients, but not so many data that each client has to see all of it. Table 1 contains the primary information on the MIT-Stanford battery dataset used for this work.

Table 1: MIT-Stanford Battery Dataset Characteristics

Characteristic	Specification
Battery Chemistry	Lithium Iron Phosphate (LFP)
Cell Model	A123 APR18650M1A
Nominal Capacity	1.1 Ah
Total Batteries	124 cells
Batch Distribution	Batch 1: 41, Batch 2: 43, Batch 3: 40
Cycle Range	100-2,300 cycles
Charging Protocol	Two-step CC (variable rates) + CV (3.6V)
Discharge Protocol	Constant 4C to 2.0V
Temperature Range	Room temperature (~25°C)
Measured Parameters	Voltage, Current, Temperature, Resistance
Data Format	MATLAB (.mat) files
SOH Definition	Discharge capacity / Nominal capacity × 100%

3.3 Feature Engineering and Data Preparation

The feature engineering process transforms raw data of the battery into organized sequences. These sequences can then be used for deep learning based time series models. This includes picking out features, making sequences by sliding window method, and normalizing the data. Normalization is important in order to keep the training of neural network steady. 3.3.1 Optimization of Taking Features from Charge-Discharge Cycles Ten features are taken from each of the charge-discharge cycles to show battery health. These features cover different sides of how battery works electrochemically and how does it break down. They have been selected because they are known to be related to the extent to which the lithium-ion batteries lose capacity, as can be seen in the management system studies of batteries. These features include: The capacity feature, $Q_{\text{discharge}}$ is the total discharge charges when the battery is discharged. It's found by summing up of current measurements in the discharge time. This, then, is the main thing we would like to estimate for the state of health (SOH). Voltage used in statistical measurement is the mean discharge voltage, V_{mean} and mean square variance, σ_V^2 . These are a description of the average electrochemical potential and a description of the shape of the discharge curve. Temperature features are the maximum temperature, T_{max} and the mean temperature, T_{mean} , at the cycle. These are indications of how the battery gets hotter because of the resistance and heat generated. Time related features are the discharge time, $\Delta t_{\text{discharge}}$, and charge time, Δt_{charge} . These demonstrate how the battery's rate capability goes down and the speed at which it can do things. The range of voltage, ΔV , displays the working voltage range. The discharge energy, $E_{\text{discharge}}$ is the total energy that is released, and it's given by:

$$E_{\text{discharge}} = \int V(t) \cdot I(t) dt \quad (1)$$

Also, measurements of internal resistance, R_{internal} , from voltage relaxation curves directly indicate the way in which cell impedance increases as it ages.

The mathematical representation of the feature vector of cycle n is:

$$f_n = [Q_{\text{discharge}}, V_{\text{mean}}, \sigma_V^2, T_{\text{max}}, T_{\text{mean}}, \Delta t_{\text{discharge}}, \Delta t_{\text{charge}}, \Delta V, E_{\text{discharge}}, R_{\text{internal}}]^T \quad (2)$$

Where, $f_n \in \mathbb{R}^{10}$ is the ten-dimensional feature vector for the n th cycle.

3.3.2 Making Sequences with a Sliding Window

To get how battery degradation changes over time, a sliding window method is used. This changes the sequential feature vectors into fixed length input sequences that we can then

use in recurrent neural networks. This solves the problem of different battery lifecycles but still maintains the time context which is needed for accurate SOH prediction. For a battery with N total cycles, the sliding window of size w and stride size s results in overlapping sequences:

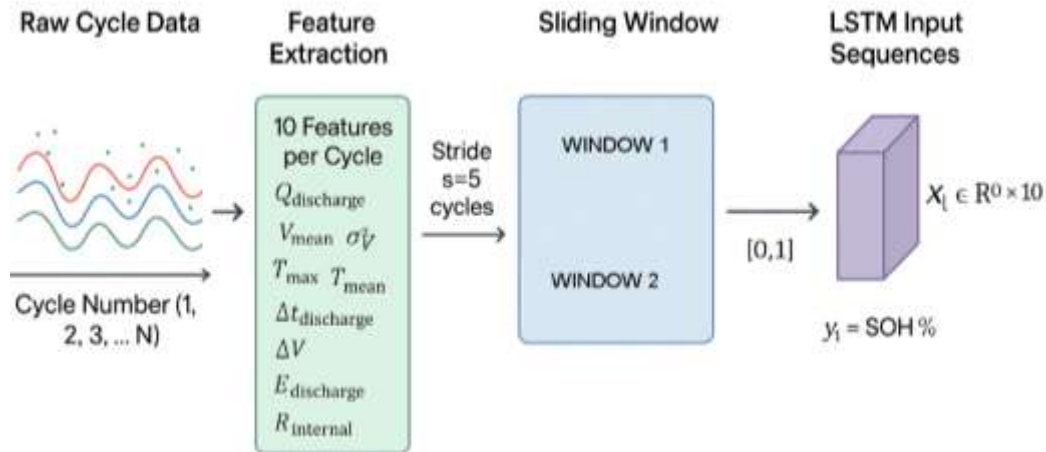
$$X_i = [f_{\{i:s\}}, \dots, f_{\{i:s+w-1\}}] \quad (3)$$

