ARNS: A Data-Driven Approach for SoH Estimation of Lithium-Ion Battery Using Nested Sequence Models With Considering Relaxation Effect

THIEN PHAM, TRUNG LE, DAT DANG, HUNG BUI, HOANG PHAM, LOI TRUONG, MAO NGUYEN, HUNG VO, AND QUAN THANH THO

Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City 740030, Vietnam
Vietnam National University Ho Chi Minh City, Ho Chi Minh City 720325, Vietnam

Corresponding author: Quan Thanh Tho (qttho@hcmut.edu.vn)

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

ABSTRACT

In recent years, lithium-ion batteries (LIB) have been used widely in portable electronic devices because of their advantages of durability, stability, high-capacity, low-cost, light-weight and small-scale. It makes LIB also deployed in various complex systems, in which efficient prediction of battery data, especially state-of-health (SoH), becomes crucial to ensure that the systems work stably without risks of power interruptions. With the recent improvement of Artificial Intelligence (AI) technologies, many works have been reported using deep learning (DL) models to investigate this problem, since such models can potentially increase their performance with more training data. This is also our direction in this research, which introduces a novel data-driven approach so-called Autoregression Nested Sequence (ARNS). On one hand, we come up with a nested sequence model to efficiently aggregate channel-wise and cycle-wise information, both of which are closely related to the operations of LIB. On the other hand, we incorporate relaxation effects into the model operations to handle peak prediction. To the best of our knowledge, ARNS is the first sophisticated deep learning model that combines all those features into a whole predictive system. The experimental results obtained using the NASA and CALCE datasets confirm significant improvement of ARNS, especially when dealing with peak periods in different SoH of multiple cycles.

INDEX TERMS
Lithium-ion battery, nested sequence models, SoH estimation, relaxation effect.

I. INTRODUCTION

In recent years lithium-ion batteries (from now on referred to as LIB) are dominating in portable electronics and penetrating the electric vehicle market due to an unmatchable combination of high energy density, high voltage, and long life cycle [1]. In [2], the high energy efficiency of LIB may also allow their use in electric grid energy storage and other Internet of things (IoT) applications, including improving the quality of energy harvested from wind, solar, geothermal, and other renewable sources, thus contributing to their more widespread use and building an energy-sustainable economy. Especially, as shown in [3], grid energy storage applications relying on LIB have continuously been increased nowadays.

However, LIB performance declines over time (calendar aging) and use (cycle aging), which can lead to degraded performance, costly replacement, operational impairment, or even catastrophic consequences [4]. Hence, it is necessary to take measures to improve the reliability and safety of lithium-ion batteries. Engineering practice shows that the Battery Management System (BMS) [5] is essential for ensuring the safe, effective and reliable operation of LIB. BMS can also provide the best performance management and life extension solution for energy storage. The main tasks of the BMS include online monitoring of the battery’s voltage and...
current, as well as providing real-time State-of-Health (SoH) and State-of-Charge (SoC) [6].

In this paper, we focus on the SoH aspect. Generally, SoH estimation is a critical metric in a BMS to quantify the extent of degradation. Basically, SoH reflects the condition of a battery, compared to its ideal conditions. This indicator is the battery capacity, which is defined as the ratio of current maximum available capacity $C_{\text{max}}$ over the rated value given by the manufacturer $C_{\text{rated}}$ as:

\[
\text{SoH} = \frac{C_{\text{max}}}{C_{\text{rated}}} \times 100\% \tag{1}
\]

Therefore, the issue of SoH estimation can be converted to capacity estimation. In actual use, the $C_{\text{max}}$ of LIB declines over time, which is the overall trend of the data. However, due to the time-varying external environment variables and the complexity of internal electrochemical performance in a practical application, the battery’s capacity degrades irregularly (Figure 1a and Figure 1b), which increases the difficulty of SoH estimation.

Many studies have been conducted to determine an accurate estimation of SoH for lithium-ion batteries. These methods include impedance spectroscopy and electrochemical techniques, or model-based methods that rely heavily on estimating or identifying the characteristic parameters of LIB [9], [10], [11]. More recently, with the advancement of Artificial Intelligence (AI), in particular machine learning, data-driven methods based mainly on data using analytical, statistical, and machine learning models, without prior knowledge of the battery’s inner workings [12] have gained more attention. Especially, Park et al. [13] proposed a deep learning model based on Long Short-Term Memory (LSTM) that achieved a significant result in capacity and Remaining Useful Life (RUL) prediction, which is still considered state-of-the-art in this area. In this research, Park et al. leveraged internal information of the batteries, or channels, including voltage, current, and temperature throughout the batteries working lifecycles, known as multi-channel information. Nevertheless, apart from such channel-wise data, this work has not leveraged aggregated cycle inputs (cycle-wise information). In our work, we aim to exploit both channel-wise and cycle-wise information about the battery, presenting a nested sequence model as subsequently discussed. The method proposed by Park et al. [13] is therefore adopted as the baseline of this paper.

One of the main difficulties of SoH estimation is that the LIB’s capacity degrades after each cycle but occasionally spikes up. This may have been a result of LIB relaxation effect, which can be expressed as follows. If the battery rests some time longer than the regular break time between two consecutive cycles, it will lead to recovery of the battery. This dependency could increase the available capacity for the next cycle [6]. The relaxation effect is a typical feature of the LIB degradation process and should be considered in SoH estimation.

When considering some existing popular datasets in the area such as NASA, CALCE, or Oxford datasets, deep learning architects have proved to provide high-accuracy output for SoH trend prediction. However, there are just a few models to utilize multi-channel information extracted from LIB practical usages. Meanwhile, the relaxation effect may be very helpful to predict peak cycles. To the best of our knowledge, there is no existing research to combine these two approaches. In this paper, we leverage a deep learning model into the SoH estimation of LIB that have been impacted by the relaxation effect. The main contribution of this paper is that we introduce an AI-based Data-driven SoH estimation method that also considers the relaxation effect, called Auto Regression Nested Sequence (ARNS) model. As compared to other approaches, ARNS introduces the following novel contribution:

- We introduce a mechanism of nested sequence model to handle seamlessly both channel-wise and cycle-wise data of LIB during their charge cycles.
- We leverage the incorporation of relaxation effect (RE) information into the model to specifically handle the peak prediction.

On top of that, to predict the trend of sequences, we adopt the Auto Regression branch instead of an additional LSTM layer to reduce model computation.

To the best of our knowledge, ARNS is the first work that incorporates the RE into a deep learning model. Our experiments also confirmed that ARNS outperforms other models, especially when predicting SoH at the peak cycles.

The rest of this paper is organized as follows. Section II explains the Battery Data which are analyzed in our problem. Next, Section III describes Related Works needed to be
TABLE 1. Summary of NASA Battery dataset [7].

| Battery No. | Length of Capacity Data (hr) | Constant Current (A) | Upper Voltage Limit (V) | Cut-off Current (mA) | Constant Current (A) | Cut-off Voltage (V) |
|-------------|------------------------------|----------------------|-------------------------|----------------------|----------------------|---------------------|
| #5          | 168                          | 1.5                  | 4.2                     | 20                   | 2                    | 2.7                 |
| #6          | 168                          | 1.5                  | 4.2                     | 20                   | 2                    | 2.5                 |
| #7          | 168                          | 1.5                  | 4.2                     | 20                   | 2                    | 2.2                 |
| #18         | 133                          | 1.5                  | 4.2                     | 20                   | 2                    | 2.5                 |

understood, prior to the application of SoH prediction. In Section IV, preliminary knowledge of sequence-based deep learning models is presented. Later on, Section V proposes and discusses in detail our ARNS model. In Section VI, the experimental results obtained using benchmarking datasets are presented, together with quantitative data and comparison. Finally, Section VII concludes the paper.

