Bi-LSTM based LIB RUL prediction

1st Haejun Kim
Dept. of Mechanical Engineering
Chungbuk National University
Cheongju, Korea
skh1111@chungbuk.ac.kr

2rd Kibum Cheon
Dept. of Mechanical Engineering
Chungbuk National University
Cheongju, Korea
cjsrlqja99@chungbuk.ac.kr

3nd Jongho Shin
Dept. of Mechanical Engineering
Chungbuk National University
Cheongju, Korea
jshin@chungbuk.ac.kr

***Abstract*—Accurate Remaining Useful Life (RUL) prediction for Lithium-Ion Battery (LIB) is essential for ensuring the reliability and safety of a Battery Management System (BMS). The non-linear degradation patterns, caused by diverse operating conditions and complex internal chemical reactions, are a key factor that makes accurate prediction difficult. In this study, we propose a prediction model based on a Recurrent Neural Network (RNN) to effectively learn the non-linear and complex degradation characteristics of these batteries. The proposed model features a dual-branch architecture that simultaneously utilizes both temporal patterns and statistical features from State of Health (SOH) sequence data. This architecture is designed to concurrently capture long-term degradation trends and local fluctuations. One branch learns temporal dependencies by directly taking the SOH sequence as input through a Long Short-Term Memory (LSTM) or Bidirectional Long Short-Term Memory (Bi-LSTM) network, while the other branch receives statistical features, such as the mean and standard deviation, extracted from that sequence as input. The feature vectors extracted from the two branches are combined and used for the final RUL prediction. To validate the model's performance and structural effectiveness, we compared and analyzed the prediction performance of LSTM and Bi-LSTM models applied under the same dataset and dual-branch architecture. The experimental results indicated that the Bi-LSTM model demonstrated superior prediction accuracy, achieving lower Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) compared to the LSTM model. This suggests that the bidirectional architecture, which utilizes both past and future contexts of the SOH time-series data, is more effective in learning battery degradation patterns with greater precision. The findings of this study are expected to contribute to the development of more reliable battery management strategies.**

***Index Terms*—Lithium-Ion Battery, Remaining Useful Life, State of Health, Deep Learning, Bi-LSTM**

I. Introduction

As energy-intensive industries such as electric vehicles, distributed energy systems, and mobile robotics rapidly expand, the demand for efficient energy sources is surging. In current commercial systems, the Lithium-Ion Battery (LIB) has been established as the standard energy storage device across various fields due to its high gravimetric and volumetric energy density, excellent power density, high efficiency, and long cycle life. [1]–[4]. However, the performance and lifespan of batteries are subject to significant uncertainty due to diverse operating conditions, manufacturing variations, and complex degradation mechanisms. [5]. Therefore, the ability to accurately predict the Remaining Useful Life (RUL) during the operational phase plays a crucial role in ensuring safety, optimizing maintenance and replacement schedules, and establishing strategies for post-use battery recycling. [6], [7].

Current studies on battery RUL prediction are broadly classified into physics-based, electrochemical model-driven and machine learning-driven approaches [8]. Initially, models based on electrochemical equations, such as the P2D model and the Newman model, are representative physics-based/electrochemical models [9]–[11]. These models can precisely analyze internal state variables—such as lithium diffusion within the electrodes, electrolyte concentration distribution, and interfacial reaction kinetics—offering high interpretability and generalizability of the battery's internal state. However, they are limited by the need to identify a large number of parameters and their high computational cost, making them difficult to implement for real-time, on-board applications [12], [13].

On the other hand, with advancements in computational performance and data acquisition technologies, RUL prediction using machine learning techniques has emerged as a promising alternative [14]. While traditional methods like Support Vector Machine and Gaussian Process Regression have been employed, recent research shows a strong trend towards machine learning techniques such as Recurrent Neural Network (RNN)-based models, capable of learning long-term dependencies, and Convolutional Neural Network (CNN)-based models, effective for processing local patterns [15]–[17].