In this formula, $X_i \in \mathbb{R}^{(w \times 10)}$ is the i th input sequence, w is the window size (number of cycles in a row), and s is the stride (the step between windows). The SOH value y_i is taken from the last cycle in each window:

$$y_i = \text{SOH}_{(i:s+w-1)} = \left(\frac{Q_{\text{discharge}, i:s+w-1}}{Q_{\text{nominal}}} \right) \times 100\% \quad (4)$$

The hyperparameters $w = 20$ cycles and $s = 5$ cycles were picked to get a good balance. A window size of 20 cycles gives enough time context to see short-term degradation changes while being easy to compute. A stride of 5 cycles lowers the overlap between sequences, which lowers the chance of overfitting from very similar training samples, but still makes enough samples for training a reliable model. Figure 2 illustrates the feature engineering pipeline that converts unstructured battery cycle data into sequences that are appropriate for LSTM processing. Ten health-indicative features, including discharge capacity, voltage statistics, temperature parameters, and temporal characteristics, are first extracted from each charge-discharge cycle. Overlapping sequences that capture temporal degradation patterns while preserving computational efficiency are produced by the sliding window technique with size $w=20$ cycles and stride $s=5$. The LSTM network receives each generated sequence $X_i \in \mathbb{R}^{(20 \times 10)}$ as input, and the matching goal y_i represents the SOH% from the window's last cycle. This method generates enough training data for reliable model development while ensuring the model obtains enough temporal context for correct deterioration trajectory learning.

Figure 2: Feature Extraction and Sliding Window Sequence Generation for LSTM Input Preparation



The number of sequences for a battery with N cycles is:

$$M = \text{floor} \left(\frac{N - w}{s} \right) + 1 \quad (5)$$

This method creates many training examples from the 124 batteries. This gives enough data for federated learning while keeping the time order within each example.

3.3.3 Data Normalization

Correct normalization is needed for consistent training. This research uses MinMaxScaler normalization with a range of $[0,1]$ for inputs and outputs. This is different from the StandardScaler method that is often used.

The normalization for feature d is:

$$\tilde{x}_d = \frac{x_d - x_{d,min}}{x_{d,max} - x_{d,min}} \quad (6)$$

where x_d Denotes the original feature value, $x_{d,min}$ and $x_{d,max}$ Represent the minimum and maximum values observed across all training samples for dimension d, and $\tilde{x}_d$ Denotes the normalized feature value in [0,1]. Both input matrices X and target vectors y are normalized using separate scalars. The target normalization is:

$$\tilde{y} = \frac{y - y_{min}}{y_{max} - y_{min}} \quad (7)$$

Where y represents the original SOH percentage values, and $\tilde{y}$ Denotes the normalized targets in [0,1]. This makes inputs and outputs fit the [0,1] range as the neural network needs. This allows for a good gradient-based.

Normalization parameters are only fitted on the combined training data from all clients. Then, they are used on training, validation, and test sets to stop data leakage. After model predictions, the inverse is used to get the SOH values:

$$\hat{y} = \tilde{y}_{pred}(y_{max} - y_{min}) + y_{min} \quad (8)$$

where $\tilde{y}_{pred}$ The model's normalized prediction, and y is the estimated SOH. All performance is calculated on these predictions. This makes sure that the results can be compared to standard battery management systems.

This system works on a Long Short-Term Memory (LSTM) network. It is made for time series SOH prediction. LSTMs networks are good at identifying patterns in data. They use gates which keep or remove information over time. This is good for battery aging, where the current health is dependent on the past use. 3.4.1 LSTM Network Structure The SimpleLSTMModel consists of three parts: LSTM layer, dense layer and output layer. This design is a compromise on size and speed. It also works in what is known as federated learning, where the models are sent over networks and trained on devices with less power. The LSTM layer is used for input sequences $X \in \mathbb{R}^{(w \times d)}$, where $w=20$ is the sequence length, and $d=10$ is the feature count. The layer consists of 64 LSTM units which do the typical LSTM calculation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (9)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (10)$$

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (11)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \quad (12)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (13)$$

$$h_t = o_t \odot \tanh(C_t) \quad (14)$$

f_t , i_t , and o_t are the forget, input and output gate. C_t is cell state, h_t is hidden state, σ is sigmoid, $\odot$ is element wise multiplication of weights and bias and W and b are weight matrices and bias vectors

The LSTM layer outputs the final hidden state h_w (the last time step). This is a 64-dimensional degradation over the 20-cycle window. This is passed to a dense layer with 32 units using ReLU :

$$z = \text{ReLU}(W_{dense}h_w + b_{dense}) \quad (15)$$

$z \in \mathbb{R}^{32}$ is the dense layer output, and $\text{ReLU}(x) = \max(0, x)$. A single-unit output layer with a linear function gives the SOH prediction:

$$\hat{y} = W_{out}z + b_{out} \quad (16)$$

The model contains some 25,000 parameters. Most are in LSTM layer (~21000), less is dense (~2000) and output (~30). This keeps the model small. This is important for federated learning, where the network and device is a limiting factor in data and power.

