Lithium-Ion Battery Aging Estimation with Federated Deep Learning

Luan L. Santos *, Guilherme A. Thomaz*, Lucas A. C. Souza *, Marcelo L. D. Lanza*,
Robson F. S. Dias *, Luís Henrique M. K. Costa *, Miguel Elias M. Campista *
Universidade Federal do Rio de Janeiro (UFRJ)*
Email: {luansantos, dias}@lafae.ufrj.br, {guiaraujo, airam, lanza, luish, miguel}@gta.ufrj.br

***Abstract*—A reliable method for predicting the State of Health (SOH) of lithium-ion batteries is crucial for extending their lifespan. However, using traditional machine learning models for this task requires large, centralized datasets, which creates significant privacy issues. This work proposes a federated learning (FL) framework with a temporal autoencoder architecture that estimates SOH directly from partial charging cycles, removing the need for full capacity curves.**

Our systematic evaluation against a centralized baseline revealed a trade-off between accuracy and privacy. The centralized model achieved superior performance, while the FL model showed a 25.9% higher loss and a 10.9% higher Mean Absolute Error (MAE). However, increasing the number of clients in the FL setup slowed model training convergence. These results validate FL as a practical, privacy-preserving solution for SOH prediction, despite a reduction in performance.

***Keywords*—lithium-ion batteries, federated learning, machine learning, deep learning**

I. Introduction

The use of Lithium-Ion Batteries (LIB) introduces challenges related to their lifespan, as batteries are typically discarded once they reach 80% of their initial capacity [1]. This not only raises environmental concerns due to the significant natural resources required, but also highlights the need to extend battery lifetime, ensuring consistent energy delivery even at capacities below the current threshold [2]. The aging mechanisms of batteries, including the capacity “knee,” are multifaceted and described in detail in [3]. Factors such as lithium plating, solid–electrolyte interphase growth, mechanical deformation, and other mechanisms, arising from mechanical, chemical, or thermal causes, contribute to this phenomenon [4]. Modeling each of these aging mechanisms individually is a complex and computationally demanding task, requiring deep expertise in chemical engineering [5].

To address this complexity, the initial approach is to employ deep learning techniques to estimate the state of health of the cell. This approach requires a substantial amount of data to train a deep neural network capable of capturing the subtle ways in which batteries age. In real applications, there are concerns not only about estimating battery aging but also about the process by which such estimation occurs, since operational safety is critical in real-world deployments [6]. This implies keeping data private as it supports safe deployment by preserving sensitive data [7], though restricting data access complicates traditional machine learning models training [8]. Therefore, to address this issue, Federated Learning (FL) emerged as a solution [9], enabling model training without sharing input data and output labels. Instead, federated learning requires only sharing model parameters. Nevertheless, it introduces new challenges such as communication overhead, reduced performance compared to centralized training, and gradient inversion attacks [8].

For the evaluation of these aging prediction approaches, a generic metric used to assess battery aging, without identifying the primary mechanism responsible for capacity loss, is the state of health (SOH). It is computed as:

$$\mathrm{SOH}(\%) = \frac{C_{current}}{C_{nominal}} \times 100, \tag{1}$$

where $C_{current}$ is the LIB remaining capacity in the current cycle and $C_{nominal}$ is its nominal capacity.

In [10], the authors propose a FL encoder-decoder LSTM to predict full capacity degradation. in different chemistries for assessing its generalization ability. This approach attained a mean absolute percentage error (MAPE) lower than 5%, but requires capacity curve information that limits applicability in real-world scenarios due to the lack of periodic reference performance tests. The proposal of [11] employs a convolutional neural network (CNN) coupled with a channel attention module combining FL pre-training with transfer learning customization. Although this work attains a MAPE of 0.81% across two datasets it requires knowledge of the battery capacity curve. The work [12] tackles aging prediction accounting for periodic fluctuations in battery capacity. It employs a temporal autoencoder coupled with an attention mechanism, federated with a contribution-aware strategy considering data volume and local performance. The work attains a Mean Absolute Error (MAE) of 0.87%, but it only evaluates four lithium-ion battery cells from the same dataset. In [13], the authors address privacy issues by combining a CNN-LSTM health predictor federated with privacy-preserving techniques, singular value decomposition, and hierarchical noise injection. This framework achieves a coefficient of determination of 0.8035, improving capacity prediction performance over the FL baseline, even though it lacks evaluation of generalization across different chemistries.