Furthermore, advanced machine learning–based approaches for state diagnosis and prediction—such as Transformer-based models, CNN–BiLSTM–Attention hybrids, and convolution–recurrent combined models—have been reported in various fields [18]–[20]. These approaches automatically learn the correlations between input and output data, achieving high prediction accuracy without requiring physical models, and exhibit robustness against measurement noise. In this study, to apply such advanced machine learning techniques to RUL prediction, a deep learning model is proposed that uses only a

This research was supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF), Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, Republic of Korea 2020M3C1C1A01083162), and by the Regional Innovation System & Education(RISE) program through the (Chungbuk Regional Innovation System & Education Center), funded by the Ministry of Education(MOE) and the (Chungcheongbuk-do), Republic of Korea.(2025-RISE-11-014-03)

portion of the previous SOH sequence—rather than the entire historical sequence—based on the current prediction time. By focusing on an arbitrary number N of SOH sequences prior to the prediction point, the proposed model achieves high prediction accuracy while mitigating sample inefficiency, prediction latency, and storage costs, thereby improving its applicability to real systems. Additionally, to enhance prediction accuracy, statistical features of the SOH sequence are incorporated into the input data. These statistical features include the mean, standard deviation, skewness, kurtosis, maximum, minimum, and median values. This is implemented by adding an auxiliary neural network that learns the correlation between the statistical features of the SOH sequence and the RUL, thereby compensating for the limited predictive capability of RNN-based models that rely solely on time-series data.

In this study, Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM) architectures, both based on recurrent neural networks, are adopted to compare their performance under the same dataset and training conditions. LSTM processes input sequences in a unidirectional manner, predicting the current state based on past information, and is particularly effective at learning long-term dependencies in time-series data. In contrast, Bi-LSTM consists of two LSTMs that operate in both forward and backward directions, enabling the model to utilize information from both the past and the future. This bidirectional structure helps mitigate temporal uncertainty in RUL prediction by allowing the model to determine the relevance of specific data points using context from both directions. Therefore, this study quantitatively compares the real-time applicability of the unidirectional LSTM architecture and the temporal uncertainty mitigation capability of the bidirectional Bi-LSTM architecture.

The remainder of this paper is organized as follows. Section 2 describes the architectures of LSTM and Bi-LSTM. Section 3 introduces the dataset used for training and details the specific structure of the RUL prediction model. Section 4 presents the prediction results of the proposed models and analyzes their performance through a comparative study. Finally, Section 5 summarizes the conclusions of this study and discusses future work and its expected impact.

II. LSTM AND BI-LSTM

RUL prediction for LIB is performed based on State of Health (SOH) sequence data. Recurrent Neural Network (RNN) are widely used to effectively learn the features of such sequential data. However, basic RNN have a limitation known as the long-term dependency problem, where initial information fails to propagate through later time steps as the sequence length increases. This section describes the architectures and features of LSTM and Bi-LSTM, which are representative recurrent neural network models proposed to solve this problem.

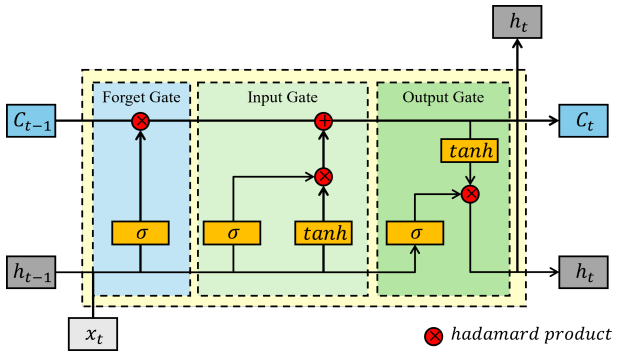

Fig. 1. LSTM cell structure

A. Long Short-Term Memory

LSTM, proposed by Hochreiter and Schmidhuber [21], is a model designed to solve the long-term dependency problem found in traditional RNN [21]. The core idea of LSTM is to effectively control the flow of information by introducing a cell state and several gates. The cell state acts like a conveyor belt, selectively carrying information throughout the entire sequence. This architecture efficiently solves the long-term dependency problem by allowing important information to be retained over long sequences while discarding irrelevant information. In the following subsections, we will describe the three types of gates that constitute the LSTM architecture and their roles, and analyze the advantages and disadvantages of this structure.

