

Article

Remaining Useful Life Estimation of Lithium-Ion Batteries Using Alpha Evolutionary Algorithm-Optimized Deep Learning

Fei Li ^{1,†}, Danfeng Yang ^{1,†}, Jinghan Li ², Shuzhen Wang ^{1,*}, Chao Wu ^{1,*}, Mingwei Li ¹, Chuanfeng Li ¹, Pengcheng Han ¹ and Huafei Qian ³

¹ School of Power Electrical Engineering, Luoyang Institute of Science and Technology, Luoyang 471023, China; lifei@lit.edu.cn (F.L.); 18625468187@163.com (D.Y.); lcf@lit.edu.cn (C.L.); birdhpc@my.swjtu.edu.cn (P.H.)

² College of Information Engineering and Artificial Intelligence, Henan University of Science and Technology, Luoyang 471023, China

³ Harbin Shenkong Technology Co., Ltd., Harbin 150028, China

* Correspondence: wangsz@lit.edu.cn (S.W.); wuchao@lit.edu.cn (C.W.)

† These authors contributed equally to this work.

Abstract

The precise prediction of the remaining useful life (RUL) of lithium-ion batteries is of great significance for improving energy management efficiency and extending battery lifespan, and it is widely applied in the fields of new energy and electric vehicles. However, accurate RUL prediction still faces significant challenges. Although various methods based on deep learning have been proposed, the performance of their neural networks is strongly correlated with the hyperparameters. To overcome this limitation, this study proposes an innovative approach that combines the Alpha evolutionary (AE) algorithm with a deep learning model. Specifically, this hybrid deep learning architecture consists of convolutional neural network (CNN), time convolutional network (TCN), bidirectional long short-term memory (BiLSTM) and multi-scale attention mechanism, which extracts the spatial features, long-term temporal dependencies, and key degradation information of battery data, respectively. To optimize the model performance, the AE algorithm is introduced to automatically optimize the hyperparameters of the hybrid model, including the number and size of convolutional kernels in CNN, the dilation rate in TCN, the number of units in BiLSTM, and the parameters of the fusion layer in the attention mechanism. Experimental results demonstrate that our method significantly enhances prediction accuracy and model robustness compared to conventional deep learning techniques. This approach not only improves the accuracy and robustness of battery RUL prediction but also provides new ideas for solving the parameter tuning problem of neural networks.



Academic Editor: Yong-Joon Park

Received: 5 September 2025

Revised: 23 September 2025

Accepted: 15 October 2025

Published: 20 October 2025

Citation: Li, F.; Yang, D.; Li, J.; Wang, S.; Wu, C.; Li, M.; Li, C.; Han, P.; Qian, H. Remaining Useful Life Estimation of Lithium-Ion Batteries Using Alpha Evolutionary Algorithm-Optimized Deep Learning. *Batteries* **2025**, *11*, 385. <https://doi.org/10.3390/batteries11100385>

Copyright: © 2025 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/>).

Keywords: lithium-ion battery; remaining useful life (RUL); Alpha evolutionary algorithm; deep learning

1. Introduction

In the current era of rapid development in renewable energy and electric vehicle industries, the performance and lifespan assessment of lithium-ion batteries, as core energy storage components, have become key issues in both scientific research and industrial fields [1–3]. Due to complex factors such as the number of charge–discharge cycles, operating temperature, and charge–discharge rates, lithium-ion batteries inevitably experience capacity decline, power reduction, and shortened lifespan. Therefore, accurately assessing the state of health (SOH) of lithium-ion batteries and reliably predicting their remaining

useful life (RUL) is of significant practical importance for enhancing equipment reliability, optimizing maintenance strategies, and reducing operational costs.

In recent years, with the rapid development of machine learning and artificial intelligence technologies, data-driven methods for predicting the lifespan of lithium-ion batteries have attracted widespread attention. Many scholars have utilized deep learning techniques to construct various models for RUL prediction, such as Long Short-Term Memory networks (LSTM), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN). Han et al. [4] innovatively proposed a denoising Transformer-based neural network (DTNN) model, which demonstrated significant advantages over traditional machine learning models and other deep learning architectures in terms of the accuracy and reliability of lithium-ion battery RUL prediction—its coefficient of determination (R^2) reached 0.991, with a mean absolute percentage error (MAPE) of only 0.632% and an absolute RUL error as low as 3.2 cycles. Gu and Liu [5] addressed the individual differences in lithium-ion batteries' practical applications by proposing a RUL prediction method based on transfer learning: through the application of extreme learning machines (ELM) twice, they effectively bridged the performance gaps between different batteries, significantly enhancing the accuracy and efficiency of predictions. Akram et al. [6] presented an innovative method for battery state of health (SOH) estimation by analyzing distribution of relaxation times parameters across varying state of charge levels. A LSTM learning model demonstrates significantly enhanced SOH prediction accuracy compared to conventional electrochemical impedance spectroscopy-based approaches. Chen et al. [7] designed a neural network integrating a denoising autoencoder (DAE) and Transformer: by preprocessing the original capacity data with DAE, it effectively mitigated the interference of noise on the prediction results, outperforming existing mainstream methods in RUL prediction tasks. Zhang et al. [8] utilized deep transfer learning technology to propose a lithium-ion battery lifespan prediction method applicable to multiple discharge strategies: by transferring the Fta features of batteries under different discharge strategies, it effectively addressed the issue of large data distribution differences and achieved real-time personalized assessment of battery health status. Bellomo et al. [9] proposed methodologies for addressing LSTM regression tasks involving input and output sequences with heterogeneous lengths. This work systematically examined the Autoregressive one-step prediction framework and subsequently presented a novel one-time multi-step prediction approach. This innovative method, grounded in a customized loss function architecture, enables simultaneous prediction of all future temporal steps within a unified computational framework. Saleem et al. [10] introduced a novel TransRUL framework to advance battery RUL prediction. A dual-encoder transformer design with hybrid positional encoding and multi-head attention mechanisms are integrated for time-series feature extraction. The convolutional long short-term memory deep neural network model and the temporal transformer model have demonstrated superior performance in RUL prediction [11].

Although the above studies have made phased progress in lithium-ion battery RUL prediction, there is still room for improvement in terms of prediction accuracy, model stability, and adaptability to complex operating conditions. To further enhance the accuracy and reliability of RUL prediction, this study proposes a hybrid method integrating deep learning and Alpha Evolution (AE) algorithm [12]: by leveraging the powerful global search capability of the AE algorithm, it systematically optimizes the weights, biases, and hyperparameters of deep learning models to construct a more superior lifespan prediction model. Through comparative evaluations with baseline models such as CNN, Temporal Convolutional Network (TCN), and Attention, the superiority of the proposed AE based CNN-TCN-BiLSTM-Attention architecture in lithium-ion battery lifespan estimation is verified. The experimental results show that the parameter optimization driven by AE

not only significantly improves the prediction accuracy but also enhances the stability and generalization ability of the model through a meticulous parameter exploration process.

2. Materials and Methods

2.1. Main Algorithms of Deep Learning

In the field of lithium-ion battery life prediction, deep learning has demonstrated great potential. Among various algorithms, CNN, RNN, and long LSTM networks are crucial for modeling and predicting battery data [13–15]. CNN is a deep learning model designed for image processing and has significant advantages in the field of computer vision. In the prediction of lithium battery life, historical data such as battery voltage, current, and temperature can be used as features for extraction and preprocessing to predict the battery's lifespan. RNN has unique advantages in handling time series data and is suitable for modeling continuous data. When applied to lithium battery life prediction, it can capture long-term dependencies in the data, which helps understand the changes in battery performance over time. Its common variants, LSTM and GRU, introduce gating mechanisms to address the problem of gradient vanishing or explosion. Traditional RNN can effectively learn long-term dependencies when processing long sequences. LSTM is an improved RNN algorithm. By introducing memory units and gating mechanisms, it overcomes the defects of traditional RNN in handling long sequence data, such as gradient vanishing and explosion. In lithium battery life prediction, LSTM can study and remember the performance of the battery in different states, which can more accurately predict the future lifespan and is an ideal choice in this field.

