Proceedings of CECNet 2021
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210385

Sequence-in-Sequence Learning for SOH Estimation of Lithium-Ion Battery

Thien PHAM [a], Loi TRUONG [a], Mao NGUYEN [b], Akhil GARG [c], Liang GAO [c], and Tho QUAN [a1]

[a] *Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam National University Ho Chi Minh City, Vietnam*
[b] *Faculty of Computer Science, Ho Chi Minh City University of Computer Science, Vietnam National University Ho Chi Minh City, Vietnam*
[c] *School of Mechanical Science and Engineering, Huazhong University of Science and Technology Wuhan, China*

Abstract. State-of-Health (SOH) prediction of a Lithium-ion battery is essential for preventing malfunction and maintaining efficient working behaviors for the battery. In practice, this task is difficult due to the high level of noise and complexity. There are many machine learning methods, especially deep learning approaches, that have been proposed to address this problem recently. However, there is much room for improvement because the nature of the battery data is highly non-linear and exhibits higher dependence on multidisciplinary parameters such as resistance, voltage and external conditions the battery is subjected to. In this paper, we propose an approach known as bidirectional sequence-in-sequence, which exploits the dependency of nested cycle-wise and channel-wise battery data. Experimented with real dataset acquired from NASA, our method results in significant reduction of error of approximately up to 32.5%.

Keywords. Lithium-ion Batteries, BiLSTM, Auto Regression.

1. Introduction

Nowadays, *Lithium-ion Batteries* (LIB) have been used in many electronic devices as the main power supply to avoid power interruption. Especially, the recent growing quantity of moving objects explodes the demand for LIB. In particular, LIB has been used enormously due to its advantages of durability, stability, high-capacity, low-cost, light-weight and small-scale. In another approach, users need to have recent batteries monitoring data during using them. This demand comes from the need to lookahead malfunction, foresee the risks, avoid explosion or optimize the battery life. Therefore, it is essential to predict LIB capacity accurately and promptly in order to provide the exact input for above demands, making the issue of *State-of-Health* (SOH) estimation for LIB highly essential to maintain proper functioning [1].

[1]Corresponding Author: Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam National University Ho Chi Minh City, Vietnam; Email: qttho@hcmut.edu.vn.

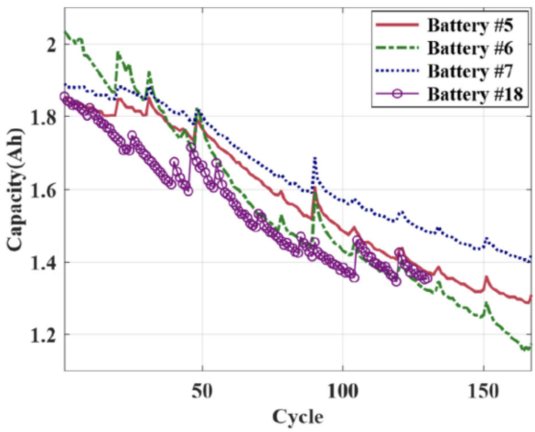

Figure 1. Capacities of real LIBs, in Ampere Hour (Ah), recorded during charge cycles increases [3].

Basically, SOH is a metric to reflect the condition of a battery, compared to its ideal conditions. In the actual lifetime, the capacity of a battery is normally reduced during the discharge cycle and needs a charge cycle afterward. These charge and discharge cycles, empirically, influence the battery SOH over a long time. There are a number of metrics used for SOH estimation of a battery. Perhaps the most popular one is introduced in [2], where SOH is calculated as $SOH = \frac{C_{full}}{C_{nom}} \times 100\%$, where C_{nom} (*Nominal Capacity*) is the original capacity given when the battery was manufactured and C_{full} (*Full Capacity*) is the full capacity practically charged at the current cycle. Theoretically, C_{full} is gradually degraded after every charge and discharge cycles and the SOH estimation task is thus equivalent to that of C_{full} prediction, given the current cycle of the battery. However, this estimation is a non-trivial exercise due to the nonlinear nature of battery capacity change one practically used. Moreover, different batteries introduce different rates of SOH decay. Figure 1 shows an example of SOHs measured from four batteries [3]. It can be observed that the SOHs of the cycles create a time-series trend and there are clear degradations with various rates over around 150 cycles of charge and discharge of all the four modules. Moreover, the batteries C_{full} have not always strictly decreased but occasionally increased in a periodical manner, making *spikes* over the trends.

