XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

LSTM and State Space Modeling for Predicting Lithium-Ion Battery Health and Ensuring Safety in Real-World Applications

Thiago Schmidt[1], Fernanda Cristina Corrêa[2]

[1]*Dept. of Electrical Engineering, Federal University of Technology – Paraná*
R. Dr. Washington Subtil Chueire, 330, Ponta Grossa, 84217-220, Parana, Brazil
thiagos.1994@alunos.utfpr.edu.br
[2]*Dept. of Electrical Engineering, Federal University of Technology – Paraná*
R. Dr. Washington Subtil Chueire, 330, Ponta Grossa, 84217-220, Parana, Brazil
fernandacorrea@utfpr.edu.br

Abstract. Accurate estimation of battery state of charge (SOC), state of health (SOH), and remaining useful life (RUL) is essential for the safe and efficient operation of lithium-ion energy storage systems. Understanding and predicting the capacity and degradation behavior of individual battery cells and packs in real time is critical to reducing operational risk and extending service life. In real-world applications, such as electric vehicles, mobile electronics, and aerospace systems, undetected battery aging can cause performance loss, sudden shutdowns, or even catastrophic thermal events. The ability to anticipate failure modes through accurate SOH and RUL estimation enables manufacturers and users to implement timely maintenance, load balancing, and safe charging strategies. This not only increases energy efficiency and reliability but also plays a pivotal role in preventing accidents and ensuring user safety. This study proposes a hybrid evaluation framework using two advanced sequence modeling approaches, Long Short-Term Memory (LSTM) networks and the recently introduced MambaLithium framework, a selective state-space model, to predict key battery health indicators. The NASA's dataset was used during this research, consisting of the discharge cycles and getting the values of voltage, current, temperature and capacity. LSTM networks are applied for their ability to learn short- and mid-range dependencies using gated memory units, while the Mamba model integrates state-space representations to encode long-range sequence information effectively. Both models are trained and evaluated using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Pearson correlation (R^2) to assess predictive accuracy. This work underscores the importance of integrating advanced machine learning models into battery management systems and demonstrates how intelligent diagnostics can support safer, longer-lasting energy storage technologies.

Keywords: State of Health, State of Charge, Remaining Useful Life, State Space Model, Long Short-Term Memory.

1 Introduction

Technological advances and growing energy demands have positioned batteries as key energy storage systems due to their efficiency, energy density, and versatility. Modern battery technology began in 1836 with the Daniell cell and evolved slowly until the 1990s, when lithium-ion batteries (LIBs) emerged as a paradigm shift, becoming the standard for portable electronics. However, their widespread use also raised safety concerns, leading to several large-scale product recalls.

The rise of electric vehicles (EVs) has further increased reliance on LIB packs, intensifying concerns about their safety, reliability, and lifespan. Consequently, accurate monitoring of State of Charge (SOC), State of Health (SOH), and Remaining Useful Life (RUL) is essential to ensure safe and efficient operation in demanding automotive environments.

Artificial Neural Networks (ANNs) play a key role in improving predictive management of LIBs, helping to mitigate risks associated with accelerated degradation. ANNs consist of interconnected processing elements or artificial neurons that mimic biological neural networks [1, 2]. Input layers receive battery data, hidden layers extract and transform features through weighted combinations and nonlinear activations (e.g., ReLU, sigmoid), and the output layer generates predictions. Deeper networks enable more complex representations [3], and training uses gradient descent and backpropagation to minimize prediction errors [4].

Neural networks can be used to classify and safely repurposed used batteries, giving them a ”second life” instead of being discarded. This approach improves environmental sustainability by extending the batteries' life. Common second-life applications include energy storage for off-grid solar homes and powering low-power electric vehicles.

This study aims to evaluate and compare the performance of two distinct neural network architectures in predicting battery degradation parameters. Using SOC, SOH, RUL, and temperature and capacity as known inputs, we analyze the models' effectiveness, thereby improving the predictability of battery behavior and enabling proactive measures to be taken before any risk exposure to the user.

Section 2 presents the theoretical framework developed in this study, including detailed descriptions of the LSTM and MambaLithium models. Section 3 reports and discusses the results, focusing on the performance and behavior of the models. Finally, Section 4 summarizes the conclusions and outlines future research directions.