2.2. Convolutional Neural Network

CNN is an improved feedforward Backpropagation (BP) network derived from the theory of visual receptive fields, capable of processing multi-dimensional data such as images, time series, and text. Its parameter sharing mechanism reduces the number of parameters and model complexity while enhancing generalization ability, in contrast, fully connected networks extract high-level features through dense connections. A typical CNN structure consists of convolutional layers, pooling layers, and fully connected layers, with each layer performing distinct functions, as shown in Figure 1.

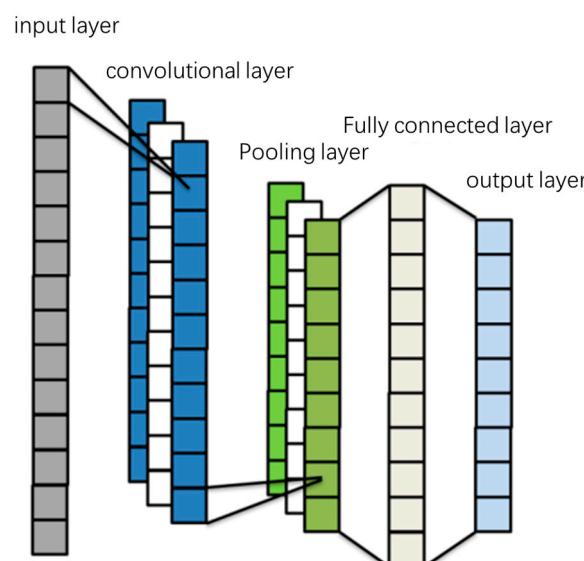


Figure 1. Schematic Diagram of the Convolutional Neural Network Model.

- Convolutional Layer

CNN is an improved feedforward backpropagation network based on the theory of visual receptive field [16,17], capable of processing multi-dimensional data such as images, time series, and text. Its parameter sharing mechanism enhances the generalization ability while reducing the number of parameters and model complexity. In contrast, fully connected networks extract high-level features through dense connections. The typical structure of CNN consists of convolutional layers, pooling layers, and fully connected layers. Each layer performs different functions, as shown in Figure 1.

$$x_t^l = \sum_{i=1}^{N_{l-1}} \text{conv2D}(w_{it}^{l-1}, s_i^{l-1}) + b_t^l \quad (1)$$

In the above formula, and respectively represent the input and bias of the t -th neuron in the l -th layer, is the convolution kernel between the i -th neuron node in the $(l - 1)$ -th layer and the t -th neuron node in the l -th layer, is the output of the i -th neuron node in the $(l - 1)$ -th layer, is the number of neuron nodes in the $(l - 1)$ -th layer, and conv1D denotes the one-dimensional convolution operation.

- Pooling Layer

The pooling layer performs dimensionality reduction on the output of the convolutional layer. Max pooling or average pooling is adopted according to the window size. The stride determines the moving interval of the sliding window: a larger stride results in a more significant reduction in the size of the feature map and higher computational efficiency, whereas a smaller stride retains more details. Equation (2) illustrates the max pooling process, which enhances model efficiency and robustness by compressing spatial dimensions.

$$s_t^l = \max_{(t-1)H+1 \leq j \leq tH} (s_j^{l-1}) \quad (2)$$

- Fully Connected Layer

The fully connected layer flattens the feature maps into one-dimensional vectors, integrates global features through the weight matrix ω and bias b , and outputs the prediction results. Each neuron is connected to all neurons in the previous layer, and the weights are optimized by minimizing the loss function. Equation (3) presents the specific content of the operation steps for the fully connected layer.

$$y = \sigma(wc_i + b) \quad (3)$$

In the above equation, ω represents a non-linear activation function (ReLU), while b denotes the bias term. The output result of the fully connected layer is characterized by y . Here, the weight coefficient is assigned the symbol ω .

Backpropagation in neural network models is used to calculate gradients, optimizing the parameter weights of convolutional layers and fully connected layers. Figure 1 is the schematic diagram of the convolutional neural network model. Its goal is to minimize the loss function and improve model performance. The dropout technique is frequently employed in this process, with specific operations manifested as: randomly dropping neuron outputs during training to reduce model complexity and enhance generalization ability. In addition to the aforementioned methods, regularization techniques introduce penalty terms into the loss function to constrain the scale of parameters and suppress overfitting.

2.3. Temporal Convolutional Neural Network and Feature Extraction

TCN is an improved time series modeling architecture based on CNN [18,19]. Its three core components—causal convolution (strictly ensuring temporal causality), dilated causal convolution (expanding the receptive field through dilation factors), and residual connection (optimizing the transmission of deep information)—work together: They efficiently capture long-term historical dependencies while preventing the leakage of future information, significantly enhancing the modeling ability for time series data.

- Causal Convolution

The TCN, characterized by strict causal constraints, produces outputs that depend solely on data at and before the current time t , with no inclusion of future information. The fundamental principle of “every effect has a cause” is fully embodied in this network.

- Dilated Causal Convolution

As the length of time-series sequences increases, traditional causal convolution requires stacking multiple layers of networks to capture longer historical information, leading to a sharp rise in computational complexity. To address this issue, a dilation factor d is introduced to form dilated causal convolution: by inserting interval sampling (with a stride of d) into the convolution operation, the receptive field is significantly expanded, as shown in Equation (4).

$$F = (k - 1) \times d + 1 \quad (4)$$

Here, F represents the size of the receptive field, k represents the size of the convolution kernel, and d represents the dilation factor. The operation formula is as follows:

$$F(x_t) = (F_d * X)(x_t) = \sum_{i=0}^{k-1} f_i x_{(t-d*i)} \quad (5)$$

where f_i represents the i -th value in the convolution kernel, and $t - d * i$ denotes the direction toward the past.

- Weight Normalization

Weight normalization refers to normalizing the weights, which is beneficial for accelerating convergence. Let the vector form of the convolution kernel be ω , the vector form of the receptive field be ω , and the bias size be b . Then, the output y of a neuron can be expressed as Equation (6):

$$y = \phi(\vec{w} \times \vec{x} + b) \quad (6)$$

where

$$\vec{w} = \|\vec{w}\| \cdot v / \|\vec{v}\| \quad (7)$$

- ReLU

The ReLU activation function is shown in Figure 2. As a piece-wise function, ReLU can be expressed as Equation (8):

$$f(x) = \max(0, x) \quad (8)$$

Its sparsity enables the output of true zero values, accelerating training and simplifying the model. Figure 2 is a schematic diagram of the ReLU activation function.

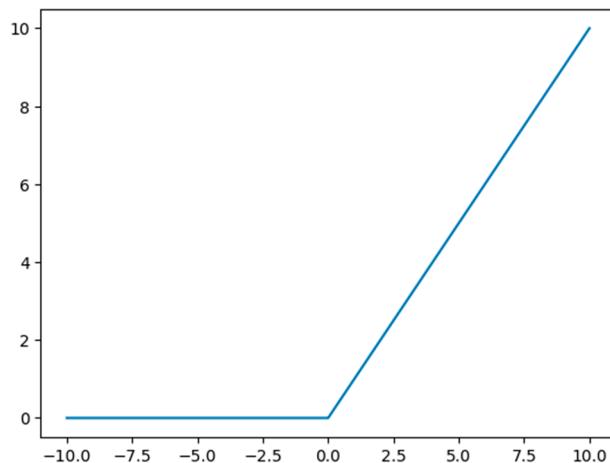


Figure 2. Schematic Diagram of the ReLU Activation Function.

- Dropout Layer

During training, neuron outputs are randomly dropped out. Through the effect of ensemble learning, this suppresses overfitting and improves generalization performance. Figure 3 shows a schematic diagram of its workflow.