II. BATTERY DATA

In this study, we use the two LIB datasets known as NASA and CALCE. Data from the NASA repository was collected from a custom-built battery prognostics testbed at the NASA Ames Prognostics Center of Excellence (PCoE)\(^1\) [7]. This dataset contains the test results of commercially available lithium-ion 18650-sized rechargeable batteries [14] in which we selected 4 batteries #5, #6, #7 and #18 that are widely used in SoH estimation [6], [15]. Similarly, the CALCE data was given for the usages of battery state estimation, remaining useful life prediction, accelerated battery degradation modeling, and reliability analysis. This battery type has a rated capacity of 1.1 Ah with the discharging current (1.1 A), and the cycling test of the batteries is implemented under ambient temperature. The 4 batteries #35, #36, #37, #38 are focused in comparison in this paper based on similar works in this area [16]. Figure 1a and 1b summarize the capacity degradation of the batteries through their working cycles.

A. BATTERY DEGRADATION DATA

Typically, the lithium-ion batteries were run through 3 different operational profiles (Charge cycle, Discharge cycle, and Impedance Measurement), in which Impedance Measurement can be used in some typical experimental methods to predict SoH by analyzing the battery’s electrochemical impedance spectroscopy [11].

In the NASA dataset, the Charge cycle was carried out in Constant Current (CC) mode at 1.5 A until the battery voltage reached 4.2 V and then continued in a Constant Voltage (CV) mode until the charge current dropped to 20 mA. The Discharge cycle was carried out at a constant current (CC) level of 2 A until the battery voltage fell to 2.7 V, 2.5 V, 2.2 V and 2.5 V for batteries #5, #6, #7, and #18 respectively. However, in the CALCE dataset, degradation tests were performed for a set of cells with LiCoO\(_2\) cathode of 1.1 Ah capacity rating.

The cells were charged and discharged repeatedly subjected to a charging profile using CC-CV protocol in ambient temperature (\(1^\circ C\)). The current was maintained at 1 A till the voltage reached 4.2 V. The charge was maintained at 4.2 V until the charging current was reduced to 0.05 A. The failure threshold for the cells was configured to be 0.88 Ah. The specification for each batteries data are presented in Table 1 and Table 2.

B. REST PERIODS

Normally, the LIB data include two-time scales, calendar time and a number of cycles [6]. From now on, we denote \(t^c\) and \(t^s\) to indicate the calendar time and the index number of cycles, respectively. Thus, for each cycle, the time scale \(t^c\) can be mapped to a specific calendar time scale \(t^s\). From Figure 2, it can be observed that the capacity regeneration is often caused by the calendar time gap between two cycles exceeding some threshold [17]. Therefore, an effective SoH estimation model should try to encode this information.

In the past, various works researched the correlation between rest period and lithium-ion battery’s capacity regeneration, in other words relaxation effect. Paper [18] showed that rest durations provided to a Li-ion cell after every discharge and charge normalize the gradients of concentration and potential in the electrolyte and electrodes. This leads to better utilization of cyclable lithium and therefore regeneration of the battery’s capacity. Moreover, the research also showed that a sufficiently long relaxation of the cell at the end of discharge could result in (a) a higher concentration of lithium in the solid matrix of the negative electrode; and (b) a lower concentration of lithium in the positive electrode, both leading to a higher cell potential, which has significant influence over battery performance in the subsequent cycle.

Furthermore, [19] proved that the amount of capacity regenerated is not linear to the length of the previous rest period. Moreover, experiments with various breaks showed that the duration of the relaxation time is more important than the sum of rest periods in deciding the cycle capacity. In this research, we aim to incorporate the relaxation effect with modern deep learning models to improve the SoH prediction of LIB, especially at peak periods.

III. RELATED WORKS

In this paper, we focus on data-driven methods for SoH prediction. The advantages of those methods are that they do not require specific knowledge about the battery working principles, as such knowledge may be hard to be obtained in

---

\(^1\)The PCoE serves an umbrella for prognostic technology development, especially isolating prognostic technology differences within the aeronautics and space exploration areas.
some specific situations. Instead, those methods only depend on the collected aging data and thus can be automatically processed without much expert knowledge [6].

Figure 3 presents basic steps commonly found in data-driven approaches for SoH estimation [20], which generally produce output as prediction models. Such models are normally initialized with random values. In order to train the model, the historical data from recorded cycles are extracted. The training process has taken place by repeatedly performing model specific calculation based on the difference between the intermediate and measured ground truth values. The model is then adjusted gradually (i.e., fine-tuning its parameters) until it has converged and can be used for providing the estimation for the new SoH data.

Based on the insight of the model training process, one can classify data-driven methods into two major categories: (i) classical machine learning methods, which include statistical approaches and other shallow learning methods; and (ii) deep learning, which were recently developed based on the advancement of this field.

A. STATISTICAL AND SHALLOW LEARNING APPROACHES

Generally, the typical approaches for SoH estimation attempt to fit the degradation law of capacity generally through machine learning methods based on degradation data of LIB and estimating the SoH by extrapolating the capacity in the next cycle.

Statistical methods can be applied to the whole process of SoH prediction. On the one hand, these methods are effective to extract and re-produce effective signals and waveforms from data that suffer from much noise signals. Hence, statistical approaches can be adopted to perform noise filtering to obtain clean waveform data on which models are generated for further predictions. On the other hand, statistical methods can be used directly for SoH prediction as a machine learning approach. It is also noted that currently there is still no clear distinction between statistical and machine learning methods in the literature review. However, the statistical and classical/conventional learning machine, e.g., Bayes Naive Classification or Support Vector Machine share some common points as follows: (i) those methods try to produce output from analysis of training/input data (the learning process); and (ii) the learning process of those methods do not generate a large degree of hidden (or latent) features like deep learning.

### TABLE 2. Summary of CALCE battery dataset [8].

| Battery | Length of Capacity Data | Ambient Temperature (°C) | Upper Voltage Limit (V) | Charge Current (A) | Discharge Current (A) | Rated Capacity (Ah) |
|---------|-------------------------|---------------------------|--------------------------|-------------------|----------------------|---------------------|
| #CS2_35 | 882                     | 1                         | 4.2                      | 0.5               | 1                    | 1.1                 |
| #CS2_36 | 936                     | 1                         | 4.2                      | 0.5               | 1                    | 1.1                 |
| #CS2_37 | 969                     | 1                         | 4.2                      | 0.5               | 1                    | 1.1                 |
| #CS2_38 | 996                     | 1                         | 4.2                      | 0.5               | 1                    | 1.1                 |
models, and therefore they also do not use too many trainable parameters like deep learning models. Hence, we regard them as the same group of shallow learning.

Commonly used statistical filtering methods are mainly divided into Kalman filtering (KF) [21], Particle filtering (PF) [22] and Single Particle model [23]. To enhance the model with higher accuracy and robustness, once proposed Adaptive Hybrid model [24], and Constraint Propagation for LIB data [25] for the optimal design parameters enhancement of identifiability and states observability. There have been some approaches with these methods that take into account the relaxation effect, one of such is proposed in [17].

After the noise filtering steps, some process modeling methods may be further applied using probability theory and stochastic processes to analyze the variation law of historical monitoring data [6], [26]. This type of method often estimates the SoH by establishing a stochastic degradation model of battery capacity [26]. Using stochastic process modeling methods, Xu et al. [6] also introduced a method based on the Wiener Process with modeling the relaxation effect.

Recently, there are many studies using the classical machine learning methods (known as shallow learning) to deal with SoH estimation. Some examples of shallow learning approaches include Bayesian predictive model proposed by Hu et al. [27], Hidden Markov model by Piao et al. [28], Liu et al. [29] used Gaussian process regression and Relevance-vector machine [30]. The internal computations of those shallow learning approaches are still based on statistics, but their fine-tuning processes are performed step-wise until the final convergence is reached.

