batteries

MDPI

Article

A Study on the Differences in Optimized Inputs of Various Data-Driven Methods for Battery Capacity Prediction

Kuo Xin [1], Fu Jia [1], Byoungik Choi [2] and Geesoo Lee [3,*]

[1] Department of Mechanical System Engineering, Tongmyong University, Busan 48520, Republic of Korea; 20237010@g.tu.ac.kr (K.X.); 20238004@g.tu.ac.kr (F.J.)
[2] Department of Aerospace Engineering, Pusan National University, Busan 46241, Republic of Korea; hull@pusan.ac.kr
[3] Department of Automotive Engineering, Tongmyong University, Busan 48520, Republic of Korea
* Correspondence: gslee@tu.ac.kr

Abstract: As lithium-ion batteries become increasingly popular worldwide, accurately determining their capacity is crucial for various devices that rely on them. Numerous data-driven methods have been applied to evaluate battery-related parameters. In the application of these methods, input features play a critical role. Most researchers often use the same input features to compare the performance of various neural network models. However, because most models are regarded as black-box models, different methods may show different dependencies on specific features given the inherent differences in their internal structures. And the corresponding optimal inputs of different neural network models should be different. Therefore, comparing the differences in optimized input features for different neural networks is essential. This paper extracts 11 types of lithium battery-related health features, and experiments are conducted on two traditional machine learning networks and three advanced deep learning networks in three aspects of input differences. The experiment aims to systematically evaluate how changes in health feature types, dimensions, and data volume affect the performance of different methods and find the optimal input for each method. The results demonstrate that each network has its own optimal input in the aspects of health feature types, dimensions, and data volume. Moreover, under the premise of obtaining more accurate prediction accuracy, different networks have different requirements for input data. Therefore, in the process of using different types of neural networks for battery capacity prediction, it is very important to determine the type, dimension, and number of input health features according to the structure, category, and actual application requirements of the network. Different inputs will lead to larger differences in results. The optimization degree of mean absolute error (MAE) can be improved by 10–50%, and other indicators can also be optimized to varying degrees. Therefore, it is very important to optimize the network in a targeted manner.

Keywords: neural networks; SVM; PSO-BP; CNN; LSTM; GRU; input; self-attention

check for updates

Academic Editor: Federico Baronti

Received: 23 November 2024
Revised: 8 January 2025
Accepted: 10 January 2025
Published: 13 January 2025

Citation: Xin, K.; Jia, F.; Choi, B.; Lee, G. A Study on the Differences in Optimized Inputs of Various Data-Driven Methods for Battery Capacity Prediction. *Batteries* **2025**, *11*, 26. https://doi.org/10.3390/batteries11010026

1. Introduction

As human society pays more attention to climate and the environment, the use of lithium-ion batteries has grown significantly. Lithium-ion batteries themselves have advantages such as high energy density and long life cycles [1], and are widely used in various electric devices, including electric vehicles. With the popularization of lithium-ion battery applications, research on the state of lithium-ion batteries and related parameters has gained attention. With the popularization of lithium-ion battery applications, research on the state of lithium-ion batteries and related parameters has gained attention. Take the health status of lithium-ion batteries as an example, As the mileage continues to increase, the number of charge and discharge cycles also increases, repeating electrochemical reactions inside the battery. These reactions will lead to irreversible loss of materials, resulting in gradual performance degradation, most notably reflected in the reduction of maximum available capacity. The state of health (SOH) of a battery is typically defined as the ratio between the current maximum available capacity to that of a new battery. This ratio reflects the potential output power of the battery. Therefore, an accurate assessment of the battery status is of great significance to ensure the safety and stability of its operation [2]. Research on battery SOH has gone through different stages of development. There are currently three primary ways to estimate battery SOH, namely experimental method, model building, and data-driven method.

The experimental method involves conducting tests on lithium-ion batteries under specific conditions, typically charge-discharge and pulse experiments, and obtaining the battery capacity through experimental data [3]. While this method is applicable across various battery types, it is constrained by the limitations of the experimental environment [4]. For instance, Scipioni et al. [5] obtained the aging process of lithium-ion batteries by analyzing the battery structure including a positive electrode, negative electrode, electrolyte, etc. Liu et al. [6] studied the measurement of battery SOH under conditions that do not affect the integrity of lithium-ion batteries.

Model-based methods for estimating battery SOH typically involve equivalent circuit models [7] and electrochemical models [8]. These models simulate the chemical changes, current, and voltage changes during the operation of lithium batteries. They are used to studying the relationship between key factors such as material concentration and reaction rate and battery SOH. In this process, optimization methods such as Kalman filtering are openly applied to improve the robustness of prediction and reduce data noise. Lüders et al. [9] conducted in-depth research on the process of lithium plating and lithium stripping and built models based on this process. Lai et al. [10] analyzed a variety of equivalent circuit models and performed global optimization based on different parameter identification. Yang et al. [11] used constant voltage charging current measurement to assess battery SOH online. Eddahech et al. [12] used impedance spectroscopy measurement combined with recurrent neural networks to monitor the health status of lithium-ion batteries. Although model-based methods offer explainability of battery behavior, it is difficult to construct itself, the calculation is relatively complex, and it has high requirements for data accuracy and quality. Consequently, their practical implementation can be challenging.

Data-driven methods have received extensive attention and research due to their simplicity and accuracy. Since a series of historical data related to battery recycling includes the entire battery degradation process, a series of health features extracted from them are used as neural network inputs to predict the battery SOH through neural networks. These health features are generally categorized into direct features and introductory features [13].

Traditional machine learning methods for battery SOH include Backpropagation (BP) neural networks [14], Support Vector Machine (SVM), extreme learning machines, random forests and annealing algorithms, etc. Building on these approaches, deep learning methods such as Convolutional Neural Network (CNN) [15], Long Short-Term Memory (LSTM) [16,17], Gated Recurrent Unit (GRU) [18], etc. have been combined to achieve more accurate predictions of battery SOH. Weng et al. [19] used support vector regression for incremental capacity analysis to predict the health status of vehicle batteries. Dong et al. [20] introduced particle filters into support vector regression to predict battery health status. Sbarufatti et al. [21] introduced particle filters into radial basis functions to predict battery SOH. Lipu et al. [22] conducted a comprehensive analysis of deep learning for battery state estimation in a battery management system (BMS). Chen et al. [23] focused on the relevant features of the constant voltage stage and combined features to complete the health status assessment of lithium-ion batteries. Raman et al. [24] studied recurrent neural networks (RNNs) and their variants to predict the battery SOH. Bao et al. [25] employed swarm intelligence optimization techniques to improve the accuracy of prediction.

Different studies vary significantly in the selection of input health features for battery parameter prediction. Li et al. [26] extracted features such as voltage change differences at the same time as input based on the charge and discharge curves. Jia et al. [27] introduced indirect features for Gaussian process regression to improve the accuracy of prediction. Lu et al. [28] extracted health features with slope as the main object as input. In the process of using neural networks, many different health feature extraction methods are employed.

As more and more methods are proposed and improved in the data-driven field, researchers will verify the performance of the methods by comparing the accuracy of the final prediction results. Data-driven methods have become prevalent in various fields due to their impressive predictive capabilities. A comprehensive review of previous studies on battery-related parameters reveals a common trend among data-driven research: despite employing novel algorithms or optimization methods, such studies frequently rely on identical input data when conducting comparative analyses with other methods. For example, Zhang et al. [29] demonstrated their findings on battery State of Health (SOH) prediction, as shown in Table 1. Similarly, Lin et al. [30] reported their results on battery capacity prediction, which are summarized in Table 2.

Table 1. Battery SOH prediction comparison.

Method	MAE	RMSE	R^2
KAN-LSTM	0.2043	0.2896	97.21
LSTM	0.2368	0.3368	96.23
CNN-LSTM	0.5045	0.6249	87.01
BILSTM	0.2810	0.3538	95.84
CNN-BILSTM	0.3480	0.4475	93.34

Table 2. Comparison of battery capacity predictions.

Method	RMSE	MSE	MAE	MAPE
LSTM	0.04473	0.00200	0.04080	0.03947
BiLSTM	0.03237	0.00105	0.03035	0.02964
CNN-BiLSTM	0.02874	0.00083	0.02543	0.02463
SSA-CNN-BiLSTM	0.02555	0.00065	0.02216	0.02143

Despite their success, these models often have low interpretability, leading to their categorization as “black box” approaches. These characteristics highlight the importance of understanding how different network architectures interact with input data. Given the inherent differences in their internal structures, different machine learning models and neural networks exhibit varying dependencies on specific features, This implies that the optimal input configuration for different networks may vary, and the best input setup for one model may not align with the requirements of another. Therefore, comparing the performance of different networks under identical input conditions may lead to biased or incomplete conclusions. However, there has been a lack of comprehensive comparative studies addressing this issue. In previous research, almost all papers on algorithm optimization have employed identical input conditions, including the types of input features, their dimensionality, and data volume, among other factors.

For instance, from the perspective of input dimensionality, certain networks may exhibit overfitting as the dimensionality increases, whereas others demonstrate greater adaptability to higher-dimensional input data, achieving more accurate results as the dimensionality grows.

From the perspective of health indicator categories, most researchers rely on Pearson correlation coefficients to assess the strength of the relationship between health features and the target parameter, implicitly assuming a linear correlation. Some researchers employ Grey Relational Analysis (GRA) for feature evaluation, with the common goal of identifying features most closely related to the predicted parameter. Nonetheless, such criteria may not be universally applicable to all networks, underscoring the need for differentiated studies. This study investigates the dependency of different networks on correlation coefficients, aiming to determine the most suitable correlation criterion for each type of network.

From the perspective of input data volume, some networks are expected to achieve relatively accurate experimental results with a smaller amount of data, provided that a certain application standard is met. Investigating and analyzing the performance differences of various networks under different data input volumes can provide valuable insights into optimizing the time and process of data collection in practical applications.

Battery capacity can be used to calculate the battery health status and can intuitively reflect the relevant physical characteristics of the battery. This study explores the battery capacity prediction problem by comparing the performance differences of various networks when selecting different input features. The focus of the research lies in identifying the optimal input configuration for each type of network, with an emphasis on three key influencing factors: the correlation of input features, input dimensionality, and input data volume.

This study represents an investigation into input variability. So we selected both classical machine learning algorithms and advanced deep learning models. This will give a more representative result. For machine learning, the Particle Swarm Optimization Backpropagation (PSO-BP) algorithm and the Support Vector Machine (SVM) algorithm were selected, representing well-established approaches known for their effectiveness in handling structured and moderately sized datasets. PSO-BP leverages particle swarm optimization to enhance the convergence and accuracy of the backpropagation network, while SVM excels in solving classification and regression problems with clear boundaries.

On the deep learning front, the study utilized several hybrid architectures designed to capture complex relationships in high-dimensional data. These included the CNN-LSTM-Attention network, the CNN-GRU-Attention model, and the CNN-BiLSTM-Attention network. The CNN layers in these models extract spatial features from raw data, while the LSTM and BiLSTM layers specialize in processing temporal sequences, capturing long-term dependencies. The Attention mechanism further refines the models by assigning dynamic weights to input features, enabling the networks to focus on the most relevant aspects of the data. This combination of feature extraction, sequence modeling, and attention-based optimization allows the deep-learning models to achieve enhanced accuracy and robustness in capacity estimation.

The main contributions of this paper are as follows:

(1) The input health features were divided into three different categories. The classification was based on data availability and computational complexity. The Person and Spearman correlations between each health feature and the battery capacity were calculated and used to create various input groups with varying correlation coefficients. Study the dependency differences of different networks on various correlation coefficients.