Research in this field shows that even though each battery possesses a different decay rate, historical data of the energy status of the battery, can be used to predict the energy status in the next few cycles [4]. With the recent advancement of AI methods, much work considers leveraging sequence-based *deep learning* models to handle this issue. In [5], a model using importance sampling and neural networks is introduced. Perhaps the most widely used sequence-based deep learning model for this problem is based on the well-known deep learning architecture of *Long Short-term Memory* (LSTM) [6], [7], [2], which was applied in various works with remarkable results [5], [6].

Traditionally, in LSTM-based approaches, the previous information of the battery capacity in some last cycles will be used to predict the corresponding capacity statuses in the next cycles. However, some additional information from other *channels*, such as voltage, current and temperature, can be complementarily used for insightful analysis of the battery status. Thus, the collected channel-wise data of a battery become those of *multivariate time series* [2]. In [8], a multi-channel charging model is proposed for the prediction task. Other partial data also used in [9] with similar purpose.

Notably, perhaps the most remarkable achievement so far is reported in the work of [2]. In this work, the authors proposed an importance sampling strategy of 11 points in a charge cycle. To construct a predictive model, other channel-wise information, as above discussed, has been incorporated to introduce a multivariate time series data model, which is then eventually handled by an LSTM architecture. This work is considered state-of-the-art now in this area.

Thus, in this study, we exploit an observation that not only the cycle-wise relationship during a lifetime of a battery, but also its internal channel-wise information. To realize this idea, we suggest a novel architecture, known as *sequence-in-sequence* (seq-in-seq) model, where the battery energy is analyzed during two nested sequences, including local sequence and global sequence, which reflect the channel-wise and cycle-wise information respectively. Moreover, we also present a combined architecture of Bi-LSTM [10] and *Autoregression* [11], applied in each local and global sequences, to better represent their sequential relationships in different views, resulting in our ultimate model, socalled *BiAR-SeqInSeq*.

In the rest of paper, we summarize preliminaries in Section 2. The overall architecture and details of BiAR-SeqInSeq model are given in Section 3. Section 4 discusses our experimental results when verifying our model with real NASA battery dataset. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. Charge Cycle of a Lithium-ion Battery

In common working conditions, a LIB lifetime spans up to 2-3 years or equivalently, 300-500 charge and discharge cycles. In the prior cycle, LIB internal channels (current, voltage, temperature) vary and LIB capacity increases gradually to reach a fully-charged state. This charge cycle can be roughly divided into 2 separate periods: *Constant Current* (CC) one with gradually increased voltage and *Constant Voltage* (CV) one with gradually decreased current, as depicted in Figure 2. A charge cycle starting with CC period as the current is kept unchanged while voltage is gradually increased. By the end of CC period, the charge cycle is continuously controlled under CV period as the current is gradually decreased and the voltage remains stable. This CC-CV process is completed when the current reaches a threshold constant value [12]. The battery then goes for a later discharge phase to accumulate SOH.

In [4], there are 5 crucial features that can be extracted during this CC-CV process to infer battery lifetime as follows (Figure 2).

- x_1: Initial charge voltage
- x_2: CC charge capacity, calculated by accumulation the rectangle ICCt, where Icc is the constant current in the charge, and t the duration of CC period
- x_3: CV charge capacity, accumulated by the integral of the area of decreasing current during CV period
- x_4: final charge voltage (voltage measured at battery terminal)
- x_5: final charge current (current measured at battery terminal)