Even though some remarkable results have been reported, shallow learning methods encounter a common issue when dealing with massive datasets, not only in LIB data but also in all other major domains, is that they hardly take full advantage of large volume and high-dimensional information. To be precise, it is commonly observed that shallow learning methods get saturated after handling a certain volume of training data. After this point, the models hardly improve with more training data provided. In addition, recent surveys on various domains also show that the shallow learning models hardly scale due to the overfit [31], [32]. With the advancement of deep learning methods nowadays, the deep learning models are expected to be capable to work with far more training data before getting saturation points and thus potentially getting better performance [33]. Hence, there are many recent works reported in this direction, which are discussed subsequently.

B. DEEP LEARNING METHODS

Deep learning (DL) is a machine learning concept based on Artificial Neural Networks (ANN) and other advanced variants, such as RNN, CNN, GNN, etc, that typically consist of more than one hidden layers, organized in deeply nested network architectures [34].

An advantageous point of deep learning models is that they are particularly more helpful than shallow learning in domains with large and high-dimensional data [34]. With recent advancements of deep learning, many researchers have also considered leveraging these models in data-driven approaches to SoH estimation. Khumprom and Yodo [14] presented a method using deep neural networks to predict the SoH and the RUL of the lithium-ion battery. Xia and Abu Qahouq [35] proposed an adaptive SoH estimation method utilizing a Feedforward Neural Network (FNN) and online AC complex impedance. She et al. [36] proposed a method for a battery aging assessment using a Radial Basis Function (RBF) neural network. Eddahech et al. [37] showed that Recurrent Neural Networks (RNN) can be used to predict performance decline in batteries. Tian et al. [38] proposed a deep neural network to estimate the charging curves of batteries from which SoH can be computed. Shen et al. [39] proposed a method to predict the usable capacity of a battery using a deep Convolutional Neural Network (CNN) with current and voltage measurement data.

Perhaps the most noteworthy approach in this direction is the use of recurrent networks like LSTM [13], [40], because the nature of this network is very suitable to handle sequence or time-series data. Thanks to the capability of time-series processing, LSTM-based approaches have been proven as one of the most effective methods for SoH prediction.

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 an insightful analysis of the battery status. Thus, the collected channel-wise data of a battery become those of multivariate time series [13]. In [13], a multi-channel charging model is proposed for the prediction task. The paper also incorporates the relationship between capacity and charging profiles, archived by trainable neural networks. Similarly, other partial data is used in [41] using a statistical feature selection and engaging both charge and discharge cycles. These deep learning studies provide an alternative approach that can potentially improve the accuracy of SoH prediction, since deep learning may scale better with large datasets as previously discussed. Then, deep learning would enjoy more benefits in the future, once more data can be gathered and computational infrastructure can also be further upgraded to support more computational ability.

In particular in the work of Park et al. [13], an approach that combined multi-channel charging profiles and an LSTM model to improve the accuracy of the SoH prediction task is introduced. So far, this work is considered as the state-of-the-art result in this area and also adopted as the baseline of this paper, from herewith referred to as BaseLSTM model.

It is also noted that recent studies in this area also introduced some advanced DL models. In [42], the authors introduced a seq2seq model which can foresee the SoH for one cycle. The seq2seq (or sequence-to-sequence) is a DL model including an encoder and a decoder, each of those is a sequence-based deep learning architecture. Normally, the encoder will encode the input as a context vector, based on
which the decoder infers the corresponding output. In [43], a stacked CNN model was presented for the aging prediction of Lithium-ion batteries. However, neither of them concerned the contribution of the relaxation effect, which is the focus of our work.

IV. SEQUENCE-BASED DEEP LEARNING MODELS

A. RECURRENT NEURAL NETWORKS (RNN)

Recurrent Neural Network (RNN) is a deep learning model commonly used for time series problem solving and other kinds of sequence data [44]. In an RNN model, the past information is preserved through the hidden layer and then passed through each cell of the RNN. The RNN is considered one of the most common and earliest models for sequence-based prediction. Depending on the number of inputs and outputs, RNN is classified into different types, such as one-to-one, one-to-many, many-to-one, and many-to-many.

RNN has various practical related applications such as speech-to-text, sentimental classification, machine translation, video recognition, etc. However, the RNN performance is limited by some constraints noticed as sequential and slow training, vanishing and exploding gradient, and long-sequence processing difficulties. Therefore, the next version of RNN, which is LSTM, was introduced to improve the above limitation.

B. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) is an upgraded version of RNN. 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, by a means of a memory cell. This process is controlled by three gates namely input gate \(i\), output gate \(o\), and forget gate \(f\). Basically, a cell of an LSTM is depicted in Figure 4. Firstly, forget gate is used to determine the level of information rejection in the cell:

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

Secondly, the cell will get the input gate to update the current memory cell:

\[
i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)
\]
\[
\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)
\]

Thirdly, it will continue to calculate and cell state the \(C_t\) of the current memory cell:

\[
C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t
\]

The last thing is to calculate the output gate for the current memory cell:

\[
o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)
\]
\[
h_t = o_t \ast \tanh(C_t)
\]

In the above equations \(i, f, o, C, h\) 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 cell, in the order given. Concretely, the activation function \(\text{sigmoid}\) 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 \(\text{sigmoidal}\) activation function then the weights will be updated by the element-wise multiplication of the output gate and internal state activated by the non-linearity \(\tanh\) function. With the pivotal component is the memory cell accommodating three gates, one input, one forget and one output gate, LSTM has overcome the limitations of RNN by enhancing the ability to remember values over an arbitrary time interval by regulating the flow of information inside the memory cell.

C. BIDIRECTIONAL LONG SHORT-TERM MEMORY

Bidirectional Long Short-Term Memory (Bi-LSTM) is an upgraded version of LSTM. Bi-LSTM is proven especially helpful in the occasions where the inputs are of sequence form and could be processed in both forward and backward directions to infer more information [45]. In a Bi-LSTM network, there are two unidirectional LSTMs in opposite directions, as shown in Figure 5. Bi-directional LSTM information flows from backward to forward to enrich information encoded in each step and better understand the context. The model adopts the combination of the input data from the past and the future to give the output of the current time frame. First, the input will go through a forward LSTM layer to get the output of this layer:

\[
h^f_t = \tanh(W^f_{h_t}[h_{t-1}, x_t] + b^f_h)
\]

Then also from this input through the backward LSTM layer to get the output of this LSTM layer:

\[
h^b_t = \tanh(W^b_{h_t}[h_{t+1}, x_t] + b^b_h)
\]

Finally, concatenating these outputs, we get the final output:

\[
y_t = W^f_y h^f_t + W^b_y h^b_t + b_y
\]

In the above equations \(h^f_t, h^b_t\) are outputs of the forward hidden and backward hidden layers considering the input in
ascending and descending order respectively. Finally, the bidirectional LSTM network.

**FIGURE 5.** The bidirectional LSTM network.

We also note that the details of the above-mentioned channel-wise information and cycle-wise information will be given in the subsequent sections.

**B. FEATURE EXTRACTION FROM BATTERY DATA**

In the common working mode, existing LIB in NASA [7] and CALCE [8] datasets stretch up to approximately 1-3 years, equivalent to 100-2000 charge and discharge cycles throughout the working lifespan. During each cycle, LIB internal channels such as current, voltage, temperature or internal resistance are measured, marking multi-channel charging profile information. It is also noted that a battery capacity can be quantified in the discharge stage by measurement of stored coulombs [48]. Data collected during these cycles will be adopted for the later task of battery SoH prediction.

We use such information to extract features of battery data for further processing. The battery data can be classified into two types as follows.

- **Channel-wise data:** they are data collected from the above-mentioned multi-channel charging profile of the battery, represented as a channel-wise feature vector.
- **Cycle-wise data:** they are useful aggregated data of the battery in a cycle, represented as a cycle-wise feature vector.

