

Article

Accurate Capacity Prediction and Evaluation with Advanced SSA-CNN-BiLSTM Framework for Lithium-Ion Batteries

Chunsong Lin ^{1,3}, Xianguo Tuo ^{1,3,*}, Longxing Wu ^{2,4,*} , Guiyu Zhang ^{1,3} and Xiangling Zeng ³

- ¹ School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China; lynchunsong@suse.edu.cn (C.L.); zhangguiyu@suse.edu.cn (G.Z.)
² College of Mechanical Engineering, Anhui Science and Technology University, Chuzhou 233100, China
³ School of Mechanical Engineering, Sichuan University of Science and Engineering, Yibin 644000, China; 31808521022@stu.suse.edu.cn
⁴ Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan 430068, China
* Correspondence: tuoxianguo@suse.edu.cn (X.T.); batterywu@163.com (L.W.)

Abstract: Lithium-ion batteries (LIBs) have been widely used for electric vehicles owing to their high energy density, light weight, and no memory effect. However, their health management problems remain unsolved in actual application. Therefore, this paper focuses on battery capacity as the key health indicator and proposes a data-driven method for capacity prediction. Specifically, this method mainly utilizes Convolutional Neural Network (CNN) for automatic feature extraction from raw data and combines it with the Bidirectional Long Short-Term Memory (BiLSTM) algorithm to realize the capacity prediction of LIBs. In addition, the sparrow search algorithm (SSA) is used to optimize the hyper-parameters of the neural network to further improve the prediction performance of original network structures. Ultimately, experiments with a public dataset of batteries are carried out to verify and evaluate the effectiveness of capacity prediction under two temperature conditions. The results show that the SSA-CNN-BiLSTM framework for capacity prediction of LIBs has higher accuracy compared with other original network structures during the multi-battery cycle experiments.

Keywords: lithium battery; sparrow search optimization algorithm; CNN; BiLSTM; capacity prediction



Citation: Lin, C.; Tuo, X.; Wu, L.; Zhang, G.; Zeng, X. Accurate Capacity Prediction and Evaluation with Advanced SSA-CNN-BiLSTM Framework for Lithium-Ion Batteries. *Batteries* **2024**, *10*, 71. <https://doi.org/10.3390/batteries10030071>

Academic Editor: Matthieu Dubarry

Received: 26 December 2023

Revised: 11 February 2024

Accepted: 18 February 2024

Published: 21 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

LIBs are widely used in energy storage systems in various industries and electric vehicles due to their high energy density, light weight, and no memory effect [1–3]. Nevertheless, LIBs in operation are inevitably accompanied by irreversible side reactions, such as the structural degradation of the cathode material [4], the oxidative decomposition and interfacial reaction of the electrolyte [5], the dissolution of the active substance, and the precipitation of the lithium metal [6], which results in a decline in the storage capacity. In these cases, effective and accurate estimation of battery health status can provide reference information for battery users to maintain the normal operation of the system [7]. Particularly, it has high guiding value for prolonging battery life, reducing equipment maintenance costs, and executing efficient battery system management (BMS). For this reason, capacity estimation of LIBs is a critical research topic in the current BMS [8].

A large number of scholars have conducted in-depth research on capacity estimation of LIBs, which can be broadly divided into model-based methods and data-driven methods [9]. The model-based method is to predict the battery capacity degradation trend by establishing a battery physical model and identifying the corresponding relationship between observable quantities and various indicators. For instance, the co-estimation of SOC, capacity, and resistance of lithium-ion batteries was realized in [10] by establishing a high-fidelity electrochemical model. In addition, considering the formation of the solid electrolyte interface (SEI) layer and the crack propagation caused by the volume expansion

of the particles in the active material, a simplified single-particle model was established to achieve high-precision and rapid prediction of capacity attenuation and voltage curve changes with the number of cycles [11]. Although electrochemical models have advantages in providing physical mechanisms, their complex model construction and coupling algorithms pose great challenges for researchers to achieve on-board and efficient battery capacity estimation in BMS.

Data-driven methods, such as artificial neural network (ANN) [12,13], support vector machines (SVM) [14], and Gaussian process regression (GPR) [15], have been widely used and have achieved great success in the battery health management fields. In fact, this method regards the battery as a black box model and relies on experimental data collected to train models that can make predictions about battery aging behavior [16]. Superior to model-based methods, there is no need to explore complex capacity attenuation mechanisms and establish a physical model of LIBs [16]. On the contrary, the feature indicators for aging characterization are only selected to obtain real-time data, which can achieve high estimation accuracy of battery capacity. Among them, CNN technology has achieved great success in the field of battery health estimation because it accommodates more layers with the same number of parameters [16]. For example, based on the method of CNN, Ref. [17] used impedance spectrum as input to achieve high-precision estimation of capacity degradation, but the cost of impedance spectrum acquisition was high in practice. Ref. [18] uses a backpropagation (BP) neural network for SOH estimation, achieving higher estimation accuracy. However, due to the BP network's limited capability in feature extraction, which necessitates manual extraction of certain features, CNN offers significant advantages in this context. In [19], the CNN method was used to estimate the state of health (SOH) of lithium-ion batteries from the voltage, current, and temperature measurements during charging, and good estimation results were obtained. Additionally, the long short-term memory (LSTM) neural network avoiding gradient vanishing and exploding problems has also begun to be applied in battery health estimation in recent years. In Ref. [20], based on the data of the optimal voltage part, the LSTM neural network was used for realizing LIB capacity estimation, and the correlation between the health factor of the voltage part and the complete capacity was analyzed to achieve higher estimation accuracy. In [21], the bidirectional LSTM (BiLSTM) method based on attention mechanisms is used in lithium battery capacity estimation and achieves good results. Gradually, the CNN-LSTM neural network appeared and was utilized for investigating the remaining service life of LIBs, and its life judgment was still based on capacity [22]. Ref. [23] has enhanced the CNN-LSTM algorithm, resulting in improved accuracy for SOH estimation. Ref. [24] employs a BiLSTM network combined with an attention mechanism to estimate SOH. This approach allows the attention mechanism to enhance the BiLSTM network's efficiency, resulting in improved accuracy for SOH estimation. Similarly, Ref. [25] applied the CNN with an additional LSTM layer to predict the remaining useful life (RUL) of LIBs, which extracted the characteristics of the relaxed voltage curve to achieve battery capacity estimation without additional cyclic information. In Ref. [26], a CNN-BiLSTM approach was employed for estimating lithium battery capacity. However, the hyperparameters of neural networks typically require manual tuning, which can be subject to subjective human bias. To mitigate this, swarm intelligence optimization methods have been applied for network optimization. Ref. [27] reported the use of the Particle Swarm Optimization (PSO) algorithm to enhance the Bi-LSTM network, achieving a network with higher accuracy. Similarly, Ref. [28] utilized the PSO algorithm to optimize key parameters of the LSTM, resulting in improved performance in state of charge (SOC) prediction for LIBs. Unfortunately, traditional intelligent optimization algorithms often face challenges such as slow convergence, poor stability, and a tendency to become trapped in local optima. More importantly, in terms of evaluating prediction results, current methods predominantly rely on one or multiple metrics like Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). However, these methods lack a comprehensive description of curve-following scenarios, as discussed and evaluated in Ref. [29].

In summary, the current research progress, whether it be the estimation of battery health status or remaining life, essentially revolves around the issue of capacity estimation. Despite the extensive application of deep learning methods in LIB state estimation, there are still shortcomings that need addressing, which includes the need for further improvement in prediction accuracy, reliance on manual expertise for network parameter tuning, difficulties in feature extraction, and the singularity of evaluation methods. To address these issues, this paper carries out an accurate capacity prediction and evaluation with advanced SSA-CNN-BiLSTM framework for LIBs. Firstly, this method mainly utilizes CNN for automatic extraction of input parameter features, enabling autonomous feature acquisition. Furthermore, the bidirectional mechanism of neural networks is used to enhance the predictive capability of the LSTM network. More innovatively, the SSA is employed for optimizing network parameters, thus overcoming the limitations of traditional intelligent optimization algorithms and achieving an optimal selection method for neural parameters that reduces dependence on manual expertise. Ultimately, an SSA-CNN-BiLSTM network is constructed to achieve higher prediction accuracy. To evaluate the prediction results, four composite assessment metrics are introduced along with an error-based multi-criteria evaluation methodology. Two groups of experiments with distinct capacity degradation patterns under 24 °C and 4 °C are conducted to further validate the performance of SSA-CNN-BiLSTM for capacity prediction of LIBs. And the main contributions and innovations in this paper can be attributed to the following aspects.

1. The LSTM network's performance is augmented through a bidirectional network mechanism. Combining the CNN approach with the BiLSTM algorithm, a CNN-BiLSTM method for estimating lithium battery capacity has been proposed and experimentally validated, proving its feasibility and effectiveness.
2. The SSA optimization algorithm optimizes the CNN-BiLSTM network without human intervention, achieving automated selection of optimal network model parameters.
3. Four composite evaluation metrics are introduced along with an error-based multi-criteria assessment methodology. Comprehensive algorithm performance benchmarking is conducted on experimental data under different temperature conditions.

The rest of this article is organized as follows. Section 2 provides a detailed discussion of the relevant basic background theories. Section 3 describes basic issues such as experimental settings, data sources, and evaluation methods. Section 4 gives the experimental results and discusses the proposed method of capacity prediction. Section 5 gives the conclusions.