In the context of FL, a common issue for LIB SOH prediction is the lack of privacy-preserving approaches. The aforementioned works rely on knowledge of the capacity curve, limiting the deployment of these models in real-world applications, where periodic reference performance tests are rarely available. In summary, the main contributions of this work are:

- a temporal autoencoder architecture to estimate SOH directly from partial charging cycle data, removing the dependence on full capacity curves required by previous FL approaches;
- a systematic federated learning evaluation with comparison against centralized and baseline models.

This paper is organized as follows. Section II describes the centralized machine learning algorithm and the FL implementation. The experimental results are presented in Section III. Section IV concludes this study and provides future directions.

II. Federated learning for battery aging

This section details the methodology for training a deep learning model in an FL context to predict battery aging. The approach is structured into five main components: data pre-processing, the proposed machine learning architecture, the implemented federated learning algorithm, the evaluation metrics, and the experimental setup. We use a publicly available dataset of lithium-ion battery cells to demonstrate the effectiveness of the model. The dataset used aging experiments with cells of identical specifications [4], summarized in Table I. These cells received the same aging profile under consistent conditions. The investigated battery cells are the commercially available Sanyo/Panasonic UR18650E cylindrical cells, manufactured using a well-established process. These cells use a carbon anode and Nickel-Manganese-Cobalt (NMC) as the cathode material.

A. Data pre-processing

The data preprocessing pipeline prepares the raw battery data for machine learning model training. The process begins with data cleaning and feature engineering. Initially, raw current and voltage measurements are clamped to the battery's specified operating range to remove sensor outliers. The capacity for each full cycle is then calculated by integrating the current over time. We filter out any cycles with physically impossible capacity values or those that are incomplete, ensuring data integrity. Next, we address the variable-length time-series data. The first-life data, defined as the period during which the battery's SOH degrades from 100% to 80%, is extracted for each cell, as shown in Figure 1. However some cells did not achieve the inferior limit due to testing time limit.

TABLE I: Specifications of the battery cell in the dataset.

Battery Parameters	**Specifications**
Type	Cylindrical
Positive Electrode	$LiNiMnCoO_2$
Negative Electrode	carbon
Nominal Capacity	1.85 Ah
Nominal Voltage	3.9 V
Charge Cutoff Voltage	4.0 V
Discharge Cutoff Voltage	3.5 V
Cutoff Current	40 mA

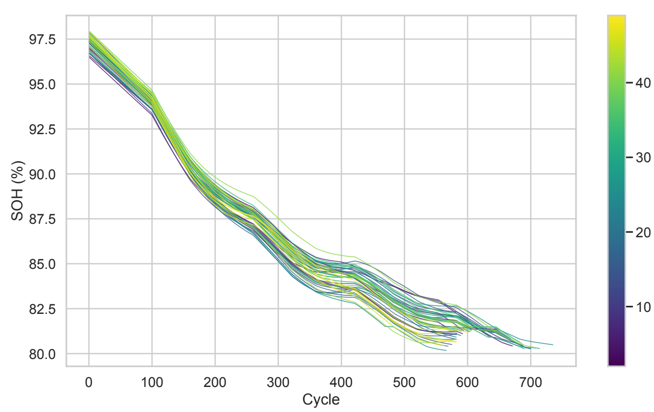

Fig. 1: SOH vs. Cycle for all cells.

To ensure uniform input dimensions for the model, all sequences are truncated to a standard length. This length is defined as the average length of the constant current (CC) section of the charging cycle, calculated across all cells. The model's input features are the voltage (Figure 2), current (Figure 3), and temperature (Figure 4) measurements drawn from this specific truncated CC section.