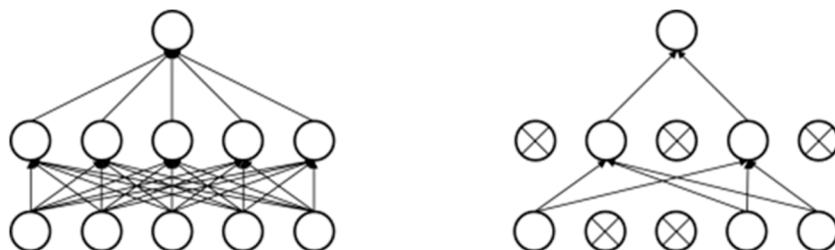


Figure 3. Schematic Diagram of the Dropout Layer Principle.

2.4. Bidirectional Long Short-Term Memory Recurrent Neural Network

- LSTM Network Structure

LSTM is specifically designed to address the problem of gradient vanishing/exploding in long sequence training. Its core lies in simulating the information processing process of the brain through a gating mechanism (input gate, forget gate, output gate), dynamically adjusting the cell state, retaining key features while selectively forgetting irrelevant historical information, and enabling the gradient to propagate stably over a long period in the time dimension [20–22]. This enables it to effectively capture long-range dependencies and significantly enhance the modeling ability for distant temporal patterns. In the time series problem of lithium battery life prediction, the calculation process of LSTM is as follows:

The forget gate aims to discard previously useless information. It generates $f_t \in [0, 1]$, through the sigmoid function based on h_{t-1} and x_t , which controls the discarding or retention of historical memory C_{t-1} . The calculation formula is shown in (9):

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (9)$$

The role of the input gate is to filter important features. The input filter i_t determines the content to be updated through the sigmoid layer, the candidate memory \tilde{C}_t generates new information through the tanh layer, and \tilde{C}_t in state update integrates historical and current key features. The calculation formulas are as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (10)$$

$$\bar{C}_t = \sigma([h_{t-1}, x_t] + b_c) \quad (11)$$

$$C_t = f_t C_{t-1} + i_t \bar{C}_t \quad (12)$$

The role of the output gate is to determine how much information at the current moment is worth outputting. O_t (sigmoid) controls the output intensity, and h_t generates the current hidden state. Through a parameterized selection mechanism, the gate enables the modeling of long-range dependencies and the stable propagation of gradients. The calculation formulas are as follows:

$$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \quad (13)$$

$$h_t = O_t \cdot \tanh(C_t) \quad (14)$$

- Principles of the BiLSTM Neural Network

BiLSTM is an improved architecture of LSTM. It synchronously captures both historical and future information of sequences through bidirectional LSTM units (forward propagation + backward propagation), fuses bidirectional features (as shown in Equation (15)) to explore deep-seated temporal patterns, and significantly enhances prediction accuracy and data utilization efficiency.

The BiLSTM connects two backward recurrent neural networks to the same output layer, breaking through the limitations of the traditional LSTM's one-way update [23–25]. It adds a data path from the future to the past, enabling the output layer to obtain complete sequence information. Moreover, the information in the hidden layer is independent of each other, which gives it a significant advantage in extracting time series features. Its final output result can be expressed by Formula (15):

$$h_t = \vec{h}_t \bullet \overleftarrow{h}_t \quad (15)$$

In the formula, \vec{h}_t represents the forward output of BiLSTM, and \overleftarrow{h}_t represents the backward output of BiLSTM. As shown in Figure 4, the structure of the BiLSTM network is presented.

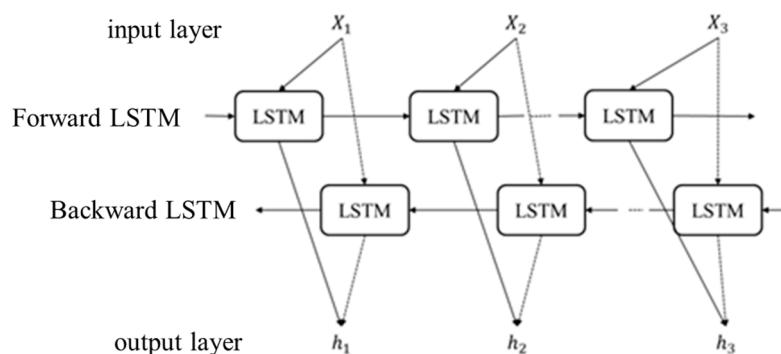


Figure 4. Structure of BiLSTM Network Structure Diagram.

- The AE algorithm

The TCN-BiLSTM model is composed of a TCN layer and a BiLSTM layer stacked together. After data preprocessing, the TCN layer extracts local and remote temporal features, while the BiLSTM layer further captures context and global features. The specific workflow of the model is as follows:

TCN layer: The preprocessed data is input into the TCN layer, which combines causal convolution and expansion convolution. Causal convolution strictly ensures that the

model only uses the data of the current time step, avoiding future information leakage, and expansion convolution effectively expands the receptive field by introducing dilation factors, making it possible to capture long-term battery degradation patterns.

BiLSTM layer: The output of the TCN layer (containing rich local time features) serves as the input of the BiLSTM layer. The BiLSTM layer consists of forward and backward propagation units, which synchronously capture the historical and future context information of the sequence. This design compensates for the relatively weak ability of the TCN layer to integrate global features, further integrating key information related to battery life, and enhancing the model's understanding of the overall degradation trend.

Fully connected layer: After the output of the BiLSTM layer, a fully connected layer with hidden neurons is introduced. This layer realizes the transformation from the high-dimensional feature space (extracted by TCN and BiLSTM) to the target RUL space, bridging the gap between feature representation and prediction output.

Output layer: For lithium battery RUL prediction, a linear activation function is directly used to output the predicted RUL value.

The TCN-BiLSTM model combines the unique advantages of TCN and BiLSTM. The advantage of TCN in extracting local and remote temporal features, and the advantage of BiLSTM in capturing context dependencies. It demonstrates strong feature extraction capabilities and high adaptability to battery operation data and can fully utilize the multi-dimensional features emphasized in this study, achieving high prediction accuracy.

2.5. Multi-Scale Attention Mechanism

To effectively integrate the output features of the CNN and TCN-BiLSTM models, this paper introduces the Multi-Scale Temporal Attention Mechanism (MTAM). By dynamically allocating weights to focus on key temporal information and correlate it with cross-model features, it enhances the model's ability to capture complex temporal patterns [26,27]. This module takes the concatenated sequences of the CNN output features and the TCN-BiLSTM output features as its input. It achieves multi-scale feature fusion through two layers of multi-head self-attention mechanisms and completes the final prediction through a fully connected layer.

1. Multi-head self-attention mechanism

To capture the dependencies at different scales in the feature sequence, the module adopts a two-layer multi-head self-attention structure and learns diverse feature correlation patterns through parallel attention heads (AH).

- The first layer self-attention (fine-grained feature correlation)

The first layer self-attention focuses on the fine-grained temporal local dependencies. It calculates the correlation weights between features through multiple parallel attention heads. For the input feature X , first, a linear transformation is applied to generate the query, key, and value matrices:

$$Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V \quad (16)$$

Here, W_i^Q, W_i^K, W_i^V represents the learnable parameter of the i -th attention head, and h is the hidden dimension of a single attention head.

The attention weights are calculated by scaling the dot product:

$$\text{Attention}(Q_i, K_i, V_i) = \text{soft} \left(\frac{Q_i K_i^T}{\sqrt{h}} \right) V \quad (17)$$

Here, \sqrt{h} is the scaling factor, which is used to alleviate the problem of gradient vanishing caused by the excessively large inner product values in high dimensions.