(2) Study the dependency of different networks on input dimensions and identify the optimal input dimensions for different networks. Investigate the changes in results caused by variations in input dimensions across different networks. This analysis demonstrates that different networks have differences in their optimal input dimensions.

(3) Evaluate the input training data requirements for different networks, focusing on the amount of data needed to achieve a specified level of predictive accuracy. Study the dependency of different networks on input data volume.

2. Database

This data set performs cyclic charging, discharging, and impedance experiments on four groups of batteries at an ambient temperature of 4 degrees Celsius. The database is from the National Aeronautics and Space Administration (NASA) [31]. The charging process uses a constant current-constant voltage (CC-CV) mode, first charging with a constant current of 1.5A, and then changing to constant voltage charging when the battery voltage reaches 4.2 V until the charging current decreases to 20 mA. The discharge process is conducted at a fixed load current level of 1 A and stops when the discharge voltage reaches a fixed value. The electrochemical impedance spectroscopy (EIS) experiment involves a frequency scan from 0.1 Hz to 5 kHz. The experimental protocol continues until the capacity of the battery drops to 1.4 Ahr, representing 30% of the initial capacity. The dataset's cycle curve is shown in Figure 1. The maximum capacity of the battery will decrease as the number of cycles increases. In order to increase the accuracy of the conclusions. Two sets of experiments were conducted using different battery data from the NASA dataset. The B5 battery dataset was designated as the experimental group, while the B18 battery dataset served as the control group.

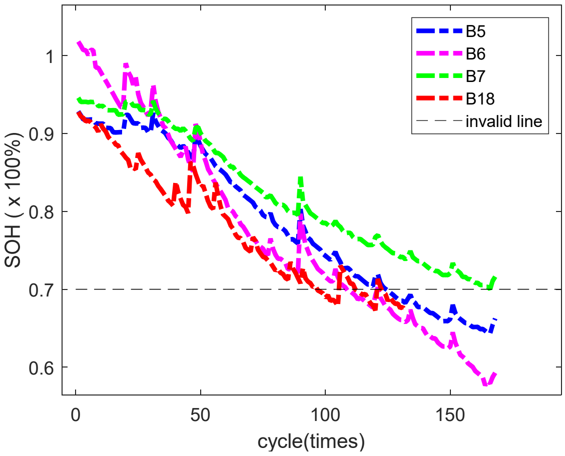


Figure 1. Dataset cycle curve.

To calculate the battery capacity and each health characteristic, it is important to note that the *SOH* of the battery decreases from 100% to between 70% and 80% over its operational lifespan, ultimately reaching what is termed the end-of-life (EOL). Therefore, the remaining capacity of the battery during each cycle can represent the battery SOH.

$$SOH = \frac{Q_{cycle}}{Q_{initial}} \times 100\%, \tag{1}$$

where, Q_{cycle} represents the maximum capacity of the battery in each charge and discharge cycle, and $Q_{initial}$ represents the initial capacity of the battery.

Therefore, the health status of the battery can be obtained from the maximum capacity of the battery during different cycles. Maximum capacity has many uses in different aspects, so the experiment studies the maximum capacity of the battery during the cycle.

3. Methodology

3.1. Feature Extraction

The extraction of the health index (HI) is essential in the data-driven method for battery capacity estimation, which can convert raw data into meaningful features for machine learning. Since the capacity estimation model needs to input multi-dimensional features, the constructed HI needs to accurately describe the battery capacity.

Regarding the classification of health features, researchers have adopted various classification criteria, most of which are based on physical characteristics such as time, current, voltage, and temperature. In this study, the classification criterion is based on the practicality and the complexity of the feature extraction process. Features that can be directly obtained or computed with simple calculations are categorized as the first type. Features requiring more complex calculations, such as integration operations, are classified as the second type. Features that involve deeper extraction of data like the IC curve are grouped into the third type.

The first category of data is the following five types, constant voltage rise time (3.9–4.1 V) (HF1), maximum temperature during discharge (HF2), constant current charging time (HF3), constant voltage charging time (HF4), and constant voltage drop discharge time (HF5) as shown in Figure 2.

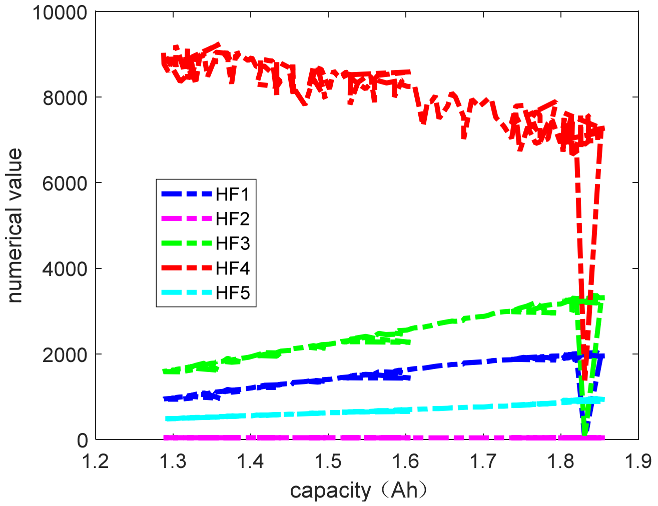


Figure 2. First category features.

The second category of data is the following four types, the proportion of constant current charging time to total charging time (HF6), extracting the area enclosed by the current-time curve of the charging process (HF7), extracting the area enclosed by the constant current charging current-time curve (HF8), extract the area enclosed by the constant voltage charging current-time curve (HF9) as shown in Figure 3.

The third category of data related to the IC curve: Extract the peak value of the IC curve (HF10) Extract the voltage corresponding to the peak value of the IC curve (HF11) as shown in Figure 4. The IC curve requires the calculation of the battery capacity change and the battery terminal voltage change, which are calculated using the following formula:

$$\frac{dQ}{dV} \approx \frac{\Delta Q}{\Delta V} = \frac{Q_k - Q_{k-N}}{V_k - V_{k-N}} = \frac{\int_{t_{k-N}}^{t_k} idt}{V_k - V_{k-N}}, \quad (2)$$

where, Q_k, V_k, t_k are the amount of charge stored in the battery at K sampling points, the battery terminal voltage and time [32]. When the sampling frequency is fixed, the following formula can be obtained:

$$\frac{dQ}{dV} \approx \frac{\int_{t_{k-N}}^{t_k} idt}{V_k - V_{k-N}} \approx \frac{I \cdot (t_k - t_{k-N})}{V_k - V_{k-N}} = i \cdot \Delta t \cdot \frac{N}{V_k - V_{k-N}}, \tag{3}$$

where, i is the current in the current charging period, and Δt is the sampling interval. When the voltage rises faster, a peak will appear in the curve, and when the voltage rises slowly, a trough will appear in the curve. The IC curve can well reflect the characteristics of voltage change.

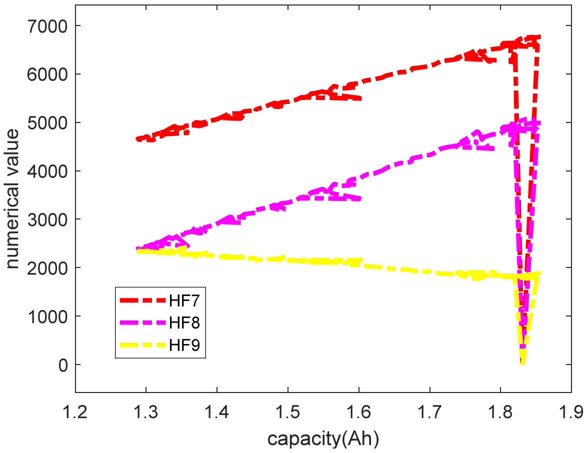


Figure 3. Second category features.

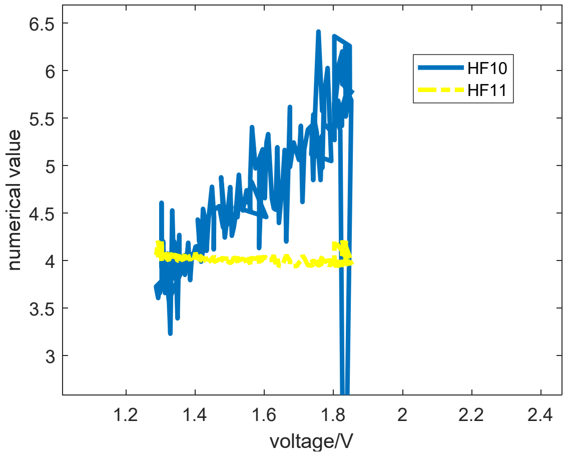


Figure 4. Third category features.

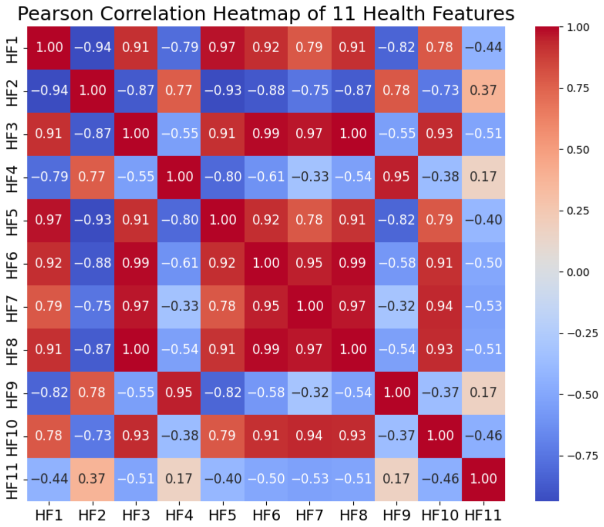
3.2. Feature Preprocessing

In the research of battery SOH or capacity prediction, comprehensive analysis of statistical metrics and data preprocessing are crucial steps. These steps help replace outliers, reduce errors, and make the input data more scientifically robust. Chen and Xia et al. [33] developed an online two-dimensional filtering framework and applied it to both incremental capacity analysis and differential thermal voltammetry. To better utilize the extracted data, this study implements the following data preprocessing procedures:

For the first and second types of features, outlier detection and attribution methods were employed. The mean, standard deviation, minimum, maximum, 25th percentile (Q1), and 75th percentile (Q3) of each feature category were calculated. The interquartile range (IQR) was defined as Q3 − Q1, and outliers were identified as data points falling outside the range [Q1 − K × IQR, Q3 + K × IQR], The initial value of K is selected as 1.5 and adjusted according to the data range. Upon identifying outliers, values exceeding the boundaries were replaced with the nearest boundary value. This method preserves the overall structure of the data and minimizes the introduction of new biases caused by measurement errors during data collection.

For the third type of feature, Gaussian Smoothing (GS), was applied. This technique reduces high-frequency noise, resulting in smoother signals or data. While reducing noise, the method aims to retain the overall trends and characteristics of the data as much as possible.

In this experiment, the Pearson and Spearman correlation coefficients of each characteristic parameter and the maximum capacity of the battery per cycle were calculated. The Pearson Correlation Heatmap of all features is shown in Figure 5.