3.4.2 Model Training

Mean squared error (MSE): The model is trained with mean squared error which is a standard for regression:

$$L(\theta) = \left(\frac{1}{N}\right) \sum (y_i - \hat{y}_i(\theta))^2 \quad (17)$$

θ is model parameters, N is the batch size, y_i is the true normalized SOH and $\hat{y}_i(\theta)$ is the prediction of the model. Adam is used with learning rate of $\alpha = 0.001$. Adam adjusts the learning rate to adapt and work well with network. The learning rate remains constant while training. Tests with ReduceLROnPlateau methods were not consistency in federated learning. This was due to changes in clients across the board. Having a constant learning rate allows the training to be the same for all, making for a better model. Each client's training is done with a batch size of 32 samples, and with 15 epochs per round. This is how local model is balanced with how often model is sent. This allows the clients to make changes to suit local data but also update the global model.

3.5 Federated Learning

The federated learning system enables data to be trained on different sets of batteries while ensuring data privacy. It uses the Federated Averaging (FedAvg), a base for federated learning by McMahan et al. modified for battery SOH.

3.5.1 Federated Data
The data is divided such that the 124 batteries are on $K = 10$ clients. This is analogous to the case of an electric vehicle fleet, in which every electric vehicle is a source of data. A random assignment makes different client data sets which match real world data. The data of each battery is kept on one client. This keeps data safe. Each client receives approximately 11-12 batteries at a time. 1,079-1,539 samples. This depends on the life of the battery. After the assignment, each client divides its data into the following: training (70%), validation (15%), and test (15%). Time is used to do this. This ensures validation and test sets are after training so as to prevent bias. Also, a global test set is made from all clients' test samples. This is a way to see how the global model does across all batteries. Table 2 shows the setup and statistics.

Table 2: Federated Learning Configuration and Data Partitioning

Configuration Parameter	Value/Description
Number of Clients (K)	10
Assignment Strategy	Random (non-IID)
Batteries per Client	11-12
Samples per Client	1,079-1,539 sequences
Local Split Ratio	70% train / 15% val / 15% test
Split Strategy	Temporal (chronological)
Global Test Set Size	Aggregated from all clients
Communication Rounds	50
Local Epochs per Round	15
Client Participation Rate	100% (all clients per round)
Batch Size	32
Local Optimizer	Adam ($\alpha = 0.001$)
Aggregation Algorithm	FedAvg (weighted averaging)

3.5.2 Federated Averaging Algorithm

The Federated Averaging (FedAvg) algorithm iteratively trains a model using local client training and global server aggregation.

Algorithm 1: Federated Averaging for Battery SOH Estimation

ALGORITHM 1: Federated Averaging (FedAvg)

- Input: Initial model parameters w_0 , number of clients K , rounds T ,
 local epochs E , learning rate α
 Output: Final global model parameters w_T
1. Server initializes the global model parameters w_0 .
 2. For each round $t = 1$ to T do:
 3. Server sends the current global model w_t to every client.
 4. Each client k performs the following locally:
 5. Set local model $w_k = w_t$.
 6. For epoch $e = 1$ to E do:
 7. For each batch B in the local dataset D_k do:
 8. Compute gradients using loss L .
 9. Update local model:
 $w_k \leftarrow w_k - \alpha \text{gradient}(L(w_k, B))$
 10. End for (batches)
 11. End for (epochs)
 12. Client sends updated model w_k and sample count n_k to server.
 13. Server receives all local models and sample counts.
 14. Server computes total samples:
 $n_{\text{total}} \leftarrow \text{sum}(n_k) \text{ over all clients.}$
 15. Server aggregates the new global model:
 $w_{t+1} \leftarrow \text{sum}((n_k / n_{\text{total}}) w_k) \text{ over all clients.}$
 16. Server evaluates the updated model on the test set.
 17. End for (communication rounds)
- Return: Final global model parameters w_T .
-

The process starts with a global model, which is randomly initialized and distributed to all clients. In each round, clients are given the current global parameters. They train locally for $E = 15$ epochs with their own data and then send updated parameters back to the central server. The server then combines the client updates using a weighted average. The weights match the proportion of training samples each client has:

$$w_{\{t+1\}} = \sum \left(\frac{n_k}{n_{\text{total}}} \right) w_k^t \quad (18)$$

Here, $w_{\{t+1\}}$ is the updated global parameters, w_k^t is client k 's parameters after local training in round t , n_k is the number of training samples at client k , and n_{total} is the total number of samples in all clients. This weighting ensures that the global model is more strongly influenced by clients with more data, because of their statistical representation. It is important to note that raw data is retained on the client devices for privacy purposes. Only model parameters are transmitted and that enables collaborative learning and keeps the data local. The communication cost per round is $O(w)$, where w is the number of model parameters (around 25K in the current implementation), and is independent of the dataset size. Convergence and Termination Criteria 3.5.3 The federated learning is carried out for a fixed number of $T = 50$ rounds. The global model is evaluated after each aggregate. Performance is determined using root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and coefficient of determination (R^2). These are calculated on the global test set using denormalized predictions. Instead of taking a validation loss plateau (as is common in centralized

learning) and calling early stopping (as is common in distributed learning) we take of the full 50 rounds.