1) Forget Gate: The Forget Gate decides what information to discard from the previous cell state, $\boldsymbol{C}_{t-1}$. It passes information from the hidden state through a sigmoid layer to produce gate values between 0 and 1, which are then applied to the cell state via an element-wise product. This process determines which information is to be retained. The corresponding equation is as follows:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f\,[\boldsymbol{h}_{t-1},\,\boldsymbol{x}_t] + \boldsymbol{b}_f) \tag{1}$$

2) Input Gate: The Input Gate determines how much of the new information, based on the current input $\boldsymbol{x}_t$ and the previous hidden state $\boldsymbol{h}_{t-1}$, should be stored in the cell state, $\tilde{C}_t$.

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i\,[\boldsymbol{h}_{t-1},\,\boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{2}$$

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_c\,[\boldsymbol{h}_{t-1},\,\boldsymbol{x}_t] + \boldsymbol{b}_c) \tag{3}$$

The cell state for the current time step is updated using the outputs from the Forget and Input Gates. In this process, the Forget Gate regulates the proportion of information to retain from the previous cell state, while the Input Gate regulates the proportion of the new candidate cell state to incorporate. The equation is as follows:

$$\boldsymbol{C}_t = \boldsymbol{f}_t \cdot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \cdot \tilde{\boldsymbol{C}}_t \tag{4}$$

3) Output Gate: The Output Gate generates the hidden state for the current time step based on the updated cell state, $\boldsymbol{C}_t$. The output is calculated by taking the element-wise product of two terms: the output of a sigmoid layer that processes the previous hidden state, and the updated cell state after being passed through a hyperbolic tangent layer. This result is the hidden state for the current time step,$\boldsymbol{h}_t$.

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o\,[\boldsymbol{h}_{t-1},\,\boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \cdot \tanh(\boldsymbol{C}_t) \tag{6}$$

Fig. 1 illustrates the structure of an LSTM memory cell. Through this gated architecture, LSTM compensates for the structural limitations of conventional RNN and mitigates the long-term dependency problem. LSTM excels at learning long-term trends from time-series data while filtering out irrelevant information, making it well-suited for capturing the gradual degradation patterns present in SOH sequences. However, due to its unidirectional structure, LSTM is constrained to using only information from past time steps to predict the output at the current time step. This causal structure introduces temporal uncertainty, as it cannot leverage the context from the entire sequence.

B. Bidirectional-LSTM

Bidirectional LSTM (Bi-LSTM) is a model proposed to overcome the unidirectional structural limitations of LSTM. As its name suggests, it aims to utilize contextual information more broadly by processing the input sequence in two directions [22], [23]. Fig. 2 illustrates the specific architecture of Bi-LSTM. Bi-LSTM is designed to simultaneously incorporate bidirectional information from the sequence by arranging identical LSTM cells in both forward and backward passes. The output hidden state at time step t is defined as follows:

$$\boldsymbol{h}_t \;=\; \left[\overrightarrow{\boldsymbol{h}}_t\,;\,\overleftarrow{\boldsymbol{h}}_t\right] \in \mathbb{R}^{2H} \tag{7}$$

Through this bidirectional architecture, Bi-LSTM processes the same sequence in both forward and backward directions, resulting in a representation for each time step that is an aggregate of both past and future contexts. In time-series data, it is difficult for a unidirectional LSTM structure to discern whether a change at a specific point is short-term noise or the initial sign of a cumulative pattern shift. Bi-LSTM addresses this problem of temporal uncertainty by adopting a bidirectional structure. Furthermore, for the same sequence length, it can improve prediction accuracy by leveraging a richer set of hidden layer computations compared to a standard LSTM. On the other hand, the increased number of parameters and computations leads to longer training times and can heighten the risk of overfitting, particularly when the dataset size is limited. In this study, we apply Dropout and Layer Normalization to reduce this risk and improve the model's generalization performance [24].