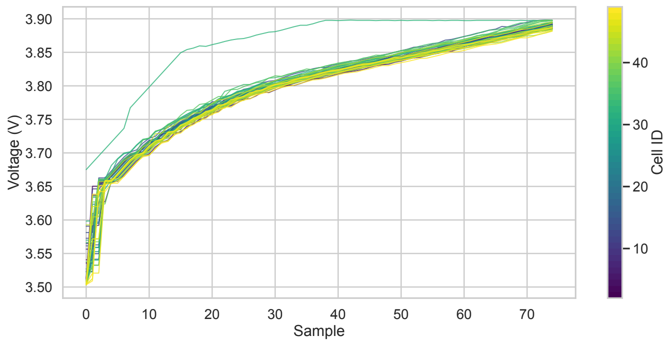

Fig. 2: Voltage vs. Sample for all cells.

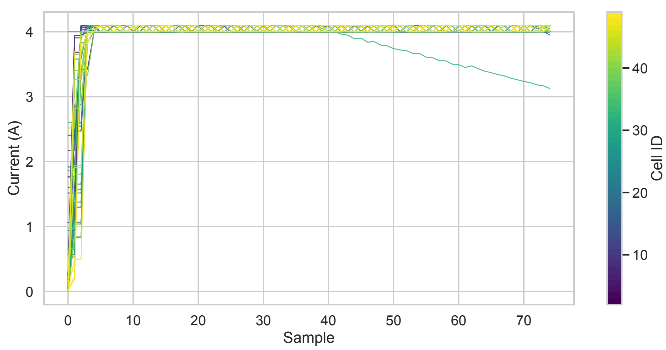

Fig. 3: Current vs. Sample for all cells.

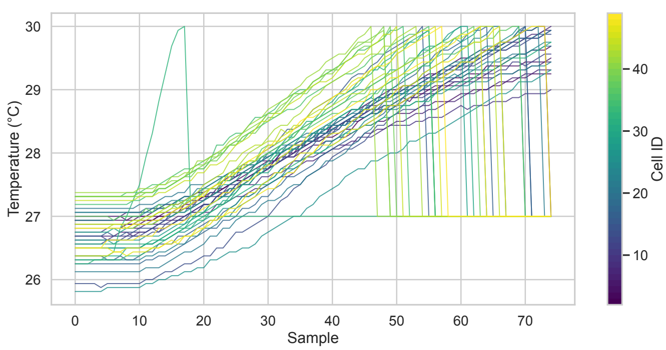

Fig. 4: Temperature vs. Sample for all cells.

Since each the first life of each battery ends after a different number of cycles, and each cycle has a different duration, we perform right padding with zeros on both the input feature sequences and the target SOH sequences. This aligns all sequences to the same length, a necessary step for batch processing and training for most machine learning models.

Finally, the preprocessed input features are normalized using Min-Max scaling to a range of $[0, 1]$. This scaling step prevents features with larger magnitudes from dominating the learning process. The target SOH curve is smoothed with a rolling average window of size of 50, selected by a sensitivity analysis to reduce noise and highlight the underlying degradation trend.

B. Machine Learning Architecture

The proposed machine learning model uses a deep learning architecture designed to estimate the SOH of a battery over its lifetime. The architecture is composed of two main stages: an intra-cycle feature extractor and an inter-cycle sequence model. The architecture's layers and corresponding output dimensions are summarized in Table II.

TABLE II: Architecture of the SOH estimator.

#	Layer	Output	Parameters
1	Input	(B, T, P, F)	–
2	Masking	(B, P, F)	value=0
3	GRU	$(B, 32)$	units=32
4	MLP	$(B, 64)$	ReLU
5	Time distributed	$(B, T, 64)$	–
6	LSTM	$(B, T, 128)$	units=128
7	MLP	$(B, T, 64)$	linear
8	MLP (SOH)	$(B, T, 1)$	linear

1) Intra-cycle Feature Extraction: The model's input is a tensor $X \in \mathbb{R}^{B\times T\times P\times F}$, where B is the batch size, T is the number of cycles (timesteps), P is the number of points per cycle ($P = 75$), and F is the number of features per point ($F = 3$, corresponding to voltage, current, and temperature).