2. Methodology

2.1. Algorithm

SSA [30], proposed in 2020, is a group intelligence optimization algorithm inspired by the predation characteristics of sparrows. This method is based on sparrows observing their group members while foraging. Sparrows will observe other individuals in the group at the same time when they carry out predation activities. Based on individual predatory behavior, members of the sparrow group can be categorized into discoverers and followers. The followers obtain food resources under the guidance of the discoverers, and some sparrow individuals compete with other sparrows for quality food resources, replacing their position. The sparrow search algorithm, developed from the predation characteristics of sparrows, boasts advantages such as excellent global performance and rapid convergence, making it a novel swarm intelligence algorithm. Its basic principles are as follows:

- (1) The energy level of the individual in the population depends on the fitness of the individual. The discoverer, with higher fitness and more substantial energy, plays a critical role in providing optimization directions for other individuals in the population by finding food sources. This role is significant in population optimization.

- (2) When the sparrow encounters a dangerous situation during the predation process, it will send an alarm to other individuals in the population. If the alarm value exceeds the safety threshold, the discoverer will lead the population to a safe area to escape quickly.
- (3) In the sparrow population, the ratio of followers to discoverers remains constant, but the identity of individual sparrows can change. When a follower has sufficient energy, it will transform into a discoverer. Similarly, if a discoverer no longer ranks high in energy, it will become a follower to access better food resources, following the lead of other discoverers.
- (4) When a follower's fitness level is too low, there is a certain probability that the follower will leave the discoverer and move to other areas to achieve a higher fitness level.
- (5) Followers constantly observe the movements of the discoverer, tracking their footprints and approaching them according to specific rules. They either follow the discoverer's path or search for food in the nearby vicinity.
- (6) Individuals on the periphery of the population are more susceptible to attacks by natural predators. Therefore, these peripheral individuals must continuously change and update their positions to avoid such attacks and to test the fitness of various locations.

The SSA algorithm updates the position information of food finders, followers, and alerters through iteration based on individual fitness, ultimately leading to the identification of the global optimizers' position information. The solution process of the SSA algorithm is shown in Figure 1. Initially, the algorithm initializes the population according to optimization variables, subsequently segregating them into discoverers and followers. It then updates their positions based on individual fitness scores. A subset of individuals is randomly chosen as vigilant, with their positions also being updated. The process iterates until a termination criterion is met. If criteria are not met, iterations proceed; otherwise, the algorithm halts and delivers the optimal solution. The specific description of the individual position update method is illustrated in Equations (1)–(3).

The discoverer's location update status is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T_{max}^2}\right) & \text{if } R_2 < ST \\ X_{i,j}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \quad (1)$$

where $X_{i,j}^{t+1}$ represents the position of the i -th sparrow in the j dimension in the $t + 1$ iteration; R_2 is the vigilance value and both α and R_2 are a random number from 0 to 1; T_{max} is the maximum number of iterations of the population; Q is a random number from a standard normal distribution; L is a unit vector of $1 \times d$; ST is the alert threshold, which represents the safe range. When the vigilance value R_2 is less than the vigilance threshold ST , it means that the feeding environment is safe, the sparrow population is not threatened, and the finder can search extensively. When the vigilance value R_2 is greater than or equal to the vigilance threshold ST , it means that some sparrows in the population have sensed the presence of danger in the surrounding area and warned the remaining sparrows, and then they will immediately move to a safe area to forage.

The follower's location update status is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{i,j}^t}{i^2}\right), & i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L, & i \leq \frac{n}{2} \end{cases} \quad (2)$$

where X_p^{t+1} denotes the optimal position of the discoverer at the $t + 1$ iteration; n represents the number of sparrows in the population; A represents a vector of $1 \times d$ and the random value of the element is 1 or -1 ; and $A^+ = AT(AAT) - 1$. When $i > n/2$, it indicates that the sparrow with lower fitness is the i -th follower of the hungry state and does not need to obtain food. Therefore, it is necessary to move the position to find food.

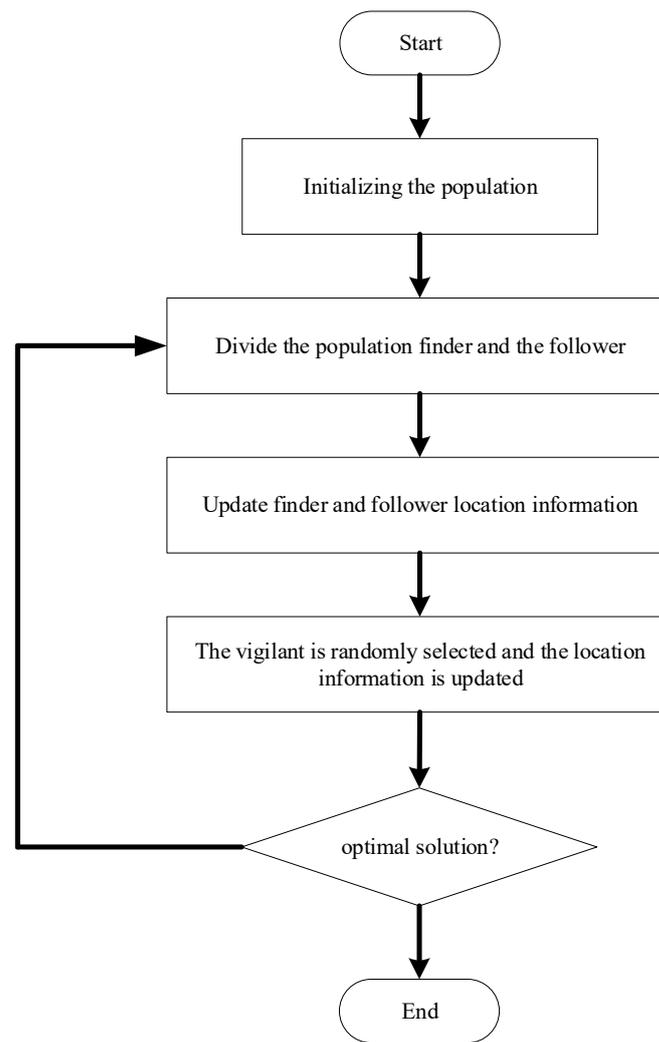


Figure 1. SSA algorithm solving process.

The vigilant's location update status is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t|, & f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g \end{cases} \quad (3)$$

where X_{best}^t represents the global optimal position in the t -th generation; X_{worst}^t represents the global worst position in the t generation, and β is a random number that obeys the standard normal distribution; K is a random number from -1 to 1 ; f_i is the individual fitness value; and f_g and f_w are the current optimal and worst fitness values, respectively. ε is a constant that prevents the denominator from being zero. When $f_i > f_g$, the sparrow is located at the edge of the population and is vulnerable to threats. And when $f_i = f_g$, it means that the sparrow is aware of the danger and approaches the sparrow in a safe position to escape the danger.

2.2. CNN Layer

CNN [31] is a widely used neural network architecture in the field of deep learning, primarily designed for extracting deep-level features without relying heavily on prior knowledge for validation. A typical CNN usually consists of layers such as the input layer, convolutional layer, activation layer, pooling layer, fully connected layer, and output layer.

Specifically, the convolutional layer is one of the core components of CNN. Its primary function is feature extraction. The convolution operation involves sliding a convolutional kernel over the input data, multiplying and summing the features of local regions with the kernel, thereby generating new feature maps. This process effectively captures local patterns and features within the input data. The computational formula can be described as:

$$O^l = f\left(\sum_{i=1}^N X * K_i + b_i^l\right) \quad (4)$$

where the input data are represented as X , K_i is the i -th convolutional kernel, $f(\cdot)$ represents activation function, N is the number of convolutional kernels used in the convolutional layer, b_i^l is the i -th deviation of the feature mapping of layer l , and $*$ is the convolution operation. The activation function is the Relu function.

However, with the addition of more data, the computational burden escalates, necessitating the incorporation of a pooling layer. This layer serves to diminish the computational load and further reduces the parameter count by discarding non-essential samples from the dataset. The model can be efficiently trained using a backpropagation algorithm. Compared to other shallow or deep neural networks, CNNs are distinguished by their reduced parameter requirements, making them a compelling choice in deep learning architectures, as illustrated in Figure 2. The formula for computing CNN feature extraction can be articulated as follows:

$$O^l(j) = \max_{(j-1)w \leq t \leq jw} \{X^{l-1}(t)\} \quad (5)$$

where $O^l(j)$ is the output of the j -th pooled region of the l layer, w denotes the width of the pooled region, and $X^{l-1}(t)$ is the pooled region.

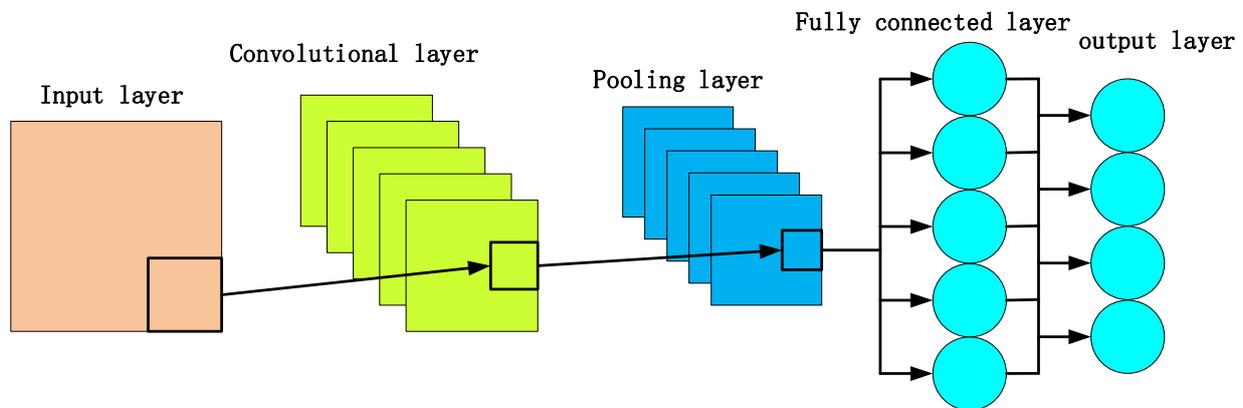


Figure 2. Schematic diagram of CNN structure.