This study evaluates the structural characteristics and practical applicability of LSTM and Bi-LSTM by comparing them under identical input data and training environments. While

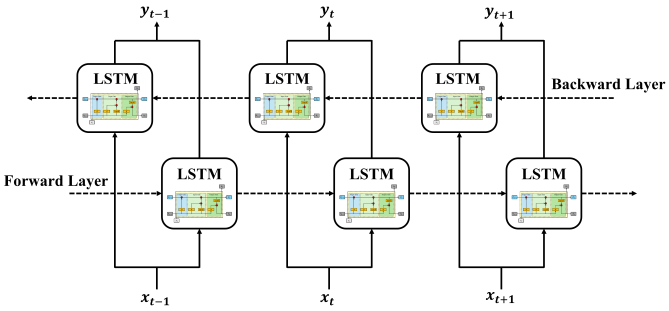

Fig. 2. Bi-LSTM structure

LSTM offers the advantage of shorter training times due to lower computational costs, Bi-LSTM can leverage both past and future contexts to reduce temporal uncertainty and achieve more stable prediction performance. Notably, as this study uses only the 100 most recent sequences leading up to the prediction point, the comparison highlights LSTM's capability for unidirectional learning of recent trends against Bi-LSTM's interpretive strength in considering the full bidirectional context. Through this experimental comparison, we analyze the results within a practical battery RUL estimation scenario and aim to propose a model architecture that is well-suited for operational deployment.

III. RUL Prediction Model Design

This chapter describes the detailed architecture of the model designed for the RUL prediction of lithium-ion batteries. First, we describe the composition and preprocessing of the dataset used for training, and then we introduce in detail the architecture of the proposed model, which utilizes both SOH sequence data and statistical features.

A. LIB Dataset

In this study, a charge-discharge test is conducted on eight SAMSUNG INR 18650-25R lithium-ion cells in a thermo-hygrostatic chamber maintained at 25 °C and 60% relative humidity. The cells are charged using a 0.5C-rate Constant Current–Constant Voltage (CC–CV) method and discharged with random currents. Over 950 cycles are repeated, during which the voltage and current time-series data for each cycle are collected. Prior to model training, the collected data undergoes preprocessing to extract input features effective for Remaining Useful Life (RUL) prediction. First, after extracting the Cycle-SOH curve for each cell, a sliding window technique is applied to construct the dataset for model training. By setting the window size to N, sequences composed of N consecutive SOH values are sequentially extracted from the entire SOH dataset. Each generated sequence is used as the first model input (the sequence input). Simultaneously, seven statistical features—mean, standard deviation, skewness, kurtosis, maximum, minimum, and median—are calculated from the sequence to form the second model input (the feature input). Finally, the ground truth RUL for each sequence is