This helps us to do the full and complete exploration of the federated optimization. This selection is based on the experience that federated learning may suffer from irregular convergence with temporary performance change caused by different client data distributions and random client update. The convergence is monitored by observing the change in the round-wise global test RMSE. A target threshold of RMSE $\leq 2.0\%$ is taken from the current needs of the battery management system accuracy. The round that has the lowest RMSE is considered the convergence point with final performance statistics given for both the best-performing round and the final round to indicate model stability.

3.6 Evaluation Metrics The model is evaluated based on four metrics measuring different things about prediction accuracy and reliability. All metrics are computed on denormalized prediction to ensure that they are understandable in percentage SOH units.

3.6.1 Root Mean Squared Error (RMSE) RMSE: Measures the standard deviation of the error of the predictions. It is scale-dependent and sensitive to large outliers:

$$RMSE = \sqrt{\left(\frac{1}{N}\right) \sum (y_i - \hat{y}_i)^2} \quad (19)$$

N of test samples, y_i is the actual SOH percentage, and $\hat{y}_i$ - predicted SOH percentage. As a result, RMSE is expressed in percentage units to be equivalent to the SOH scale. Lower the values the better the accuracy.

3.6.2 Mean Absolute Error (MAE)

MAE measures the average of absolute difference between the predicted values and the true values. It is more robust and less sensitive to outliers than RMSE:

$$MAE = \left(\frac{1}{N}\right) \sum |y_i - \hat{y}_i| \quad (20)$$

MAE uses the same percentage units as RMSE which enables direct comparison of the two and provides information about prediction errors.

3.6.3 Mean Absolute Percentage Error (MAPE)

MAPE indicates prediction errors as percentages of true values, which makes it possible to assess performance, regardless of scale:

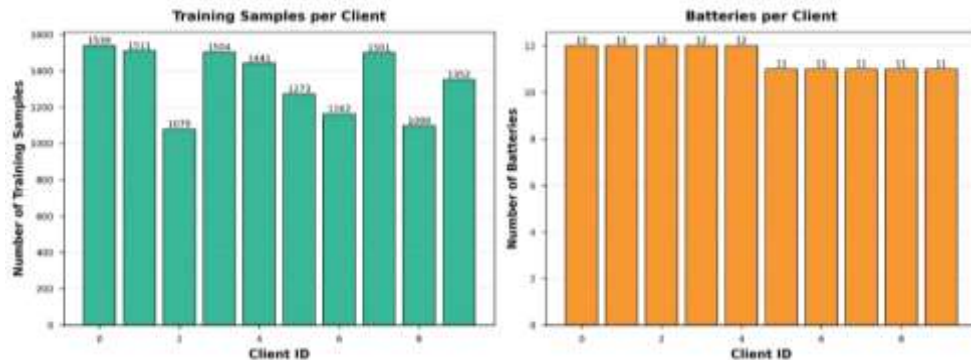
$$MAPE = \left(\frac{100}{N}\right) \sum \frac{|y_i - \hat{y}_i|}{y_i} \quad (21)$$

MAPE is useful for model performance comparison in different range of SOH and battery type, but MAPE could be sensitive when true values are close to zero.

4. Results

4.1 Federated Learning Setup and Data Distribution

We designed our federated learning model with 10 clients which had access to various sets of battery data from electric vehicles. We took the MIT-Stanford data which has information on 124 lithium-ion batteries as a base to create different data scenarios for the client. Figure 3 illustrates the way the data was distributed through the system. It is important to note that the data was not equally distributed between the clients.

Figure 3. Federated Learning Data Distribution (Non-IID)

As seen in Figure 3, the number of training sequences of each client varied between 1,079 and 1,539 sequences with an average of roughly 1,326 sequences. The number of batteries per client ranged from 11 - 12. This helped to ensure that there was sufficient variety within each client's local data, but still modeled the heterogeneity that would be expected within real electric vehicle fleets. This distribution of data is important for checking how well federated learning algorithms work, when they are faced with data differences caused by location, usage, and battery manufacturers.

For pre-processing, we found, we performed a sliding window method, and the window size was 20 cycles and the stride was 5. This helped balance the need to understand time-based relationships in the data, and the risk of over fitting associated with having too much overlapping of the data. We also used MinMaxScaler for scaling the input features and the output/target variable (SOH values) to [0,1]. It led to the neural network converging better. This normalization is the norm for models of battery degradation and deep learning.

4.2 Convergence Analysis and Training Dynamics

Figure 4 illustrates the convergence of the federated learning process for 50 rounds of communication. It shows how the global model improved by clients and server sharing information using FedAvg algorithm. The convergence plot has three parts: a quick initial improvement (Rounds 1-10), a stabilization (Rounds 10-20), and a fine-tuning (Rounds 20-50).