The outputs of H_1 attention heads are concatenated and passed through a linear transformation to obtain the output of the first layer of self-attention. To enhance the stability of the model, the output of the first layer is subjected to layer normalization (LN) and residual connection (RC):

$$Z'_1 = LN(Z_1 + X) \quad (18)$$

- Second-layer self-attention (cross-scale feature fusion)

The second layer of self-attention is designed to capture cross-scale global dependencies. By reducing the number of attention heads $H_2 < H_1$ and lowering the hidden dimension $h' = \frac{h}{2}$, it focuses on more abstract feature correlations. Its calculation process is similar to that of the first layer, but the input is the normalized output Z'_1 of the first layer, and the final output is:

$$Z_2 = LN(MH(Z'_1) + Z'_1) \quad (19)$$

Here, $d_{att1} < d_{att2}$ achieves the aggregation of cross-scale information through dimension compression.

- Feature Fusion and Output Layer

After being processed by two layers of attention mechanisms, the feature sequence already contains multi-scale temporal dependencies and cross-model feature correlations. To generate the final prediction result, the module performs feature fusion and dimension mapping through a fully connected layer:

$$Y = \text{ReLU}(Z_2 W_f + b_f) y = Y W_o + b_o \quad (20)$$

Here, W_f, b_f represents the parameters of the fusion layer, W_o, b_o represents the parameters of the output layer, ReLU is the activation function of the regression layer, and y represents the final predicted sequence.

- Training Configuration

The multi-scale attention network adopts the Adam optimizer, with a maximum training epoch set to 50, a batch size of 32, and an initial learning rate of 0.0005. To prevent overfitting, a piecewise learning rate schedule (with a learning rate decay factor of 0.2 and a decay period of 10 epochs) is employed, along with L2 regularization (with a coefficient of 0.0005) and gradient clipping (with a threshold of 0.5). During the training process, early stopping is performed based on the performance of the validation set (with a patience value of 100) to ensure the generalization ability of the model.

This multi-scale attention mechanism captures local and global temporal dependencies in a hierarchical manner, dynamically fuses the complementary features of CNN and TCN-BiLSTM, and effectively improves the prediction accuracy of complex time series data, especially for the demand for focusing on key information in multi-model fusion scenarios.

2.6. Model Optimization Based on Alpha Evolutionary Algorithm

The AE algorithm is a new evolutionary algorithm that updates the solution using an alpha operator with adaptive basis vectors and random and adaptive step sizes [12]. Firstly, candidate solutions are selected to construct the evolutionary matrix. Through diagonal or weighted operations of the evolutionary matrix, the population state is estimated. To enhance the correlation of each generation's estimation, the cumulative estimation results of two evolutionary paths are achieved to realize adaptive basis vectors. Secondly, the composite differential operation is used to construct adaptive step sizes to estimate the

problem gradient, which is used to accelerate the convergence of acoustic emission; finally, the attenuation factor alpha is adjusted adaptively according to the search space to generate random step sizes, to balance exploration and exploitation. The AE algorithm introduces a dynamic adaptive search strategy, achieving a balance between global exploration and local exploitation capabilities through a unique Alpha parameter regulation mechanism. Compared with traditional genetic algorithms (GA), differential evolution (DE), and particle swarm optimization (PSO), the core innovation of the AE algorithm lies in its adaptive search strategy, which can dynamically adjust the search step size and direction according to the evolutionary process, effectively avoiding the premature convergence or low search efficiency problems caused by fixed parameters in traditional algorithms. The AE algorithm performs better than conventional algorithms in multiple algorithm tests of CEC. This paper uses this optimized algorithm to optimize the hyperparameters of neural networks and compares with conventional optimization algorithms. The performance of the method in this paper is better. Figure 5 below is the flowchart of the AE algorithm.

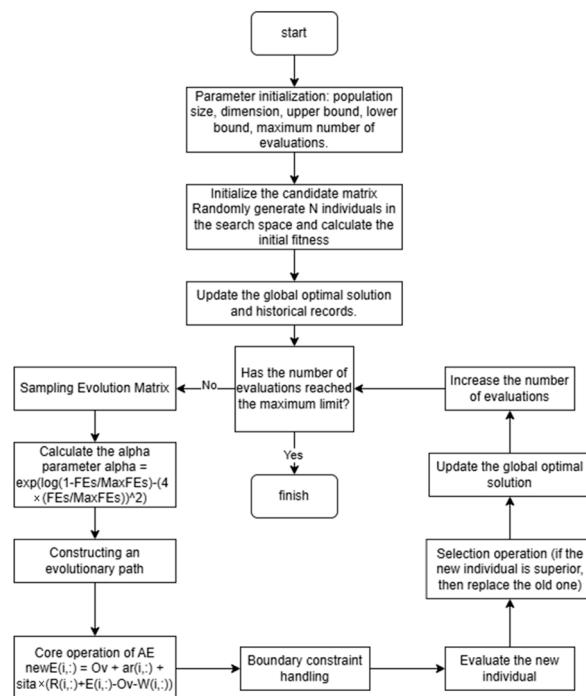


Figure 5. Flowchart of the AE algorithm.

1. Population Initialization

The strategy generates a set of candidate solutions based on the problem space, providing the evolutionary algorithm with an initial guess of promising information. In the search space D , the candidate solution X_i is usually uniformly initialized using the following equation:

$$X_i = lb + (ub - lb) * rand(0, 1, [1, D]), i = 1, 2, \dots, N \quad (21)$$

Here, X_i represents the candidate solution, and lb and ub , respectively, represent the lower and upper bounds of the search space.

2. Operators

In order to find the optimal solution to the problem, it is necessary to improve the candidate solutions, which is a key step in the optimization algorithm. In this work, the set of candidate solutions is defined as the candidate matrix X , while the evolutionary

matrix E is a subset of the candidate matrix. Please note that the overall concept is usually a metaphor for the solution set, and the solution set is essentially a matrix. During the evolution process, instead of directly evolving each solution X_i in the current candidate matrix, a sampling replacement operation is performed on the matrix to obtain the matrix to be evolved. The following Figure 6 describes this relationship:

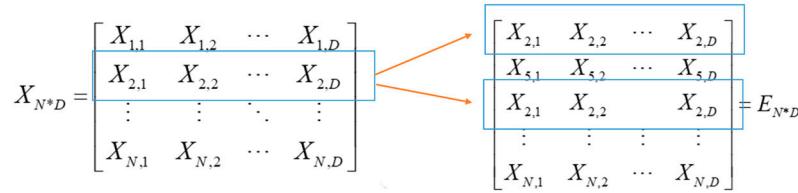


Figure 6. The evolving matrix.

This algorithm achieves efficient search solely by relying on an alpha operator. In this operator, multiple steps of extracting and utilizing evolutionary information are carried out simultaneously within the operator. The mathematical model of the operator is as follows:

$$E_i^{t+1} = P + \alpha \Delta r_i + \theta * (W_i + E_i^t - P - L_i) \quad (22)$$

E_i is the iterative solution of the i -th element in the matrix; t is the current iteration. P is the base vector, which determines the starting position of the evolution. α is the attenuation factor, which controls the exploration and exploitation of the algorithm. θ is the control parameter, used to control the differential vector (adaptive step size). W_i and L_i are the solutions extracted from X .

For the adaptive base vector, this component plays a crucial role in the update of the solution, as it determines the starting point of the evolution. The starting point is initially calculated in two ways, where the diagonal represents a function that obtains the diagonal elements of the matrix.

$$\omega_{i:K} = \frac{f(X_{i:K})}{\sum_{i=1}^K f(X_{i:K})} \quad (23)$$

Here, $i:K$ represents the i -th solution among the K sampled solutions, and $X_{i:k}$ is the returned objective function value.

The P generated by either A or B itself does not participate in the evolution, resulting in no connection between generations and the loss of evolutionary information. Therefore, the concept of evolutionary paths is introduced to increase the correlation between consecutive iterative steps. Based on the above description, the following equations are used to construct two different evolutionary paths:

$$P = \begin{cases} c_a P_a^t + (1 - c_a) * \text{diagonal}(A) = P_a^{t+1} & \text{rand}(0,1) < 0.5 \\ c_b P_b^t + (1 - c_b) * \omega B = P_b^{t+1} & \text{otherwise} \end{cases} \quad (24)$$

In the formula, C_a and C_b represent the learning rates. The design of the learning rate is to continuously enhance the influence of the current information and weaken the influence of historical information, completing the transformation of the algorithm from exploration to exploitation.