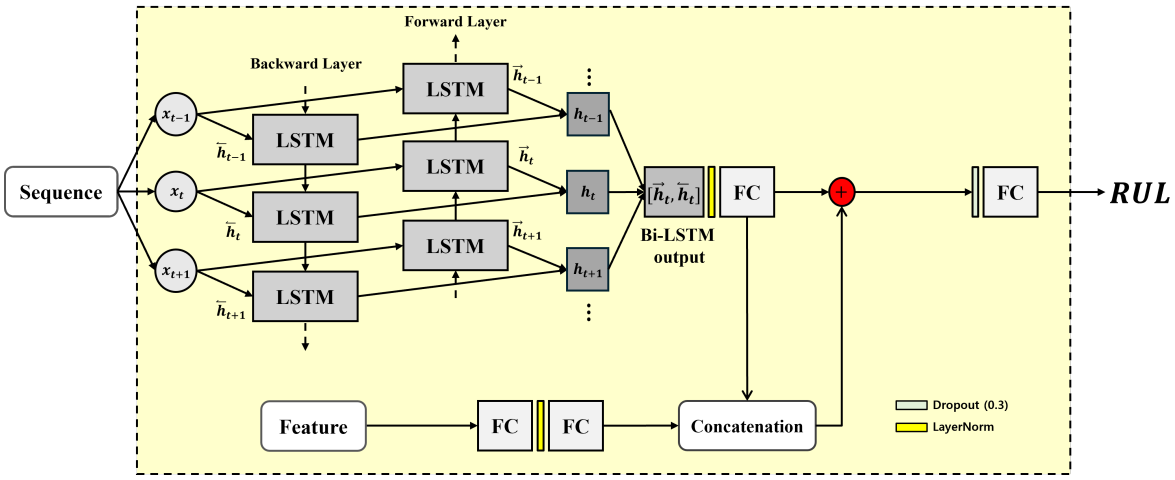

Fig. 3. Proposed RUL prediction model

calculated as the difference between the battery's end-of-life (EOL) cycle number and the cycle number of the last data point in that sequence.

Here, the State of Health (SOH), a key indicator representing the degree of performance degradation in lithium-ion batteries, is defined in this study as a capacity-based metric. The available capacity, Q_k, at the k-th cycle is calculated by integrating the current. The equation is as follows:

$$Q_k = \int_{t_{\text{start}}}^{t_{\text{end}}} I(t)\, dt, \qquad Q_k \approx \sum_i \frac{I_i + I_{i+1}}{2}\, \Delta t_i. \tag{8}$$

The reference capacity, Q_{ref}, is defined as the nominal capacity of the cell. Consequently, the capacity-based SOH is calculated as follows:

$$\text{SOH}_k = \frac{Q_k}{Q_{\text{ref}}} \times 100\%. \tag{9}$$

Additionally, the battery's end-of-life (EOL) is defined as the point when the SOH reaches 80%, and for each cell, this cycle index is recorded as k_{EOL}. Accordingly, the Remaining Useful Life (RUL) at cycle k is calculated as follows and used as the target variable for training the LSTM and Bi-LSTM models.

$$\text{RUL}(k) = k_{\text{EOL}} - k \tag{10}$$

B. RUL Prediction Model

The RUL prediction model proposed in this study features a dual-branch architecture designed to simultaneously learn both the dynamic patterns and static features of the SOH time series. The overall model architecture is illustrated in Fig. 3.

The first path is the *sequence branch*, which takes the SOH sequence data as input and uses the most recent N SOH samples prior to the prediction point. In this study, the proposed model is validated with $N = 100$. In Fig. 3, x_t denotes the SOH value at the time step t within the SOH sequence data, and h_t represents the hidden state learned by the Bi-LSTM model from the SOH sequence that includes x_t, which serves as a high-dimensional feature vector that captures temporal dependencies around the time step t. The input sequence is transformed into hidden states through the LSTM and Bi-LSTM layers and then compressed into a final feature vector via the fully connected (FC) layer.

The second path is the *feature branch*, which uses a statistics-based feature vector extracted from the same SOH sequence segment as the sequence branch. Specifically, from each sequence, statistical properties such as mean, standard deviation, skewness, and kurtosis are calculated, along with positional statistics including the minimum, maximum, and median values. This feature vector is processed through two FC layers and a normalization step before being combined with the vector obtained from the sequence branch.

The fusion of these branches is achieved through both concatenation and addition operations, a design that ensures both the time-series contextual information and the statistical features are jointly reflected. Specifically, the feature vector from the sequence branch is utilized in both the concatenation and addition layers, a design choice that allows the features from the sequence data to carry more weight in the final representation. After these fusion steps, the resulting vector is passed through a final FC layer and transformed into a single scalar value, which represents the predicted RUL.