2 Development

The conceptual framework for neural networks was first established in the 1940s, as scientists began to model the human nervous system. However, significant advancements in the field were only reignited in the 2010s with the rise of machine learning and deep learning, which were accelerated by the availability of large-scale datasets for training novel algorithms.

To achieve the goal of this work, we selected two neural network models to compare in the context of battery health estimation, with a specific focus on predicting key indicators such as SOC, SOH and RUL. The first model is LSTM, a type of RNN designed to capture long-term dependencies in sequential data. LSTMs are particularly effective in time-series applications due to their ability to retain information over long input sequences, which makes them well-suited for modeling battery degradation patterns over charge-discharge cycles.

The second model is the MambaLithium architecture, a more recent development that leverages the Mamba framework, a state-space-based deep learning model known for its high efficiency and performance in long-sequence modeling. MambaLithium adapts this architecture specifically for the battery domain, aiming to capture complex non-linear dependencies and dynamic behavior inherent in lithium-ion cells through a more structured and scalable modeling approach.

This work aims to evaluate and compare the performance of both models using real-world datasets, to determine which architecture is more accurate and reliable in forecasting battery degradation metrics. The comparative analysis will offer insights into the advantages and limitations of traditional recurrent models such as LSTM versus newer, state-space-inspired frameworks like MambaLithium in the context of battery health management.

2.1 Dataset

Data for the model were obtained from the NASA battery repository. Battery discharge cycles B0005, B0005, and B0007 batteries (614 cycles each) were used, including cell temperature, voltage, current, and elapsed time. Battery B0018 was excluded due to fewer cycles. Capacity data were included to track degradation. All three batteries followed identical acquisition conditions, detailed in the repository's README. Data extraction and organization were performed with a Python script available on the author's GitHub github.

2.2 LSTM

LSTM architecture was introduced to overcome the limitations of traditional RNNs, particularly their inability to retain information over long sequences due to vanishing or exploding gradients. LSTM units are designed with specialized gates, namely, the input gate, forget gate, and output gate, that regulate the flow of information and maintain a memory cell capable of learning long-term dependencies. This makes LSTMs particularly effective for sequential tasks such as time series prediction, speech recognition, and battery health estimation [5].

The forget gate is responsible for determining which parts of the previous cell state should be discarded. It uses a sigmoid activation function to output values between 0 and 1, which are then multiplied element-wise with the previous state. The forget gate is defined as:

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

The input gate decides what new information should be added to the cell state. It consists of two parts: a

sigmoid layer that determines which values to update, and a tanh layer that creates a vector of candidate values to be added to the state and the equations are:

$$i_t = \sigma\big(W_i[h_{t-1}, x_t] + b_i\big), \tag{2}$$
$$c_t = \tanh\big(W_c[h_{t-1}, x_t] + b_c\big). \tag{3}$$

The cell state update is computed by combining the information retained from the previous state with the newly scaled candidate values. This allows the network to remember long-term dependencies [5]:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t. \tag{4}$$

The output gate determines which part of the cell state should be output as the new hidden state. It uses a sigmoid function to control the amount of information and applies a tanh activation to the updated cell state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \tag{5}$$
$$h_t = o_t \odot \tanh(c_t). \tag{6}$$

The sigmoid activation function, denoted as σ, is central to these gates because it produces outputs between 0 and 1. This range effectively serves as a filter or "valve," regulating the extent to which information passes through the gate. Mathematically, the sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{7}$$

where x is the input to the function, which typically consists of a weighted sum of the current input vector and the previous hidden state plus a bias term. Complementing the sigmoid, the hyperbolic tangent function is used as the activation function for generating candidate values and scaling the cell state before producing the hidden state. The tanh function outputs values between -1 and 1, allowing for both positive and negative activations, which is important for capturing nuanced patterns. It is defined as:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{8}$$

In the LSTM, tanh is used to create the candidate cell state ct and to scale the updated cell state ct before multiplying by the output gate activation to produce the new hidden state h_t, as presented in equations **??** and 6.

Together, the sigmoid function controls how much information passes through the gates (acting as selective filters), while the tanh function defines how the neuron activates, producing values that can express richer, signed information necessary for learning complex temporal patterns.

Each gate (forget, input, output, and candidate state) has its own set of weight matrices W and bias vectors b. These parameters are learned during training using backpropagation through time (BPTT), and they define how the input and previous hidden state are transformed into gate activations.