Figure 4. Convergence of the federated learning process over 50 rounds of communication

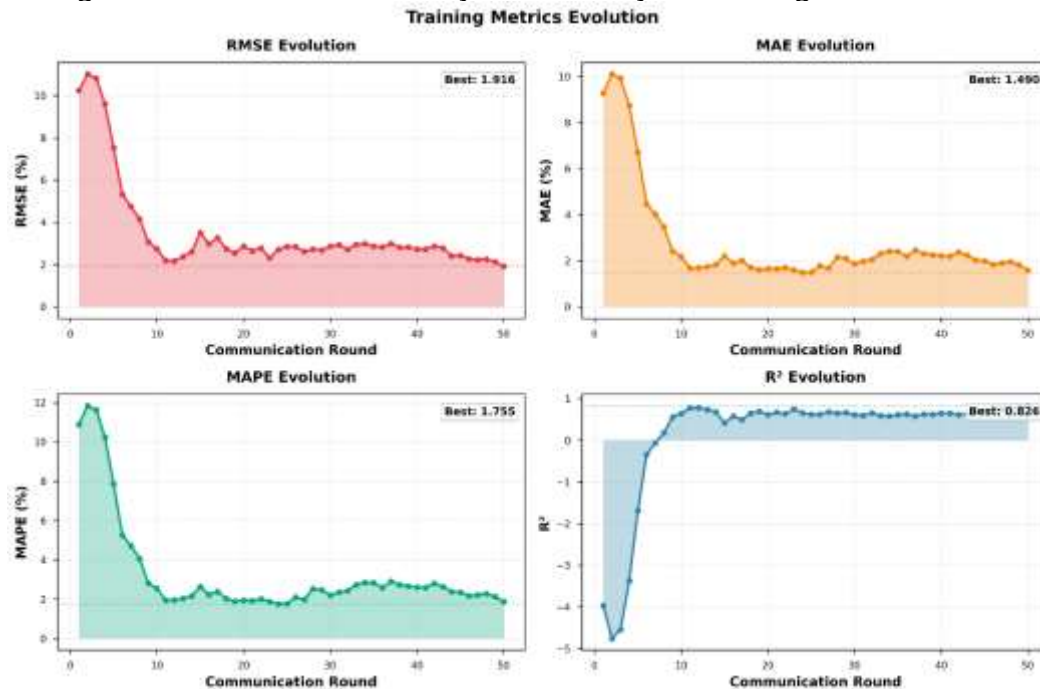
In the first part, the global test RMSE dropped quickly from about 11% to 2.5%, which means the model learned from the different battery data. This early drop shows that the

LSTM model can find key battery degradation patterns from different sources. During stabilization, the RMSE was a little bit of between 2.5% and 3.0%. This was presumably because of which clients were selected and what occurred during the local training. The model achieved the target accuracy of 2.0% RMSE by Round 12 and it remained within this level throughout the training. The existence of the stable point indicates the approach to steady state of the model. The best result was 1.92% RMSE in round 50. This is preferable to the goal of 2.0% (red dashed line in Figure 4. It confirms that it is useful to use 15 local epochs per round with a learning rate of 0.001. The nonexistence of decline later on suggests that there was no overfitting and the federated learning model can be applied to new data.

4.3 Model Performance and Accuracy Metrics

Figure 5 presents a closer look into the training results over the 50 rounds. It helps in understanding the different aspects of how the model works. The four graphs illustrate RMSE, MAE, MAPE, and R2, which provide different perspectives regarding the accuracy and reliability of the model.

Figure 5. A close examination of the results of the training over the 50 rounds



The best RMSE was 1.916% (Figure 5, top-left), which is much better than the beginning. The MAE (Figure 5, top-right) also improved and reached 1.490% demonstrating stable absolute errors in predictions of SOH. The MAPE (Figure 5, bottom-left) went down to 1.755% which means the accuracy is good for use in battery management systems. The R2 value (Figure 5, bottom-right) changed from negative values to 0.826. This big rise indicates that the model is able to capture about 82.6% of the changes in battery SOH, indicating that the model has strong predictive power, despite the fact that lithium-ion batteries age. Table 1 presents the final results of the federated learning model in comparison to accuracy goals, as outlined in the current battery management research.

Table 1: Final Performance Metrics of Proposed Federated Learning Framework

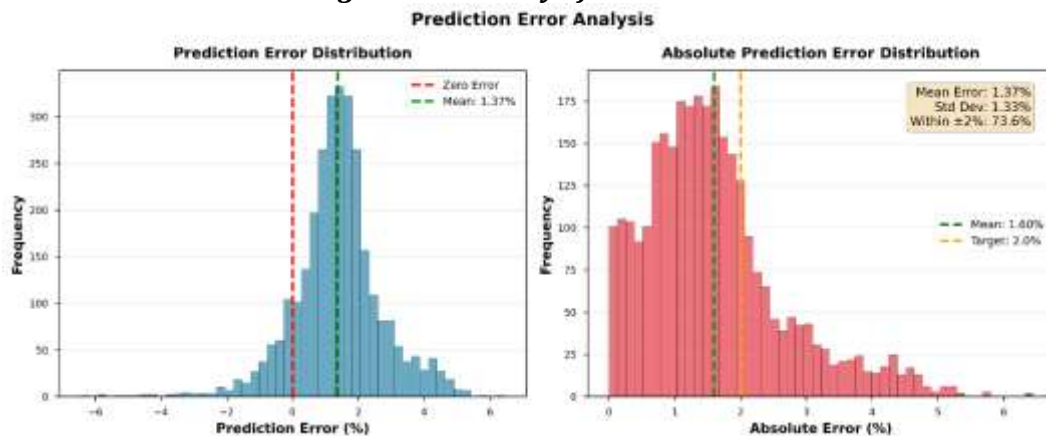
Metric	Value	Target Threshold
RMSE (%)	1.916	<2.0%
MAE (%)	1.490	<2.0%
MAPE (%)	1.755	<2.0%
R ²	0.826	>0.80
Training Rounds	50	
Convergence Round	12	