IV. Model Validation

This chapter validates and analyzes the performance of the RUL prediction model designed in Chapter 3. First, we define the dataset partitioning method, the training environment, and the performance evaluation metrics used in the experiments. Subsequently, under the same dual-branch architecture, we compare the prediction results of the models applying LSTM and Bi-LSTM to the sequence branch, respectively, and quantitatively evaluate the accuracy of each model using error metrics.

A. Experimental Environment and Evaluation Metrics

The performance validation of the proposed model was conducted using the MATLAB Deep Learning Toolbox [25]. Out of the total eight battery cell datasets, six (batteries 1-6) were used for model training and validation, while the remaining two (batteries 7-8) were set aside as a completely unseen test set to evaluate the model's generalization performance. The Adam optimizer [26] was used for model training, with the following key hyperparameters: a maximum of 600 epochs, a mini-batch size of 128, an initial learning rate of 3e-4, and L2 regularization of 1e-4. Additionally, an early stopping strategy was employed, halting the training if the validation performance did not improve for 100 consecutive epochs.

To quantitatively evaluate the prediction accuracy of the models, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were used as performance metrics. The equation for each metric is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}i|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where, n is the total number of test data points, y_i is the actual RUL value, and $\hat{y}_i$ is the RUL value predicted by the model.

B. Experimental Results and Analysis

Table I presents the quantitative comparison of prediction errors for each model across different batteries. The Bi-LSTM-based model outperformed the LSTM model in all evaluation metrics for both test batteries. Specifically, for Battery 7, which was not used during training, the MAE decreased from 8.62 to 6.53, and the RMSE decreased from 10.58 to 8.19, demonstrating a significant improvement in performance. Similarly, for Battery 8, the Bi-LSTM model achieved lower errors, confirming its stable prediction capability. In particular, the noticeable reduction in RMSE is noteworthy. While the MAE represents the average magnitude of prediction errors, the RMSE assigns greater weight to larger errors. Therefore, the lower RMSE of the Bi-LSTM model indicates a reduced likelihood of severe prediction failures, implying that Bi-LSTM provides higher reliability for practical applications such as Battery Management Systems (BMS), where preventing system malfunctions or safety issues is critical.

TABLE I
PERFORMANCE COMPARISON OF RUL PREDICTION MODELS ACROSS ALL BATTERIES

Battery	Model	MAE	RMSE
Training Batteries (1–6)			
Battery 1	LSTM-based Model	4.99	6.67
	Bi-LSTM-based Model	**4.74**	**6.16**
Battery 2	LSTM-based Model	9.03	13.19
	Bi-LSTM-based Model	**7.57**	**10.38**
Battery 3	LSTM-based Model	7.39	11.46
	Bi-LSTM-based Model	**5.10**	**7.19**
Battery 4	LSTM-based Model	5.87	8.53
	Bi-LSTM-based Model	**5.74**	**7.98**
Battery 5	LSTM-based Model	10.32	13.64
	Bi-LSTM-based Model	**8.38**	**10.44**
Battery 6	LSTM-based Model	10.91	12.84
	Bi-LSTM-based Model	**9.76**	**11.64**
Validation Batteries (7–8)			
Battery 7	LSTM-based Model	8.62	10.58
	Bi-LSTM-based Model	**6.53**	**8.19**
Battery 8	LSTM-based Model	6.67	9.02
	Bi-LSTM-based Model	**6.23**	**8.01**