2.3 MambaLithium

Recent developments in sequential modeling, particularly the Structured State Space Sequence Model (S4) and its extension S6, have led to the emergence of the Mamba architecture. By combining selection and scanning mechanisms, Mamba effectively captures complex, non-linear dependencies in sequential data—challenges that conventional time-series models often struggle with. Given the long-term and non-linear degradation behavior of lithium-ion batteries (LIBs), Mamba represents a promising approach for accurate state prediction and improved battery management strategies.

MambaLithium employs the Mamba architecture to predict the health and charge states of lithium-ion batteries. Unlike Transformer models, which process sequences in parallel, Mamba operates sequentially, handling data step by step. This approach reduces computational overhead and enhances efficiency, making it particularly suitable for time-series applications.

At the core of Mamba lies a Selective State Space Model (SSM), where the internal state h serves as a dynamic memory representing the evolving condition of the battery. At each time step, the model receives new input data (e.g., voltage or temperature readings), updates its internal state, and generates a prediction based on

this updated representation. A key innovation in Mamba is its selective mechanism, which enables the model to dynamically identify and prioritize the most relevant portions of the input sequence for accurate prediction [6].

The key innovation of Mamba is making the model's parameters data-dependent. Instead of being fixed, the step size delta and the matrices B and C are dynamically generated from the current input xt. This allows the model to "select" what information to focus on at each step.

In the selection mechanism, $B \in \mathbb{R}^{\bar{B}\times L\times N}$, $C \in \mathbb{R}^{\bar{B}\times L\times N}$, and $\Delta \in \mathbb{R}^{\bar{B}\times L\times D}$ can be learned from $X \in \mathbb{R}^{\bar{B}\times L\times D}$ using fully connected layers.

By discretization, for each batch and each dimension, let x_t, h_t, and y_t denote the input, hidden state, and output at time $t = 1, 2, \ldots, L$, respectively. The S4 model can be written as follows [6]:

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x_t. \tag{9}$$

$$y_t = C_t h_t. \tag{10}$$

The scan module operates by applying a series of trainable operations to subsections, known as windows, of the input sequence. These operations, which are optimized during the training process, may consist of convolutional filters, recurrent mechanisms, or various other transformation techniques. As the window moves progressively across the entire sequence, the module can extract temporal features and identify relationships that extend across multiple time steps. This sliding-window approach allows the model to learn and represent patterns effectively within sequential data.

Applying the mamba lithium model, the features are normalized with the z-score method. The output of the model is reduced to a one-dimensional representation for regression prediction. The loss function selected to use in the model is L1-loss, which can measure the average difference between the predicted values and the measured values.

2.4 Work Proposal

This study evaluates the performance of two models—an LSTM network and the Mamba framework—for predicting battery degradation. Both models are optimized using **Optuna**, an automatic hyperparameter optimization framework that efficiently explores the search space using the Tree-structured Parzen Estimator (TPE) algorithm, allowing for faster convergence and improved model performance compared to manual tuning [7]. Hyperparameters such as learning rate, hidden size, and weight decay are tuned for each model. Discharge data from the NASA battery repository, including cell voltage, current, temperature, and capacity, are used, with batteries B0005 and B0006 for training and B0007 for testing. The LSTM captures temporal dependencies in sequential data, while Mamba combines physics-informed state-space modeling with data-driven learning. The goal is to compare their predictive accuracy in estimating battery state of charge, state of health and remaining useful life.

3 Results

To evaluate and compare the performance of the models, we adopted standard regression metrics, including Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). These metrics are widely recognized in the literature for assessing predictive accuracy.

3.1 SOC Estimation

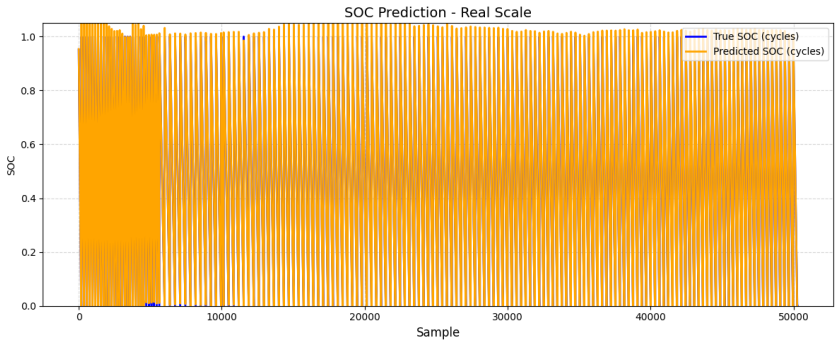
For SOC estimation, both models perform exceptionally well, as shown in Table 1. The comparison in Figure 1 highlights different behaviors: while both models closely follow the true values, the Mamba model produces a smoother response, indicating slightly more stable and consistent predictions compared to the LSTM model.