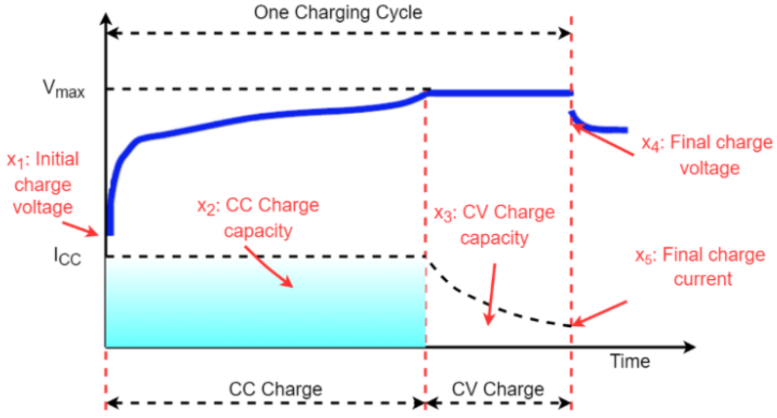

Figure 2. Charge cycle and handcrafted features in that cycle [4].

Obviously, those x_1 values are usually unstable considering many different cycles. Moreover, x_1 is taken in the non-sampling area [5] and its value practically plays a very small impact on the output. Therefore, in this study, we will ignore x_1 contribution in the calculation. The other 4 values x_2, x_3, x_4, x_5 are encoded as *cycle-wise* data to predict the battery SOH, as subsequently discussed.

Additionally, we also adopt the information of *peak* status, which represents that the SOH value of the current cycle is staying at the peak comparing to its neighbour using z-scores or not [13] as a cycle-wise information, alongside with the C_{full} information of the considered cycle.

2.2. A Baseline Sampling-based Model

Obviously, current and voltage are essential in the learning model to predict battery lifetime or SOH. However, these values are continuous and one would naturally consider sampling them. In [2], temperature values are additionally sampled at these timepoints to create multi-channel input for their prediction. Those attributes of current, voltage and temperature introduce channel-wise information of a LIB.

Figure 3 depicts the channel-wise raw values of a battery. Moreover, in [5], the authors proposed 11 important timepoints in a charge cycle to extract the sampling current, voltage and temperature as depicted. The authors explained the need to maintain important data points which carry the shape of the derivative ΔVoltage (V) at 11 positions in the charge cycle. Those 11 points of corresponding to the $35^{th}, 52^{th}, 70^{th}, 88^{th}, 90^{th}, 92^{th}, 94^{th}, 96^{th}, 98^{th}, 99^{th}, 100^{th}$ produce the corresponding sampled channel-wise data.

Figure 4 illustrates how channel-wise data are encoded in [2]. The sampled values of voltage V_j, current I_j and temperature T_j at the above mentioned 11 data points are formed together as 11×3 matrix $\xi_n = [[V_1, I_1, T_1][V_2, I_2, T_2], ..., [V_S, I_S, T_S]], S = 11$. The authors also use the C_j, which is the SOH value at the j^{th} cycle, as other additional information in their final encoded feature vectors of a cycle. The features vectors of the past cycles are then fed into a LSTM model to make the prediction for the next cycle, as subsequently discussed. In our study, we consider this model as the baseline model in our subsequent experiments.

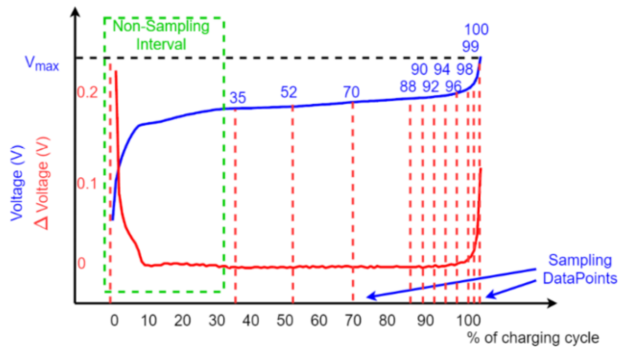

Figure 3. Sampling important values of current and voltage during a charge cycle [5].