	HF1	HF2	HF3	HF4	HF5	HF6	HF7	HF8	HF9	HF10	HF11
HF1	1.00	−0.94	0.91	−0.79	0.97	0.92	0.79	0.91	−0.82	0.78	−0.44
HF2	−0.94	1.00	−0.87	0.77	−0.93	−0.88	−0.75	−0.87	0.78	−0.73	0.37
HF3	0.91	−0.87	1.00	−0.55	0.91	0.99	0.97	1.00	−0.55	0.93	−0.51
HF4	−0.79	0.77	−0.55	1.00	−0.80	−0.61	−0.33	−0.54	0.95	−0.38	0.17
HF5	0.97	−0.93	0.91	−0.80	1.00	0.92	0.78	0.91	−0.82	0.79	−0.40
HF6	0.92	−0.88	0.99	−0.61	0.92	1.00	0.95	0.99	−0.58	0.91	−0.50
HF7	0.79	−0.75	0.97	−0.33	0.78	0.95	1.00	0.97	−0.32	0.94	−0.53
HF8	0.91	−0.87	1.00	−0.54	0.91	0.99	0.97	1.00	−0.54	0.93	−0.51
HF9	−0.82	0.78	−0.55	0.95	−0.82	−0.58	−0.32	−0.54	1.00	−0.37	0.17
HF10	0.78	−0.73	0.93	−0.38	0.79	0.91	0.94	0.93	−0.37	1.00	−0.46
HF11	−0.44	0.37	−0.51	0.17	−0.40	−0.50	−0.53	−0.51	0.17	−0.46	1.00

Figure 5. The Pearson correlation heatmap.

3.3. Methods Introduction

Different types of health features, varying dimensions of these health features, and the amount of input data can all impact the performance of different networks. The experiment in this paper selected two classic machine learning methods, PSO-BP and SVM, as well as three deep learning methods, CNN-LSTM-Attention network, CNN-LSTM-Attention, and CNN-BiLSTM-Attention network.

3.3.1. PSO-BP

The BP neural network is among the most used models in neural networks, utilizing the error back propagation algorithm for training. It is based on the principles of gradient descent and gradient search, aiming to reduce the mean square error between the network's actual output and its expected output.

Typically, a BP neural network comprises three layers: an input layer, an output layer, and one or more hidden layers in between. While the hidden layers do not interact directly with external inputs or outputs, their state changes influence the relationship between the input and output. Each layer contains multiple nodes.

The BP algorithm operates through two phases: signal forward propagation and error back propagation. In the forward phase, the network processes the input to generate output, while in the backward phase, errors are propagated from the output layer back to the input layer, guiding the adjustment of weights and thresholds to improve performance. The formula for forward propagation is as follows:

$$a^n = \sigma(z^n) = \sigma\left(W^n a^{n-1} + b^n\right), \tag{4}$$

where, a^n is the output matrix of layer n. W^n and b^n are the weights and biases between different nodes in different layers, and $\sigma(\cdot)$ represents the activation function.

By adjusting the weights and thresholds between the input layer and the hidden layer, as well as between the hidden layer and the output layer, the error is reduced along the gradient direction, and the loss function is expressed as follows:

$$J(W,b,x,y) = \frac{1}{2}\| a^L - y \|_2^2 = \frac{1}{2}\| \sigma\left(z^L\right) - y \|_2^2, \tag{5}$$

where a^L represents the output of the last layer, y represents the sample, and $\| C \|_2$ represents the L2 norm of C.

The calculation formula of the loss function for the weight and error of the Lth layer is as follows:

$$\frac{\partial J}{\partial b^L} = \frac{\partial J}{\partial z^L}\frac{\partial z^L}{\partial b^L} = \frac{\partial J}{\partial z^L} = \left(a^L - y\right) \odot \sigma'\left(z^L\right), \tag{6}$$

$$\frac{\partial J}{\partial W^L} = \frac{\partial J}{\partial z^L}\frac{\partial z^L}{\partial W^L} = \frac{\partial J}{\partial b^L}\left(a^{L-1}\right)^T, \tag{7}$$

The gradients of the loss function at different layers are calculated as follows, where m represents any number of layers:

$$\begin{aligned} \frac{\partial J}{\partial b^m} = \frac{\partial J}{\partial z^m}\frac{\partial z^m}{\partial b^m} = \frac{\partial J}{\partial z^m} = \frac{\partial J}{\partial z^{m+1}}\frac{\partial z^{m+1}}{\partial z^m} \\ = \frac{\partial J}{\partial h^{m+1}} \odot \sigma'(z^m)\left(W^{m+1}\right)^T \end{aligned}, \tag{8}$$

$$\frac{\partial J}{\partial W^m} = \frac{\partial J}{\partial z^m}\frac{\partial z^m}{\partial W^m} = \frac{\partial J}{\partial b^m}\left(a^{m-1}\right)^T, \tag{9}$$

After multiple iterations, the training process is stopped after the weight matrix W and the deviation matrix b corresponding to the minimum error are determined. At this point, the trained neural network can process the input information of similar samples and output the information with the minimum error after nonlinear transformation.

After several iterations, the training process halts once the weight matrix W and the bias matrix b corresponding to the minimum error are identified. At this stage, the trained neural network is capable of processing input data from similar samples and producing outputs with minimal error through nonlinear transformations.

At present, there are many optimization methods for BP networks, and particle swarm optimization (PSO) is a commonly used method as shown in Figure 6. The PSO algorithm simulates biological populations through particles, in which particles have two basic properties: speed and position. Speed represents the speed of movement, and position represents the direction of movement. Each particle moves freely in the solution space to seek the optimal solution. In the process of seeking the optimal solution, each particle has individual behavior and group behavior. Each particle will learn from two values, one is the individual's historical optimal solution *pbest*, and the other is the group's historical optimal solution *gbest*. The particle will continuously adjust its speed and position through this learning process, and the quality of each position is determined by fitness. The fitness function serves as the objective function for the optimization process.

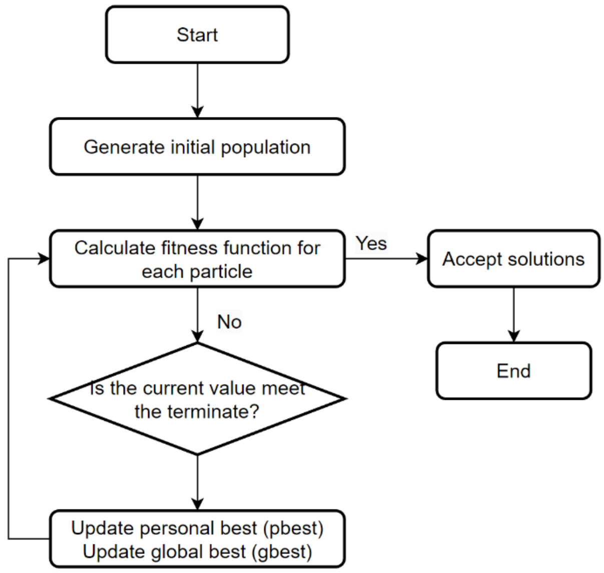

Start
Generate initial population
Calculate fitness function for each particle
Yes
Accept solutions
End
No
Is the current value meet the terminate?
Update personal best (pbest)
Update global best (gbest)

Figure 6. Principle of PSO.

The PSO-BP network in this study has three hidden layers, the maximum number of training epochs is 1500, with a target error of 1 × 10^{-6}, a learning rate of 0.01, a population size of 35, and a maximum number of iterations set to 35. The velocity boundaries are defined as a maximum of 1.0 and a minimum of −1.0, while the search range boundaries are set to a maximum of 1.0 and a minimum of −1.0. The learning factors are c1 is 0.01 and C2 is 0.01.

3.3.2. SVM

An SVM is a machine-learning technique used for both classification and regression tasks. It is capable of handling both linearly separable and non-linearly separable problems by constructing an optimal decision boundary in a high-dimensional space.

The key idea behind SVM is to identify a hyperplane that divides different classes of sample points. This hyperplane, known as the maximum margin hyperplane, maximizes the distance between the closest sample points from each class. By mapping samples into a higher-dimensional space, SVM makes it possible to separate data that may not be linearly separable in the original space. To manage computational complexity in high-dimensional spaces, SVM employs kernel functions, such as linear kernels, polynomial kernels, and Gaussian kernels, which allow complex transformations to be performed in the original input space.

During the optimization, SVM primarily focuses on the critical sample points near the decision boundary, referred to as support vectors, which play a crucial role in determining the model’s final classification.

The SVM network in this study is configured with a penalty factor c is 4.0, a kernel function parameter of 0.8, and an epsilon-loss function precision of 0.01.

3.3.3. 1D CNN

CNN is widely used for data extraction in the modern deep learning process. For the processing of health features in the process of battery capacity estimation, one-dimensional to three-dimensional CNN layers can be selected for convolution operations. After the CNN layer completes the feature extraction, a richer feature input will be obtained. CNN layers are usually composed of input layers, convolution layers, activation layers, pooling layers, fully connected layers, and output layers.

Convolution is the most important operation in a CNN. The process of convolution operation is to slide a convolution kernel of set specifications on the processed input data, multiply the local area features and the kernel over which the convolution kernel slides, and combine all the results obtained in this process. This process effectively expands and captures the features of the input data. The calculation formula can be described as:

$$O^l = f\left(\sum_{t=1}^{N} X \times K_t + b_t^l\right), \tag{10}$$

The input data is denoted as X, where K_t represents the convolution kernel, and $f\ (\cdot)$ symbolizes the activation function. N refers to the total number of convolution kernels utilized in the convolution layer. b_t^l is the bias term corresponding to the feature map in the 1st layer, and the symbol $*$ denotes the convolution operation. The activation function typically employed in this process is the ReLU (Rectified Linear Unit) function.

Because the expansion and extraction of data by the convolution layer greatly increases the amount of data obtained, resulting in a complex calculation process, a pooling layer is usually added to simplify the obtained features and discard some features. There are also many pooling methods.

CNN has a wide range of applications, the formula for calculating CNN feature extraction can be expressed as follows:

$$O^l(j) = \underset{(j-1)w \le t \le fw}{max} \left\{X^{l-1}(t)\right\}, \tag{11}$$

where $O^l(j)$ represents the output of the j-th pooled region in the l-th layer, the variable w refers to the width of the pooling area, and $\left\{X^{l-1}(t)\right\}$ denotes the corresponding region from the previous layer $(l-1)$ that is being pooled.

3.3.4. Channel Self-Attention Module

The introduction of the attention mechanism can weigh features or feature channels, allowing the model to focus on more meaningful or relevant features for the current task or the stronger correlation within the data. This significantly improves the model's emphasis on important information, thereby improving the model's performance and the accuracy of the results. It also reduces the impact of noise and increases the robustness of the model.

$$C_{out} = BN(C_{tn}) = \gamma \frac{C_{tn} - \mu_C}{\sqrt{\sigma_C^2 + \epsilon}} + \beta, \tag{12}$$

where μ_C and σ_C represent the mean and standard deviation, β is the shift parameter, BN is the batch normalization operation, C_{tn} is the input of the attention module, and ϵ is a hyperparameter to prevent the denominator of the formula from being zero [34], respectively. The formulas for the weight and output steps of the self-attention module are as follows:

$$\omega_t = \frac{\gamma_0}{\sum\limits_{j=0} \gamma_f}, \tag{13}$$

$$M_c = tanh(\omega_t(BN(C_{tn}))), \tag{14}$$

where ω_t is the weight, γ is the channel scaling factor, and tanh represents the activation function.

3.3.5. BiLSTM Layer

The feature maps extracted by the CNN from the original data are input into the subsequent long short-term memory (LSTM) layer to achieve capacity prediction. LSTM can achieve better prediction for time-related features by learning short-term and long-term dependencies. The basic structure of LSTM is illustrated in Figure 7.