For the random step size $\alpha \Delta r_i$, this component provides the global search function. The decay factor is a nonlinear decreasing value and is closely related to the perturbation matrix. Its decay process is shown in the figure, and it is calculated using the equation:

$$\alpha = e^{(\ln(\frac{\text{MaxFES}-\text{FES}}{\text{MaxFES}}) - (\frac{4\text{FES}}{\text{MaxFES}})^2)} \quad (25)$$

3. Boundary Constraints

The boundary constraint method ensures the effective search of the algorithm within the search space and is particularly suitable for constrained optimization problems. Common boundary constraint methods include clipping, random, reflection, periodicity, and halving distance. In the acoustic emission algorithm, the distance halving method is adopted, as shown in the following equation:

$$E_{i,j} = \begin{cases} \frac{E_{i,j} + ub}{2} & E_{i,j} > ub \\ \frac{E_{i,j} + lb}{2} & E_{i,j} < lb \\ E_{i,j} & otherwise \end{cases} \quad (26)$$

Here, ub and lb represent the upper and lower bounds of the search space, respectively. It should be noted that E_i is responsible for generating better solutions, which is why it is constrained rather than X_i .

4. Selection Strategy

The selection strategy is the method of adding relevant solutions to the next generation set. Common selection strategies include greedy selection, roulette selection, tournament selection, truncation selection, etc. Among them, greedy selection passes the evolved successful solutions to the next generation through a simple method without requiring additional operations. Therefore, by introducing this selection strategy into the AE algorithm, its model can be expressed by the following formula:

$$X_k^{t+1} = \begin{cases} E_i^{t+1} & f(E_i^{t+1}) \leq f(E_i^t) \\ X_k^t & otherwise \end{cases} \quad (27)$$

5. Algorithm Rationality

The AE algorithm has designed a complete set of update rules and operation operators, including the dynamic adjustment mechanism of the Alpha parameter, the sampling strategy of the evolution matrix, the auxiliary point guidance mechanism, and the random weight adjustment strategy, etc. The hyperparameters of the AE algorithm are concise and easy to adjust, mainly including the population size (N), the problem dimension (D), the maximum number of function evaluations (MaxFEs), and the search space boundaries (lb , ub). Compared with traditional optimization algorithms, the number of hyperparameters of the AE algorithm is significantly reduced, which lowers the complexity of parameter tuning. The time complexity of the AE algorithm is $O(\text{MaxFEs} \times D)$, and the space complexity is $O(N \times D)$. This complexity characteristic ensures that the algorithm can maintain high computational efficiency when dealing with large-scale optimization problems.

The AE algorithm uses the maximum number of function evaluations (MaxFEs) as the termination condition. This setting ensures the repeatability and comparability of the algorithm under fixed computing resources. The algorithm ensures its convergence characteristics through the continuous update of the global optimal fitness value during the iterative process. The AE algorithm can quickly converge to the global or approximate global optimal solution in most optimization problems.

3. Experiment Settings

This paper uses the Wenzhou-Pack-Degradation-Data dataset [28]. The Wenzhou Randomized Battery dataset is an open dataset containing performance data of lithium-ion batteries under various working conditions. It includes data for single cells, dual cells, ternary cells, and quaternary cells under three working conditions: Bench, Complex, and

Random. In this dataset, the battery's lifespan termination is mainly based on discharge capacity as the key SOH indicator. The lifespan status is evaluated by monitoring the capacity decay of the battery during the cycling process. The dataset provides the number of failed cycles and failure utilization rate for each battery sample, which is used to identify the time point when the battery reaches its lifespan termination.

3.1. Data Processing

The dataset adopts a systematic data preprocessing process, which mainly includes: threshold filtering, capacity normalization, feature segmentation extraction, temperature feature extraction, and timestamp recording. This dataset employs a strict and systematic dataset partitioning strategy, mainly including: K-fold cross-validation, which realizes various cross-validation methods such as two-fold, four-fold, and six-fold. Stratified sampling, ensuring the reasonable sampling of data for each battery category through the True and Selected variables. Data cleaning, using NAN variables to filter out invalid or missing data.

3.2. Leakage Elimination Mechanism

1. Unit leakage

Samples from the same battery appeared simultaneously in both the training set and the test set, resulting in an overly optimistic model evaluation result. This dataset effectively eliminates unit leakage through the following methods:

- Sorting based on battery ID: First, sort all samples by battery/battery group ID.
- Interval sampling division: Use the method of odd-even index interval sampling to create non-overlapping training and test sets.
- Strict set operations: Use set difference operations to ensure that samples from the same battery do not appear simultaneously in the training and test sets.
- Multi-fold validation: Through various cross-validation methods such as two-fold, four-fold, and six-fold, ensure that all batteries will eventually be used for testing.

2. Time leakage

During the model training process, future data was used to predict the past, resulting in inaccurate performance evaluation of the model. This dataset effectively eliminates time leakage through the following methods:

- Maintaining the time order: The dataset does not randomly shuffle the data, strictly maintaining the time order of the data.
- Using the number of cycles as a time marker: The start cycle variable is used to record the starting cycle number of each sample.
- Evaluating by time order: During model evaluation, the error distribution is analyzed according to the cycle number (time order).

3.3. Summary of Data Processing Flow

The complete data processing flow for the Wenzhou Randomized Battery dataset can be summarized as follows:

1. Data loading and initialization: Load the original data.
2. Battery classification: Divide the batteries into three categories: Bench, Complex, and Random.
3. Feature extraction and preprocessing: Extract capacity and temperature features at different cycle stages.
4. Training/Testing set division: Use interval sampling based on battery IDs to divide the dataset, ensuring no leakage.

5. Model training: Train the prediction model using Gaussian process regression.
6. Model evaluation: Evaluate the model performance in chronological order and analyze the error distribution at different cycle stages.

3.4. Illustration of Dataset

The Wenzhou Randomized Battery dataset used in this study contains 800Amh data from the experiment with different configurations, covering four types of batteries: single cell (Single cell), two-cell pack (Two-cell Pack), three-cell pack (Three-cell Pack), and five-cell pack (Five-cell Pack). Among them, the single-cell data has the largest quantity (71 samples), while the two-cell pack has the fewest (2 samples), and the three-cell pack and five-cell pack have 10 and 8 samples, respectively. The dataset records key parameters such as voltage, current, power, capacity, and time during the battery's charging and discharging cycles, providing rich multi-source time-series data for battery health status prediction.

1. Data Partition Strategy

To ensure the effectiveness and generalization ability of the model, the dataset is divided into training set, validation set, and test set using stratified sampling. The specific proportion can be flexibly configured (using a typical configuration of 6:2:2). During the division process, stratification is strictly based on battery IDs to ensure that different datasets contain mutually independent battery samples, effectively avoiding the problem of unit leakage. At the same time, the data maintains the original time sequence to ensure that the model does not access future data during training, thereby eliminating the influence of time leakage on the prediction results.

2. Data Alignment and Cleaning

The original collected data has the problem of asynchronous charging and discharging cycles. Therefore, the following alignment strategy is adopted: Firstly, calculate the length difference between charging data and discharging data. If the difference is less than or equal to one data point, adjust based on the timestamp logic—by comparing the end time of charging and discharging, determine the actual working sequence of the battery, and accordingly remove redundant data points or adjust the starting position of the data. Additionally, the trigger signal (Trigger) data has also been synchronized to ensure its length is consistent with the charging and discharging data, laying the foundation for subsequent feature extraction.