2.3. BiLSTM Layer

The feature map extracted by the CNN from the original data is fed into a subsequent BiLSTM to achieve capacity prediction. It is widely recognized that Recurrent Neural Networks (RNNs) struggle to establish dependable long-term dependencies during training, which limits their performance on long sequential data. These difficulties include problems associated with long-range dependencies, gradient vanishing, and gradient exploding. LSTM aims to overcome these problems by being capable of learning both short-term and long-term dependencies. The structure of the base unit of an LSTM is shown in Figure 3.

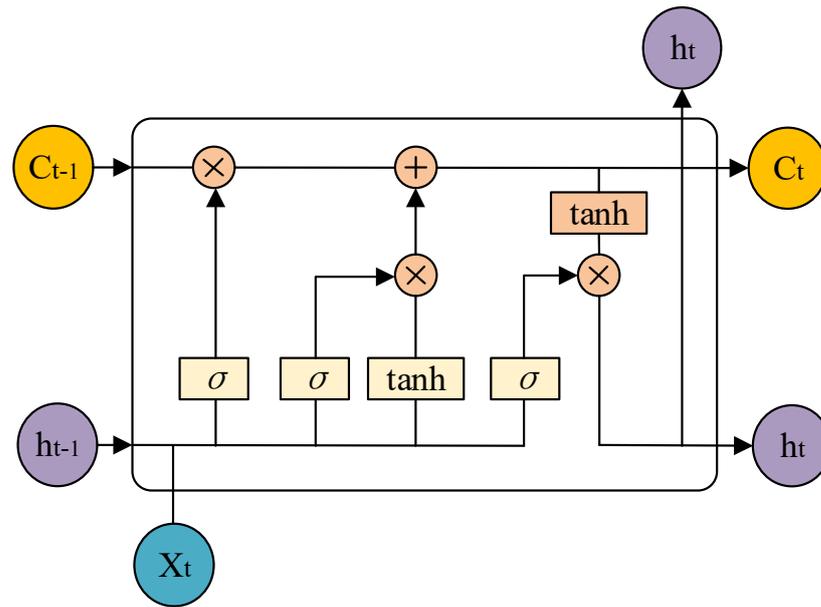


Figure 3. Schematic diagram of LSTM structure.

In the LSTM neural network, information is transmitted through the utilization of forgetting gates, input gates, and output gates. These gates work together synergistically to regulate the level of memory and forgetting of both previous and current information. The specific process of information transmission in the LSTM network can be delineated as follows:

- (1) The forgetting and memory of information. The input information and storage information are multiplied by the weight matrix, respectively, and the bias is added. After the sigmoid function is normalized, the final input information is obtained.

$$f_t = \text{sigmoid}(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \quad (6)$$

where h_{t-1} is the output of the storage unit, $W_{f,x}$ and $W_{f,h}$ are the weight matrix, $W_{f,x}$ is the input of this round, and $W_{f,x}$ is the offset of the forgetting gate.

- (2) New information input. The process of information input requires the input data to be filtered through the weight matrix and then multiplied with the activation matrix to obtain the information input to the memory unit.

$$\tilde{s}_t = \text{tanh}(W_{\tilde{s},x}x_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}}) \quad (7)$$

$$i_t = \text{sigmoid}(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \quad (8)$$

where b_s and b_i represent the bias, $W_{s,x}$, $W_{s,h}$, $W_{i,h}$, and $W_{o,h}$ represent the weight matrix, \tilde{s}_t is the new information candidate value, and i_t is the activation matrix of the input gate.

- (3) Unit status update and information output. The updated unit state is obtained by adding the results of the first and second steps. Subsequently, this unit state is multiplied by the output matrix to obtain the updated information output.

$$s_t = f_t \cdot s_{t-1} + i_t \cdot \tilde{s}_t \quad (9)$$

$$o_t = \text{sigmoid}(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \quad (10)$$

$$h_t = o_t \times \text{tanh}(s_t) \quad (11)$$

where b_o is the bias vector, $W_{o,h}$ and $W_{o,h}$ represent the weight matrix, o_t is the output gate activation matrix, and s_t is the unit state. The information input is achieved by performing the above operation for each moment of information, and the output value of the sequence is obtained after all the moments of information have been processed.

However, in dealing with many complex modeling aspects, BiLSTM is an improvement on LSTM. Since the performance of LSTM depends heavily on the temporal order of the data, interfering with the time step or temporal order will seriously affect the feature extraction and data learning of LSMT. Therefore, BiLSTM is chosen as the prediction model for battery capacity. As shown in Figure 4, the BiLSTM layer integrates forward and reverse LSTM components. By evaluating current time values in each direction, these contribute to the ultimate output. Such an arrangement bolsters network resilience and broadens applicability, proving adept at managing extensive and fluctuating datasets. The sequence of generating network outputs is outlined below:

$$\vec{h}_t = f(x_t, \vec{h}_{t-1}) \tag{12}$$

$$\overleftarrow{h}_t = f(x_t, \overleftarrow{h}_{t-1}) \tag{13}$$

$$h_t = \vec{w}_t \vec{h}_t + \overleftarrow{w}_t \overleftarrow{h}_t + b_t \tag{14}$$

where \vec{w}_t and \overleftarrow{w}_t represent the weights of the forward and reverse LSTM hidden layer outputs, respectively; h_t is a linear combination of the forward and reverse hidden layer states.

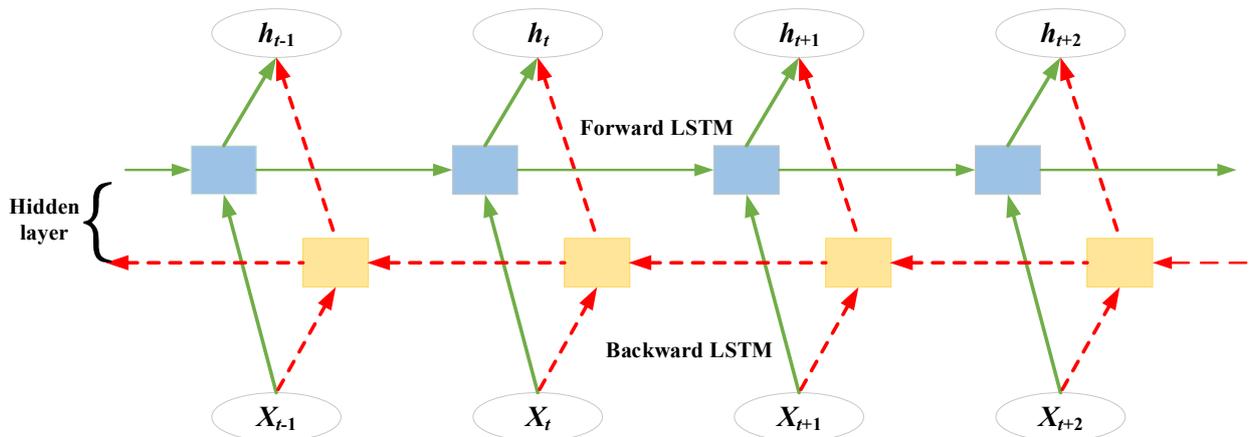


Figure 4. Schematic diagram of BiLSTM structure.

2.4. SSA-CNN-BiLSTM Optimized Framework

The overall framework of the algorithm is illustrated in Figure 5, which uses battery capacity as the input data. The data are preprocessed using sliding window techniques and normalization methods, after which they are divided into training and testing datasets. The CNN layer is used to automatically extract features, which are then processed by the main network, BiLSTM, to compute the prediction results. The SSA algorithm is employed to optimize the hyperparameters of the BiLSTM network. Finally, the optimal hyperparameters are output, resulting in the optimized SSA-CNN-BiLSTM algorithm. This optimized algorithm is then applied to the test dataset for experimentation. The specific process is as follows:

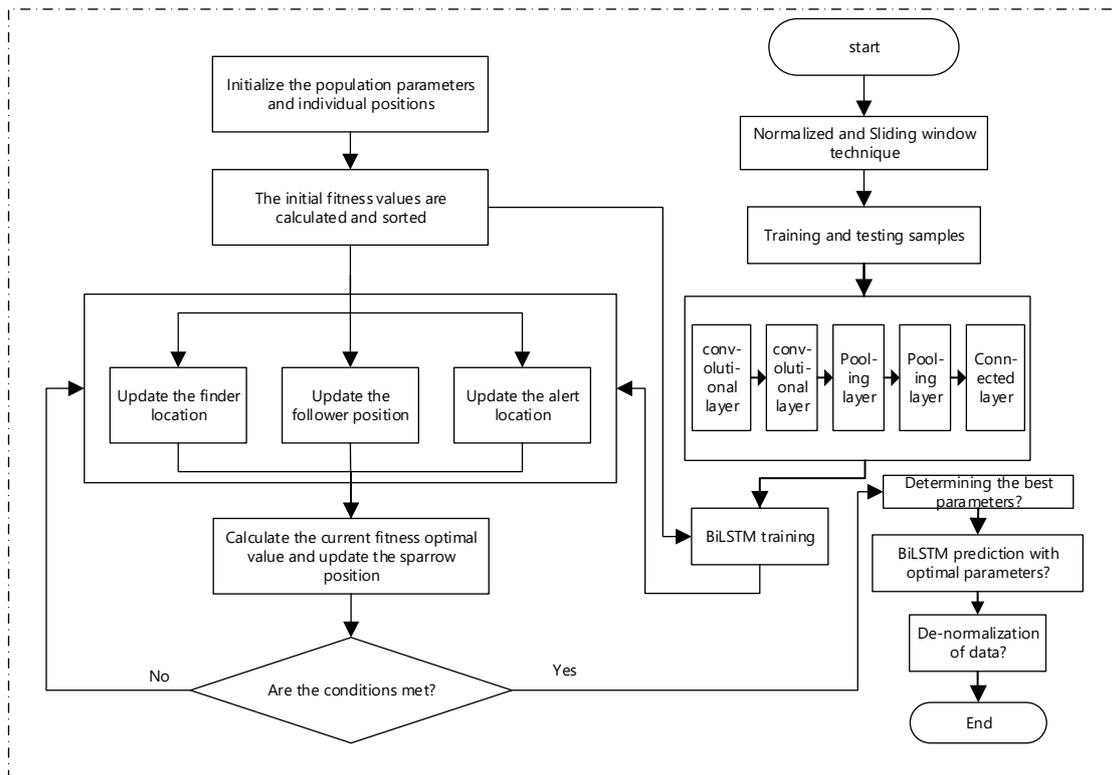


Figure 5. SSA-CNN-BiLSTM optimized framework.