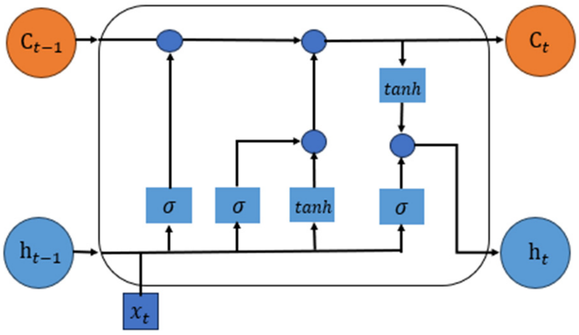


Figure 7. Schematic of LSTM.

In the LSTM neural network, information transmission is completed through the forget gate, input gate, and output gate. LSTM can save information in a long sequence. These different types of gates work together to realize the memory and forgetting functions of past and current information.

These gates include sigmoid and tanh activation functions, which can realize the passing, filtering, and control operations of information flow. The process of information transmission can be described as follows:

1. Memory and forgetting of information: The input information and stored information are multiplied by the corresponding weights, combined with the bias, and then passed through the sigmoid function for normalization. The process results for the next step are shown below.

$$f_t = sigmoid\left(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f\right), \tag{15}$$

2. The data is first processed by the weight matrix, then multiplied by the activation matrix to extract the relevant information as it moves from the input to the memory unit

$$\tilde{s}_t = tanh\left(W_{\tilde{s},x}x_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}}\right), \tag{16}$$

$$i_t = sigmoid(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i), \tag{17}$$

where $b_{\tilde{s}}$ and b_i represent the bias, $W_{\tilde{s},x}$, $W_{\tilde{s},h}$, $W_{i,x}$, $W_{i,h}$ represent the weight matrix, $\tilde{s}_t$ is the candidate value for new information, and i_t refers to the activation matrix of the input gate. These elements are used to control the flow of information into the memory unit, determining what new information should be stored based on the current input.

3. Update of unit state and information output: Through the calculation of the previous two processes, the unit state is updated. The updated unit state and the output matrix are calculated to obtain the updated output:

$$s_t = f_t \cdot s_{t-1} + i_t \cdot \tilde{s}_t, \tag{18}$$

$$o_t = sigmoid(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o), \tag{19}$$

$$h_t = o_t \times tanh(s_t), \tag{20}$$

where b_o is the bias vector, $W_{o,h}$ represent the weight matrix, o_t is the activation matrix of the output gate, and s_t refers to the unit state. These components are responsible for controlling how much of the information in the memory unit is passed through the output gate at each time step, ultimately influencing the final output of the network. The principle is shown in Figure 7.

Bidirectional Long Short-Term Memory (BiLSTM) is an advanced variation of LSTM. By combining LSTM components in two directions, the BiLSTM network can capture bidirectional dependencies and consider the information before and after the sequence at the same time. It has been widely used in many natural language processing (NLP) and time series prediction tasks. The output order of the generated network is as follows:

$$\overrightarrow{h}_t = f\left(x_t, \overrightarrow{h}_{t-1}\right), \tag{21}$$

$$\overleftarrow{h}_t = f\left(x_t, \overleftarrow{h}_{t-1}\right), \tag{22}$$

$$h_t = \overrightarrow{w}_t \overrightarrow{h}_t + \overleftarrow{w}_t \overleftarrow{h}_t + b_t, \tag{23}$$

where, $\overrightarrow{w}_t$ $\overleftarrow{w}_t$ represents the output weights of the hidden layers in two directions, and $\overrightarrow{h}_t$ represents the states of the hidden layers in two directions.

The network structures of CNN-LSTM-Attention, CNN-GRU-Attention, and CNN-BILSTM-Attention used in this study are illustrated in the Figure 8 below:

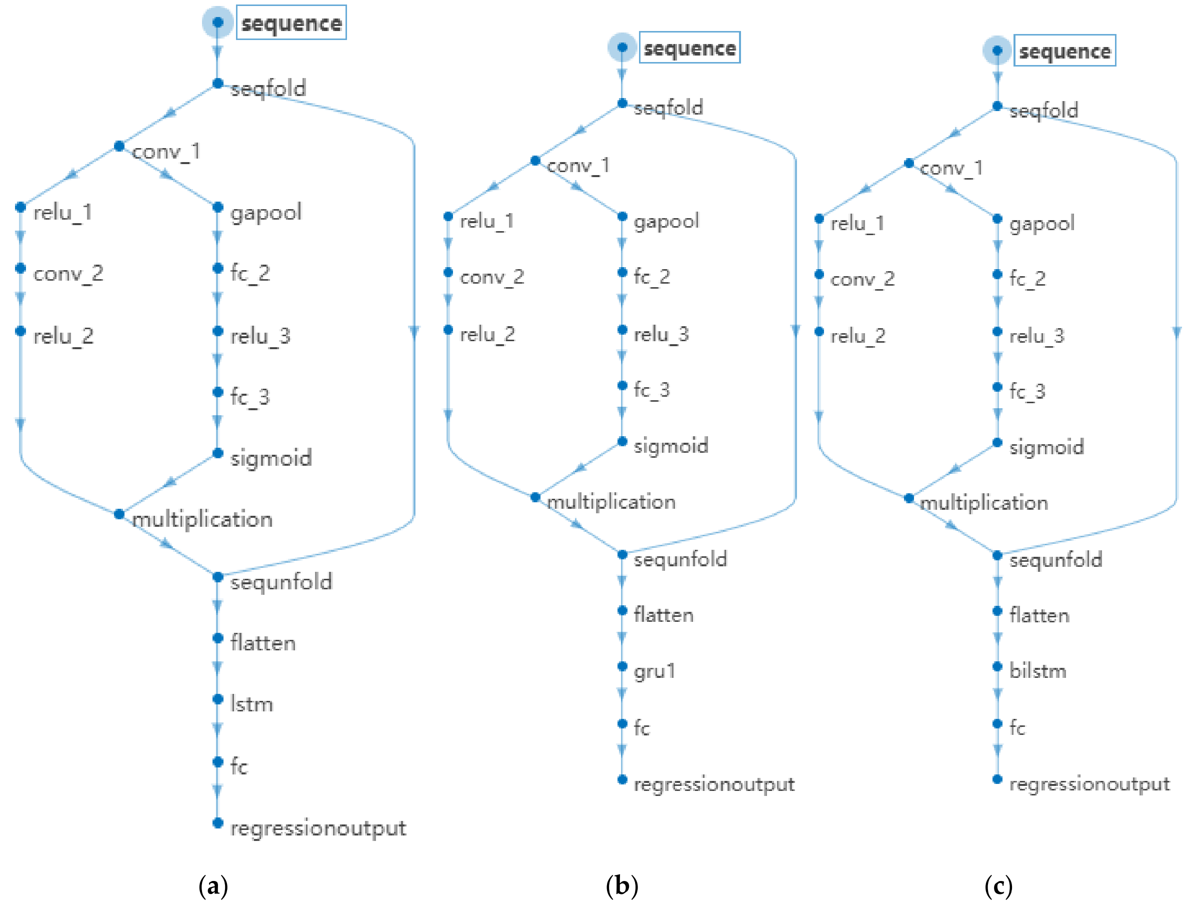


Figure 8. Deep learning network structure. (**a**) CNN-LSTM-Attention; (**b**) CNN-GRU-Attention; (**c**) CNN-BILSTM-Attention.

In these networks, the number of channels in the first convolutional layer is 32, and the second convolutional layer has 64 channels. The fully connected layers in the attention mechanism produce outputs of 16 and 64. All three networks share the same maximum number of iterations, set to 1500, with an initial learning rate of 0.01. The learning rate follows a piecewise strategy, with a decay factor of 0.1 and a decay period of 500 iterations. During training, data shuffling is applied to enhance the robustness of the models.

3.3.6. Evaluation Criteria

The evaluation criteria for the experiment will be based on the following indicators. The mean error (MAE) determines the average error between the true value and the predicted value. The mean square error (MSE) squares the error value before averaging it. The root mean square error (RMSE) calculates the square root of the MSE to gain insight into the accuracy of the results. The percentage of variance in a dependent variable's actual values that can be accounted for by the projected values of a regression model is expressed as the R-squared (R^2) value. y_t is the predicted value of the experiment, $\hat{y}$ represents the actual battery capacity, and $\overline{y}$ is the average value.

$$MAE = \frac{1}{n}\sum\nolimits_{t=1}^{n} |y_t - \hat{y}|, \tag{24}$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y})^2, \tag{25}$$

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{t=1}^{n} (y_t - \hat{y})^2}, \tag{26}$$

$$R^2 = 1 - \frac{\sum(y_t - \hat{y})^2}{\sum(y_t - \overline{y})^2} \tag{27}$$

4. Experiment

This part constructs three different types of experiments to study the impact of different inputs on different neural networks and determine the optimal input for different networks.

4.1. Correlation Experiment

Now in the field of battery parameter prediction, almost all studies use the Pearson correlation coefficient of health features as the main reference for evaluating health characteristics.

This article employs both the Pearson correlation coefficient and the Spearman correlation coefficient to evaluate each health feature. Pearson correlation coefficient assesses the linear correlation between two data sets, while Spearman correlation coefficient does not care whether the two data sets are linearly correlated but monotonically correlated. Spearman correlation coefficient is also called rank correlation or rank correlation. Pearson correlation coefficient requires that the statistical data must be continuous variables and conform to normal distribution, while the Spearman correlation coefficient does not have this requirement; Pearson correlation coefficient has poor stability and is not very reliable when singular values or long-tail distribution appear, while Spearman is relatively more robust.

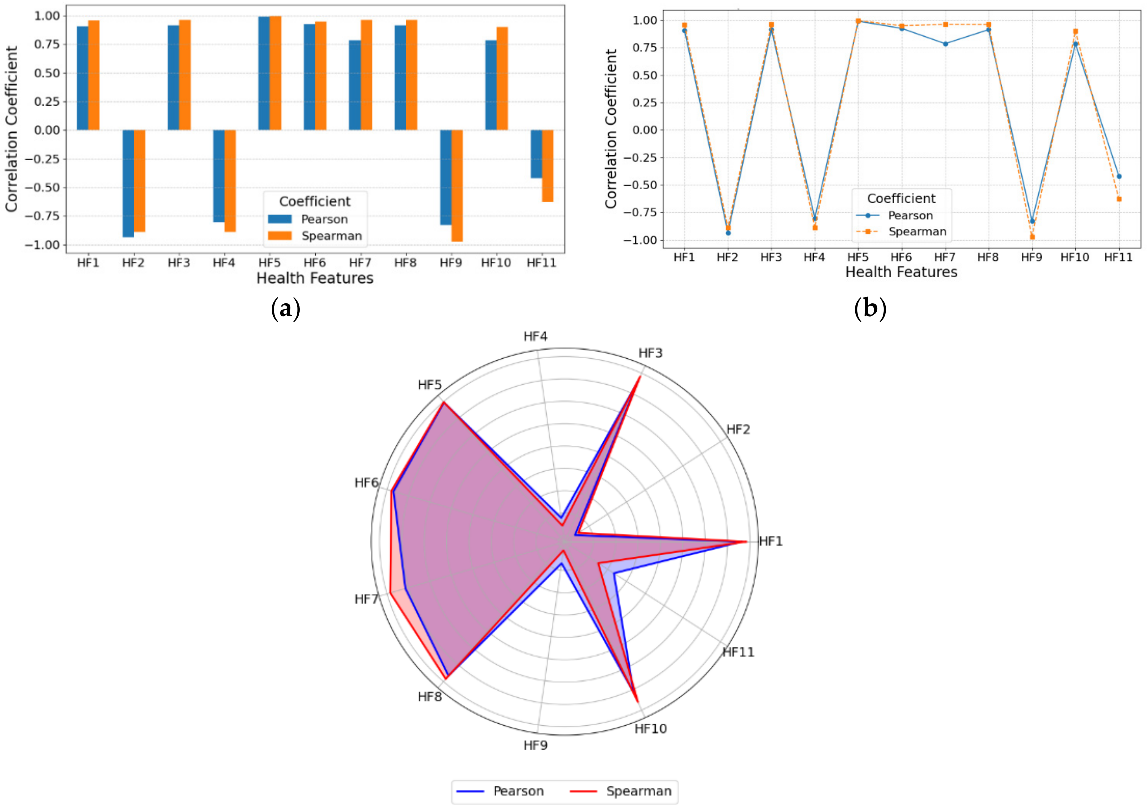
In this experiment, the input data was first grouped based on Pearson and Spearman correlation coefficients, respectively. The study then analyzed whether the impact of different input groups on various networks was consistent. This approach aims to identify the dependency differences of different networks on correlation coefficients, serving as the basis for subsequent experiments.