The first stage processes each cycle independently to extract a representative feature vector. For a given cycle $t \in \{1, \ldots, T\}$, the input is a matrix $X_t \in \mathbb{R}^{B\times P\times F}$. A masking layer is applied to handle the variable length of cycles, effectively setting padded values to zero and preventing them from influencing the subsequent layers.

The masked input for each cycle is then passed to a Gated Recurrent Unit (GRU) layer [14]. The GRU processes the sequence of points within each cycle to produce a fixed-size feature vector. The output of this GRU for a single cycle is a hidden state $h_t \in \mathbb{R}^{B\times 32}$, where 32 is the number of units,

$$h_t = \text{GRU}(X_t). \tag{2}$$

This feature vector h_t is then fed into a two-layer Multi-Layer Perceptron (MLP) [15]. The first MLP layer has a ReLU [16] activation function and an output dimension of 64,

$$z_t = \sigma(\mathbf{W}_1 h_t + \mathbf{b}_1), \tag{3}$$

where σ is the ReLU function. The second MLP layer is a linear layer. Its function is described in the next stage. A Time Distributed layer is used to apply the GRU and the first MLP to each cycle in the sequence, producing a sequence of feature vectors $Z \in \mathbb{R}^{B\times T\times 64}$.

2) Inter-cycle Sequence Modeling: The second stage of the architecture models the temporal dependency between cycles. The sequence of feature vectors Z is fed into a Long Short-Term Memory (LSTM) layer [17]. This layer learns the long-term degradation patterns of the battery by processing the sequence of cycle-level features. The LSTM's output is a sequence of hidden states $H \in \mathbb{R}^{B \times T \times 128}$, where 128 is the number of units,

$$H = \text{LSTM}(Z). \tag{4}$$

Finally, the output of the LSTM is passed through two subsequent MLP layers to predict the SOH for each timestep. The first MLP layer is a linear layer with an output dimension of 64. The final MLP layer is a linear layer with a single output neuron, predicting the SOH value for each cycle,

$$\hat{y} = \text{Linear}_2(\text{Linear}_1(H)), \tag{5}$$

where $\hat{y} \in \mathbb{R}^{B \times T \times 1}$ is the predicted SOH curve for the batch. The mean squared error (MSE) is used as the loss function to train the model, comparing the predicted SOH values $\hat{y}$ with the ground-truth SOH values y,

$$\mathcal{L}_{\text{MSE}} = \frac{1}{BT} \sum_{i=1}^{B} \sum_{t=1}^{T} (\hat{y}_{i,t} - y_{i,t})^2. \tag{6}$$

C. Federated Learning (FL) Framework

The FL framework is implemented to collaboratively train a global battery aging prediction model without sharing raw data from individual clients. The process, detailed in Algorithm 1, is an iterative cycle of local model training on client devices and a central server aggregation.

The process is initiated by a central server that maintains and updates the global model. At the start of each global communication round r, the server selects a fraction of clients, $m = f \cdot n$, to participate. The server then dispatches the current global model weights, $\Theta_r = w_r$, to each of these selected clients.

The core of the FL process lies in the client function. Each client c, upon receiving w_r, performs local optimization for e_{local} epochs using its private dataset P_c. The local training uses the ADAM optimizer. The local learning rate, η_{local}, is initially set to $5 \cdot 10^{-5}$ for the first 5 global rounds to ensure stable convergence and then aligns with the global learning rate, η_{global}. This local optimization step produces updated weights, w_{r+1}^c, which are then sent back to the server.

Upon receiving the updated weights from all participating clients in the round, the server aggregates them to form the new global model. The aggregation is performed using a weighted average, as shown in the Server function of the algorithm. The weight for each client's model is proportional to the size of its local training data, N_c, relative to the total number of samples from all participating clients, N_p. The updated global model weights $\Theta_{r+1} = w_{r+1}$ are calculated as:

$$w_{r+1} = \sum_{c=1}^{m} \frac{N_c}{N_p} w_{r+1}^c. \tag{7}$$

This iterative process, repeated for e_{global} global epochs, allows the global model to learn from the distributed data while ensuring client data privacy.