Step 1: Data preprocessing.

Initially, the input data are transformed into multidimensional data using the sliding window method. Subsequently, Formula (15) is applied to standardize the original dataset, converting data values into the [0, 1] range, which facilitates more effective training. Following this, the dataset is partitioned into two subsets: a training set and a testing set, to be utilized for model training and evaluation. This step is instrumental in enhancing the convergence speed and stability of the model.

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{15}$$

where x refers to the input dataset, with x_{min} indicating the minimum value in this collection of data, and x_{max} highlighting the maximum value found within it.

Step 2: Feature extraction layer.

The CNN layer is alternately arranged by convolution operation and pooling layer to extract feature vectors from the input data. These feature vectors will be used as inputs to the subsequent BiLSTM network for training the model. The goal of CNN is to efficiently capture spatial features in the data.

Step 3: Initialize search algorithm parameters.

The initial parameters of the SSA algorithm mainly include the number of iterations, the population size, the percentage of discoverers, the percentage of the number of joiners, the percentage of vigilantes, and the warning value. In this paper, the number of iterations is set to 50, the number of populations to 30, the percentage of discoverers to 30%, the number of joiners to 70%, the percentage of vigilantes to 10%, and the warning value to 0.8. In this step, an initialized BiLSTM network is also created for optimization in subsequent steps.

Step 4: Define the fitness function.

The CNN-BiLSTM method is used to calculate the prediction results, and the Root Mean Square Error (RMSE) was used as the individual fitness function evaluation value.

Step 5: Sparrow Search Algorithm Optimization.

In this step, SSA finds the optimal hyperparameters based on the position-updating mechanisms of different sparrow types. According to the rules of the algorithm, the positions of the finders, joiners, and warners in the group of sparrows are constantly updated to find the combination of parameters that optimize the fitness value. Optimization of the fitness value usually means that the model has the highest prediction accuracy.

Step 6: Determine the iteration termination condition.

Check whether the fitness value reaches the maximum number of iterations. If the maximum number of times is reached, the iterative process is terminated and the current BiLSTM optimal hyperparameters are obtained, including the range of the learning rate, the range of the number of neurons, and the range of the regularization coefficients. Otherwise, continue with step 5 to keep optimizing the parameters.

Step 7: Model the optimal network model.

Using the obtained optimal hyper-parameters, the optimal CNN-LSTM network model is established. This optimal model will be used for the battery RUL prediction task with optimized parameters to provide higher prediction performance.

3. Experimental Design

3.1. Dataset Description

The experimental data used in this study come from the National Aeronautics and Space Administration Prognostics Center of Excellence (NASA PCoE) [32]. The first group (B0005, B0006, B0007, B0018) and the sixth group (B0053, B0054, B0055, B0056) batteries were selected as the subjects of study. The first group of batteries underwent charge–discharge testing at a room temperature of 24 °C, with their impedance being measured. The charging process was divided into two steps. The first step involved charging with a constant current of 1.5 A until the battery voltage reached 4.2 V. The second step involved charging with a constant voltage, maintaining the battery voltage at 4.2 V, and reducing the current to 20 mA. The discharge process was carried out with a continuous current of 2 A until the voltage dropped to 2.7 V, 2.5 V, 2.2 V, and 2.5 V, respectively. The sixth group was tested under the room temperature of 4 °C, with the same charging and discharging methods. Due to the different environmental conditions, the degradation curve of the capacity showed substantial fluctuations. The attenuation curve of the battery capacity is indicated in Figure 6, where some abnormal data have been removed.

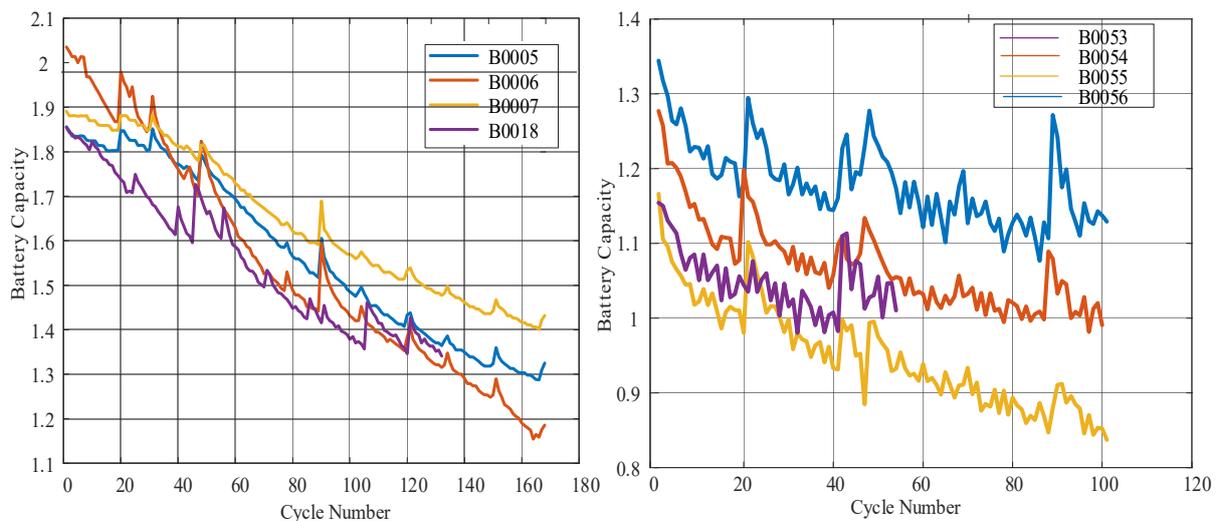


Figure 6. Battery capacity degradation curves.

3.2. Optimization Parameter

Some hyper-parameters of the neural network are selected as the optimization parameters of the SSA algorithm, as shown in Table 1.

Table 1. Optimization parameters.

Optimized Parameters	Parameter Range
number of neurons	10~200
learning rate	0.001~0.01
L2 regularization coefficient	1×10^{-10} ~ 1×10^{-2}

3.3. Evaluation Indicators

To avoid the singularity of evaluation metrics, we concurrently utilize four metrics: MAPE, MAE, RMSE, and MSE, as expressed in Formulas (16)–(19).

$$\text{MAPE} = \frac{1}{n} \sum_{k=1}^n \left| \frac{\widehat{Q}_k^* - Q_k}{Q_k} \right| \times 100\% \quad (16)$$

$$\text{MAE} = \frac{1}{n} \sum_{k=1}^n \left| \frac{\widehat{Q}_k^* - Q_k}{Q_k} \right| \quad (17)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{k=1}^n (Q_k - \widehat{Q}_k^*)^2} \quad (18)$$

$$\text{MSE} = \frac{1}{n} \sum_{k=1}^n (Q_k - \widehat{Q}_k^*)^2 \quad (19)$$

In the above formulas, Q_k , \widehat{Q}_k^* , \overline{Q}_K represent the actual, predicted, and average values of the actual battery capacity, respectively. n represents the total number of predictions. If the RMSE, MSE, MAPE, and MAE are closer to 0, the prediction results of the proposed method are more accurate and the prediction performance is more stable.

4. Experiment and Result Analysis

The experiments were conducted on a Hasee laptop (Shenzhen, China) featuring an Intel i7-10750H quad-core processor, NVIDIA GeForce RTX 3060 Laptop GPU, and 16 GB RAM. All experiments were carried out in a MATLAB 2022b environment, using a single GPU setup, with the training period set to 500 rounds. Experimental programs for LSTM, BiLSTM, CNN-BiLSTM, and SSA-CNN-BiLSTM were constructed, respectively.

The experiments were built in MATLAB with LSTM, BiLSTM, CNN-BiLSTM, and SSA-CNN-BiLSTM experimental programs, respectively. Experimental data were obtained using the first group of NASA data, with capacity as the experimental variable. The initial 70% of the data were utilized as training data, while the remaining 30% were allocated for testing purposes. The sliding window method was used to process the data. The sliding window was set to 9, and the prediction step was 1. The evaluation metrics for the experimental results are presented in Table 2. The iteration plot for the SSA algorithm is illustrated in Figure 7. For a visual comparison of performance metrics, the graphical analysis of performance indicators is depicted in Figures 8 and 9. Finally, the prediction curves of the four optimization algorithms are graphically analyzed for their tracking behavior, as shown in Figures 10 and 11.

Table 2. Comparison of error indicators for B0005, B0006, B0007, B0018 (24 °C).