The experiment groups the experimental data according to two different correlation coefficients of battery capacity, and first studies the difference in the performance of different network models for different input groups. Obtain the dependence of different networks on different correlation coefficients. The following table is the correlation coefficient data:

From Table 3, it is evident that the Pearson correlation of each health feature is HF5, HF2, HF6, HF3, HF8, and HF1 from high to low, and the Spearman correlation is HF5, HF9, HF7, HF8, HF3, and HF1 from high to low. Figure 9 shows the coefficient differences between different plots.

Table 3. Correlation coefficient table.

Category	**Coefficient**	**HF1**	**HF2**	**HF3**	**HF4**	**HF5**
1Category	Pearson Spearman	0.9067 0.9566	−0.9353 −0.8900	0.9151 0.9612	−0.8021 −0.8897	0.9905 0.9954
Category	Coefficient	HF6	HF7	HF8	HF9	
2Category	Pearson Spearman	0.9250 0.9488	0.7860 0.9623	0.9135 0.9609	−0.8266 −0.9728	
Category	Coefficient	HF10	HF11			
3Category	Pearson Spearman	0.7842 0.9013	−0.4181 −0.6260			



(**a**)

(**b**)

(**c**)

Figure 9. Correlation comparison diagram. (**a**) Pearson and Spearman coefficient grouped bar chart; (**b**) Pearson and Spearman coefficients; (**c**) Pearson and Spearman correlation coefficients radar chart.

To verify the degree of dependence on different networks with different correlations of health features. This step of the experiment divides the input data into two groups, one group is the three health features with the highest Pearson correlation, and the other group is the three health features with the highest Spearman correlation. These two groups of data are used to train PSO-BP, SVM, CNN-LSTM-Attention network, CNN-LSTM-Attention, and CNN-

BILSTM-Attention respectively. The training set accounts for 80% and the test set accounts for 20%. All experiments are completed on Matlab2022 [35]. All the experiments were conducted on a Dell G16 7630 laptop, equipped with a 13th Gen Intel® Core™ i7-13650HX processor operating at 2.60 GHz. The graphics card utilized is an Nvidia GeForce RTX 4060 Laptop GPU.

Because the machine learning algorithm is relatively simple, MAE and MSE are selected for analysis. B5 is the experimental group and B18 is the control group. The results are presented in Tables 4 and 5.

Table 4. Machine learning experiment results (B5).

PSO-BP	**MAE**	**RMSE**
Pearson Sperman √	0.0063839 0.0045212	0.0081825 0.0055251
SVM	MAE	RMSE
Pearson Spearman √	0.010132 0.0096875	0.011728 0.011095

Table 5. Machine learning experiment results (B18).

PSO-BP	**MAE**	**RMSE**
Pearson Sperman √	0.031932 0.016178	0.038874 0.018226
SVM	MAE	RMSE
Pearson Spearman √	0.042804 0.023332	0.05041 0.023332

Then, the deep learning algorithm experiment was conducted and selected MAE, MSE, RMSE, and R^2 for evaluation of the experimental results and obtained the following experimental Tables 6 and 7. B5 is the experimental group and B18 is the control group.

Table 6. Deep learning experiment results (B5).

CNN-LSTM	**MAE**	**MSE**	**RMSE**	**R^2**
Person Spearman √	0.0055257 0.0046942	6.26×10^{-05} 3.02×10^{-05}	0.007911 0.0055	0.99801 0.99904
CNN-GRU	MAE	MSE	RMSE	R^2
Person √ Spearman	0.0056249 0.0083583	6.60×10^{-05} 0.00013731	0.008126 0.011718	0.99833 0.99652
CNN-BiLSTM	MAE	MSE	RMSE	R^2
Person Spearman √	0.0066974 0.0057694	6.46×10^{-05} 5.36×10^{-05}	0.008039 0.007324	0.99821 0.99865

Through the analysis of experimental results from Tables 4–7, for PSO-BP, SVM, CNN-LSTM, and CNN-BiLSTM networks, the input of health features with high Spearman correlation coefficients can obtain more accurate results. For CNN-GRU networks, the input of health features with relatively high person coefficients can obtain more accurate results. And this influence has different degrees of influence on different networks. It has a greater impact on the PSO-BP and CNN-GRU networks, with a 45% reduction in MAE, while it has a smaller impact on other networks. The SVM network has the least impact, with an MAE deviation of about 10%. The results can be clearly shown in Figure 10. The construction of neural networks has different algorithmic logic for processing different tasks. Therefore, different networks have different correlation emphases on the input health features, which has a greater impact on the accuracy of the results. In practical applications, selecting an appropriate correlation coefficient based on the characteristics of the network can enhance the accuracy of the results.

Table 7. Deep learning experiment results (B18).

CNN-LSTM	**MAE**	**MSE**	**RMSE**	**R^2**
Person	0.009469	1.29×10^{-04}	0.011345	0.99753
Spearman √	0.008062	1.05×10^{-04}	0.010262	0.99798
CNN-GRU	MAE	MSE	RMSE	R^2
Person √	0.012533	0.000266	0.016319	0.99584
Spearman	0.016539	0.000644	0.025381	0.99001
CNN-BiLSTM	MAE	MSE	RMSE	R^2
Person	0.010401	1.51×10^{-04}	0.012268	0.99691
Spearman √	0.009494	1.63×10^{-04}	0.012766	0.99741

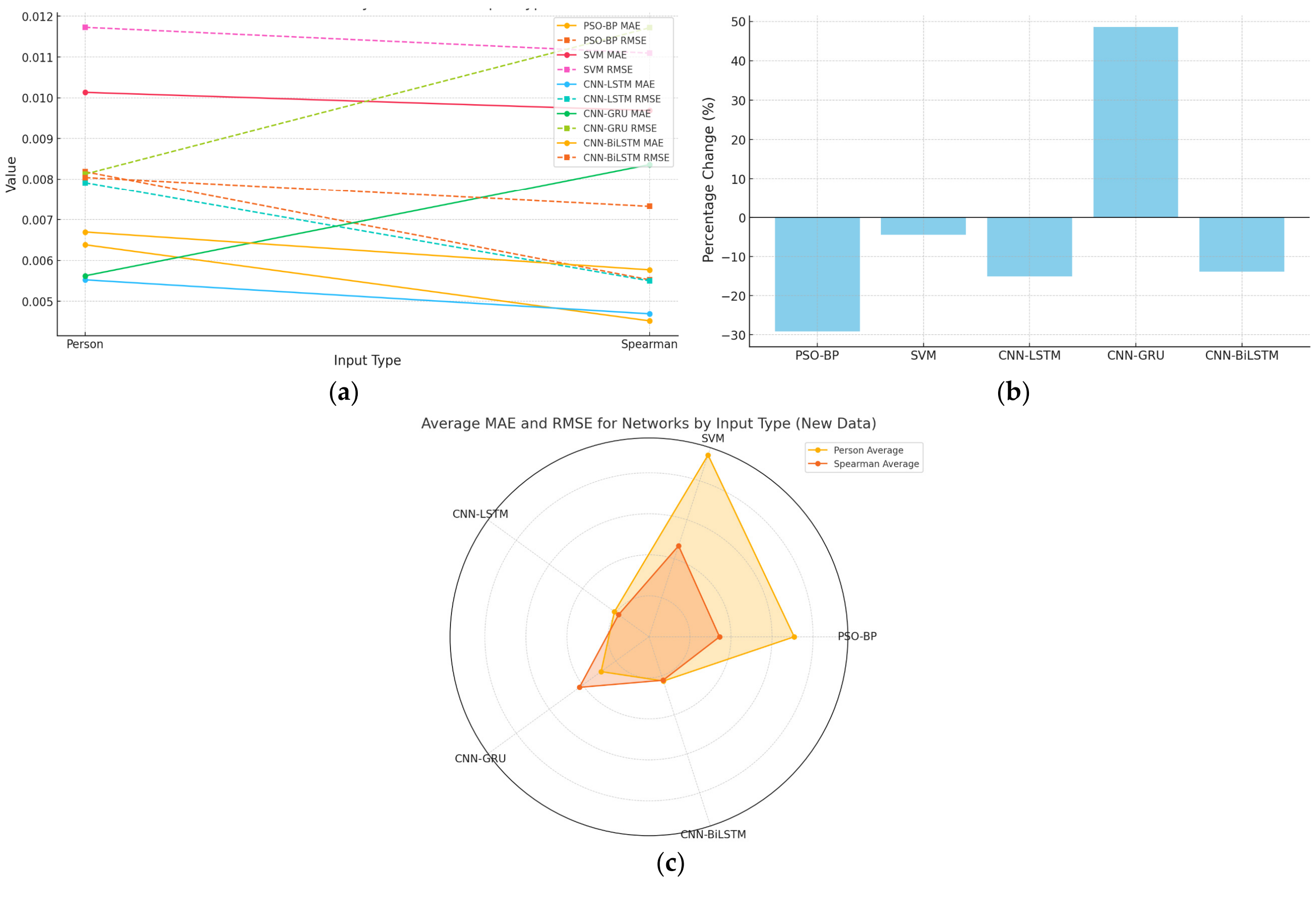


Figure 10. B5 Results comparison chart. (**a**) MAE and MSE trends line chart; (**b**) Percentage Change in MAE Bar Chart; (**c**) Average MAE and MSE Radar Chart.

4.2. Input Dimension Experiment

In current research in related fields, researchers often compare the performance of different neural networks using the same input features. However, different neural networks may have different optimal input dimensions.

Building on the first experiment, a second experiment was conducted by gradually increasing the number of input features for each network. This step is aimed at observing the relationship between the degree of overfitting and the increased input numbers for different algorithms and obtaining the following table. In terms of the order of increasing the dimension, when increasing to the sixth dimension, the difference in type was introduced, and the first, second, and third types of features were added respectively. B5 is the experimental group and B18 is the control group. The results of BSO-BP and SVM were obtained in Tables 8–11.

Table 8. BSP-BO dimensional experiment result (B5).

Input Size Number	MAE	RMSE
3	0.0045212	0.0055251
4	0.0074008	0.0084692
5	0.03395	0.038234

Table 9. BSP-BO dimensional experiment result (B18).