1) **CHANNEL-WISE DATA**

Channel-wise data is extracted from the multi-channel charging profile of the battery. Figure 7 shows the charging profiles of voltage, current, and temperature for different states-of-health in NASA dataset (90%, 80% and 70%) [13]. Obviously, the charging profiles of voltage, current, and temperature change for each sample points of voltage, current and temperature for different states-of-health in NASA dataset (90%, 80% and 70%) [13].

However, as the multi-channel information of the charging profiles is continuous, we perform sampling to extract the channel-wise feature vectors eventually. As channel-wise data is only used as the input of our Local Sequence Network in the non-linear branch, we will discuss the sampling process and the channel-wise feature vectors in this section. The sampled channel-wise $D_{k}^{\text{chan}}$ is represented as three vectors of length S described in Eq. 11.

$$D_{k}^{\text{chan}} = \begin{cases} V_{1}^{k}, V_{2}^{k}, \ldots, V_{S}^{k} \\ I_{1}^{k}, I_{2}^{k}, \ldots, I_{S}^{k} \\ T_{1}^{k}, T_{2}^{k}, \ldots, T_{S}^{k} \end{cases} \ (or \ R_{1}^{k}, R_{2}^{k}, \ldots, R_{S}^{k})$$

where $V_{s}^{k}$ and $I_{s}^{k}$ are the s-th sample points of voltage, current at k-th cycle, respectively. The $T_{s}^{k}$ or $R_{s}^{k}$ are the temperature (for NASA dataset) or the internal resistance (for CALCE...
It is noted that when developing predictive models in the CALCE dataset, we also use internal resistance as an input parameter. This had been proposed in some studies such as [49] and [50]. In this paper, we do not use the current internal resistance as the input. Instead, these above values of previous cycles are used in SoH prediction of the current cycle, similar to the way that SoH itself of previous cycles is used as a part of the input to infer the SoH for the current cycle [13].

2) CYCLE-WISE DATA
Apart from the timeline-based information collected from each channel, the aggregated information of voltage was also proved to be useful for SoH prediction. As discussed in [51], during the CC-CV process there are 5 significant cycle-wise features as follows.

- $x^1$: Initial charge voltage.
- $x^2$: CC charge capacity, calculated by accumulating the rectangle of $ICC \times t$, where ICC is the constant current in the charge, and $t$ is the duration of the CC period.
- $x^3$: CV charge capacity, accumulated by the integral of the area of decreasing current during the CV period.
- $x^4$: Final charge voltage (Voltage measured at battery terminal).
- $x^5$: Final charge current (Current measured at battery terminal).

The meanings of those five cycle-wise features are also illustrated in Figure 8. However, as confirmed in our experiment, we found that $x^1$ and $x^4$ are voltages measured at the starting and ending periods of the cycle that are not stable and often noisy. Moreover, $x^2$ is found to be correlated with SoH of the current cycle. Thus, we exclude $x^1, x^2, x^4$ from our ultimate cycle-wise feature vectors.

Additionally, besides the SoH at the current cycle $C_k$, in order to take into account the relaxation effect in our cycle-wise feature vectors, we also introduce a new feature of $T_{rest}$, which represents the LIB rest period before every charging cycle. Rest period of a battery at the $k$-th cycle ($T_{rest}^k$) is calculated by Eq. 12 where $t_{s}^k$ is the calendar timestamp of the $k$-th cycle.

$$T_{rest}^k = t_{s}^k - t_{s}^{k-1}$$

(12)

This is the approach to incorporating rest period information into the overall architecture and will later be mentioned in the next section. In addition, we also use $P$, a boolean peak indicator, to represent the peak status of the battery capacity. $P$ is calculated by comparing the current value with the mean value of five previous battery capacities, to decide whether the current capacity is a peak or not.

Thus, our cycle-wise feature vector of a $k$-th cycle ($D^{cycle}_k$) can be ultimately represented as 5-tuple vector as described in Eq. (13).

$$D^{cycle}_k = (C_k, P_k, T_{rest}^k, x^3_k, x^5_k)$$

(13)

Then we use a sliding window of size $L$ (from $k + 1$ to $k + L$) to predict the battery capacity in the $k + L + J$ cycle, where $J$ is the prediction interval. To accurately predict the...
battery capacity, the rest time of the $k + L + J$ cycle is also considered. Due to difficulties in obtaining the starting time of the predicting charge cycle in a real application, we choose a prediction interval of 1 ($J = 1$). Ultimately, our model, as shown in Eq.14 and Eq.15 takes in a sequence of battery data, each represented by a tuple of channel-wise and cycle-wise data along with the rest time of the predicting cycle as input.

$$D_k = (D_{chan}^k, D_{cycle}^k)$$

$$D_{k+1}, D_{k+2}, \ldots, D_{k+L}, T_{rest}^{k+L+J} \rightarrow C_{k+L+J} \tag{15}$$

### C. NON-LINEAR BRANCH (NESTED SEQUENCE MODEL)

1) LOCAL SEQUENCE NETWORK

As discussed, we use Local Sequence Network to handle channel-wise data. It is carried by a Bi-LSTM network, each
unit of which processes channel-wise data of a cycle, represented as a channel-wise feature vector.

As previously mentioned, raw channel-wise data is continuous information collected from three channels: voltage, current and temperature. To make a channel-wise feature vector, we perform sampling from those data. Our sampling points are derived from the work of [52], which introduced the important sampling method of \( S \) points that represent the voltage curve during the CC process of the charging cycle. In this paper, we choose \( S = 11 \) together with the detail of the sampling method depicted in Table 3, and further illustrated in Figure 10.

As observed in Figure 10, there exists a non-sampling interval, which is typically determined because the battery is usually not fully discharged when charging begins in actual use, sampling headroom from 0-30% of the charging process. The closer to the end of the CC charging process, the more the voltage curve changes, so its significance is greater for that charging cycle. Thus, we make more sampling points toward the end of the CC process (i.e., 35%, 52%, 70%, 88%, 90%, 92%, 94%, 96%, 98%, 99%, 100% of the overall CC charge phase of the batteries). Not only voltage value, at each sampling point we also get corresponding values of voltage, current and temperature (for NASA dataset) or internal resistance (for CALCE dataset) of the batteries.

Thus, our input of the Local Sequence Network includes 11 units of the Bi-LSTM network, each of which corresponds to a sampling point. Finally, among the 11 Bi-LSTM units, the output vector of the last one, from now on referred to as charge profile features for convenience, will be taken as the output of the Local Sequence Network.

2) GLOBAL SEQUENCE NETWORK

After the charge profile features vector has been extracted and learned, we concatenate it with the cycle-wise features vector, which includes the battery capacity data and rest time of the charge cycle, to create a single flattened vector as the input for our Global Sequence Network. Then we use a Bi-LSTM many-to-one model similar to the local sequence networks to model the contribution of external changes (charging profile and rest time) to the non-linear changes of battery capacity. As demonstrated in Figure 6, the input of our Global Sequence Network is \( L \) units of a Bi-LSTM network, each corresponds to a charge cycle features vector and the return of the last Bi-LSTM unit is selected as the output of our nested sequence model, from now on referred to as \( C^{\text{non-linear}} \).

D. LINEAR BRANCH

In theory, the nested sequence model is capable of learning linear trends. However, since our nested sequence model is fairly complex the process of learning these trends can be time-consuming. So in order to better train our model we use a simple autoregression (AR) model, which is a linear model that uses observations from previous time steps as input to a regression equation to predict the value at the next step. The inputs of this branch are the capacity data of \( L \) consecutive charge cycles (similar to the non-linear branch), and the output of this branch is a scalar value, from now on referred to as \( C^{\text{linear}} \) (Eq.16).

\[
C^{\text{linear}}_{k+L+j} = \sum_{i=1}^{L} \phi_i \ast C_{k+i}
\]  

where \( \phi_1, \ldots, \phi_L \) are the parameters of the model, \( C_{k+i} \) is the capacity measured in cycle number \( k+i \) and \( C^{\text{linear}}_{k+L+j} \) is the prediction capacity with a prediction interval of \( L \).