BatteryID	Method	RMSE	MSE	MAE	MAPE	Training Time (s)	Testing Time (s)
B0005	LSTM	0.07160	0.00513	0.06661	0.05034	21.41	0.26
B0005	BiLSTM	0.05488	0.00301	0.05063	0.03827	26.28	0.24
B0005	CNN-BiLSTM	0.02586	0.00067	0.02239	0.01650	26.76	0.28
B0005	SSA-CNN-BiLSTM	0.01568	0.00025	0.01346	0.01004	~	~
B0006	LSTM	0.09817	0.00964	0.08573	0.07011	24.53	0.30
B0006	BiLSTM	0.08019	0.00643	0.06966	0.05700	28.19	0.30
B0006	CNN-BiLSTM	0.05537	0.00307	0.04912	0.03802	30.18	0.32
B0006	SSA-CNN-BiLSTM	0.02832	0.00080	0.02350	0.01834	~	~
B0007	LSTM	0.06237	0.00389	0.05716	0.03980	20.33	0.25
B0007	BiLSTM	0.05310	0.00282	0.04528	0.03161	25.27	0.24
B0007	CNN-BiLSTM	0.04107	0.00169	0.03612	0.02437	27.56	0.30
B0007	SSA-CNN-BiLSTM	0.02613	0.00068	0.02275	0.01607	~	~
B0018	LSTM	0.09087	0.00826	0.08217	0.05982	20.52	0.25
B0018	BiLSTM	0.05586	0.00312	0.04907	0.03578	24.87	0.26
B0018	CNN-BiLSTM	0.04067	0.00165	0.03072	0.02174	25.40	0.27
B0018	SSA-CNN-BiLSTM	0.03275	0.00107	0.02873	0.02040	~	~

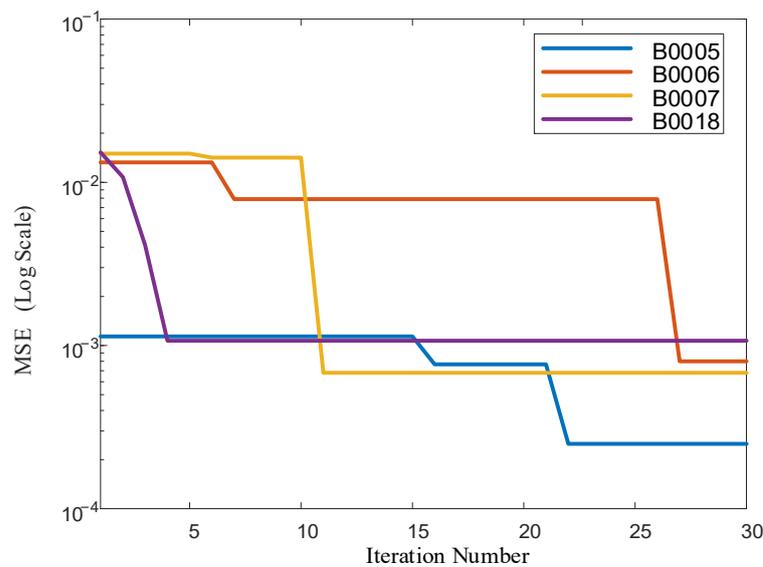


Figure 7. The SSA fitness function changes of B0005, B0006, B0007, B0018.

Based on Table 1, it is evident that LSTM, serving as the baseline model, demonstrates the lowest predictive accuracy. In Figure 8, the optimization percentage is based on LSTM as the comparative baseline. BiLSTM slightly outperforms the standard LSTM across the four performance metrics, with only a few percentage points of improvement. This can be attributed to the bidirectional structure, which allows the model to simultaneously consider past and future information, thereby enhancing predictive accuracy. By combining the models of CNN, CNN-BiLSTM significantly outperforms both standalone LSTM and BiLSTM across all indicators, with performance improvements ranging up to 74.5%. The incorporation of CNN layers enables the extraction of local features from time series data. When combined with BiLSTM, it further captures time-dependent features within the sequence. Among the four, the SSA-CNN-BiLSTM algorithm exhibits the best performance metrics. All error metrics show a significant improvement over the other three algorithms, with MSE optimized by 95.7% compared to LSTM, and the other three metrics surpassing 77%. This suggests that by combining the SSA with CNN and BiLSTM, the predictive capabilities of the model are maximally enhanced.

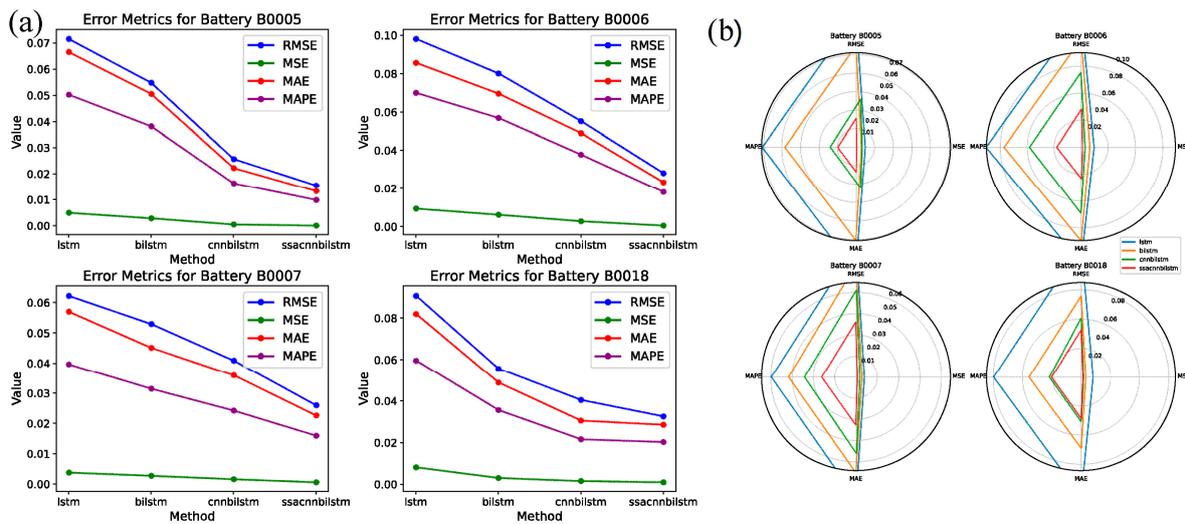


Figure 8. Comparison of error metrics for B0005, B0006, B0007, B0018 (24 °C): (a) line diagram for error composite index; (b) radar diagram for error composite index.

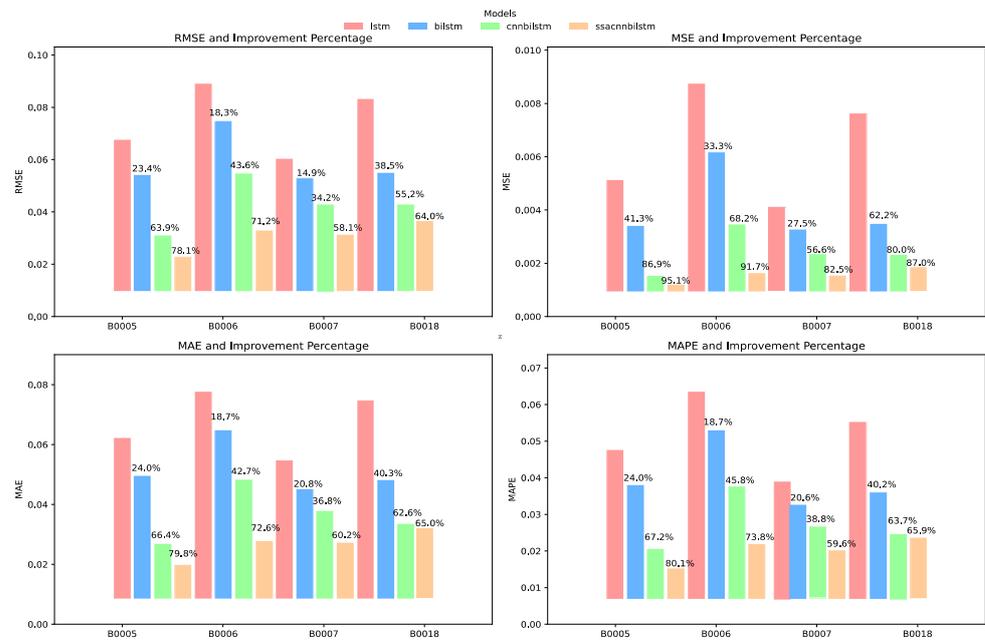


Figure 9. Compared to LSTM model optimization percentage of B0005, B0006, B0007, B0018 (24 °C).

To analyze the computational burden of various algorithms, training and testing times were employed as evaluation metrics, as shown in Table 2, where the results are the average of five runs. Given that the SSA algorithm is a swarm optimization algorithm, its computational burden is N (population size) \times M (number of iterations) times greater than that of CNN-BiLSTM, representing a considerable time expense. Consequently, the time cost for the SSA-CNN-BiLSTM algorithm was not examined, with its base time reference being that of CNN-BiLSTM. In terms of training and testing times, the LSTM model carries the least computational burden, while the CNN-BiLSTM model bears the greatest, and the BiLSTM model falls in between. Nevertheless, the variance among them is minimal. Due to the inherently short testing times, the differences between the algorithms are even smaller, essentially negligible, thus obviating the need for further analysis.