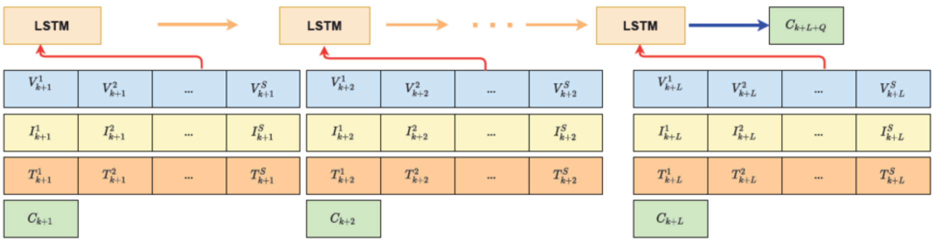

Figure 4. Sampling-based features for SOH estimation [2].

2.3. *Long Short-Term Memory*

As discussed, these sequences of feature vectors over cycles will be used to train LSTM [14], which is virtually a sequence of multiple logical cells, unfolded from a physical cell. Figure 5 presents an architecture of a basic LSTM cell, which was introduced in [14]. LSTM is in fact an upgraded version of Recurrent Neural Network (RNN) [15]. At the time t, this network will produce output $y(t)$ from the input $x(t)$. However, the output of this network at the previous iteration will also be used as part of the input of the next step, or recurrent input, together with new actual input. In the standard LSTM model, processing information is more complicated when modules containing computational blocks are repeated over many timesteps to selectively interact with each other in order to determine what information will be added or removed. This process is controlled by three gates namely *input gate*, *output gate*, and *forget gate*. Controlling the flow of information inside an LSTM model can be described as the following mathematical equations:

$$i = \partial(W_i x_t + W_{hi} h_{t-1} + b_i) \tag{1}$$

$$f = \partial(W_f x_t + W_{hf} h_{t-1} + b_f) \tag{2}$$

$$o = \partial(W_o x_t + W_{ho} h_{t-1} + b_o) \tag{3}$$

$$(C)_t = (C)_{t-1} \otimes f + i \otimes tanh(W_C x_t + W_{hC} h_{t-1} + b_C) \tag{4}$$

$$h_t = o \otimes tanh((C)_{t-1}) \tag{5}$$

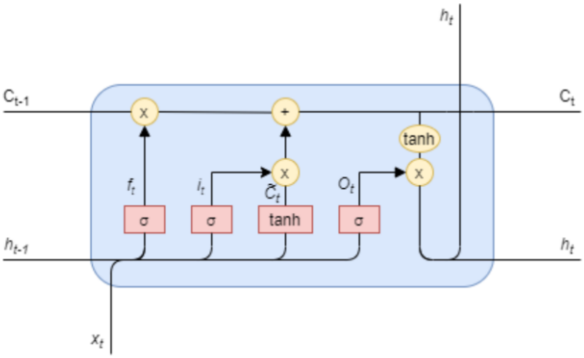

Figure 5. A basic LSTM cell [14].

In Eq. (1) - Eq. (5), $i, f, o, (C), h$ denote input gate, forget gate, output gate, internal state, and hidden layer, respectively. Here, W_i, W_f, W_o, W_C, and b_i, b_f, b_o, and b_C represent the weights and bias of three gates and a memory cell, in the order given. Concretely, the activation function *sigmoid* helps an LSTM model to control the flow of information because the range of this activation function varies from 0 to 1 so that if the value is 0 then all of the information will be cut off, otherwise the entire flow of information will pass through. Similarly, the output gate will allow information to be revealed appropriately due to the sigmoidal activation function then the weights will be updated by the element-wise multiplication of output gate and internal state activated by non-linearity *tanh* function.

In the LSTM neural network architecture proposed by [2], input vector x_i at each timestep is feature vector of each cycle as described above. This LSTM architecture uses L previous cycle to predict C_{full} of next cycle. As discussed, this work has enjoyed the best performance so far in terms of SOH prediction.

3. The BiAR-SeqInSeq Model

In this section, we present our proposed BiAR-SeqInSeq model. For the sake of convenience, technical details of our model is presented in separate subsections. Firstly, we present the *Autoregression* technique. Then, we present our feature encoding strategy for each charge cycle as the premise for our next discussion about the sequence-in-sequence bidirectional architecture. Finally, we present how to combine these techniques into our ultimate model.