E. FINAL DENSE LAYERS

Finally, we concatenate the linear change (\( C^{\text{linear}}_{k+L+j} \)), non-linear change (\( C^{\text{non-linear}}_{k+L+j} \)) and the rest time value of the predicting cycle (\( T^{\text{rest}}_{k+L+j} \)) to create a \( 3 \times 1 \) vector as the input for a artificial neural network consisting of three dense layers with \( tanh \) activation function in the hidden layers. The structure of those three layers is similar to the common multi-layer artificial neural network architect and each node value is described by Eq.17.

\[
Z^i = W_0^i + W_1^i \ast X_1 + W_2^i \ast X_2 + \ldots + W_n^i \ast X_n
\]  

where,

- \( Z^i \) is the symbol for denotation of the value of a node in the \( i \)-th dense layer.
- \( W_i^j \) are the weights corresponding to that node.
- \( X_k^i \) are the independent variables or the inputs.

This network aims to combine linear change and non-linear change to estimate capacity at the next step. The network also considers the rest time value of that step \( T^{\text{rest}} \) to predict a more accurate result.

VI. EXPERIMENTS

A. DATASETS

To verify our proposed methods, we will use two datasets NASA [7] and CALCE [8]. The first dataset was gathered in the accelerated tests under controlled laboratory conditions, and with regular cycling patterns. Meanwhile, the second dataset also comes from lab tests with regular cycling patterns.
TABLE 5. Estimation error compare to baseline (executed on last 30 cycles of NASA dataset).

| Model       | RMSE  | MAPE  | Prediction Interval |
|-------------|-------|-------|---------------------|
| BaseLSTM [13] | 0.0168 | 1.050% | 30                  |
| MC-LSTM [54]  | 0.0130 | 0.694% | 1                   |
| ARNS         | 0.0091 | 0.590% | 1                   |

(a) Battery #5

| Model       | RMSE  | MAPE  | Prediction Interval |
|-------------|-------|-------|---------------------|
| BaseLSTM [13] | 0.0085 | 0.470% | 30                  |
| MC-LSTM [54]  | 0.0135 | 0.658% | 1                   |
| ARNS         | 0.0065 | 0.303% | 1                   |

(c) Battery #7

TABLE 6. Accuracy comparison for all testing cycles on NASA dataset.

| Method | Overall RMSE, MAPE | Peak RMSE, MAPE | Non-Peak RMSE, MAPE |
|--------|-------------------|-----------------|--------------------|
| GF     | 0.0359, 1.8294    | 0.0357, 1.7158  | 0.0363, 0.6530     |
| MLP    | 0.0548, 6.2998    | 0.0543, 6.2025  | 0.0555, 6.4501     |
| SVR    | 0.0378, 2.1377    | 0.0383, 2.1195  | 0.0370, 2.1659     |
| MC-LSTM [54]  | 0.0123, 0.4824    | 0.0116, 0.4128  | 0.0133, 0.5820     |
| ARNS   | 0.0123, 0.5739    | 0.0141, 0.6386  | 0.0092, 0.4812     |

(b) Battery #5

| Method       | Overall RMSE, MAPE | Peak RMSE, MAPE | Non-Peak RMSE, MAPE |
|-------------|-------------------|-----------------|--------------------|
| BaseLSTM [13] | 0.0388 | 1.880% | 30                  |
| MC-LSTM [54]  | 0.0364 | 1.665% | 1                   |
| ARNS         | 0.0280 | 1.419% | 1                   |

(d) Battery #18

TABLE 7. Average accuracy comparison on both NASA and CALCE datasets.

| Method | Overall RMSE, MAPE | Peak RMSE, MAPE | Non-Peak RMSE, MAPE |
|--------|-------------------|-----------------|--------------------|
| GF     | 0.1154, 6.7512    | 0.1147, 6.6994  | 0.1185, 7.1346     |
| MLP    | 0.0297, 2.8748    | 0.0282, 2.7067  | 0.0359, 3.6807     |
| SVR    | 0.1143, 6.6857    | 0.1167, 6.8289  | 0.1021, 5.9992     |
| MC-LSTM [54]  | 0.0111, 0.3029    | 0.0118, 0.3366  | 0.0071, 0.1394     |
| ARNS   | 0.0119, 0.5286    | 0.0126, 0.5660  | 0.0078, 0.3468     |

(c) Battery #7

| Method       | Overall RMSE, MAPE | Peak RMSE, MAPE | Non-Peak RMSE, MAPE |
|-------------|-------------------|-----------------|--------------------|
| BaseLSTM [13] | 0.1066, 6.2992    | 0.0988, 5.8741  | 0.1220, 7.1462     |
| MLP         | 0.0755, 7.8289    | 0.0833, 9.0185  | 0.0543, 5.1885     |
| SVR         | 0.1142, 6.6374    | 0.1062, 6.1575  | 0.1303, 7.7026     |
| MC-LSTM [54]  | 0.0319, 1.4154    | 0.0365, 1.6584  | 0.0211, 0.9467     |
| ARNS        | 0.0245, 1.1916    | 0.0231, 1.0867  | 0.0266, 1.3939     |

(b) Battery #6

| Method       | Overall RMSE, MAPE | Peak RMSE, MAPE | Non-Peak RMSE, MAPE |
|-------------|-------------------|-----------------|--------------------|
| BaseLSTM [13] | 0.0316, 0.1603    | 0.0330, 0.1636  | 0.0297, 0.1558     |
| MLP         | 0.0254, 0.0322    | 0.0267, 0.0365  | 0.0231, 0.0255     |
| MC-MLP      | 0.0266, 0.0253    | 0.0268, 0.0274  | 0.0259, 0.0218     |
| MC-MLP (RE) | 0.0278, 0.0266    | 0.0281, 0.0287  | 0.0271, 0.0233     |
| MC-LSTM [54]  | 0.0251, 0.0231    | 0.0253, 0.0213  | 0.0248, 0.0241     |
| ARNS        | 0.0196, 0.0216    | 0.0183, 0.0188  | 0.0204, 0.0235     |

ARNS (%) improvement 23.17, 17.47 28.34, 17.44 23.07, 12.85

(b) Average accuracy of four batteries on CALCE dataset

Any missing data is replaced with the average of subsequent previous $n$, $(n = 5)$ values for both charge and discharge data. After data cleansing, we performed standard feature normalization so that every model feature has the same range and gradient descent converges faster. In this research, we used min-max normalization as described by Eq. 18. The normalization is performed before splitting the data into training, validation, and test sets.

$$\frac{x_k^s - \min(x)}{\max(x) - \min(x)}$$

where $x_k^s$ is the collection of selected features from the datasets, $x_k^s$ is the $s$-th sampled feature of the $k$-th charge phase.

just as the first dataset. Nevertheless, both datasets have variant and natural features of the lithium-ion data distribution. Especially, both datasets have some peaking movements in SOH degradation, which create challenges in our prediction. The specification of each train and test sets are presented in Table 4.