Input Size Number	MAE	RMSE
3	0.016178	0.018226
4	0.021651	0.025713
5	0.035144	0.041175

Table 10. SVM dimensional experiment result (B5).

Input Size Number	MAE	RMSE
3	0.0096875	0.011095
4	0.014118	0.016002
5	0.018886	0.021408

Table 11. SVM dimensional experiment result (B18).

Input Size Number	MAE	RMSE
3	0.023332	0.023332
4	0.027322	0.035099
5	0.033324	0.042639

The B5 results of BSO-BP and SVM were obtained in Figure 11. The comparison results show that the overfitting phenomenon of the PSO-BP network and the SVM network is the smallest when the input health feature is three, and the MAE and RMSE values are also the smallest. The accuracy of these two networks decreases as the input dimensionality increases. Therefore, in practical applications, it is recommended to select three features as input to achieve optimal performance.

For the PSO network, the results caused by the input of different dimensions are relatively different, and the MAE even has a difference of about 30%, while the input deviation of different dimensions of the SVM network is small, and the deviation of MAE is about 10%.

The experimental results analyzed using the CNN-GRU network showed that the optimal input dimensionality for these networks is also three, this is similar to the PSO-BP and SVM networks. The experimental results are presented in Tables 12 and 13. Under which the results achieve the highest accuracy. The comparison results of the three networks are shown in Figure 12.

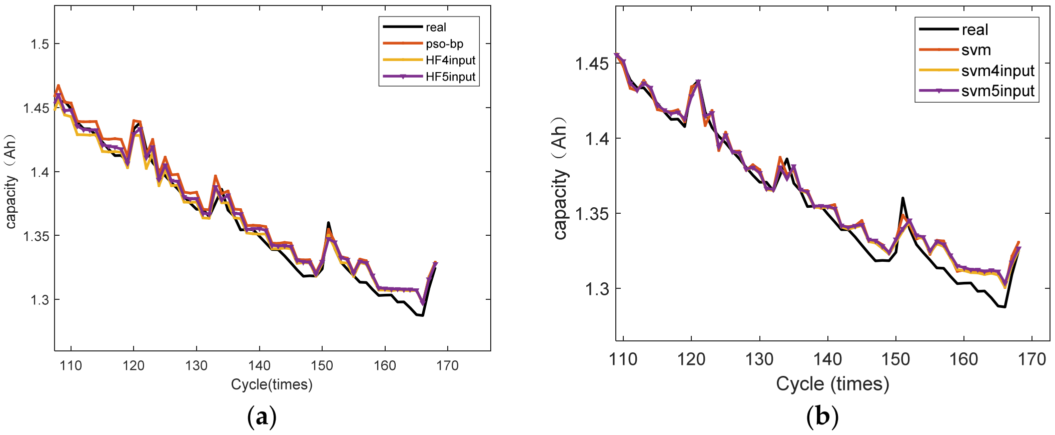


Figure 11. PSO-BP and SVM dimensional experiment for B5. (**a**) PSO-BP results for different input size; (**b**) SVM results for different input sizes.

Table 12. CNN-GRU dimensional experiment result (B5).

Input Size Number	MAE	MSE
3	0.005092	3.96×10^{-05}
4	0.008212	2.12×10^{-04}
5	0.005949	5.39×10^{-05}

Table 13. CNN-GRU dimensional experiment result (B18).

Input Size Number	MAE	MSE
3	0.005092	3.96×10^{-05}
4	0.008212	2.12×10^{-04}
5	0.005949	5.39×10^{-05}

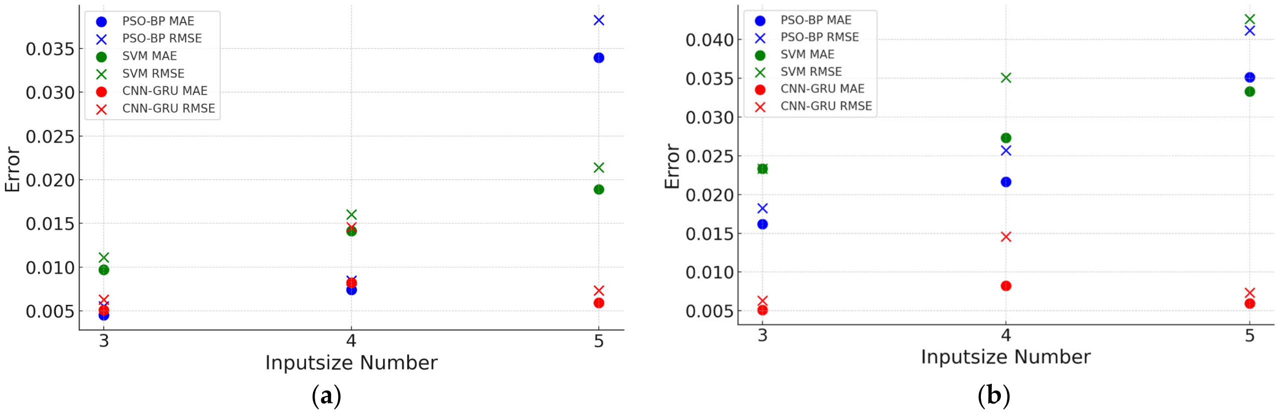


Figure 12. Results comparison scatter plot for different networks (**a**) B5 result; (**b**) B18 result.

Then the experiment of deep learning network was carried out, and the number of health features was gradually increased to study the relationship between the degree of overfitting and the increased dimension. In terms of the order of increasing the dimension, when the input size number is 6, the first, second, and third types of features that were previously classified are used differentially when the sixth input feature is added. B5 is the experimental group and B18 is the control group.

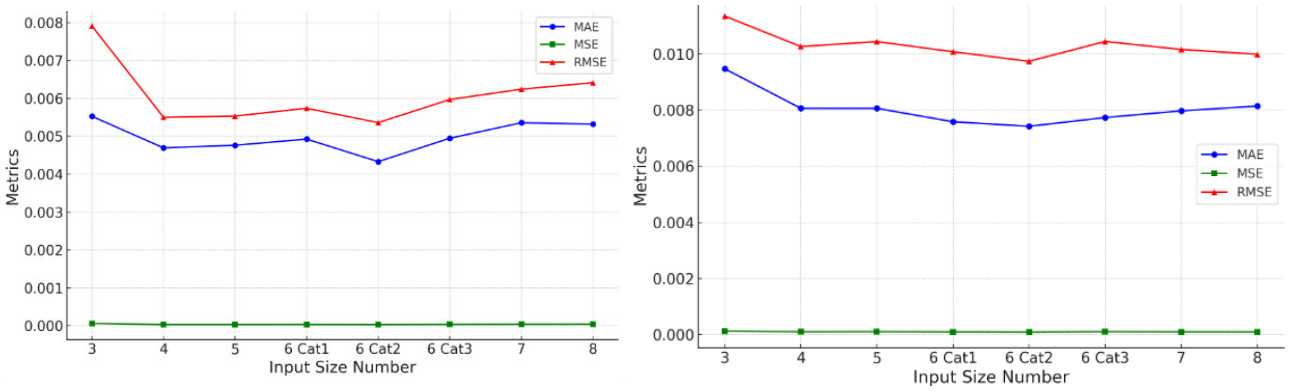
For the CNN-LSTM network, the experimental results are obtained in Tables 14 and 15, and the multi-dimensional CNN-LSTM network experiment results are displayed with a line chart, as shown in Figure 13. The analysis of the experimental results for the multi-dimensional CNN-LSTM network reveals that accurate results can be obtained when the number of input health features is 4. However, as the input dimensionality gradually increases from 6, significant overfitting becomes apparent. Comparing the results of the experimental group and the control group, the optimal outcomes in both cases were observed when the input dimensionality was 6.

Table 14. CNN-LSTM dimensional experiment result (B5).

Input Size Number	MAE	MSE	RMSE	R^2
3	0.0055257	6.26×10^{-05}	0.007911	0.99801
4	0.0046942	3.02×10^{-05}	0.0055	0.99904
5	0.0047624	3.06×10^{-05}	0.005533	0.99903
6 Category 1	0.0049245	3.29×10^{-05}	0.005739	0.99895
6 Category 2	0.0043307	2.87×10^{-05}	0.005359	0.99909
6 Category 3	0.0049479	3.56×10^{-05}	0.00597	0.99887
7	0.0053568	3.90×10^{-05}	0.006242	0.99876
8	0.0053192	4.11×10^{-05}	0.006413	0.99869

Table 15. CNN-LSTM dimensional experiment result (B18).

Input Size Number	MAE	MSE	RMSE	R^2
3	0.009469	1.29×10^{-04}	0.011345	0.99753
4	0.008062	1.05×10^{-04}	0.010262	0.99798
5	0.008061	1.09×10^{-04}	0.010438	0.99791
6 Category 1	0.007584	1.01×10^{-04}	0.010076	0.99805
6 Category 2	0.007421	9.48×10^{-05}	0.009736	0.99818
6 Category 3	0.007734	1.09×10^{-04}	0.010443	0.9979
7	0.007971	1.03×10^{-04}	0.010157	0.99802
8	0.008141	9.97×10^{-05}	0.009987	0.99808



(**a**) (**b**)

Figure 13. CNN-LSTM performance metrics for different input sizes (**a**) B5 result; (**b**) B18 result.

For 6-dimensional input, the variations among the three types of features were minimal, indicating that under multi-dimensional input conditions, reasonably accurate results can be achieved by adding features that are easier to obtain. In both groups, the highest accuracy was achieved when the input dimensionality was 6 and the sixth input feature belonged to the second type.

The specific fitting graph for the experimental group is shown in Figure 14. Therefore, in practical applications, using either 4-dimensional or 6-dimensional input for the

CNN-LSTM network is recommended to achieve the most accurate results. This differs significantly from the PSO-BP and SVM networks.

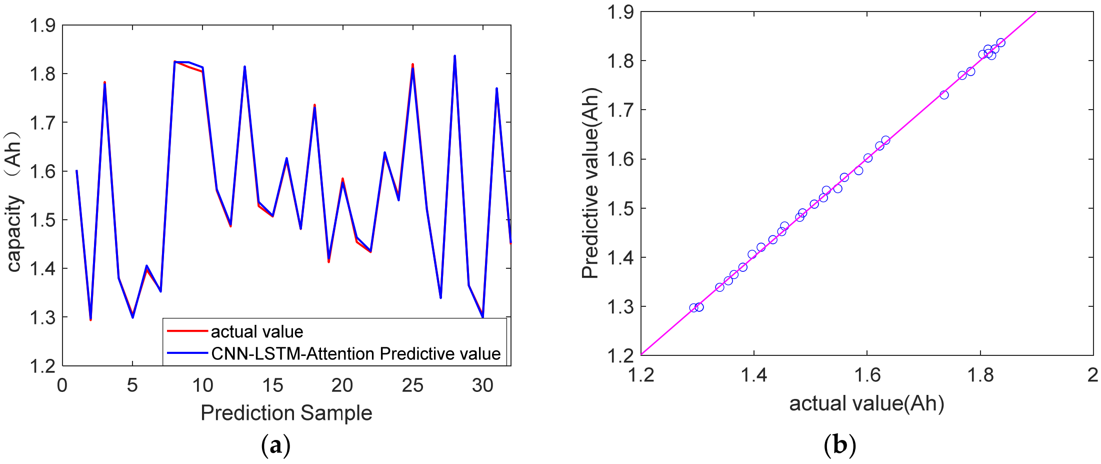


Figure 14. CNN-LSTM B6 Category 2 input experiment result (**a**) Test set fitting plot; (**b**) Error graph.