3.2 SOH Estimation

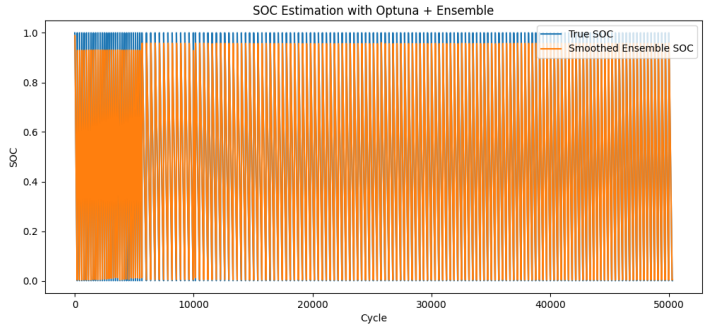
For SOH estimation, the LSTM model clearly outperforms Mamba, achieving very accurate predictions as shown in Table 2. Figure 2 illustrates that LSTM produces smooth predictions with low-errors, while Mamba

Table 1. SOC estimation performance comparison between Mamba and LSTM models.

Model	MSE	RMSE	MAE	R^2
Mamba	0.0002	0.0140	0.0105	0.9977
LSTM	0.0037	0.0608	0.0142	0.9592



(a) LSTM SOC.



(b) Mamba SOC.

Figure 1. Comparison between SOC estimation from LSTM and Mamba

struggles to capture the degradation pattern, resulting in higher errors and a lower R^2.

Table 2. SOH estimation performance comparison between Mamba and LSTM models.

Model	MSE	RMSE	MAE	R^2
Mamba	0.0009	0.0305	0.0274	0.8578
LSTM	0.0000	0.0000	0.0021	0.9989

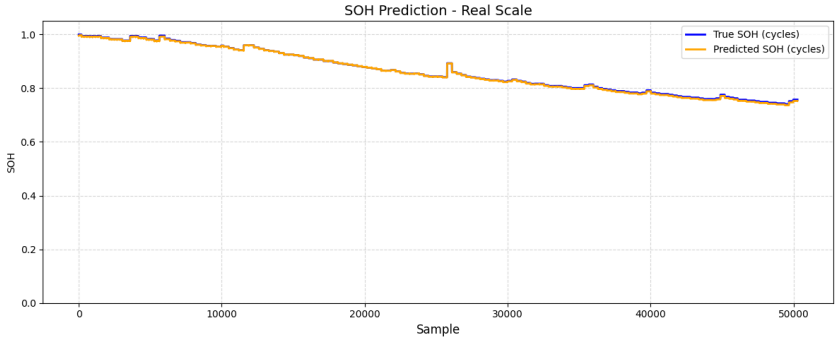
3.3 RUL Prediction

For RUL estimation, Mamba produced low absolute errors due to normalization but showed a lower R^2, Table 3, indicating that it captures the overall trend less accurately. In contrast, LSTM effectively models the temporal degradation patterns, providing highly accurate predictions. A similar behavior can be observed in the performance of the models for RUL prediction, as shown in the Figure 3

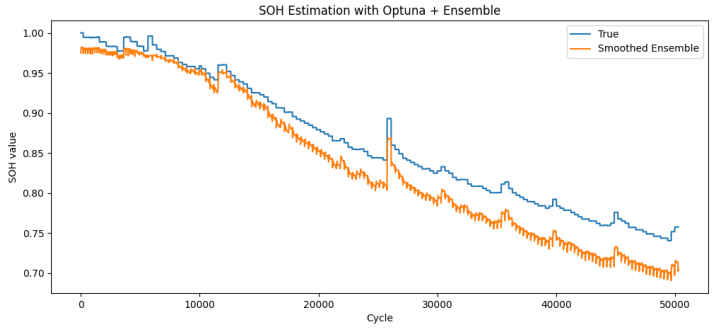
Table 3. RUL estimation performance comparison between Mamba and LSTM models.