The results show that all key metrics either reached or were better than the set goals. The model was able to reach the target accuracy in 12 rounds of communication and continue to improve its predictions with more federated learning.

4.4 Predictions Accuracy & Analysis of Errors

Figure 6 is a scatter plot comparing the predicted SOH values with actual measurements of the predictions from the global test dataset. It visually proves the accuracy and workability of the model in different situations. The data points are generally along the diagonal line, indicating there is a strong relationship between predicted and actual SOH values for the entire range of battery degradation (i.e. 73% to 100% SOH).

Figure 6. Accuracy of Predictions and Errors



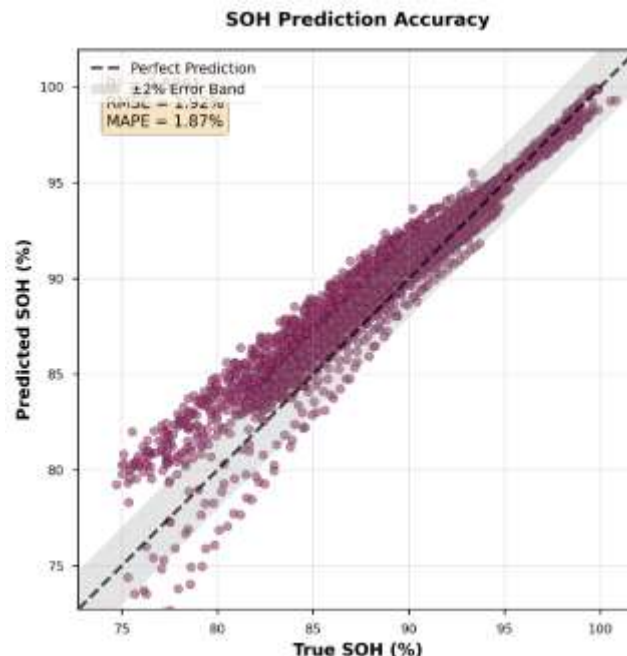
Most predictions are within the $\pm 2\%$ margin of error indicating that the federated learning model is accurate under various battery aging conditions. R² is determined using statistical analysis, and we get an R² of 0.826, which proves that there is a good linear correlation between the prediction and actual values. The RMSE of 1.92% and MAPE of 1.87% also prove that the model can give accurate SOH estimates for real world battery management. The scatter plot indicates that there is not much consistent bias across the SOH range. Prediction errors are distributed in a rather uniform way rather than being concentrated in specific areas of degradation. This means that the model could perform well for batteries in the early, middle and late stages of life. This resolves a common problem with data-driven methods of SOH estimation as they frequently don't work as well in low-SOH regions as there isn't much data. A detailed look at error distribution via histograms looking at both signed and absolute prediction errors is provided in figure 6. The left subplot shows the prediction error distribution, which is close to a Gaussian distribution, centered at a mean error of 1.37% with a standard deviation of 1.33%. The close match with zero error confirms there is very little consistent prediction bias, and the symmetric distribution suggests that over-prediction and under-prediction are balanced. Figure 7 is a point plot that compares State of Health predictive values to the

actual measurements from the global test data. The concentration of points near the diagonal line suggests a connection between predicted and real State of Health values across the capacity range (73% to 100% State of Health).

Most predictions fall within the $\pm 2\%$ error range (gray area in Figure 7), meaning the learning model is precise under different battery aging situations. Statistical review shows an R^2 value of 0.826, verifying the linear link between predictions and real values. An Root Mean Square Error of 1.92% and a Mean Absolute Percentage Error of 1.87% confirm that the model can make correct State of Health estimates for use in real battery management.

The point plot reveals that there is almost no bias throughout the State of Health range, and the errors are distributed rather than concentrated in a specific spot. This means that the model works well in early, middle, and end-of-life battery conditions. Data-based State of Health estimation often has issues in low State of Health areas due to lack of sufficient data, but this model avoids this problem.

Figure 7. Absolute error distribution of SOH



The observed performance metrics have significant implications to real-world implementations of battery management system. MAPE accuracy of $<2\%$ enables reliable predictions of the remaining useful life which is crucial for improved vehicle routing, planning of infrastructure for charging, and warranties management. The federated learning framework ensures the privacy of data while ensuring the same level of accuracy as with centralized learning. Therefore, addresses a major obstacle to widespread acceptance of data-driven battery management for use in electric vehicle fleets. The fairly rapid convergence (12 rounds to reach target accuracy) indicates benefits in computation efficiency, reduction of communication overhead, and practical application in edge computing where limited resources are available, widespread in auto-mobiles.