3.1. The Autoregression Technique

In statistics, econometrics and signal processing, an *Autoregression (AR)* model is a *random process*, which specifies that the output variable depends linearly on its own previous values and on a stochastic term. Thus, the model can be considered as a form of a stochastic difference equation evaluated on the values from the past.

In general, given the value $C_{k+1}, C_{k+2}, ..., C_{k+L}$ where C_j is the value the j^{th} timepoint and C_{k+L} the current time, the value of the next value C_{k+L+1} can be evaluated as Eq. (6)

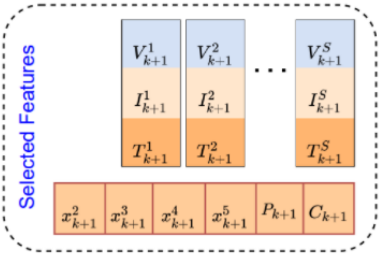

Figure 6. The features selected inside a cycle (channel-wise features).

$$C_{k+L+1} = c + \sum_{j=1}^{L} \varphi_j C_{k+j} + \varepsilon_{k+L+1} \tag{6}$$

where c is a initial constant (or bias), φ_j is the corresponding coefficient of C_{k+j} and ε_{k+L+1} is a random noise. There are different methods to evaluate φ_j, one of which is to approximate them by means of a neural network, as we further discuss and apply in our works.

3.2. Cycle Encoding

Extended from [2], we rearrange 3 channel information inside each cycle into 11 vectors $\xi_j = (V_j, C_j, T_j); j = 1..11$, corresponding to 11 sampling time steps as illustrated in Figure 6. Additionally, in each cycle, we also additionally extract an encoded vector $E_V = (x_2, x_3, x_4, x_5, peak, C)$, where the meaning of $x_2, x_3, x_4, x_5, peak, C$ are already discussed in Section 2.1. ξ_j and E_V will be used in our sequence-in-sequence training mechanism in later discussion.

3.3. Bidirectional Seq-in-seq Training Strategy

With feature vectors described above for each cycle, the features are encoded as the input using the sequence-in-sequence mechanism to train our predictive model as illustrated in Figure 7. We call this mechanism sequence-in-sequence because the gradients flow through two nested sequences during the training phase: local sequence and global sequence, or *channel-wise* and *cycle-wise* sequences.

3.3.1. Bidirectional LSTM

In both channel-wise and cycle-wise sequences, data-points are trained in forward and backward directions between time steps, hence the name bidirectional. Recall that in [2], the author used LSTM architecture to learn from sequential data.This LSTM is unidirectional, which reads input data from left to right order. In this research, we use another variation of LSTM, known as Bidirectional LSTM, or Bi-LSTM [10] to better capture sequential relationships in each sequence. As illustrated in Figure 7, Bi-LSTM neural network composed of two independent LSTM networks in opposite directions, w.r.t the order of the input data-points. The output hidden vectors to those LSTM manner are then concatenated in a cell-wise manner, producing the final output of the Bi-LSTM network.

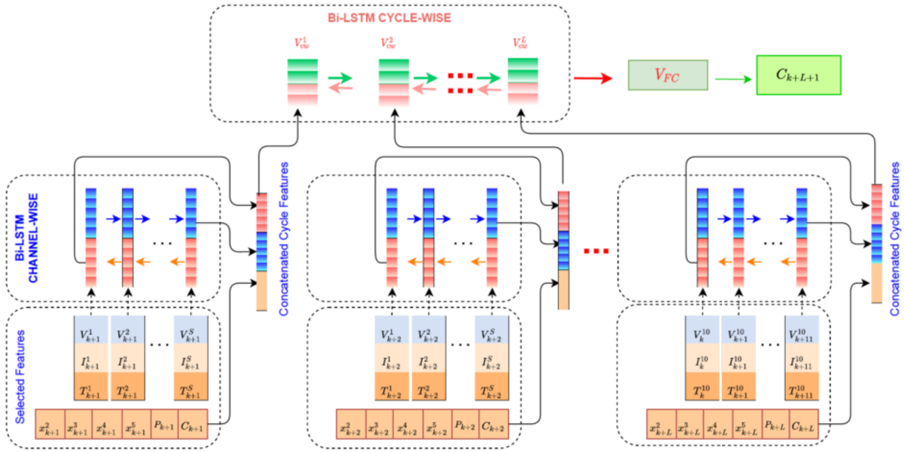

Figure 7. The Bidirectional seq-in-seq training strategy.