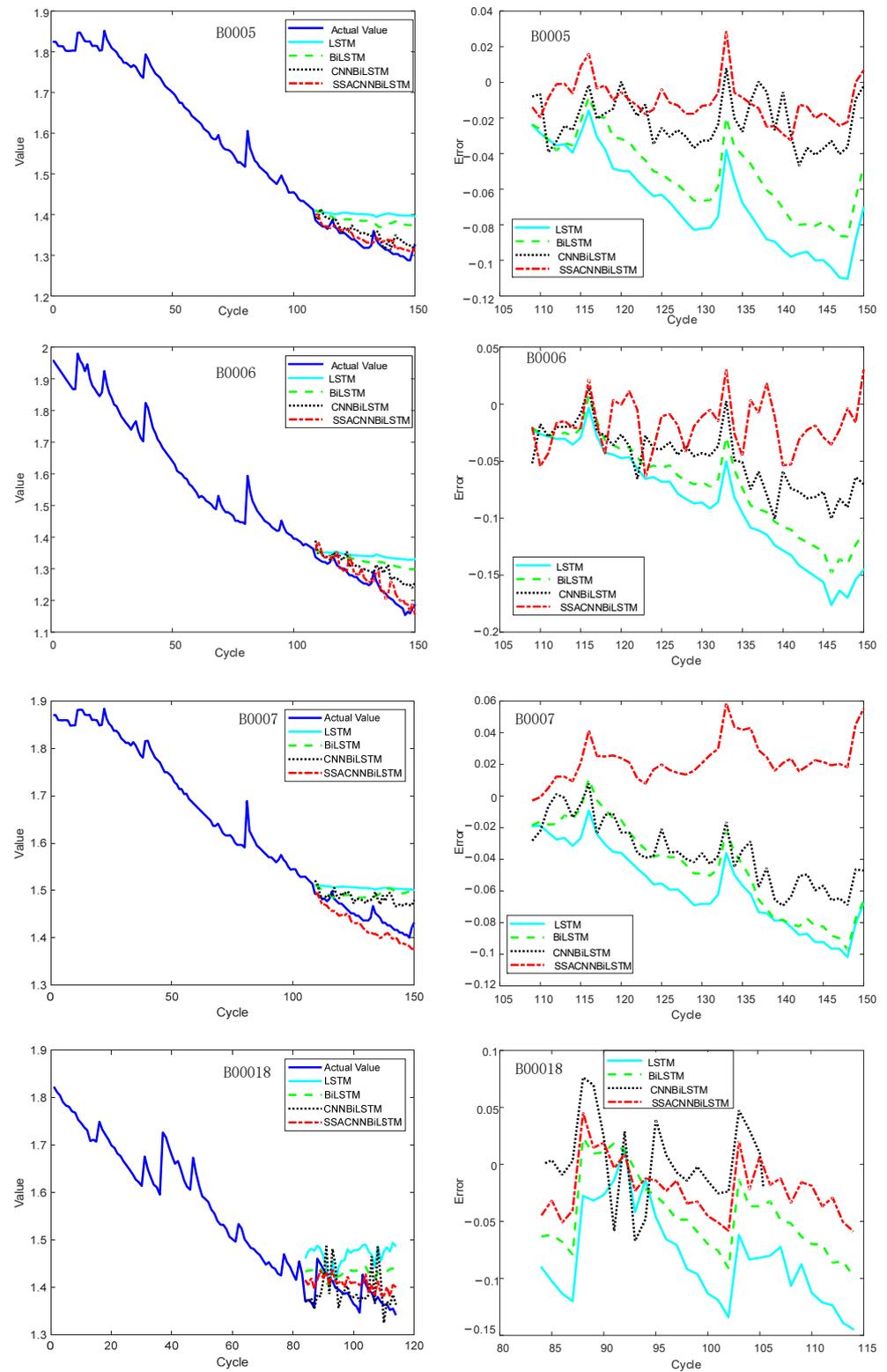


Figure 10. Estimation results of each model on B0005, B0006, B0007, B0018 (24 °C).

From Figure 10, it is visually apparent that the LSTM and BiLSTM algorithms exhibit significant prediction errors. Although the predicted trends are largely consistent, these algorithms struggle to effectively track capacity regeneration, particularly in the elliptical region of the graph. In contrast, both the CNN-BiLSTM and SSA-CNN-BiLSTM not only predict the trends correctly with smaller errors but also anticipate the capacity regeneration

phase, demonstrating superior predictive performance. The suboptimal performance of the LSTM and BiLSTM algorithms may stem from their reliance solely on capacity as input data without incorporating other indicators related to capacity estimation. The lack of diverse input information poses significant challenges for the algorithms, especially in the presence of nonlinear and unconventional pattern changes. The BiLSTM algorithm enhances the utilization of past and future information by running LSTM in two directions within the time series, resulting in some improvement in predictive performance. The CNN-BiLSTM and SSA-CNN-BiLSTM algorithms experience substantial performance improvements, attributed to the incorporation of the CNN network. The CNN automatically extracts crucial information from simple capacity indicators, enhancing algorithmic performance. CNN's automatic feature extraction is particularly beneficial in scenarios with fewer health indicators. The SSA-CNN-BiLSTM algorithm, compared to CNN-BiLSTM, achieves further performance enhancement due to the optimization of neural network hyperparameters. Reasonable hyperparameters play a significant role in improving neural network performance, and the use of the SSA optimization algorithm automates the search for optimal parameters. However, it is noteworthy that SSA is a swarm optimization algorithm that requires considerable computational resources, making it challenging for real-time applications with stringent online prediction requirements.

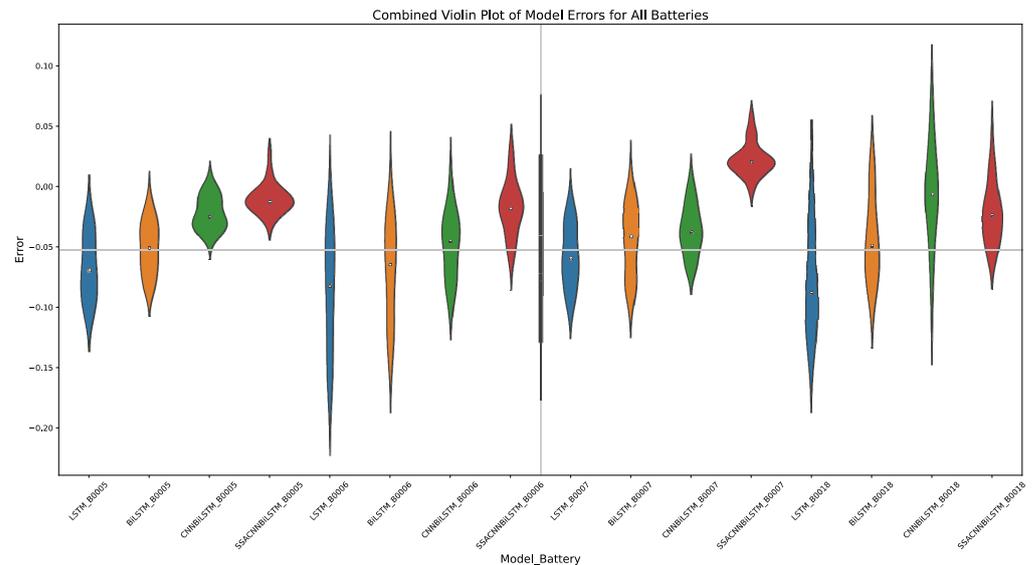


Figure 11. Raincloud diagram of absolute error for B0005, B0006, B0007, B0018 (24 °C).

Due to the more regular degradation pattern in the first group, resulting in smaller prediction errors and better curve tracking, further experiments were conducted using the sixth group, where capacity degradation exhibits greater variability. The prediction error metrics are presented in Table 3, and the visualization of error metrics is shown in Figures 12 and 13. Additionally, Figures 14 and 15 illustrate the tracking performance of the predictions and error analysis. The error metrics reveal that, compared to the conventional LSTM method, incorporating CNN and SSA optimization has reduced the error metrics to some extent, as evident from the error line chart and radar plot. However, the optimization effect is not as pronounced as in the first group, as indicated by Figure 13. Figures 14 and 15 indicate that, although the predictive results are obtained, the curve-tracking performance is inferior compared to the first group. The optimized methods can only minimize errors to some extent but do not significantly improve the fitting performance. This is attributed to the lower operating temperature of the batteries in the sixth group, resulting in larger variations in capacity degradation trends. It becomes challenging to track these changes effectively.

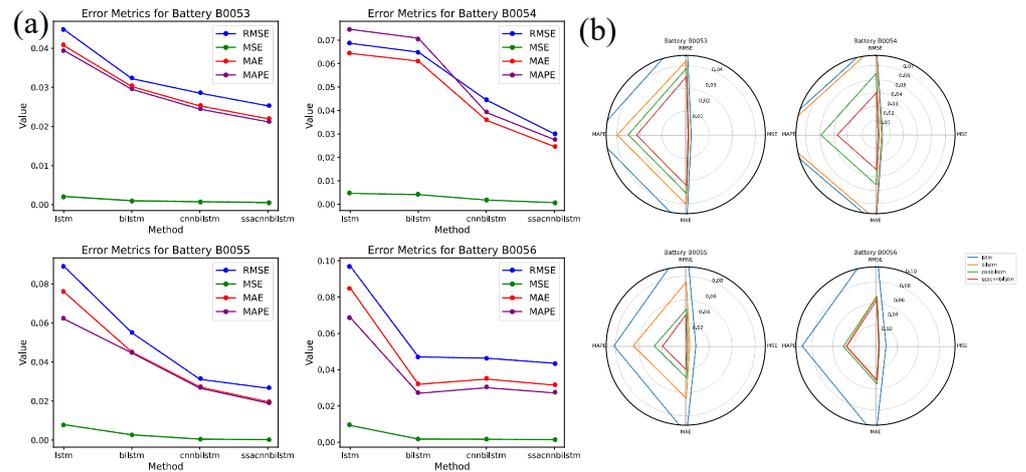


Figure 12. Comparison of error metrics for B0053, B0054, B0055, B0056 (24 °C): (a) line diagram for error composite index; (b) radar diagram for error composite index.