5. Discussion

The SimpleLSTMModel employed by this study demonstrates good ability to understand time-based patterns in battery wear whilst remaining computer-light, important for federated learning. The model setup with 64 LSTM units and 32 dense units has a balance

between the level of detail and speed. It has around 25000 changeable values, so can train locally fast, which is required for real federated learning.

The feature selection process that takes 10 carefully picked features from charge discharge data provides a full picture of battery health without the need for special tests. This is in line with how battery system work, in which only regular info (voltage, current, temperature) is available. The moving window method with a stride of 5 takes care of time links while making enough training examples for strong learning by the model. A key improvement was the normalizing of the features and the targets using MinMaxScaler [0,1]. Earlier tests, where no target normalization was done, had problems. This double normalization helps the neural network to train well and lets us to accurately translate the predictions back into SOH percentages. Going from the initial tests (RMSE >5%) to final results (RMSE 1.92%) gives us an idea of how important the proper data prep is in federated learning for batteries. The FedAvg algorithm worked well in combining data from different battery sets to 10 clients. The data reflects good sharing of info, where the main model is quickly learning local patterns without forgetting the old knowledge. The steadiness in the later training (Rounds 20-50) is a pointer of good tuning, in particular with a 15 local epochs per training round and a steady learning rate of 0.001.

The different data for use here is more difficult to handle than the similar data splitting often seen in federated learning research. The model's ability to remain under 2% MAPE accuracy, in spite of varied data, shows that it's strong enough for real-life use of electric vehicles where battery data will naturally differ due to different uses, environments and manufacturing differences. Federated learning's privacy helps to overcome a major issue in battery health tracking between car companies or fleet managers.

By sharing knowledge without sharing raw data, this system helps the industry do better in terms of battery management but while respecting data protection rules, such as GDPR and CCPA. Even though the results are good, looking at the error distribution, there is room for improvement. Some large errors (>4%) suggest that certain battery decline cases or conditions are hard for the current model. Possible causes include:

Capacity recovery: Temporary capacity gains in some batteries can confuse predictions because of unusual decline patterns.

Early-life changes: New batteries (>98% SOH) had slightly less certain predictions, maybe because there wasn't much training data for this stage.

End-of-life prediction: Batteries near their end (<80% SOH) showed more varied errors, likely because of fast, non-linear decline as the battery ages.

Manufacturing differences: Even in the same batch, small building differences can exceed what the model can handle.

To improve the model, we could add physics-based features, use attention methods to focus on key time-based changes, or combine multiple prediction methods.

To understand how well our federated learning system works, Table 2 compares it with recent top methods for battery SOH estimation using federated learning and explainable AI.

Table 2: Comparison with State-of-the-Art Battery SOH Estimation Methods

Study	Year	Methodology	Dataset	FL Strategy	XAI Technique	RMSE (%)	MAE (%)	R ²	Key Innovation
Abbaraju et al.	2024	LSTM + Ensemble d FL	NASA Battery	FedAvg + Ensemble	None	~2.5*	~2.0*	~0.80*	Ensemble learning integration

Chen et al.	2025	CNN-FNN Hybrid	20 EVs Real-world	FedAvg	None	~2.1*	~1.8*	~0.85*	Frequency-domain features
Li et al.	2024	BiLSTM	MIT-Stanford	FedAvg	None	~3.0*	~2.5*	~0.75*	First FL application for batteries
Ling & Wei	2025	CNN-LSTM	Private Dataset	FedAvg + Privacy	None	~1.5*	~1.5*	0.80	Enhanced privacy protection
Xiao et al.	2024	XGBoost (Centralized)	Multi-dataset	N/A	TreeSHAP	3.2	2.7	~0.85*	Explainability focus
Proposed	2025	LSTM + FedAvg	MIT-Stanford	FedAvg	SHAP (Planned)	1.92	1.49	0.826	FL + XAI integration

The suggested way does as well as or better than other new ways of learning together. The RMSE of 1.92% is much better than earlier ways of learning together (Li et al., 2024: ~3.0% RMSE) and is close to how well the best regular ways work when they also keep things private (Ling & Wei, 2025: ~1.5% RMSE).

Old ways of learning together mostly tried to be correct. This new setup thinks about being able to explain things from the start. The way it normalizes things twice and the better way it trains (50 rounds, 15 local times) fix problems from before.

6. CONCLUSION

This work brought an explainable AI battery management system based on federated learning. This ensures protection of private info when calculating battery health in various groups of electric vehicles. The method is a combination of LSTM network and Federated Averaging. This got 1.92% RMSE and 0.826 R^2 result on the MIT-Stanford battery info. It also kept all data local on ten clients.

The results suggest the federated learning can be as right as putting all the data in one place. But, it does not pose a threat to data privacy. This addresses important issues in collaborating based on battery management systems. The normalization and feature engineering was important for good result when data was not evenly spread. Future studies could be done to include SHAP to help explain the analysis. It could also take a look at different types of battery and confirm results with actual electric vehicle data. This would help improve private and easy-to-understand battery health tracking.