3.3.2. *Channel-wise sequence*

Channel-wise sequence is equivalent to a Bi-LSTM neural network whose input are 11 vectors $\xi_j = (V_j, C_j, T_j); j = 1..11$ above discussed. The final output of this Bi-LSTM is then concatenated with the vector $E_V = (x_2, x_3, x_4, x_5, peak, C)$ as the final output of this processing step. We denote this result vector as a channel-wise feature vector V^j_{cw}.

3.3.3. *Cycle-wise Sequence*

Cycle-wise sequence is also a Bi-LSTM neural network, whose inputs are channel-wise feature vectors $V^j_{cw}, (j = 1, .., 11)$, encoded from $L = 11$ consecutive cycles in the past, after processed in the channel-wise sequence step. The final output of this global BiLSTM, denoted as V_{FC}, will flow through a fully connected layer to predict SOH of the next cycles.

3.4. *The ultimate BiAR-SeqInSeq Model Architecture*

Figure 8 captures our ultimate model, in which bidirectional seq-in-seq architecture is combined with Autoregression technique to improve prediction results. We enhance our prediction by combining the two results from two computational branches, known as Non-linear Branch and Linear Branch, whereas Non-linear Branch output is the above mentioned V_{FC}, now is referred as C_{non_linear}. Meanwhile, in Linear-Branch, we apply Autoregression on the SOH values of the L processed cycles $(C_{k+1}, C_{k+2}, .., C_{k+L})$ to obtain the $C_{regression}$ result (Eq. (7)). Finally, the C_{non_linear} will be added up $C_{regression}$ based on the trainable combined parameter as the below Eq. (8).

$$C_{regression} = \sum_{j=1}^{L} \varphi_j C_{k+j} \tag{7}$$

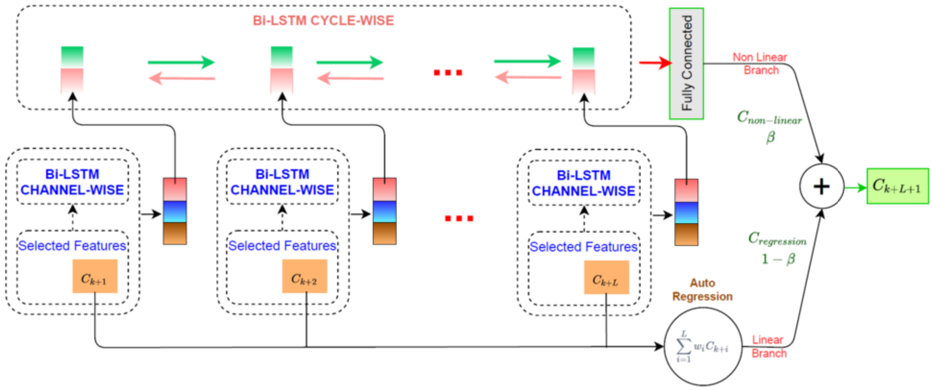

Figure 8. The ultimate BiAR-SeqInSeq mode.

$$C_{k+L+Q} = \beta C_{non\_linear} + (1-\beta)C_{regression} \tag{8}$$

As discussed, the weights φ_j and β are trainable parameters optimized in the training of the whole model to achieve the best prediction performance.

4. Experiments

In this section, we describe the experiments to confirm the performance of our proposed BiAR-SeqInSeq model. The datasets of 4 NASA batteries and the cross-validation techniques used in [2] are inherited. The quantitative comparisons will be given as below.