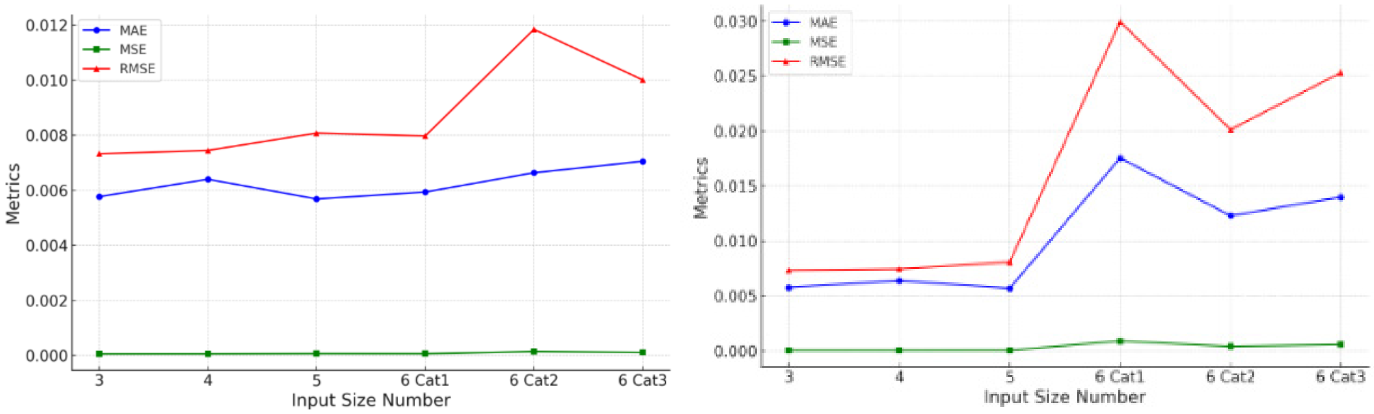
The dimensionality experiment results for the CNN-biLSTM network are presented in Tables 16 and 17 and Figure 15. The specific fitting graph for the experimental group is shown in Figure 16. When the number of the input health features ranges from 3 to 5, there is no significant overfitting phenomenon observed. When the input dimensionality ranges from 3 to 5, the differences in results are not significant. However, when the input health features become 6, The accuracy of the results decreased significantly, with the MAE value increasing by 34% compared to the input size of 3 and by 35% compared to the input size of 5. The optimal input dimensionality lies between 3 and 5, where variations in input within this range have minimal impact on result accuracy. Therefore, the best input choice for the CNN-biLSTM network should be between three and five dimensions, and it is recommended not to exceed this range.

Table 16. CNN-BiLSTM dimensional experiment result (B5).

Input Size Number	MAE	MSE	RMSE	R^2
3	0.0057694	5.36×10^{-05}	0.007324	0.99865
4	0.006398	5.55×10^{-05}	0.007447	0.9983
5	0.0056866	6.52×10^{-05}	0.008078	0.99805
6 Category 1	0.0059355	6.36×10^{-05}	0.007973	0.99805
6 Category 2	0.0066371	1.40×10^{-04}	0.011847	0.99589
6 Category 3	0.0070514	1.10×10^{-04}	0.010012	0.99747

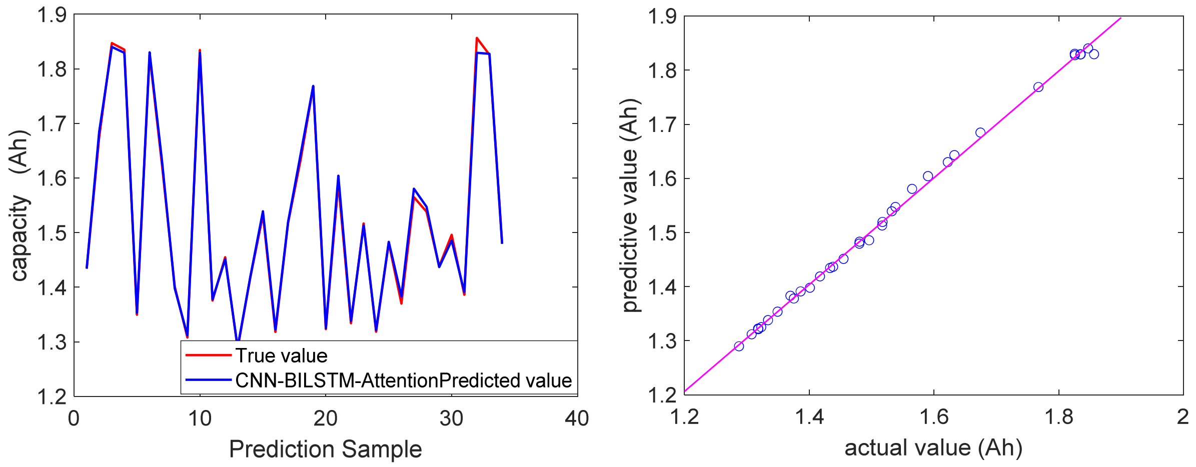
Table 17. CNN-BiLSTM dimensional experiment result (B18).

Input Size Number	MAE	MSE	RMSE	R^2
3	0.009494	1.63×10^{-04}	0.012766	0.99741
4	0.011392	0.000367	0.019156	0.9922
5	0.010866	0.000228	0.015115	0.99739
6 Category 1	0.017546	0.000897	0.02995	0.98606
6 Category 2	0.012294	0.000405	0.020131	0.99471
6 Category 3	0.013993	5.78×10^{-04}	0.025319	0.99397



(**a**) (**b**)

Figure 15. CNN-biLSTM performance metrics for different input sizes. (**a**) B5 result; (**b**)B18 result.



(**a**) (**b**)

Figure 16. CNN-biLSTM 5D input experiment result. (**a**) Test set fitting plot; (**b**) Error graph.

The second experiment demonstrates that the optimal input dimensionality varies among different networks, and the dimensionality of health features significantly influences the prediction accuracy of the results. Hence, in practical applications, it is essential to determine the most suitable input dimensionality for the selected network to improve the accuracy of prediction outcomes.

4.3. Input Data Volume Experiment

All previous experiments were conducted under the condition that the input data set accounted for 80% and the test set accounted for 20%. Because the algorithm needs to be fully analyzed during the training process, the data set should be shuffled before the training data set is divided. The experiment conducted partial tests on the shuffled data set and the non-shuffled data set, and the results also proved that the prediction accuracy after the shuffled data set was the highest. Building on these findings, the third step of the experiment was completed. For different networks, the training set proportion was gradually adjusted to 70%, 60%, and 30%, respectively, to observe the dependency of each network on the input data volume. The conclusions drawn from the experiment are presented in Tables 18–21. The drawn image is shown in Figure 17.

Table 18. CNN-GRU input volume test (B5).

Dataset Percentage	MAE	MSE	RMSE	R^2
0.8	0.0056249	6.60×10^{-05}	0.008126	0.99833
0.7	0.0069969	8.44×10^{-05}	0.0091876	0.99778
0.6	0.0077245	1.49×10^{-04}	0.01222	0.99575
0.3	0.007948	1.05×10^{-04}	0.01028	0.99708

Table 19. CNN-GRU input volume test (B18).

Dataset Percentage	MAE	MSE	RMSE	R^2
0.8	0.012096	2.52×10^{-04}	0.015881	0.99627
0.7	0.012533	2.66×10^{-04}	0.016319	0.99584
0.6	0.01305	3.49×10^{-04}	0.018691	0.99468
0.3	0.013333	3.28×10^{-04}	0.018131	0.9946

Table 20. CNN-BiLSTM input volume test (B5).

Dataset Percentage	MAE	MSE	RMSE	R^2
0.8	0.005769	5.36×10^{-05}	0.007324	0.99865
0.7	0.0063117	9.16×10^{-05}	0.0095717	0.99749
0.6	0.0067016	1.29×10^{-04}	0.011383	0.99639
0.3	0.0079478	1.43×10^{-04}	0.011991	0.99567

Table 21. CNN-BiLSTM input volume test (B18).

Dataset Percentage	MAE	MSE	RMSE	R^2
0.8	0.009494	1.63×10^{-04}	0.012766	0.99741
0.7	0.011424	2.23×10^{-04}	0.014944	0.99687
0.6	0.010827	2.60×10^{-04}	0.016121	0.99583
0.3	0.014252	4.39×10^{-04}	0.020944	0.99355

From the data in the tables, it can be observed that when the total input data volume exceeds 60%, the reduction in input data has a relatively smaller impact on the CNN-BiLSTM network compared to the GRU algorithm. This indicates that the CNN-BiLSTM network requires less data to achieve a certain level of accuracy.

When the total number of samples is sufficient, it is unnecessary to input all the data to obtain relatively accurate conclusions. In practical applications, controlling the input sample size within a specific accuracy threshold can significantly reduce processing time while maintaining prediction accuracy. This not only improves efficiency but also simplifies the data management process.

Although the training time of different networks fluctuates, the specific calculation time range is shown in Table 22.

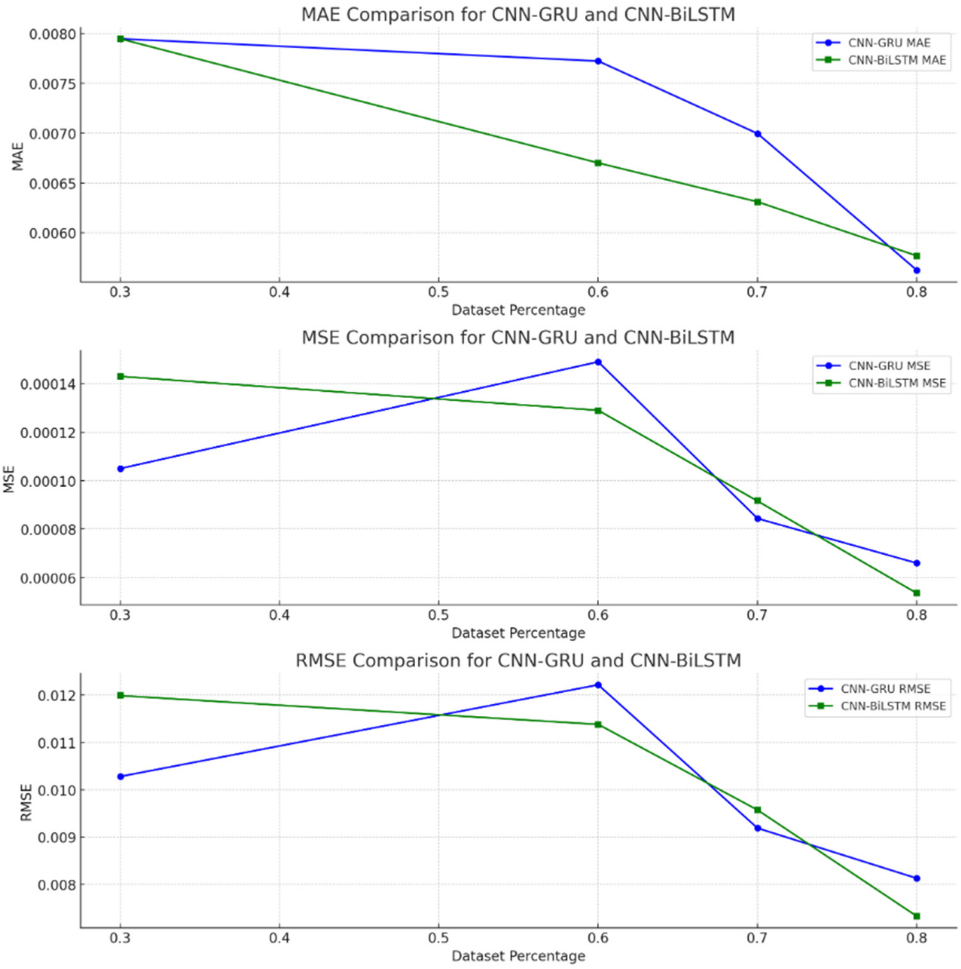


Figure 17. Input data volume comparison results.