Algorithm 1 Client-side optimization and server-side aggregation within the FL framework for the battery aging prediction.

Require: n ▷ Number of clients
Require: f ▷ Client fraction
Require: b ▷ Local minibatch size
Require: e_{global} ▷ Global epochs
Require: e_{local} ▷ Local epochs
Require: η_{global} ▷ Global learning rate
Require: N_p ▷ Number of training samples of all clients
Require: N_c ▷ Number of training samples of client c
function Server($n, f, b, e_{global}, e_{local}, \eta_{global}, N_c, N$)
  initialize w_0
  for $r \in \text{range}(e_{global})$ **do**
    $m \leftarrow \max(f \cdot n, 1)$
    $S_r \leftarrow$ random set of m clients
    for all $c \in S_r$ **in parallel do**
      $w_{r+1}^c \leftarrow \text{Client}(c, w_r, \eta_{global})$
    end for
    $w_{r+1} \leftarrow \sum_{c=1}^{S_r} \frac{N_c}{N} w_{r+1}^c$
  end for
end function
function Client(c, w_r, η_{global})
  $\eta_{local} \leftarrow [5 \cdot 10^{-5}, \eta_{global}]$
  $\mathcal{B} \leftarrow$ split client data P_c into batches of size b
  for $g \in \text{range}(e_{local})$ **do**
    for all batch $k \in \mathcal{B}$ **do**
      $w_{r+1} \leftarrow \text{ADAM}(w_r, \eta_{local})$
    end for
  end for
end function

D. Figures of Merit

The performance of the machine learning model is evaluated using two primary metrics: the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). These metrics quantify the model's ability to accurately reproduce the battery SOH curve.

The MSE, as defined in Equation 8, measures the average squared difference between the predicted and actual values. Due to the squaring term, MSE is particularly sensitive to large prediction errors.

$$\text{MSE} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2 \tag{8}$$

The MAE, defined in Equation 9, measures the average of the absolute differences between the predicted and actual values, providing a measure of error in the same units as the target variable.

$$\text{MAE} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |y_i - \hat{y}_i| \quad (9)$$

In Equations 8 and 9, N_{test} represents the number of observations in the test set, y_i is the actual SOH value, and $\hat{y}_i$ is the predicted SOH value for the i-th observation. The models were optimized to minimize both MSE and MAE to achieve high prediction performance.

E. Experimental setup

The evaluated architectures were implemented in a computer with a i9-13900K CPU, 128GB DDR4 RAM, and an RTX 3060 GPU. The software was implemented in a Python 3 environment, using TensorFlow 2.20 [18] and its high-level API, Keras [19]. 31 cells were divided between 2, 4, 6, 8 and 10 clients for training, and every client is selected at each round ($f = 1$). Since the number of cells is not divisible by the number of clients, one client uses less cells. 9 cells were separated for testing at the server after each aggregation.

III. Experimental Results

This section presents the results of centralized learning, which serves as the performance baseline for the LIB SOH estimation model. This baseline is then compared with its FL counterpart. In addition, the effect of the number of clients on FL convergence is analyzed.

Table III shows the evaluation on a test set of 9 cells. The centralized model achieves lower error for both metrics. The loss error is 25.9% higher for the FL model, while the MAE is 10.9% higher. These results indicate that centralized learning provides better accuracy. This performance gap arises because the FL model only exchanges local model parameters, without having direct access to the complete dataset.

TABLE III: Loss, MAE, and percentage of relative errors compared with centralized learning.

Method	**Loss**	**MAE**	**Loss Er.**	**MAE Er.**
Centralized	0.00640	0.0451	–	–
Federated	0.00806	0.0500	25.9	10.9

We now analyze the impact of the number of clients in the FL setup. Fig. 5 shows the training loss convergence. With two clients, the model converges fastest, though it starts from a higher initial loss. As the number of clients increases, the initial loss decreases, but the convergence becomes slower, requiring more rounds. The 6-client scenario appears to be the slowest, taking the highest number of rounds to converge compared with the other distribution scenarios.