4.1. Dataset

Dataset used in this research are provided by NASA [16], including 4 batteries named B0005, B0006, B0007, B0018. The first three battery modules provide full 168 cycles (charge and discharge) and the last B0018 123 cycles. These dataset specification is presented in Table 1. Test conditions of these batteries are presented in Table 2.

Table 1. Summary of 4 NASA battery modules over the cycles.

Battery No.	Battery Structure	Length of Capacity Data
B0005	1×616	168
B0006	1×616	168
B0007	1×616	168
B0018	1×319	123

Table 2. Test condition of NASA LIB dataset.

Battery No.	Constant Charge Current(A)	Charge Cut-off Voltage(V)	Discharge Current(A)	Discharge Cut-off Voltage(V)
B0005	1.5	4.2	2.0	2.7
B0006	1.5	4.2	2.0	2.5
B0007	1.5	4.2	2.0	2.2
B0018	1.5	4.2	2.0	2.5

We used all 4 batteries historical data to develop the models, in which 80% of the dataset was used for training and the remaining 20% was used for testing.

4.2. Evaluation results

In our experiments, we develop the following models.

- LSTM-based Multi-Channel Prediction [2]: As discussed, this is the current SOTA performance of SOH estimation problem.
- LSTM-based Multi-Channel Prediction: based specifically on [2], apply basic fine-tuning and noise-removal techniques to record the output. We have no seq-in-seq and cycle-wise features in this experiment.
- Con2DLSTM: Another additional model we develop for experiment. This architecture uses 2D CNN to extract cycle features and use LSTM to predict [17].
- Bi-SeqInSeq NoAR: similar to above BiLSTM-seq-in-seq-Autoregression but without Autoregression is applied.
- BiLSTM-seq-in-seq-Autoregression (BiAR-SeqInSeq): This is the model we presented.

Experiments results are summarized in Tab. 3. We can see that the baseline LSTM-based Multi-Channel [2] method is still outperforming deep-learning based time series analysis methods such as its baseline. However, models Conv2D_LSTM and Bi-SeqInSeq results even better than the baseline due to their capability to capture both local features and global features. Besides that, Autoregression technique helps to coordinate channel-wise and cycle-wise information most reasonably by trainable parameters learnt by back propagation mechanism when training the end-to-end model, which also significantly contributes to our best performance enjoyed.

Table 3. Summary of performance outcome.

Index	Methods	RMSE	Reduction outcome
(1)	LSTM-based Multi-Channel [2]	0.0268 - 0.030	
(2)	LSTM-based Multi-Channel (No multi-task, no Conv2D)	0.040	
(3)	Conv2D_LSTM [17]	0.033 - 0.034	Reduce 17.5% compare to (2)
(4)	Bi-SeqInSeq NoAR (Without Autoregression)	0.036	Reduce 10.0% compare to (2)
(5)	**BiAR-SeqInSeq**	**0.027**	Reduce 10% compare to (1) and 32.5% compare to (2)

The experiment result on Tab. 3 shows that the overall loss is reduced approximately 10% from 0.030 (reported in [2]) to 0.027 ((0.030-0.027)/0.030 = 10%) compared between LSTM-based Multi-Channel to BiAR-SeqInSeq. Moveover, if we compare with the result of the same method re-implemented and worked on the same data set, the loss is reduced from 0.040 (LSTM-based Multi-Channel) to 0.036 (Bi-SeqInSeq NoAR, reduce ((0.04-0.036)/0.040 = 10%) and to 0.027 (BiAR-SeqInSeq, reduce ((0.040-0.027)/0.040 = 32.5%), which are significant.

5. Conclusion

We have proposed some set of improvements on a SOTA method in Lithium-ion Battery SOH prediction using LSTM-based deep learning model. Based on the deep study of LIB data, we have developed a sequence-in-sequence advancement. Moverover, thanks to the dependency on cycle-wise SOH time-series, we keep the Autoregression portion to play its role in final prediction.

The improvements originate from the deep understanding of the nature seq-in-seq in LIB data. Before our study, the existing LIB researches have not fully investigated the dependency. Therefore, we create deep improvement on the outcome. The experiment result shows that the proposed method creates improvement in almost all cases and the ultimate model BiAR-SeqInSeq definitely gave significant improvement.