Model	MSE	RMSE	MAE	R^2
Mamba	0.0000	0.0014	0.0011	0.8241
LSTM	21.7063	4.6591	3.6772	0.9992

Overall, the comparison highlights that LSTM excels in capturing sequential battery degradation for SOH and RUL, leading to higher R^2 values and lower errors. However, Mamba demonstrates strong SOC prediction

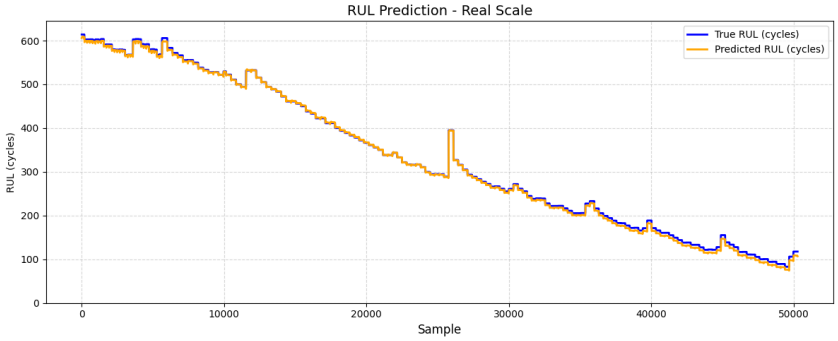


(a) LSTM SOH.

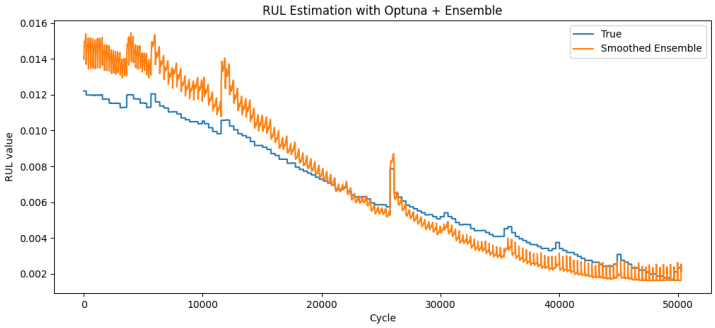


(b) Mamba SOH.

Figure 2. Comparison between SOH from LSTM and Mamba



(a) LSTM RUL.



(b) Mamba RUL.

Figure 3. Comparison between RUL from LSTM and Mamba

and offers advantages in parallelizable computation and real-time deployment, making it a valuable alternative for practical battery monitoring applications.

4 Conclusion

The comparative analysis of LSTM and Mamba models highlights complementary strengths. For SOC estimation, Mamba produces smoother and more stable predictions, with lower overall errors (MSE, RMSE, and MAE) compared to LSTM, which shows slightly higher deviations from the true values, as summarized in Table 1. In contrast, for SOH and RUL, LSTM demonstrates lower errors across all metrics, resulting in predictions that closely follow the true degradation patterns. Mamba exhibits higher errors, particularly in MAE and RMSE, which reflects its difficulty in capturing gradual temporal changes in battery health when the dataset is short, leading to less precise and slightly delayed responses, as shown in Table 2 and Figure 2. From a practical perspective, Mamba's parallelizable architecture, faster inference, and flexibility in handling diverse input features make it highly suitable for real-world applications, where efficient, real-time battery monitoring is essential. With further optimization, Mamba's performance could approach or exceed that of LSTM while maintaining these operational advantages.

Acknowledgements. The authors also acknowledge the Federal University of Technology – Paraná for the support provided for this research.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] K. Gurney. *Computer and Symbols versus Nets and Neurons*. Academic Press, 1999.
[2] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, Redwood City, CA, 1991.
[3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, vol. 323, n. 6088, pp. 533–536, 1986.
[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, n. 8, pp. 1735–1780, 1997.
[6] Z. Shi. Mambalithium: Selective state space model for remaining-useful-life, state-of-health, and state-of-charge estimation of lithium-ion batteries, 2024.
[7] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631. ACM, 2019.