**B. DATA PREPROCESSING**

Before training the model, the data cleaning is performed using the same step given in [13]. To obtain the capacity data after each charging phase, the datasets are sorted chronologically, and for each charging phase, we select the capacity measured in the next immediate discharge phase.
TABLE 8. Comparison with other RNN-based models (executed on NASA dataset).

| Sequence  | Network |   RMSE  | MAPE (%) |
|-----------|---------|---------|----------|
| ARNS-RNN  | 0.0203  | 1.1605  |
| ARNS-GRU  | 0.0195  | 1.0953  |
| ARNS-LSTM | 0.0174  | 0.8864  |
| ARNS-Bi-LSTM | 0.0131 | 0.6645  |

(a) Battery #5

| Sequence  | Network |   RMSE  | MAPE (%) |
|-----------|---------|---------|----------|
| ARNS-RNN  | 0.0097  | 0.3581  |
| ARNS-GRU  | 0.0124  | 0.5680  |
| ARNS-LSTM | 0.0154  | 0.7314  |
| ARNS-Bi-LSTM | 0.0103 | 0.4274  |

(c) Battery #7

| Sequence  | Network |   RMSE  | MAPE (%) |
|-----------|---------|---------|----------|
| ARNS-RNN  | 0.0362  | 1.7089  |
| ARNS-GRU  | 0.0264  | 1.1437  |
| ARNS-LSTM | 0.0250  | 1.0772  |
| ARNS-Bi-LSTM | 0.0269 | 1.3151  |

(d) Battery #18

FIGURE 12. SoH estimation results verified over last 30 cycles (executed on NASA dataset).

cycle. The final capacity results are denormalized for accurate and fair assessment and will be presented in Ampere Hour (Ah).

C. OUTLIER REMOVAL

To stabilize the input data before feeding it into the models, we use a Z-score [53] Outlier Removal technique. Firstly, we calculate Z-score following the formula below.

\[ Z = \frac{x - \mu}{\sigma} \]  \hspace{1cm} (19)

where \( \mu, \sigma \) are the mean and standard deviation of the population.

Secondly, we compare Z-score with the threshold range (min: -threshold, max: threshold) to decide whether a data point is an outlier.

In our experiments, the threshold \( Z_{\text{NASA}} = 3.0 \) and \( Z_{\text{CALCE}} = 2.5 \) are used separately for NASA and CALCE datasets respectively. Statistically, there are approximately 5% of the data points are excluded from the datasets. This percentage and the above setting create stable output for our experiment.

D. EXPERIMENT METHOD

To ensure a fair result we adopted the experiment method of the baseline paper [13]. In which, each battery’s data is used in training, validation, and testing separately. For example,
with NASA dataset, when battery #5 is in the test set, the other batteries (#6, #7, and #18) are used for training and validation. Similarly, for CALCE dataset, when battery CS2_35 (#35) is in the test set, other batteries CS2_36, CS2_37, CS2_38 (#36, #37, and #38) are used for validation and training procedures. This process is repeated for each of the four batteries. The specification of each training, validation, and testing scenarios are summarized in Table 4. Similarly, we apply the same separation for CALCE dataset in this section, we further compare the proposed ARNS model using Bi-LSTM networks in our ARNS model can be replaced by other RNN-based models. In this section, we further compare the proposed ARNS model using various sequential networks like RNN, LSTM, and GRU in the local and global sequence networks. The selected structure and hyperparameters are kept unchanged from the ones used in the ARNS model using Bi-LSTM. We evaluated the experiment using four batteries. The result is summarized in Table 8 and further described in Figure 14.

As can be seen in Table 8 and Table 9, Bi-LSTM model creates the high-accurate performance overall. With Bi-LSTM being the most stable, it comes in first and second for the test sets of batteries number #5 and #18. Especially for the #5 battery, the model with Bi-LSTM networks outperforms other methods with a significant margin. The ARNS models using RNN and Gated Recurrent Unit (GRU) also outperform others in the test sets of batteries number #6 and #7. Furthermore, inspection shows that our model performs marginally better at the data points where the battery’s capacity spikes or increases.

Moreover, continuously executed on NASA dataset, Table 6 describes our comparison on the accuracy performance of overall battery cycles. In this table, we separate peak and non-peak testing sets, to prove our hypothesis. It is obvious that the ARNS architect provides similar loss on non-peak sections, compared with some methods such as Gaussian Fitting (GF), Multi-Layer Perceptron (MLP), Support Vector Regression (SVR), Multi-channel [13] (MC-LSTM) but creates stable improvement peak counterparts.

Table 7 depicts average accuracy output from a list of conventional methods (GF, MLP) as well as the MC-LSTM [54] and ARNS. In Table 7a, the average calculation clearly denotes this evaluation over all four batteries executed on NASA dataset. Similarly, Table 7b describes our experiment on another CALCE dataset. This CALCE measurement was recorded in the ambient temperature and during more than 800 cycles of the four battery batteries #35, #36, #37, #38. Table 7b shows that the ARNS also stably outperforms the above-mentioned methods in all the settings.

### F. COMPARISON WITH OTHER NESTED RNN-BASED MODELS

One has to wonder if the Bi-LSTM networks in our ARNS model can be replaced by other RNN-based models. In this section, we further compare the proposed ARNS model using various sequential networks like RNN, LSTM, and GRU in the local and global sequence networks. The selected structure and hyperparameters are kept unchanged from the ones used in the ARNS model using Bi-LSTM. We evaluated the experiment using four batteries. The result is summarized in Table 8 and further described in Figure 14.

As can be seen in Table 8 and Table 9, Bi-LSTM model creates the high-accurate performance overall. With Bi-LSTM being the most stable, it comes in first and second for the test sets of batteries number #5 and #18. Especially for the #5 battery, the model with Bi-LSTM networks outperforms other methods with a significant margin. The ARNS models using RNN and Gated Recurrent Unit (GRU) also outperform others in the test sets of batteries number #6 and #7. In Table 9, the average calculation proved that ARNS with Bi-LSTM provides the best accuracy when considering all four batteries of the NASA dataset.

### G. MODEL COMPLEXITY AND RUL PREDICTION

Table 9 (parameters column) and Figure 16 show the number of parameters and estimation error (RMSE) of our ARNS model using different sequence networks. The estimation error is achieved by averaging the calculated RMSE and MAPE (Eq. 20 and Eq. 21) of the entire four NASA batteries. In addition, we include the test result of the MC-LSTM also published by Park in [54] that uses the same capacity regeneration happens irregularly.

### TABLE 9. Average accuracy improvement compare with baseline [MC-LSTM [54]] (executed on NASA dataset).

| Method          | RMSE  | MAPE (%) | Parameters | Improvem. RMSE | Improvem. MAPE |
|-----------------|-------|----------|------------|----------------|---------------|
| MC-LSTM [54]    | 0.0246| 1.0320   | 2772       |                |               |
| ARNS RNN        | 0.0231| 1.1178   | 397        | 6.10%          | -8.31%        |
| ARNS-GRU        | 0.0201| 0.9505   | 632        | 18.29%         | 7.90%         |
| ARNS-LSTM       | 0.0201| 0.9354   | 682        | 18.29%         | 9.36%         |
| ARNS-MC-LSTM    | 0.0189| 0.8517   | 1152       | 23.17%         | 17.47%        |

### TABLE 10. Summary of the hyper-parameters.

| Dataset     | Method          | Number of iterations | Number of cells | Number of LSTM layers | Size of hidden nodes |
|-------------|-----------------|----------------------|-----------------|-----------------------|---------------------|
| NASA        | GF              | 100000               | -               | -                     | -                   |
|             | MLP             | 150                  | -               | -                     | -                   |
|             | SVR             | No limit             | -               | -                     | -                   |
|             | MC-LSTM [54]    | 500                  | 10              | 10                    | 21                  |
|             | ARNS            | 500                  | 4               | 10                    | 10                  |
| CALCE       | GF              | 100000               | -               | -                     | -                   |
|             | MLP             | 1000                 | -               | -                     | 8                   |
|             | MC-LSTM         | 1000                 | -               | -                     | 128                 |
|             | ARNS-MC-LSTM    | 1000                 | 8               | 2                     | 256                 |
|             | ARNS            | 1000                 | 8               | 2                     | 256                 |

### G. MODEL COMPLEXITY AND RUL PREDICTION

Table 9 (parameters column) and Figure 16 show the number of parameters and estimation error (RMSE) of our ARNS model using different sequence networks. The estimation error is achieved by averaging the calculated RMSE and MAPE (Eq. 20 and Eq. 21) of the entire four NASA batteries. In addition, we include the test result of the MC-LSTM also published by Park in [54] that uses the same capacity regeneration happens irregularly. Furthermore, inspection shows that our model performs marginally better at the data points where the battery’s capacity spikes or increases.