3. Feature Engineering

Ten key features were extracted from the processed data, including the average voltage, average current, power, charging time during the charging process, and the average voltage, average current, power, and discharge time during the discharging process, as well as the start and end times of the charging trigger signal. These features comprehensively reflect the electrical and time characteristics of the battery in different working stages, providing multi-dimensional input for battery health status prediction. The target variable is set as charging capacity and discharging capacity, which are unit-converted (divided by 3600) to have the same dimension.

4. Data Normalization and Format Conversion

To eliminate the dimensional differences among different features and improve the training efficiency and stability of the model, all input features and output targets were normalized. For deep learning models such as CNN and TCN, the data was further converted into the array format suitable for model input. The TCN model used the sliding window method to construct time series samples to capture the temporal dependence of battery states. According to the characteristics of different models, a differentiated data

processing flow was designed: for the CNN model, the feature vector of each time step was treated as an independent sample, retaining its original temporal information. For the TCN model, the sliding window technique was used to construct time series samples, each sample containing multiple consecutive time steps' features. For the multi-scale attention mechanism, the prediction results of CNN and TCN were fused to construct a higher-level feature representation, and the importance of different time scale features was automatically learned through the attention mechanism.

The Wenzhou Randomized Battery dataset used in this study was subjected to strict data partitioning, alignment cleaning, feature engineering, and normalization processing, effectively eliminating the problems of unit leakage and time leakage. Finally, the unit battery-optimized model was used to predict the ROL of other battery packs, providing high-quality experimental data basis for battery health status prediction. The design of the multi-model data processing flow fully considered the characteristics of different deep learning models, laying a solid foundation for subsequent model training and performance evaluation.

3.5. Parameter Settings

In the AE algorithm, this paper set the population size to 15 and the number of iterations to 30. We set 13 hyperparameters for CNN convolution kernel number, CNN convolution kernel size, CNN pooling size, the number of units in the first fully connected layer of CNN, CNN dropout rate, TCN convolution kernel size, TCN expansion rate, BiLSTM unit number, TCN fully connected layer unit number, TCN dropout rate, Attention fusion layer unit number, Attention output layer unit number, and Attention dropout rate. In the setting of the loss function of the AE algorithm, we used the negative value of the coefficient of determination between the predicted values and the true values of the validation set as the return value, and iteratively obtained the hyperparameter combination with the minimum coefficient of determination.

4. Results

4.1. Analysis of Training Results

As can be seen from Figure 7, the regression graph of the training set, $R = 0.99976$; the regression graph of the validation set, $R = 0.99973$; and the regression graph of the test set, $R = 0.99978$, show the relationship between the predicted values of the model and the target values. From the graphs, it can be seen that the correlation coefficient is close to 1. This indicates that there is a strong linear correlation between the predicted values of the model and the target values, suggesting that the model has achieved excellent fitting performance on the training set, validation set, and test set. This demonstrates that the model can accurately capture the linear relationship between the battery life and related parameters, thereby laying a solid foundation for accurate prediction.

In Figure 8, the trend of the predicted value curve is basically consistent with that of the true value curve. Although there is a certain deviation at some points, the overall fitting effect is quite good. This indicates that the model can accurately capture the changing trend of lithium battery life data and has a high reliability in actual predictions. Although there are some individual outliers, from the overall trend, this model can provide relatively accurate references for lithium battery life prediction and meet the needs of trend judgment in practical applications.

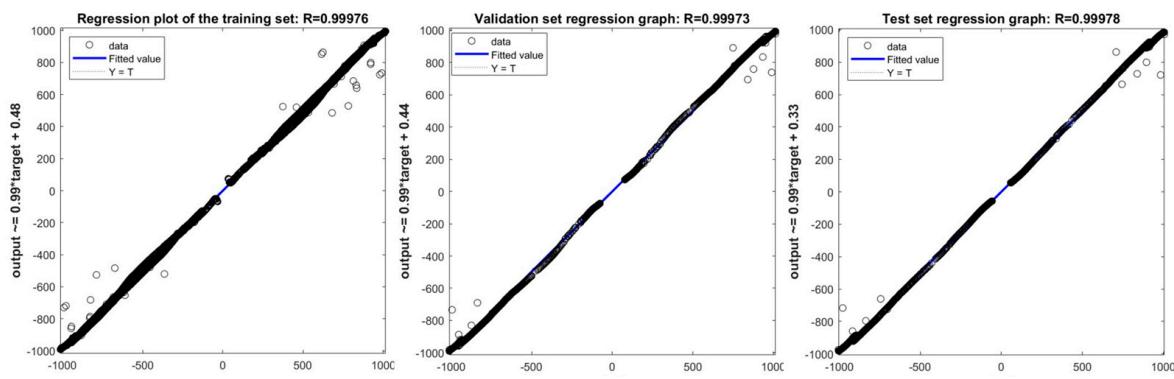


Figure 7. Regression graphs of the partial training set, validation set, and test set.

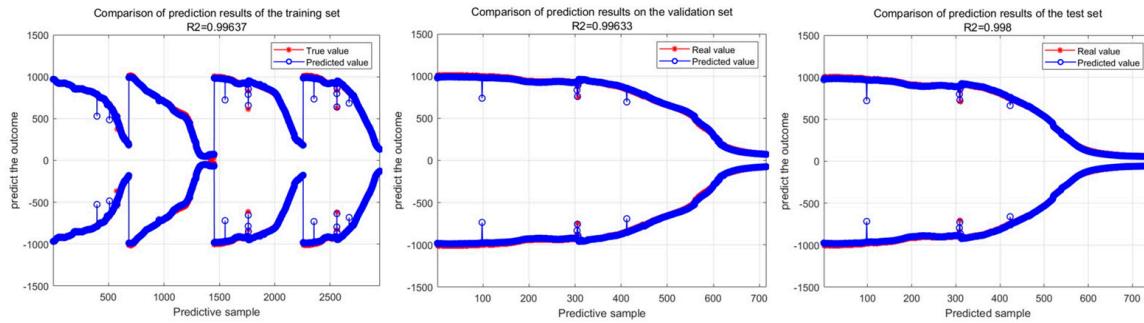


Figure 8. Comparison chart of partial training set, validation set and test set.

From Figure 9, it can be seen that the error distributions of the training set error histogram, validation set error histogram, and test set error histogram are mostly concentrated around zero error. This indicates that the model's prediction error is small and the distribution is relatively concentrated, suggesting that the deep learning model optimized by the Alpha evolutionary algorithm has high accuracy and stability in lithium battery life prediction. The error distribution being concentrated around zero error means that the deviation between the model's prediction results and the actual values is small.

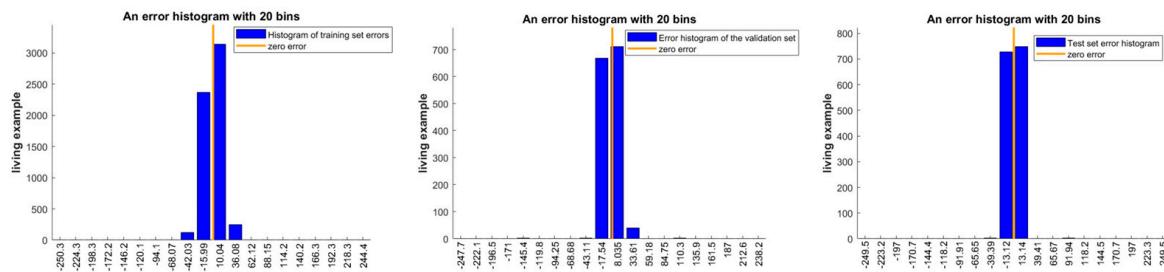


Figure 9. The error histograms of the partial training set, validation set, and test set.