Table 22. Calculation time.

Network	Training Time Range
PSO-BP SVM	1–1.2 s 0.5–1 s
CNN-LSTM-Attention	15–19 s
CNN-GRU-Attention CNN-BILSTM-Attention	15–27 s 17–23 s

5. Discussion

This paper uses comparative experiments of machine learning and deep learning methods to confirm that in the research of battery capacity prediction, different neural networks have optimal inputs corresponding to their network characteristics. There are some differences between these inputs. The specific differences include the category of input features, the dimensions of input features, and the capacity of input data. Therefore, when comparing the performance of different neural networks, in addition to factors such as the complexity of their structure and the time required for calculation, it is also necessary to consider the differences in the requirements of different networks for input features. The experimental results show that:

1. Different networks have their own optimal data input types and optimal input sizes. CNN-GRU has a strong dependence on features with high Person correlation. SVM, PSO-BP, CNN-LSTM, and CNN-BiLSTM have a strong dependence on features with

high Spearman correlation. The PSO-BP, SVM, and CNN-GRU networks have the least overfitting phenomenon when the input health feature is 3, and the CNN-LSTM network has the most accurate results and the least overfitting when the input dimension is 6. This demonstrates that the CNN-BiLSTM network exhibits strong adaptability to input dimensionality. When the number of health feature inputs for the CNN-BiLSTM network increases to 6, the prediction accuracy significantly decreases. The optimal input dimensionality for the CNN-BiLSTM network lies between 3 and 5, with minimal differences within this range.

2. Additionally, while achieving high prediction accuracy, different networks exhibit varying requirements for input data. The reduction in input data has a relatively smaller impact on CNN-BiLSTM compared to the GRU algorithm. However, differences in input data volume have a significant effect on the prediction accuracy of different networks.

Therefore, when using different types of neural networks for battery capacity prediction, it is essential to determine the type, dimensionality, and quantity of input health features based on the specific structure and category of the network, as well as the requirements of practical applications. These factors are critical to the final results. Selecting optimized inputs tailored to the characteristics of each network can significantly improve the accuracy of the results.

In future research, efforts will focus on the availability of input features and the structural complexity of networks. Combined with IoT technology, the study will explore how to perform online battery parameter evaluation using the most easily accessible features.

Author Contributions: Conceptualization, K.X.; writing—original draft preparation, K.X.; review and project administration, F.J. and B.C.; supervision, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2023RIS-007).

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

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

References

1. Jaguemont, J.; Boulon, L. A Comprehensive Review of Lithium-ion Batteries used in Hybrid and Electric Vehicles at Cold Temperatures. *Appl. Energy* **2016**, *164*, 99–114. [CrossRef]
2. Hu, X.; Xu, L. Battery Lifetime Prognostics. *Joule* **2020**, *4*, 310–346. [CrossRef]
3. Bao, Y.; Dong, W.; Wang, D. Online Internal Resistance Measurement Application in Lithium-Ion Battery Capacity and State of Charge Estimation. *Energies* **2018**, *11*, 1073. [CrossRef]
4. Braco, E.; San Martín, I.; Sanchis, P.; Ursúa, A. Fast Capacity and Internal Resistance Estimation Method for Second-Life Batteries from Electric Vehicles. *Appl. Energy* **2023**, *329*, 120235. [CrossRef]
5. Scipioni, R.; Jørgensen, P. Complementary Analyses of Aging in a Commercial LiFePO4/graphite 26650 Cell. *Electrochim. Acta* **2018**, *284*, 454–468. [CrossRef]
6. Liu, D.; Pang, J. Prognostics for State of Health Estimation of Lithium-ion Batteries based on Combination Gaussian process Functional Regression. *Microelectron. Reliab.* **2013**, *53*, 832–839. [CrossRef]
7. Topan, P.A.; Ramadan, M.N.; Fathoni, G.; Cahyadi, A.I.; Wahyunggoro, O. State of Charge (SOC) and State of Health (SOH) Estimation on Lithium Polymer Battery via Kalman Filter. In Proceedings of the 2016 2nd International Conference on Science and Technology-Computer (ICST), Yogyakarta, Indonesia, 27–28 October 2016; IEEE: New York, NY, USA, 2017; pp. 93–96.
8. Wang, D.; Zhang, Q.; Huang, H.; Yang, B.; Dong, H.; Zhang, J. An Electrochemical–Thermal Model of Lithium-Ion Battery and State of Health Estimation. *J. Energy Storage* **2022**, *47*, 103528. [CrossRef]
9. Lüders, C.; Keil, J. Modeling of Lithium Plating and Lithium Stripping in Lithium-ion Batteries. *J. Power Sources* **2019**, *414*, 41–47. [CrossRef]

10. Lai, X.; Gao, W. A Comparative Study of Global Optimization Methods for Parameter Identification of Different Equivalent Circuit Models for Li-ion Batteries. *Electrochim. Acta* **2019**, *295*, 1057–1066. [CrossRef]
11. Yang, J.; Xia, B. Online State-of-health Estimation for Lithium-ion Batteries using Constant-voltage Charging Current Analysis. *Appl. Energy* **2018**, *212*, 1589–1600. [CrossRef]
12. Eddahech, A.; Briat, O. Behavior and State-of-health Monitoring of Li-ion Batteries using Impedance Spectroscopy and Recurrent Neural Networks. *Int. J. Electr. Power Energy Syst.* **2012**, *45*, 487–494. [CrossRef]
13. 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]
14. Shen, S.; Liu, B.; Zhang, K.; Ci, S. Toward Fast and Accurate SOH Prediction for Lithium-ion Batteries. *IEEE Trans. Energy Convers.* **2021**, *36*, 2036–2046. [CrossRef]
15. Li, Y.; Li, K.; Liu, X.; Zhang, L. Fast Battery Capacity Estimation Using Convolutional Neural Networks. *Trans. Inst. Meas. Control* **2020**, *42*, 0142331220966425. [CrossRef]
16. Ren, L.; Dong, J.; Wang, X.; Meng, Z.; Zhao, L.; Deen, M.J. A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life. *IEEE Trans. Ind. Inform.* **2021**, *17*, 3478–3487. [CrossRef]
17. Vidal, C.; Kollmeyer, P.; Naguib, M.; Malysz, P.; Gross, O.; Emadi, A. Robust xEV Battery State-of-charge Estimator Design using a Feedforward Deep Neural Network. *SAE Int. J. Adv. Curr. Pract. Mobil.* **2020**, *2*, 2872–2880. [CrossRef]
18. Jiao, M.; Wang, D.; Qiu, J. A GRU-RNN Based Momentum Optimized Algorithm for SOC Estimation. *J. Power Sources* **2020**, *459*, 228051. [CrossRef]
19. Weng, C.; Cui, Y. On-board State of Health Monitoring of Lithium-ion Batteries Using Incremental Capacity Analysis with Support Vector Regression. *J. Power Sources* **2013**, *235*, 36–44. [CrossRef]
20. Dong, H.; Jin, X. Lithium-ion Battery State of Health Monitoring and Remaining Useful Life Prediction based on Support Vector Regression-particle Filter. *J. Power Sources* **2014**, *271*, 114–123. [CrossRef]
21. Sbarufatti, C.; Corbetta, M. Adaptive Prognosis of Lithium-ion Batteries based on the Combination of Particle Filters and Radial basis Function Neural Networks. *J. Power Sources* **2017**, *344*, 128–140. [CrossRef]
22. Lipu, M.S.H.; Ansari, S.; Miah, M.S.; Meraj, S.T.; Hasan, K.; Shihavuddin, A.S.M.; Hannan, M.A.; Muttaqi, K.M.; Hussain, A. Deep Learning Enabled State of Charge, State of Health and Remaining Useful Life Estimation for Smart Battery Management System: Methods, Implementations, Issues and Prospects. *J. Energy Storage* **2022**, *55*, 105752. [CrossRef]
23. Chen, S.-Z.; Liang, Z.; Yuan, H.; Yang, L.; Xu, F.; Zhang, Y. Li-Ion Battery State-of-Health Estimation Based on the Combination of Statistical and Geometric Features of the Constant-Voltage Charging Stage. *J. Energy Storage* **2023**, *72*, 108647. [CrossRef]
24. Raman, M.; Champa, V.; Prema, V. State of Health Estimation of Lithium-Ion Batteries Using Recurrent Neural Network and Its Variants. In Proceedings of the 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 9–11 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
25. Bao, Q.; Qin, W.; Yun, Z. A Multi-Stage Adaptive Method for Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Swarm Intelligence Optimization. *Batteries* **2023**, *9*, 224. [CrossRef]
26. Li, Y.; Stroe, D.I.; Cheng, Y.; Sheng, H.; Sui, X. On the Feature Selection for Battery State of Health Estimation Based on Charging–Discharging Profiles. *J. Energy Storage* **2021**, *33*, 102122. [CrossRef]
27. Jia, J.; Liang, J.; Shi, Y.; Wen, J.; Pang, X.; Zeng, J. SOH and RUL Prediction of Lithium-Ion Batteries Based on Gaussian Process Regression with Indirect Health Indicators. *Energies* **2020**, *13*, 375. [CrossRef]
28. Lu, C.; Tao, L.; Fan, H. Li-Ion Battery Capacity Estimation: A Geometrical Approach. *J. Power Sources* **2014**, *261*, 141–147. [CrossRef]
29. Zhang, Z.; Zhang, R.; Liu, X.; Zhang, C.; Sun, G.; Zhou, Y.; Yang, Z.; Liu, X.; Chen, S.; Dong, X.; et al. Advanced State-of-Health Estimation for Lithium-Ion Batteries Using Multi-Feature Fusion and KAN-LSTM Hybrid Model. *Batteries* **2024**, *10*, 433. [CrossRef]
30. 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. [CrossRef]
31. Saha, B.; Goebel, K. *Battery Data Set*; NASA Ames Prognostics Data Repository, NASA Ames Research Center: Moffett Field, CA, USA, 2007. Available online: https://phm-datasets.s3.amazonaws.com/NASA/5.+Battery+Data+Set.zip (accessed on 9 July 2024).
32. Lin, H.; Kang, L.; Xie, D.; Linghu, J.; Li, J. Online State-of-Health Estimation of Lithium-Ion Battery Based on Incremental Capacity Curve and BP Neural Network. *Batteries* **2022**, *8*, 36. [CrossRef]
33. Xia, F.; Tang, C.; Chen, J. Online Two-Dimensional Filter for Anti-Interference Aging Features Extraction to Accurately Predict the Battery Health. *Measurement* **2024**, *224*, 114758. [CrossRef]

34. Chen, K.; Wang, D.; Guo, W. A Temporal Fusion Memory Network-Based Method for State-of-Health Estimation of Lithium-Ion Batteries. *Batteries* **2024**, *10*, 286. [CrossRef]
35. MATLAB R2022b User's Guide. 2022. Available online: https://ww2.mathworks.cn/products/new_products/release2022b.html (accessed on 9 July 2024).

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.