Moreover, continuously executed on NASA dataset, Table 6 describes our comparison on the accuracy performance of overall battery cycles. In this table, we separate peak and non-peak testing sets, to prove our hypothesis. It is obvious that the ARNS architect provides similar loss on non-peak sections, compared with some methods such as Gaussian Fitting (GF), Multi-Layer Perceptron (MLP), Support Vector Regression (SVR), Multi-channel [13] (MC-LSTM) but creates stable improvement peak counterparts.

Table 7 depicts average accuracy output from a list of conventional methods (GF, MLP) as well as the MC-LSTM [54] and ARNS. In Table 7a, the average calculation clearly denotes this evaluation over all four batteries executed on NASA dataset. Similarly, Table 7b describes our experiment on another CALCE dataset. This CALCE measurement was recorded in the ambient temperature and during more than 800 cycles of the four battery batteries #35, #36, #37, #38. Table 7b shows that the ARNS also stably outperforms the above-mentioned methods in all the settings.

### E. SoH ESTIMATION

In this section, we compare the proposed model with the Multi-Channel LSTM (MC-LSTM) proposed in [13] with a prediction interval of 30 cycles. We also include our test result using the same MC-LSTM as described in [13] with the exact prediction interval as our model (1 cycle), which is also done in [54]. In all four test sets, we use the last 30 cycles to evaluate the models as done in [13]. Table 5 depicts that our models statically outperform the baseline in almost every battery test data, especially with batteries #5 and #18 where
calculation methods and prediction interval. As seen, our models outperform the baseline Multi-Channel LSTM (MC-LSTM) model on both metrics while also being less complex. This improvement confirms the efficiency of our contribution mentioned above of local and global sequence networks data extraction and $T_{\text{rest}}$ data encoding.
However, since RNN based model has to be trained sequentially, the training process of our ARNS models also follows a similar sequential manner. In addition, our model uses the predicting cycle \( (k + L + J) \) rest time feature, which restricts immediate SoH estimation when the charging cycle begins. Thus, predicting Remaining Useful Life (RUL) using this approach is challenging due to the need of rest time data for every charging cycle from the remaining battery lifetime.

After training, the final configuration afterward is summarized in Table 10. Model tuning via grid search is used to pursue the most suitable combination of hyperparameters. The Adam-optimizer [55] and learning rate \( \alpha = 0.001 \) are used in both setting (with NASA and CALCE datasets).

Figure 12 is another visualized figure for the data depicted in Table 5, which shows the prediction for the last 30 cycles in NASA dataset. Similarly, Figure 13 also describes the same information for the last 30 cycles in the CALCE dataset. As seen, the ARNS follows closely with the true data in most of the cases. In another view, Figure 14 and Figure 15 show the visualization of the prediction for all cycles. Moreover, Table 6 (NASA) and Table 7b (CALCE) summarize the quantitative output of these figures respectively, which clearly denotes the significant improvement from our method versus the baseline.

Figure 16 illustrates the comparison of various models (including ours) in two metrics: accuracy and model complexity. Obviously, the less model complexity (i.e., fewer number of parameters), the better. It is noted that there are other metrics such as computational time or memory capacity required. However, more parameters also require more
training time and consume more memory resource. Thus, comparison on number of parameters would be reasonably sufficient for complexity evaluation of models. In Figure 16, the x-axis refers to the number of parameters, and the y-axis describes the accuracy. On the x-axis, ARNS-RNN is the only model that has less than 500 parameters while all other models require more than that. The MC-LSTM has more than converged 2500 parameters and is considered as the most complex model in this set. On the other y-axis, the ARNS-Bi-LSTM can create the best RMSE loss, which is less than 0.019 on average, while all other models met higher RMSE losses correspondingly.

VII. CONCLUSION

This paper has introduced an improvement for the LIB SoH prediction, in terms of accuracy. Especially, the improvements are more visible when the prediction is carried out in peak-centered cycles. Such improvements come from the idea of leveraging multi-channel information incorporated into a suitable deep learning model. In particular, we focus on exploiting the relaxation effect for peak prediction, which has not been implemented yet in an end-to-end model. Two of the most robust contributions are: (1) We introduce a nested sequence learning model utilizing the battery’s multi-channel charging profiles of voltage, current, and temperature. And (2), we extract the crucial feature, i.e., the LIB relaxation effect by engaging the $T^{rest}$ data, which plays the key distribution in the peak periods factor of the SoH. As the result, we have archived significant improvement compared to the baseline. Through experiments on NASA’s LIB datasets, our model performed approximately 23.17% (RSME) and 17.47% (MAPE) better than the baseline. On CALCE dataset, we archived 21.91% (RMSE) and 6.49% (MAPE) better respectively, reported on Table 7a and Table 7b. Especially when considering the estimation at capacity peaks, the ARNS loss deduction are (28.34%, 17.44%) and (27.67%, 11.74%) for (RMSE, MAPE) for two different reported datasets, which are stable and significant.

The above promising output enables the approach of data-driven improvement for deep learning, specifying for SoH estimation. Moreover, a better understanding of LIB data enables us to select the proper features and efficiently optimize the prediction model. We also want to note that up to now our experiments were only conducted with existing common benchmarking datasets, as we still could not have a chance to access unseen battery datasets. It would also open rooms for transfer learning, as we intend to fine-tune our current model to continuously train with new data from additional unseen battery data that we can manage to obtain. This is also a direction we would like to pursue for the future work.

However, in contrast of the advantages of the method, it’s worth to mention the disadvantages of DL approaches used in our method. These models generally suffered higher memory consumption and longer execution time. We also illustrated this concept in Figure 16. Moreover, once the size of data is relatively small, DL models hardly outperform other machine learning methods. DL models only prove their capability when dealing with suitable-sized datasets, which can vary in different situation domains [56]. In our case with NASA datasets, other works [13] showed that the dataset sizes are fit for deep learning approaches and thus suitable for our experiments.

A visible limitation is seen at splitting the training and test sets known as data leakage [57]. This problem was caused by applying min-max normalization before splitting the dataset. As seen, it may cause over-optimistic results. However, since the above separation is inherited from the baseline, we will have the enhancement for this limitation in future works. Moreover, we also plan to include further experiments on real-condition datasets because the current approach may not work with the real LIB data with noise and incomplete cycles.

ACKNOWLEDGMENT

The authors would like to thank the support of time and facilities from the Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