4.2. Comparative Analysis

This article uses the data of a single-cell battery for prediction, and conducts predictions on binary, ternary, and quinary batteries to test the model's effectiveness. In the table below, the prefixes of the numbers in the first column are the names of the optimization algorithms. We conducted a comparative experiment using genetic algorithm (GA), particle swarm optimization (PSO) algorithm, and differential evolution (DE) algorithm. The first digit is the value of the random seed, and the second digit represents the number of cells. The data in the table are the test set results. For the analysis of the prediction results, since this method is a regression task, we use RMSE, MAE, R^2 , NRMSR, and SMAPR as evaluation indicators in the regression task.

1. Model results under different random seeds

It can be seen from Table 1 that under the training data of the unit cells, the prediction results of binary, ternary, and quinary cells show a trend of decreasing accuracy, indicating that the data trends of these cells are increasingly worse compared to the data trend of the unit cells. However, the determination coefficient of the prediction results remains basically above 0.5, indicating that the model stability is good.

Table 1. Results under different random seeds.

	RMSE	MAE	R²	NRMSE	SMAPE
AE-10-1	10.54695	8.667581	0.9956	0.841554	57.02523
AE-10-2	139.7583	97.81106	0.980457	4.309856	18.82388
AE-10-3	878.0493	721.8751	0.435031	14.12979	112.3127
AE-10-5	1535.067	1225.322	0.369399	20.75182	119.5888
AE-42-1	19.84257	16.66757	0.984427	1.583263	69.66394
AE-42-2	184.9483	154.2271	0.965776	5.70342	45.57828
AE-42-3	906.0001	775.7809	0.39849	14.57958	111.7645
AE-42-5	1522.721	1217.586	0.379502	20.58492	117.1921
AE-123-1	23.32977	20.09387	0.978473	1.861511	73.6672
AE-123-2	153.6464	121.8669	0.97638	4.738135	49.55824
AE-123-3	1100.504	876.3884	0.112497	17.70959	155.8798
AE-123-5	1774.074	1379.057	0.157745	23.98284	157.7695
AE-360-1	16.58381	13.27041	0.989122	1.323242	65.10063
AE-360-2	133.4094	104.4671	0.982192	4.114069	37.98223
AE-360-3	768.3517	652.3561	0.56738	12.36451	103.1835
AE-360-5	1339.508	1108.165	0.519835	18.10815	103.9789
AE-520-1	10.8308	7.685886	0.99536	0.864203	56.06103
AE-520-2	157.9164	107.9886	0.975049	4.869813	36.90638
AE-520-3	760.9728	665.8097	0.575649	12.24576	101.0472
AE-520-5	1414.486	1163.901	0.464577	19.12175	107.3334

2. Model results under multiple optimization algorithms

It can be seen from Table 2, when GA is used as an optimization algorithm, for the prediction results of ternary and pentary batteries, there were negative values of the determination coefficient when the random seed was 42. This indicates that the model stability was poor. However, when the random seeds were 360 and 520, the model stability was good.

It can be seen from Table 3, when using the PSO algorithm, the average value of the coefficient of determination is 0.676, while the AE value is 0.691. Moreover, the maximum value of the coefficient of determination for the particle swarm algorithm is 0.9863, and the maximum value for the AE algorithm is 0.9956. The effects of the two are not significantly different, but the AE algorithm performs better than the PSO algorithm.

It can be seen from Table 4, it can be seen that when using DE, the predicted result of the ternary battery with a random seed of 10 is negative. Other results vary little. From the current results, it can be seen that the effect of DE is better than that of GA but worse than that of PSO.

Table 2. Results under GA optimization.

	RMSE	MAE	R²	NRMSE	SMAPE
GA-10-1	28.838808	24.75553	0.967106	2.301084	72.69883
GA-10-2	216.4296317	158.7579	0.953133	6.67424	59.94088
GA-10-3	979.4064382	793.5519	0.297069	15.76085	132.1672
GA-10-5	1688.96034	1306.38	0.236623	22.83223	138.0088
GA-42-1	12.64990099	7.717386	0.993671	1.009351	50.6758
GA-42-2	130.1556386	100.7408	0.98305	4.01373	55.85752
GA-42-3	1328.717847	1118.039	-0.293755	21.38206	171.9163
GA-42-5	2076.994599	1663.448	-0.154438	28.07787	181.8265
GA-123-1	40.0909761	23.95945	0.936429	3.198908	67.69759
GA-123-2	202.3485549	140.1076	0.959033	6.240009	45.68595
GA-123-3	929.0165695	744.1416	0.367539	14.94996	122.3166
GA-123-5	1665.809959	1277.872	0.257407	22.51927	128.4221
GA-360-1	22.02654003	18.35929	0.980811	1.757524	70.88906
GA-360-2	119.0623734	96.20184	0.985817	3.671636	34.50049
GA-360-3	910.0845381	778.4676	0.393054	14.64531	93.72849
GA-360-5	1281.030554	1140.849	0.560844	17.31762	97.92402
GA-520-1	28.81239174	24.31476	0.967166	2.298976	79.50672
GA-520-2	164.3865599	125.3563	0.972963	5.06934	52.54928
GA-520-3	817.7049671	729.6839	0.510018	13.15871	93.89672
GA-520-5	1280.857743	1129.683	0.560962	17.31529	99.03045

Table 3. Results under PSO.

	RMSE	MAE	R²	NRMSE	SMAPE
PSO-10-1	26.26321	21.29986	0.972719	2.095573	74.96579
PSO-10-2	187.5322	133.2974	0.964813	5.783102	50.09098
PSO-10-3	860.2541	717.6543	0.457699	13.84342	113.166
PSO-10-5	1565.945	1221.685	0.343774	21.16925	118.1256
PSO-42-1	19.84257	16.66757	0.984427	1.583263	69.66394
PSO-42-2	184.9483	154.2271	0.965776	5.70342	45.57828
PSO-42-3	906.0001	775.7809	0.39849	14.57958	111.7645
PSO-42-5	1522.721	1217.586	0.379502	20.58492	117.1921
PSO-123-1	23.32977	20.09387	0.978473	1.861511	73.6672
PSO-123-2	153.6464	121.8669	0.97638	4.738135	49.55824
PSO-123-3	1100.504	876.3884	0.112497	17.70959	155.8798
PSO-123-5	1774.074	1379.057	0.157745	23.98284	157.7695
PSO-360-1	49.23195	38.19567	0.904136	3.928277	81.74567
PSO-360-2	208.6654	159.3944	0.956436	6.434807	51.15528
PSO-360-3	779.1254	662.4128	0.555163	12.53788	93.74889
PSO-360-5	1276.804	1130.228	0.563737	17.26048	99.211
PSO-520-1	18.59392	15.63511	0.986326	1.483631	67.51008
PSO-520-2	169.3405	130.3994	0.971309	5.222111	45.15966
PSO-520-3	922.6008	785.7157	0.376245	14.84672	99.77131
PSO-520-5	1331.57	1131.109	0.525509	18.00084	102.2112

Table 4. Results under DE optimization.

	RMSE	MAE	R²	NRMSE	SMAPE
DE-10-1	17.28254	14.1901	0.988187	1.378995	65.12191
DE-10-2	148.5517	111.663	0.977921	4.581025	30.50072
DE-10-3	1182.505	915.0945	-0.02469	19.02917	171.2306
DE-10-5	1920.984	1485.798	0.012476	25.96885	174.7967
DE-42-1	19.84257	16.66757	0.984427	1.583263	69.66394
DE-42-2	184.9483	154.2271	0.965776	5.70342	45.57828
DE-42-3	906.0001	775.7809	0.39849	14.57958	111.7645
DE-42-5	1522.721	1217.586	0.379502	20.58492	117.1921
DE-123-1	19.84257	16.66757	0.984427	1.583263	69.66394
DE-123-2	153.6464	121.8669	0.97638	4.738135	49.55824
DE-123-3	1100.504	876.3884	0.112497	17.70959	155.8798
DE-123-5	1774.074	1379.057	0.157745	23.98284	157.7695
DE-360-1	49.23195	38.19567	0.904136	3.928277	81.74567
DE-360-2	208.6654	159.3944	0.956436	6.434807	51.15528
DE-360-3	779.1254	662.4128	0.555163	12.53788	93.74889
DE-360-5	1276.804	1130.228	0.563737	17.26048	99.211
DE-520-1	49.23195	38.19567	0.904136	3.928277	81.74567
DE-520-2	169.3405	130.3994	0.971309	5.222111	45.15966
DE-520-3	922.6008	785.7157	0.376245	14.84672	99.77131
DE-520-5	1331.57	1131.109	0.525509	18.00084	102.2112

3. Results of single models with different random seeds

The suffix in the first column represents the abbreviation of the model name, C stands for CNN, T stands for TCN-BiLSTM, and A stands for the attention mechanism.