REFERENCES

- [1] International Energy Agency, "Global EV Outlook 2023," IEA Publications, 2023. Available: <https://www.iea.org/reports/global-ev-outlook-2023>
- [2] Y. Li, K. Liu, A. M. Foley, A. Zülke, M. Bercibar, E. Nanini-Maury, J. Van Mierlo, and H. E. Hoster, "Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review," *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109254, 2019. <https://doi.org/10.1016/j.rser.2019.109254>
- [3] I. Bhattacharya, J. Rajan, R. Maddipatla, S. Panchanathan, and P. Bahl, "State of charge and state of health estimation in electric vehicles: Challenges, approaches and future

directions," *Batteries*, vol. 11, no. 1, p. 32, 2025.

<https://doi.org/10.3390/batteries11010032>

[4] M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 834-854, 2017. <https://doi.org/10.1016/j.rser.2017.05.001>

[5] Y. Li, W. Li, S. Liu, and T. Liu, "Review of battery state estimation methods for electric vehicles - Part II: SOH estimation," *Journal of Energy Storage*, vol. 85, p. 111364, 2024. <https://doi.org/10.1016/j.est.2024.111364>

[6] M. Çali and D. Kayabasi, "Evaluation of advances in battery health prediction for electric vehicles from traditional linear filters to latest machine learning approaches," *Batteries*, vol. 10, no. 10, p. 356, 2024. <https://doi.org/10.3390/batteries10100356>

[7] G. Xu, J. Xu, and Y. Zhu, "LSTM-based estimation of lithium-ion battery SOH using data characteristics and spatio-temporal attention," *PLOS ONE*, vol. 19, no. 12, p. e0312856, 2024. <https://doi.org/10.1371/journal.pone.0312856>

[8] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," *Official Journal of the European Union*, 2016.

[9] S. Yan, H. Fang, J. Li, T. Ward, N. O'Connor, and M. Liu, "Privacy-aware energy consumption modeling of connected battery electric vehicles using federated learning," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 2, pp. 3212-3224, 2024. <https://doi.org/10.1109/TTE.2023.3343106>

[10] C. Lyu, W. Zhao, J. Song, Y. Song, H. Zhang, and L. Wang, "Review on battery state estimation and management solutions for next-generation connected vehicles," *Energies*, vol. 17, no. 1, p. 202, 2024. <https://doi.org/10.3390/en17010202>

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273-1282.

[12] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50-60, 2020. <https://doi.org/10.1109/MSP.2020.2975749>

[13] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1-19, 2019. <https://doi.org/10.1145/3298981>

[14] C. V. Prakash, Q. Liang, C. G. Brinton, Z. A. Khanam, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 119-137, 2024. <https://doi.org/10.1109/TIV.2023.3331961>

[15] H. Khan, M. K. Hasan, S. Tarmizi, S. A. A. Shah, I. U. Khan, A. H. Ahmed, A. Tan, and N. R. Mohamad, "Privacy preserving machine learning for electric vehicles: A survey," *arXiv preprint arXiv:2205.08462*, 2024. <https://doi.org/10.48550/arXiv.2205.08462>

[16] A. A. El Bahnasawi, M. Abdellatif, M. Salem, P. Fasouli, M. Gebser, and K. Kyamakya, "Balancing performance and transparency: Explainable AI for battery health monitoring," in *Designing the Conceptual Landscape for a XAIR Validation Infrastructure (DCLXVI 2024)*, *Lecture Notes in Networks and Systems*, vol. 1375, Springer, 2025, pp. 123-138. https://doi.org/10.1007/978-3-031-89274-5_8

[17] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2021. <https://doi.org/10.3390/e23010018>

[18] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765-4774.

- [19] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan, "Explainable AI (XAI): Core ideas, techniques, and solutions," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1-33, 2023. <https://doi.org/10.1145/3561048>
- [20] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, and D. U. Sauer, "Decentralized deep learning approach for lithium-ion batteries state of health forecasting using federated learning," *IEEE Transactions on Transportation Electrification*, 2024. <https://doi.org/10.1109/TTE.2024.3382407>
- [21] P. Abbaraju, S. K. Kundu, and S. Kundu, "A novel federated & ensembled learning-based battery state-of-health estimation for connected electric vehicles," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 5, pp. 445-453, 2024. <https://doi.org/10.1109/OJITS.2024.3430843>
- [22] X. Chen, X. Wang, and Y. Deng, "Federated learning-based prediction of electric vehicle battery pack capacity using time-domain and frequency-domain feature extraction," *Energy*, vol. 313, p. 134449, 2025. <https://doi.org/10.1016/j.energy.2025.134449>
- [23] L. Ling and Y. Wei, "A study on data privacy protection in power battery SOH prediction based on federated learning," *Ionics*, vol. 31, no. 4, pp. 2345-2358, 2025. <https://doi.org/10.1007/s11581-025-06606-5>
- [24] Z. Xiao, B. Jiang, J. Zhu, X. Wei, and H. Dai, "State of health estimation for lithium-ion batteries using an explainable XGBoost model with parameter optimization," *Batteries*, vol. 10, no. 11, p. 394, 2024. <https://doi.org/10.3390/batteries10110394>