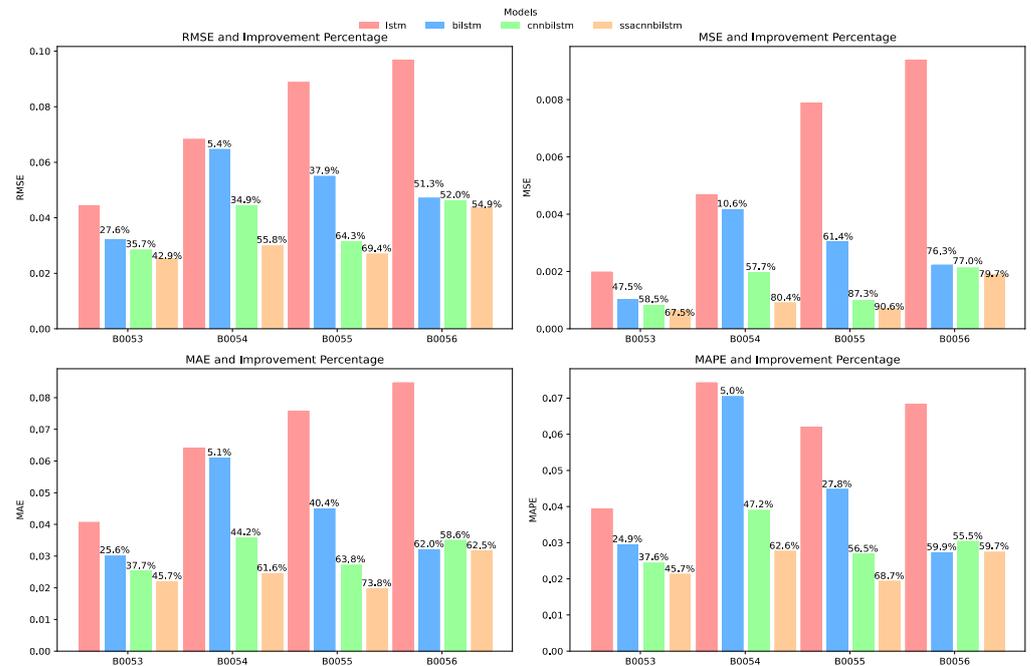


Figure 13. Compared to LSTM model optimization percentage of B0053, B0054, B0055, B0056 (4 °C).

Table 3. Comparison of error indicators for r B0053, B0054, B0055, B0056 (4 °C).

BatteryID	Method	RMSE	MSE	MAE	MAPE
B0053	LSTM	0.04473	0.00200	0.04080	0.03947
B0053	BiLSTM	0.03237	0.00105	0.03035	0.02964
B0053	CNN-BiLSTM	0.02874	0.00083	0.02543	0.02463
B0053	SSA-CNN-BiLSTM	0.02555	0.00065	0.02216	0.02143
B0054	LSTM	0.06857	0.00470	0.06443	0.07439
B0054	BiLSTM	0.06484	0.00420	0.06116	0.07066
B0054	CNN-BiLSTM	0.04462	0.00199	0.03596	0.03925
B0054	SSA-CNN-BiLSTM	0.03031	0.00092	0.02475	0.02779
B0055	LSTM	0.08889	0.00790	0.07586	0.06212
B0055	BiLSTM	0.05522	0.00305	0.04519	0.04485
B0055	CNN-BiLSTM	0.03169	0.00100	0.02743	0.02701

Table 3. Cont.

BatteryID	Method	RMSE	MSE	MAE	MAPE
B0055	SSA-CNN-BiLSTM	0.02719	0.00074	0.01990	0.01946
B0056	LSTM	0.09688	0.00939	0.08486	0.06858
B0056	BiLSTM	0.04722	0.00223	0.03228	0.02747
B0056	CNN-BiLSTM	0.04653	0.00216	0.03513	0.03049
B0056	SSA-CNN-BiLSTM	0.04373	0.00191	0.03183	0.02766

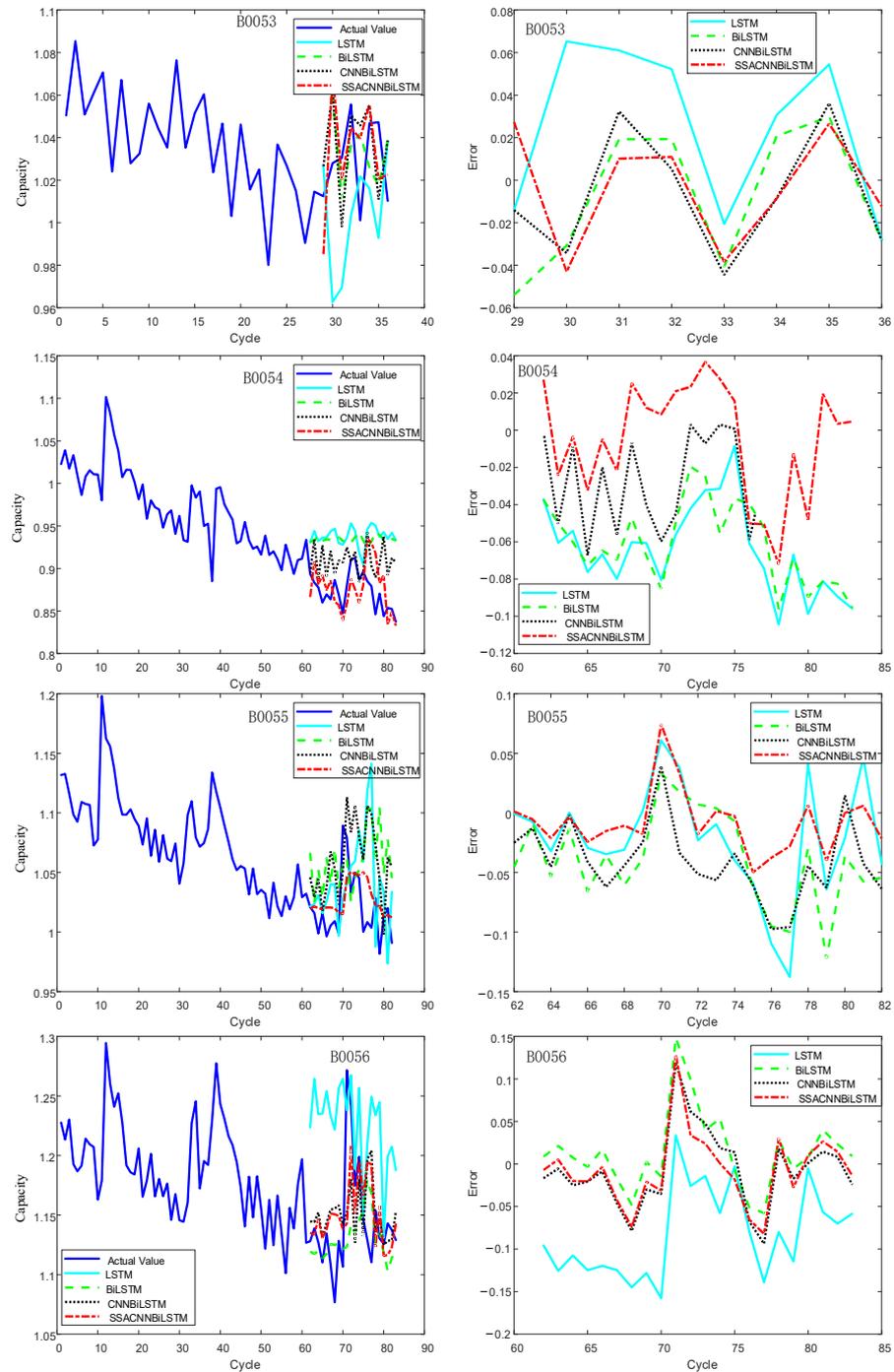


Figure 14. Estimation results of each model on B0053, B0054, B0055, B0056 (4 °C).

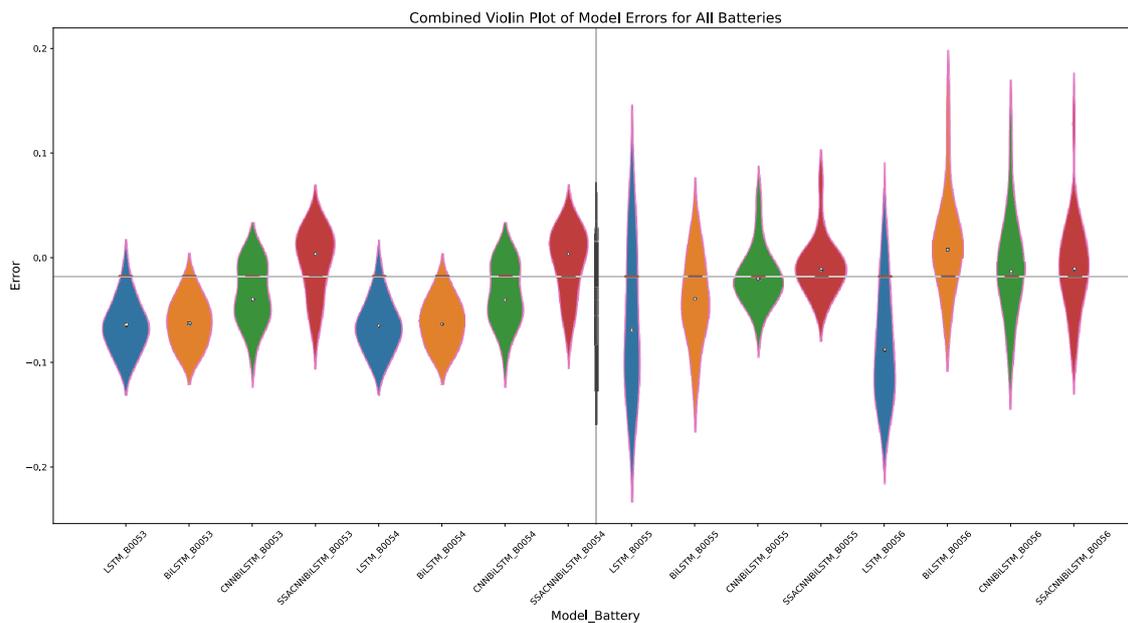


Figure 15. Raincloud diagram of absolute error for B0053, B0054, B0055, B0056 (4 °C).