From the promising result presented, we will continue data-driven investigation, applying more natural enhancements into the model. The current method, still, shows some shortcomings at the discrete 11 datapoints sampling approach. Moreover, the extracted features have room to apply an attention mechanism. As presented in [18], the attention model can extract the sequence contribution and prioritize the features contribution into the final output. We will apply the above ideas into our future research.

Acknowledgement

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number C2021-20-13.

References

[1] Xiaodong X, Chuanqiang Y,Shengjin T, Xiaoyan S, Xiaosheng S, Lifeng W. Remaining useful life prediction of Lithium-Ion batteries based on wiener processes with considering the relaxation effect. Energies, MDPI, Open Access Journal, vol. 12(9), pages 1-17; 2019.

[2] Kyungnam P, Yohwan C, Won JC, Hee-Yeon R, Hongseok K. LSTM-Based battery remaining useful life prediction with Multi-Channel Charging Profiles. IEEE Access, vol. 8, pages 20786-20798; 2020.

[3] Saha B, Goebel K. Battery data set, NASA Ames Prognostics data repository. NASA Ames, Moffett Field, Calif, USA; 2007.

[4] Chao H, Gaurav J, Schmidt CL, Carrie S, Sullivan M. Online estimation of lithium-ion battery capacity using sparse Bayesian learning. Journal of Power Sources, vol 289, pages 105-113; 2015.

[5] Ji W, Chenbin Z, Zonghai C. An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. Journal of Applied Energy, vol 173, pages 134-140; 2016.

[6] Yuefeng L, Guangquan Z, Xiyuan P, Cong H. Lithium-ion battery remaining useful life prediction with long short-term memory recurrent neural network. Proceedings of the Annual Conference of the PHM Society; 2017.

[7] Funa Z, Po H, Xiaohui Y, RUL prognostics method based on real-time updating of LSTM parameters. Chinese Control And Decision Conference (CCDC); 2018.

[8] Yohwan C, Seunghyoung R, Kyungnam P, Hongseok K. Machine Learning-Based Lithium-Ion battery capacity estimation exploiting Multi-Channel Charging Profiles. IEEE Access on Advanced Energy Storage Technologies and Their Applications, vol 7, pages: 75143 - 75152; 2019.

[9] Benvolence C, Chunhui C, Lien-Kai C, Wei-Chih S, Mi-Ching T. Long Short-Term Memory approach to estimate battery remaining useful life using partial data. IEEE Open Access, vol 8, pages 165419-165431; 2020.

[10] Alex G, Santiago F, Jürgen S. Bidirectional LSTM networks for improved phoneme classification and recognition. International Conference on Artificial Neural Networks, page 799-804; 2005.

[11] Guokun L, Wei-Cheng C, Yiming Y, Hanxiao L. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks Attention for multivariate time series. International ACM SIGIR Conference on Research Development in Information Retrieval; 2018.

[12] Peter K, Simon FS, Christian von L, Holger H, Raghavendra A, Andreas J. Lifetime analyses of Lithium-Ion EV batteries. 3rd Electromobility Challenging Issues conference (ECI), Singapore; 2015.

[13] Van Brakel. Z-score based streaming peak detection, 2014 (accessed October 5, 2020). URL https://stackoverflow.com/questions/22583391.

[14] Sepp H, Jurgen S. Long Short-Term Memory. Neural Computation, vol 9, pages 1735–1780; 1997.

[15] David E.R, Geoffrey E.H, Ronald JW. Learning representations by back-propagating errors. Nature, vol 323, pages 533–536; 1986.

[16] Nasa dataset online at: Available online: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/. Access Aug, 2021.

[17] Xingjian S, Zhourong C, Hao W. Convolutional lstm network: A machine learning approach for precipitation nowcasting. Computer Vision and Pattern Recognition; 2015.

[18] Ashish V, Noam S, Niki P, Jakob U, Llion J, Aidan N.G, Lukasz K, Illia P. Attention is All you Need. Computer Science; 2017.