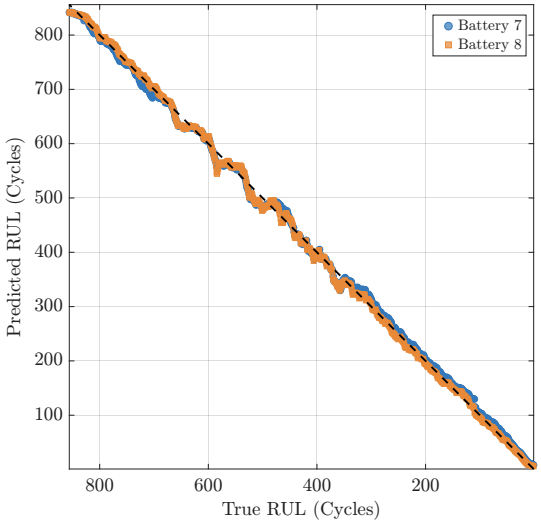

Fig. 4. Prediction results of the LSTM-based model

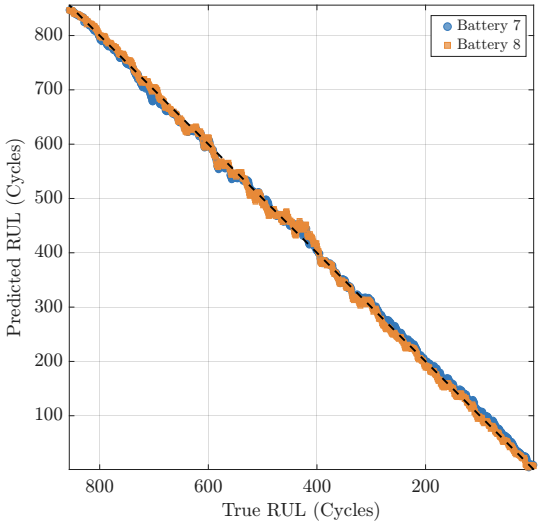

Fig. 5. Prediction results of the Bi-LSTM-based model

Furthermore, the superior performance of the Bi-LSTM model is also evident in Figs. 4-5. These figures show scatter plots comparing the actual versus predicted RUL for the LSTM and Bi-LSTM models, respectively. While both models predict the overall degradation trend effectively, it is clear that the predictions from the Bi-LSTM model are more densely

clustered around the ideal 1:1 line (dotted line).

Meanwhile, an analysis of performance across different RUL ranges reveals that both models tend to show a slight increase in prediction error as the battery approaches its end-of-life (i.e., at lower RUL values). This suggests that the models have some difficulty in perfectly capturing the accelerated degradation phenomena that often occur late in a battery's life. Nevertheless, the Bi-LSTM model's predictions remained more tightly clustered around the ideal 1:1 line, demonstrating its robustness by maintaining relatively stable predictions compared to the LSTM, especially in the end-of-life region.

In conclusion, these results confirm that by learning from both forward and backward information in the SOH time series, the Bi-LSTM can more robustly handle non-linear patterns, such as local capacity regeneration and changes in the degradation rate. This leads to the conclusion that the Bi-LSTM, which comprehensively understands bidirectional context, performs more stable and reliable predictions compared to the LSTM, which processes information only unidirectionally.

V. Conclusion

In this study, we proposed a dual-branch deep learning model for the accurate RUL prediction of lithium-ion batteries, which simultaneously utilizes both the dynamic patterns and static features of SOH sequences. Furthermore, within this proposed architecture, we compared and analyzed the performance of RUL prediction models based on LSTM and Bi-LSTM. The experimental results confirmed that the Bi-LSTM-based model, which learns from the bidirectional context of time-series data, achieved lower errors and demonstrated superior prediction accuracy compared to the LSTM-based model. In particular, the Bi-LSTM model showed a more significant improvement in the RMSE metric, which is sensitive to large errors, thus proving its robustness against outliers like major prediction failures. This implies that it can ensure higher stability and reliability in real-world system applications.

The model architecture and the applicability of Bi-LSTM proposed in this study are expected to enhance the reliability of battery management systems and contribute to the efficient and safe operation of batteries in various industrial fields.

References

[1] J.-M. Tarascon and M. Armand, "Issues and challenges facing rechargeable lithium batteries," *nature*, vol. 414, no. 6861, pp. 359–367, 2001.

[2] B. Dunn, H. Kamath, and J.-M. Tarascon, "Electrical energy storage for the grid: a battery of choices," *Science*, vol. 334, no. 6058, pp. 928–935, 2011.

[3] F. Degen, M. Winter, D. Bendig, and J. Tübke, "Energy consumption of current and future production of lithium-ion and post lithium-ion battery cells," *Nature energy*, vol. 8, no. 11, pp. 1284–1295, 2023.