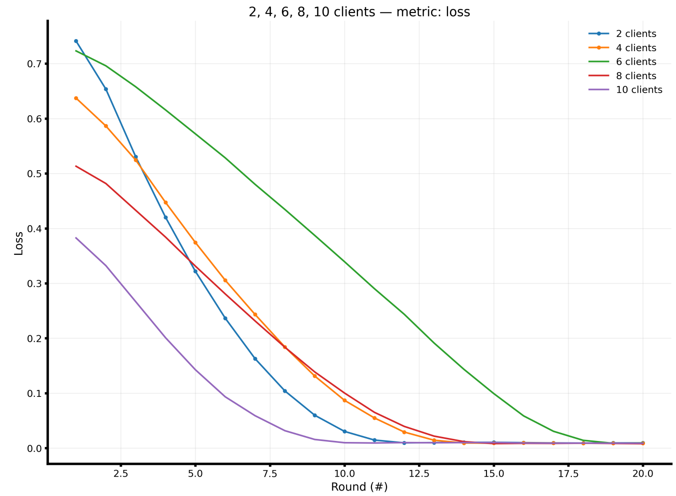

Fig. 5: Training loss convergence for different numbers of clients in FL.

Fig. 6 presents the MAE convergence under the same conditions, showing a similar trend. The slower convergence with more clients is likely due to data partitioning: as more clients participate, each client has less local data available for its training process. This can lead to less stable or representative local updates, requiring more global aggregation rounds to converge. The 6-client setup again demonstrates the slowest convergence. However, this effect was observed while using batteries with similar aging trajectories (homogeneous data). In scenarios with more heterogeneous aging patterns, increasing the number of clients might not lead to the same convergence behavior.

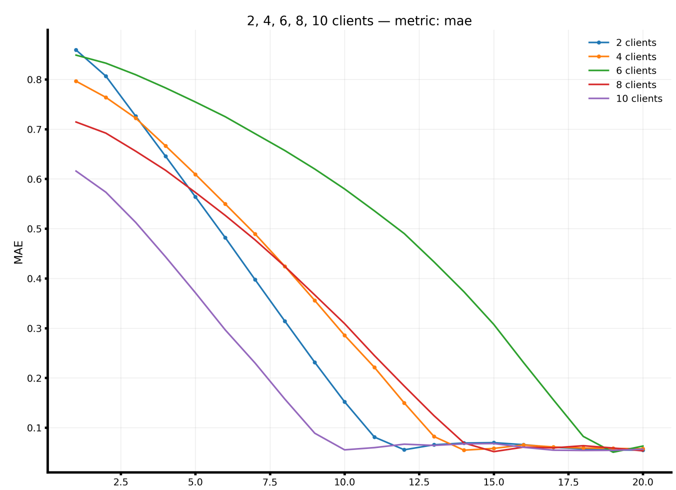

Fig. 6: Training MAE convergence for different numbers of clients in FL.

In summary, FL preserves data locality while attaining performance comparable to the centralized model.

Increasing the number of clients slows FL convergence speed when the training data is partitioned among them. Furthermore, the benefits of adding more clients are dependent on the similarity of battery degradation patterns. These results highlight the trade-off between accuracy, privacy, and convergence speed when deploying SOH prediction models in federated environments.

IV. Conclusions and Future Work

The federated learning model shows lower accuracy compared to the centralized model. Its training convergence degrades as more clients participate, since each client has less data available for training. Nevertheless, the effect of clients with heterogeneous aging trajectories remains unclear. Future work will investigate how varying degradation patterns affect FL convergence and predictive accuracy. This analysis will help to define strategies to improve FL robustness for diverse battery datasets.