5. Conclusions and Future Work

In this study, we propose an innovative SSA-CNN-BiLSTM framework aimed at accurately estimating the capacity of LIBs and effectively addressing the challenges in current battery health management systems. Firstly, the CNN applied in this framework can automatically select features, eliminating the tediousness and potential oversight of important features in manual selection processes. Moreover, our introduced SSA optimization algorithm outperforms traditional methods by overcoming the shortcomings of manual network parameter setting.

Meanwhile, the comprehensive validation of this framework using two sets of experimental data provided by NASA is also conducted under different temperature conditions. In-depth evaluation of the prediction results show that the SSA-CNN-BiLSTM framework for capacity prediction of LIBs has higher accuracy compared with traditional LSTM, BiLSTM, and CNN-BiLSTM methods during the multi-battery cycle experiments. Future work will focus on improving and expanding this research. A potential avenue for improvement is the application of additional deep neural network models to enhance the prediction accuracy of the algorithms. Thus, conducting more practical experiments is necessary to validate the algorithm.

Author Contributions: Conceptualization, methodology, and validation, writing—original draft preparation, C.L.; resources, supervision, writing—review and editing, X.T.; visualization, writing—review and editing, L.W. Software, Validation, and Data curation, G.Z.; Investigation, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Open Foundation of Hubei Key Laboratory for High-efficiency Utilization of Solar Energy and Operation Control of Energy Storage System (NO. HBSEES202308).

Data Availability Statement: The data that support the findings of this study are openly available at <https://www.nasa.gov/intelligent-systems-division/#battery> (accessed on 1 March 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Min, X.; Xu, G.; Xie, B.; Guan, P.; Sun, M.; Cui, G. Challenges of prelithiation strategies for next generation high energy lithium-ion batteries. *Energy Storage Mater.* **2022**, *47*, 297–318. [[CrossRef](#)]
2. Liu, H.; Wang, J.; Liu, X. A 4D Framework for Probing Structure-Property Relationships in Lithium Ion Batteries. *Microsc. Microanal.* **2017**, *23*, 216–217.

3. Wu, L.; Liu, K.; Liu, J.; Pang, H. Evaluating the heat generation characteristics of cylindrical lithium-ion battery considering the discharge rates and N/P ratio. *J. Energy Storage* **2023**, *64*, 107182. [[CrossRef](#)]
4. Zhou, H.; An, K.; Hu, J. Spatiotemporal-scale neutron studies on lithium-ion batteries and beyond. *Appl. Phys. Rev.* **2021**, *8*, 041318. [[CrossRef](#)]
5. Zhang, J.; Wang, Y.; Xu, K. Challenges and strategies of formulating low-temperature electrolytes in lithium-ion batteries. *Int. J. Energy Res.* **2021**, *45*, 1–25.
6. Li, Y.; Wang, Z.; Li, H. Ultrastable Orthorhombic Na₂TiSiO₅ Anode for Lithium-Ion Battery. *Adv. Energy Mater.* **2022**, *12*, 2102709.
7. Wu, L.; Lyu, Z.; Huang, Z.; Zhang, C.; Wei, C. Physics-based battery SOC estimation methods: Recent advances and future perspectives. *J. Energy Chem.* **2023**, *89*, 27–40. [[CrossRef](#)]
8. Pang, H.; Wu, L.; Liu, J.; Liu, X.; Liu, K. Physics-informed neural network approach for heat generation rate estimation of lithium-ion battery under various driving conditions. *J. Energy Chem.* **2023**, *78*, 1–12. [[CrossRef](#)]
9. Yuefeng, L.; Yingjie, H.; Hao, B.; Wei, G.; Xiaoyan, Z. A review of lithium-ion battery state of charge estimation based on deep learning: Directions for improvement and future trends. *J. Energy Storage* **2022**, *52*, 104664.
10. Zheng, L.; Zhang, L.; Zhu, J.; Wang, G.; Jiang, J. Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model. *Appl. Energy* **2016**, *180*, 424–434. [[CrossRef](#)]
11. Li, J.; Adewuyi, K.; Lotfi, N.; Landers, R.G.; Park, J. A single particle model with chemical/mechanical degradation physics for lithium-ion battery State of Health (SOH) estimation. *Appl. Energy* **2018**, *212*, 1178–1190. [[CrossRef](#)]
12. Bezha, M.; Bezha, K.; Nagaoka, N. SOH Estimation of Battery Pack Composed on Reused Li-Ion Cells based on Adaptive ANN Machine Learning Algorithm. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics-Taiwan, Taiwan, China, 28–30 September 2020; pp. 1–2.
13. Dao, V.Q.; Dinh, M.C.; Kim, C.S.; Park, M.; Doh, C.H.; Bae, J.H.; Lee, M.-K.; Liu, J.; Bai, Z. Design of an effective state of charge estimation method for a lithium-ion battery pack using extended Kalman filter and artificial neural network. *Energies* **2021**, *14*, 2634. [[CrossRef](#)]
14. Cao, X.; Zhou, Z.; Duan, B.; Gu, P.; Shang, Y.; Zhang, C. A fast capacity estimation approach for retired lithium-ion batteries. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 5836–5840.
15. Zhou, D.; Yin, H.; Xie, W.; Fu, P.; Lu, W. Research on online capacity estimation of power battery based on EKF-GPR model. *J. Chem.* **2019**, *2019*, 5327319. [[CrossRef](#)]
16. Oji, T.; Zhou, Y.; Ci, S.; Kang, F.; Chen, X.; Liu, X. Data-driven methods for battery SOH estimation: Survey and a critical analysis. *IEEE Access* **2021**, *9*, 126903–126916. [[CrossRef](#)]
17. Xiong, R.; Tian, J.; Shen, W.; Lu, J.; Sun, F. Semi-supervised estimation of capacity degradation for lithium-ion batteries with electrochemical impedance spectroscopy. *J. Energy Chem.* **2023**, *76*, 404–413. [[CrossRef](#)]
18. Wen, J.; Chen, X.; Li, X.; Li, Y. SOH prediction of lithium battery based on IC curve feature and BP neural network. *Energy* **2022**, *261*, 125234. [[CrossRef](#)]
19. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Fahmy, Y.; Emadi, A. A convolutional neural network approach for estimation of li-ion battery state of health from charge profiles. *Energies* **2022**, *15*, 1185. [[CrossRef](#)]
20. Dong, Q.; Li, X.; Tian, J.; Tian, Y. Capacity Estimation of Lithium-Ion Batteries Based on an Optimal Voltage Section and LSTM Network. In *International Conference on Energy Storage and Intelligent Vehicles, Singapore*; Springer: Singapore, 2022; pp. 1117–1127.
21. Sun, S.; Sun, J.; Wang, Z.; Zhou, Z.; Cai, W. Prediction of battery SOH by CNN-BiLSTM network fused with attention mechanism. *Energies* **2022**, *15*, 4428. [[CrossRef](#)]
22. Xie, Q.; Liu, R.; Huang, J.; Su, J. Residual life prediction of lithium-ion batteries based on data preprocessing and a priori knowledge-assisted CNN-LSTM. *Energy* **2023**, *281*, 128232. [[CrossRef](#)]
23. Xu, H.; Wu, L.; Xiong, S.; Li, W.; Garg, A.; Gao, L. An improved CNN-LSTM model-based state-of-health estimation approach for lithium-ion batteries. *Energy* **2023**, *276*, 127585. [[CrossRef](#)]
24. Guo, Y.; Yang, D.; Zhao, K.; Wang, K. State of health estimation for lithium-ion battery based on bi-directional long short-term memory neural network and attention mechanism. *Energy Rep.* **2022**, *8*, 208–215. [[CrossRef](#)]
25. Zhu, J.; Wang, Y.; Huang, Y.; Gopaluni, R.B.; Cao, Y.; Heere, M.; Mühlbauer, M.J.; Mereacre, L.; Dai, H.; Liu, X.; et al. Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation. *Nat. Commun.* **2022**, *13*, 2261. [[CrossRef](#)] [[PubMed](#)]
26. Tian, Y.; Wen, J.; Yang, Y.; Shi, Y.; Zeng, J. State-of-Health Prediction of Lithium-Ion Batteries Based on CNN-BiLSTM-AM. *Batteries* **2022**, *8*, 155. [[CrossRef](#)]
27. Kim, T.-Y.; Cho, S.-B. Optimizing CNN-LSTM neural networks with PSO for anomalous query access control. *Neurocomputing* **2021**, *456*, 666–677. [[CrossRef](#)]
28. Ren, X.; Liu, S.; Xia, Y.; Xia, D. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. *Energy* **2021**, *234*, 121236. [[CrossRef](#)]
29. Bao, Z.; Jiang, J.; Zhu, C.B.; Gao, M. A New Hybrid Neural Network Method for State-of-Health Estimation of Lithium-Ion Battery. *Energies* **2022**, *15*, 4399. [[CrossRef](#)]
30. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [[CrossRef](#)]

31. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [[CrossRef](#)]
32. Saha, B.; Goebel, K. *Battery Data Set. NASA A3mes Prognostics Data Repository*; NASA Ames Research Center: Moffett Field, CA, USA, 2008.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.