[4] N. Nasajpour-Esfahani, H. Garmestani, M. Bagheritabar, D. J. Jasim, D. Toghraie, S. Dadkhah, and H. Firoozeh, "Comprehensive review of lithium-ion battery materials and development challenges," *Renewable and Sustainable Energy Reviews*, vol. 203, p. 114783, 2024.

[5] C. R. Birkl, M. R. Roberts, E. McTurk, P. G. Bruce, and D. A. Howey, "Degradation diagnostics for lithium ion cells," *Journal of Power Sources*, vol. 341, pp. 373–386, 2017.

[6] L. Yao, S. Xu, A. Tang, F. Zhou, J. Hou, Y. Xiao, and Z. Fu, "A review of lithium-ion battery state of health estimation and prediction methods," *World Electric Vehicle Journal*, vol. 12, no. 3, p. 113, 2021.

[7] M. Berecibar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, and P. Van den Bossche, "Critical review of state of health estimation methods of li-ion batteries for real applications," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 572–587, 2016.

[8] M. S. H. Lipu, M. A. Hannan, T. F. Karim, A. Hussain, M. H. M. Saad, A. Ayob, and M. S. Miah, "A review of data-driven, physics-based, and hybrid-based prognostic methods for remaining useful life estimation of lithium-ion batteries," *Electronics*, vol. 11, no. 9, p. 1335, 2022.

[9] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of the Electrochemical society*, vol. 140, no. 6, p. 1526, 1993.

[10] T. F. Fuller, M. Doyle, and J. Newman, "Simulation and optimization of the dual lithium ion insertion cell," *Journal of the electrochemical society*, vol. 141, no. 1, p. 1, 1994.

[11] F. Brosa Planella, W. Ai, A. M. Boyce, A. Ghosh, I. Korotkin, S. Sahu, V. Sulzer, R. Timms, T. G. Tranter, M. Zyskin *et al.*, "A continuum of physics-based lithium-ion battery models reviewed," *Progress in Energy*, vol. 4, no. 4, p. 042003, 2022.

[12] Q. Xu, M. Wu, E. Khoo, Z. Chen, and X. Li, "A hybrid ensemble deep learning approach for early prediction of battery remaining useful life," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 177–187, 2023.

[13] Q. Wang, Z. Wang, P. Liu, L. Zhang, D. U. Sauer, and W. Li, "Large-scale field data-based battery aging prediction driven by statistical features and machine learning," *Cell Reports Physical Science*, vol. 4, no. 12, 2023.

[14] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.

[15] M. A. Patil, P. Tagade, K. S. Hariharan, S. M. Kolake, T. Song, T. Yeo, and S. Doo, "A novel multistage support vector machine based approach for li ion battery remaining useful life estimation," *Applied energy*, vol. 159, pp. 285–297, 2015.

[16] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Gaussian process regression for forecasting battery state of health," *Journal of Power Sources*, vol. 357, pp. 209–219, 2017.

[17] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.

[18] Z. Zhang, W. Zhang, K. Yang, and S. Zhang, "Remaining useful life prediction of lithium-ion batteries based on attention mechanism and bidirectional long short-term memory network," *Measurement*, vol. 204, p. 112093, 2022.

[19] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, and Y. Yuan, "Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network," *Applied Energy*, vol. 253, p. 113626, 2019.

[20] Y. Li, L. Li, R. Mao, Y. Zhang, S. Xu, and J. Zhang, "Hybrid data-driven approach for predicting the remaining useful life of lithium-ion batteries," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 2, pp. 2789–2805, 2023.

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

[22] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[23] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] The MathWorks Inc., "Deep learning toolbox version: R2025b," Natick, Massachusetts, United States, 2025. [Online]. Available: https://mathworks.com/help/deeplearning/index.html

[26] K. D. B. J. Adam *et al.*, "A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, vol. 1412, no. 6, 2014.