Acknowledgment

This work was partially sponsored by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Funding Code 001 and 88887.954253/2024-00, CNPq (310234/2025-5, 408255/2023-4, 405940/2022-0, and 309304/2021-0), FAPERJ (E-26/200.380/2023 and E-26/204.122/2024), FAPESP (2023/00673-7 and 2023/00811-0), Development and Research Foundation - Fundep - Rota 2030, and our partners Stellantis and Mobway.

References

[1] M. Chen, X. Ma, B. Chen, R. Arsenault, P. Karlson, N. Simon, and Y. Wang, "Recycling end-of-life electric vehicle lithium-ion batteries," *Joule*, vol. 3, no. 11, pp. 2622–2646, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S254243511930474X

[2] J. Zhu, W. Xu, M. Knapp, M. S. Dewi Darma, L. Mereacre, P. Su, W. Hua, X. Liu-Théato, H. Dai, X. Wei, and H. Ehrenberg, "A method to prolong lithium-ion battery life during the full life cycle," *Cell Reports Physical Science*, vol. 4, no. 7, p. 101464, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666386423002436

[3] P. M. Attia, A. Bills, F. B. Planella, P. Dechent, G. dos Reis, M. Dubarry, P. Gasper, R. Gilchrist, S. Greenbank, and D. Howey, "Review—"knees" in lithium-ion battery aging trajectories," *Journal of The Electrochemical Society*, vol. 169, no. 6, p. 060517, 06 2022.

[4] A. J. Warnecke, "Degradation mechanisms in nmc-based lithium-ion batteries," Ph.D. dissertation, RWTH Aachen University, Aachen, Germany, 09 2017.

[5] T. Lombardo, M. Duquesnoy, H. El-Bouysidy, F. Årén, A. Gallo-Bueno, P. B. Jørgensen, A. Bhowmik, A. Demortière, E. Ayerbe, F. Alcaide, M. Reynaud, J. Carrasco, A. Grimaud, C. Zhang, T. Vegge, P. Johansson, and A. A. Franco, "Artificial intelligence applied to battery research: Hype or reality?" *Chemical Reviews*, vol. 122, no. 12, p. 10913, 2022. [Online]. Available: https://doi.org/10.1021/acs.chemrev.1c00108

[6] N. Noura, L. Boulon, and S. Jemeï, "A review of battery state of health estimation methods: Hybrid electric vehicle challenges," *World Electric Vehicle Journal*, vol. 11, no. 4, 2020. [Online]. Available: https://www.mdpi.com/2032-6653/11/4/66

[7] M. Zheng, D. Xu, L. Jiang, C. Gu, R. Tan, and P. Cheng, "Challenges of privacy-preserving machine learning in IoT," in *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, ser. SenSys '19. ACM, Nov. 2019, p. 1–7. [Online]. Available: http://dx.doi.org/10.1145/3363347.3363357

[8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," 2021. [Online]. Available: https://arxiv.org/abs/1912.04977

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023. [Online]. Available: https://arxiv.org/abs/1602.05629

[10] T. Kröger, A. Belnarsch, P. Bilfinger, W. Ratzke, and M. Lienkamp, "Collaborative training of deep neural networks for the lithium-ion battery aging prediction with federated learning," *eTransportation*, vol. 18, p. 100294, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590116823000693

[11] Y. Liu, M. Ahmed, J. Feng, Z. Mao, and Z. Chen, "A federated transfer learning framework for lithium-ion battery state of health estimation based on fast-charging segments," *IEEE Transactions on Transportation Electrification*, pp. 1–1, 2025.

[12] B. Chen, L. Fan, X. Zeng, M. Gu, and J. Zhou, "A contribution-aware federated framework for electric vehicle batteries health estimation," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 4605–4612, March 2025.

[13] W. Fang, J. Zhang, X. Lin *et al.*, "A study on data privacy protection in power battery soh prediction based on federated learning," *Ionics*, 2025. [Online]. Available: https://doi.org/10.1007/s11581-025-06606-5

[14] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[15] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[16] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2019. [Online]. Available: https://arxiv.org/abs/1803.08375

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] A. Martín *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[19] F. Chollet *et al.*, "Keras," https://keras.io, 2015.