[1] D. Deng, “Li-ion batteries: Basics, progress, and challenges,” Energy Sci. Eng., vol. 3, no. 5, pp. 385–418, Sep. 2015.
[2] A. Manthiram, “An outlook on lithium ion battery technology,” ACS Catal., vol. 3, no. 10, pp. 1063–1069, Sep. 2013.
[3] T. Chen, Y. Jin, H. Lv, A. Yang, M. Liu, B. Chen, Y. Xie, and Q. Chen, “Applications of lithium-ion batteries in grid-scale energy storage systems,” Trans. Tianjin Univ., vol. 26, no. 3, pp. 208–217, Jun. 2020.
[4] H. E. Fernando, “Predicting remaining useful life of a lithium-ion battery accurately using AI,” J. Power Electron., vol. 2020, pp. 1–4, Dec. 2020.
[5] H. Gabbas, A. Othman, and M. Abdussamad, “Review of battery management systems (BMS) development and industrial standards,” Technologies, vol. 9, no. 2, p. 28, Apr. 2021.
[6] X. Xu, C. Yu, S. Tang, X. Sun, X. Si, and L. Wu, “State-of-health estimation for lithium-ion batteries based on Wiener process with modeling the relaxation effect,” IEEE Access, vol. 7, pp. 105186–105201, 2019.
[7] B. Saha and K. Goebel, “Battery data set,” NASA Ames Prognostics Data Repository, Washington, DC, USA, Tech. Rep., 2007.
[8] Calce Battery Research Group. Accessed: Jul. 2022. [Online]. Available: https://web.calce.umd.edu/batteries/data.html
[9] M. Daigle and C. S. Kulkarni, “End-of-discharge and end-of-life prediction in lithium-ion batteries with electrochemistry-based aging models,” in Proc. AIAA Infotech @ Aerosp., San Diego, CA, USA, 2016, doi: 10.2514/6.2016-2132.
[10] M. Safari and C. Delacourt, “Simulation-based analysis of aging phenomena in a commercial graphite/LiFePO4 cell,” J. Electrochem. Soc., vol. 158, no. 12, p. A1436, 2011.
[11] S. Jiang and Z. Song, “Estimating the state of health of lithium-ion batteries with a high discharge rate through impedance,” Energies, vol. 14, no. 16, p. 4833, Aug. 2021.
[12] C. Lin, A. Tang, and W. Wang, “A review of SOH estimation methods in lithium-ion batteries for electric vehicle applications,” Energy Prog., vol. 75, pp. 1920–1925, Aug. 2015.
[13] K. Park, Y. Choi, W. J. Choi, H.-Y. Ryu, and H. Kim, “LSTM-based battery remaining useful life prediction with multi-channel charging profiles,” IEEE Access, vol. 8, pp. 20786–20798, 2020.
[14] P. Khumporn and N. Yodo, “A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm,” Energies, vol. 12, no. 4, p. 660, Jan. 2019.
[15] A. Shahrooei (2020). Comparison of Open Datasets for Lithium-Ion Battery Testing. [Online]. Available: https://medium.com/batterybits/comparison-of-open-datasets-for-lithium-ion-battery-testing-fd0def091ca2
[16] D. Chen, W. Hong, and X. Zhou, “Transformer network for remaining useful life prediction of lithium-ion batteries,” IEEE Access, vol. 10, pp. 19621–19628, 2022.
C. She, Z. Wang, F. Sun, P. Liu, and L. Zhang, “Battery aging assessment using impedance spectroscopy and recurrent neural networks,” *Int. J. Elect. Power Energy Syst.*, vol. 42, no. 1, pp. 487–494, Nov. 2012.

I. Tian, R. Xiong, W. Shen, J. Lu, and X.-G. Yang, “Deep network method for online capacity estimation of lithium-ion batteries,” *J. Energy Storage*, vol. 25, Oct. 2019, Art. no. 100817.

M. Rashid and A. Gupta, “Cochrane review of relaxation periods over cycling performance of a Li-ion battery,” *J. Electrochem. Soc.*, vol. 162, no. 2, pp. A3145–A3153, 2015.

A. Pozzi, G. Ciaramella, K. Gopalakrishnan, S. Volkwein, and S. M. Savaresi, “Kalman filter SoC estimation for lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning,” *IEEE Access*, vol. 8, pp. 165419–165431, Dec. 2020.

B. Rzepka, S. Bischof, and T. Blank, “Implementing an extended Kalman filter for SoC estimation of a Li-ion battery with hysteresis: A step-by-step guide,” *Energies*, vol. 14, no. 13, p. 3733, Jun. 2021.

E.-S. Kim, W. Qiao, and L. Qiu, “Online SOC and SOH estimation for multicell lithium-ion batteries based on an adaptive hybrid battery model and sliding-mode observer,” in *Proc. IEEE Energy Convers. Congr. Expo.*, Sep. 2013, pp. 292–298.

D. Liu, J. Zhou, D. Pan, Y. Peng, and X. Peng, “Lithium-ion battery remaining useful life estimation using a hierarchical model with BiLSTM model for sentiment classification,” in *Proc. Int. Conf. Bangla Speech Lang. Process. (ICBSLP)*, Sep. 2018, pp. 1–4.

P. Spagnol, S. Rossi, and S. M. Savaresi, “Kalman filter SoC estimation for lithium-ion batteries,” in *Proc. IEEE Int. Conf. Control Appl. (CCA)*, Sep. 2011, pp. 587–592.

D. Liu, J. Pang, J. Zhou, Y. Peng, and D. M. Raimondo, “Interval state estimation based on constraint propagation for a lithium-ion cell using an equivalent circuit model,” *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 602–608, 2021.

X.-S. Si, W. Wang, C. Hu, and D.-H. Zhou, “Remaining useful life estimation—A review on the statistical data driven approaches,” *Eur. J. Oper. Res.*, vol. 213, pp. 1–14, Aug. 2011.

X. Hu, J. Jiang, D. Cao, and B. Egardt, “Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2645–2656, Apr. 2016.

C. Piao, Z. Li, S. Lu, Z. Jin, and C. Cho, “Analysis of real-time estimation method based on hidden Markov models for battery system state of health,” *J. Power Electron.*, vol. 16, no. 1, pp. 217–226, Jan. 2016.

D. Liu, J. Pang, J. Zhou, Y. Peng, and M. Pecht, “Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression,” *Microelectron. Rel.*, vol. 53, no. 6, pp. 832–839, 2013.

D. Liu, J. Zhou, D. Pan, Y. Peng, and X. Peng, “Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning,” *Measurement*, vol. 63, pp. 143–151, Mar. 2015.

A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, “A survey on text classification algorithms: From text to predictions,” *Information*, vol. 13, no. 2, p. 83, Feb. 2022.

H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

N. C. Thompson, K. H. Greenewald, K. Lee, and G. F. Manso, “The computational limits of deep learning,” 2020, arXiv:2007.05559.

J. Wu, X. Zhang, Z. Chen, and D. Wang, “A method for state estimation of lithium-ion battery states of health and open circuit voltage reconstruction for lithium ion batteries,” *IEEE Access*, vol. 8, pp. 165419–165431, Dec. 2020.

M. Rashid and A. Gupta, “Measurement Cell Capacity,” [Online]. Available: https://www.electronicdesign.com/technologies/test-measurement/article/21805209/measuring-cell-capacity

J. Tian, R. Xiong, W. Shen, and F. Sun, “Electrode ageing estimation and open circuit voltage reconstruction for lithium ion batteries,” *Energy Storage Mater.*, vol. 37, pp. 283–295, May 2021.

K. Severson, P. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. Chen, M. Ajkol, P. Herring, D. Fraggedakis, M. Bzantz, S. Harris, W. Chueh, and R. Braatz, “Data-driven prediction of battery cycle life before capacity degradation,” *Nature Energy*, vol. 4, pp. 1–9, May 2019.

D. Haifeng, W. Xuezhe, and S. Zechang, “A new SOH prediction concept for the power lithium-ion battery used on HEVs,” in *Proc. IEEE Vehicle Power Propuls. Conf.*, Sep. 2009, pp. 1649–1653.

G.-W. You, S. Park, and D. Oh, “Real-time state-of-health estimation for lithium-ion battery degradation models under dynamic conditions,” *Energies*, vol. 14, no. 13, p. 3733, Jun. 2021.

D. Haifeng, W. Xuezhe, and S. Zechang, “A new SOH prediction concept for the power lithium-ion battery used on HEVs,” in *Proc. IEEE Vehicle Power Propuls. Conf.*, Sep. 2009, pp. 1649–1653.

G.-W. You, S. Park, and D. Oh, “Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach,” *Appl. Energy*, vol. 176, pp. 92–103, Aug. 2016.

C. Hu, G. Jain, C. Schmidt, C. Strief, and M. Sullivan, “Online estimation of lithium-ion battery capacity usage sparse Bayesian learning,” *J. Power Sources*, vol. 289, pp. 105–113, Sep. 2015.

J. Tian, R. Xiong, W. Shen, and F. Sun, “Battery aging assessment for real-world electric buses based on incremental capacity analysis and radial basis function neural network,” *IEEE Trans. Transport. Electricit.*, vol. 16, no. 5, pp. 3345–3354, May 2020.