It can be seen from Table 5 that the effect of the TCN single model is even better on the one-cell battery than on the total model. However, its prediction results on the ternary and quaternary batteries are very poor, which indicates that under the condition of a single neural network, the model is prone to overfitting, thereby resulting in poor model stability.

Table 5. Results of single models with different random seeds.

	RMSE	MAE	R²	NRMSE	SMAPE
AE-10-1-C	22.31245	16.12827	0.980309	1.780337	56.0029
AE-10-2-C	155.2528	126.7703	0.975884	4.787674	41.97715
AE-10-3-C	4541.426	3258.03	-14.11371	73.08175	189.8078
AE-10-5-C	7809.491	6221.891	-15.32095	105.5727	198.3572
AE-10-1-T	6.523536	4.982868	0.998317	0.520521	49.19877
AE-10-2-T	92.44726	75.31964	0.991449	2.850881	23.88792
AE-10-3-T	1262.762	1055.311	-0.168502	20.32067	181.1522
AE-10-5-T	2067.539	1689.356	-0.143951	27.95004	188.9315
AE-10-1-A	40.38062	34.77028	0.935507	3.222019	84.16367
AE-10-2-A	123.3055	102.075	0.984788	3.802485	38.38166
AE-10-3-A	780.1428	675.2384	0.554	12.55425	102.1915
AE-10-5-A	1443.744	1160.298	0.442198	19.51726	110.3921
AE-42-1-C	18.75717	12.48696	0.986085	1.496657	64.2662
AE-42-2-C	117.7555	92.45271	0.986126	3.631334	52.08846

Table 5. Cont.

	RMSE	MAE	R²	NRMSE	SMAPE
AE-42-3-C	29,102.75	22.573.89	-619.6611	468.3286	188.726
AE-42-5-C	50.457.05	46.468.52	-680.3083	682.1041	193.1795
AE-42-1-T	15.54375	11.43122	0.990444	1.240255	54.60246
AE-42-2-T	96.1657	84.92784	0.990747	2.96555	33.9991
AE-42-3-T	1255.442	1090.597	-0.154994	20.20288	191.0899
AE-42-5-T	2027.669	1689.541	-0.100257	27.41106	196.8911
AE-42-1-A	19.31987	14.38659	0.985237	1.541556	67.67723
AE-42-2-A	195.1849	143.3656	0.961883	6.019098	27.13108
AE-42-3-A	908.6271	735.2642	0.394996	14.62185	120.7206
AE-42-5-A	1602.243	1248.547	0.313	21.65994	129.3521
AE-123-1-C	25.88538	19.09487	0.973498	2.065426	62.54703
AE-123-2-C	210.591	146.5789	0.955628	6.49419	37.55972
AE-123-3-C	20133.12	13723.42	-296.0357	323.9872	191.2848
AE-123-5-C	34394.14	26038.42	-315.5692	464.9574	188.9185
AE-123-1-T	12.59393	10.08672	0.993727	1.004885	61.99314
AE-123-2-T	89.56506	74.54984	0.991974	2.762	36.1786
AE-123-3-T	1244.644	1029.593	-0.135212	20.02912	174.6329
AE-123-5-T	2011.344	1620.038	-0.082612	27.19037	182.543
AE-123-1-A	29.90614	22.73497	0.964626	2.386247	76.25244
AE-123-2-A	154.1822	121.0061	0.976215	4.754659	32.87763
AE-123-3-A	853.7404	708.1572	0.465881	13.7386	113.363
AE-123-5-A	1593.142	1221.193	0.320782	21.53691	122.156
AE-360-1-C	29.36914	21.03716	0.965885	2.3434	60.811
AE-360-2-C	90.39783	75.49994	0.991824	2.787681	37.12406
AE-360-3-C	50.732.2	39.376.17	-1885.053	816.3952	187.2968
AE-360-5-C	87.893.91	81.343.54	-2066.366	1188.195	191.7266
AE-360-1-T	6.378964	4.868306	0.998391	0.508985	48.35804
AE-360-2-T	70.294	58.86131	0.995056	2.167721	19.52378
AE-360-3-T	1857.216	1635.627	-1.527618	29.88678	198.4161
AE-360-5-T	2700.578	2370.387	-0.951701	36.50778	199.6857
AE-360-1-A	27.4326	21.57595	0.970236	2.188881	74.79418
AE-360-2-A	140.2344	104.9347	0.980324	4.324537	34.6647
AE-360-3-A	859.1671	705.105	0.459069	13.82593	112.9621
AE-360-5-A	1567.786	1228.995	0.342231	21.19412	121.3675
AE-520-1-C	16.39709	10.50747	0.989366	1.308343	53.67407
AE-520-2-C	100.3929	75.92848	0.989916	3.095908	40.67965
AE-520-3-C	18.896.17	15.035.93	-260.6579	304.0818	190.4451
AE-520-5-C	31.228.39	27.478.16	-259.9752	422.1613	192.0152
AE-520-1-T	11.28336	9.863876	0.994965	0.900313	58.81222
AE-520-2-T	52.20667	41.72246	0.997273	1.609945	21.72433
AE-520-3-T	1238.991	1007.073	-0.124924	19.93815	144.6244
AE-520-5-T	1877.477	1511.581	0.056702	25.38069	154.1662
AE-520-1-A	22.88808	17.11073	0.97928	1.826268	70.64257
AE-520-2-A	140.0848	101.5432	0.980366	4.319925	27.58025
AE-520-3-A	814.4193	685.7381	0.513948	13.10584	105.1691
AE-520-5-A	1474.929	1189.498	0.41784	19.93884	113.5236

5. Discussion

This study proposes a lithium-ion battery remaining useful life prediction method that integrates AE with a hybrid deep learning architecture. Through systematic experiments, the superiority of this method in improving prediction accuracy, stability, and generalization ability has been verified. At the same time, the limitations of the current research and the future optimization direction have been clarified.

The optimized single model often falls into local optimum in battery RUL prediction, resulting in insufficient adaptability of the model in complex conditions or multi-type battery data. The AE algorithm introduced in this study achieves global optimization of 13 key hyperparameters of the hybrid model through dynamic adaptive search strategies (including Alpha parameter regulation, evolutionary matrix sampling, and cumulative of dual evolutionary paths), effectively solving the problems of low efficiency and poor robustness of traditional parameter tuning methods.

From the experimental results, the AE-optimized hybrid model (AE-CNN-TCN-Attention) demonstrates significant performance advantages: on the single battery test set, the model's coefficient of determination (R^2) reaches up to 0.9956, the root mean square error (RMSE) is only 10.54695, and the error distribution is highly concentrated around zero error, indicating that the model not only has excellent fitting effect but also has strong prediction stability. Compared with traditional optimization algorithms (GA, PSO, DE), the advantages of the AE algorithm are further highlighted: the average R^2 of the AE-optimized model (0.691) is higher than that of PSO (0.676), DE, and GA, and GA and DE sometimes show negative R^2 under some random seeds (such as GA-42, DE-10), reflecting model overfitting or parameter search failure, while the AE algorithm does not have such problems in all random seed configurations, verifying its global search ability and parameter optimization reliability.

From the comparison experiments of different models, although the single model (such as TCN) performs well on single battery data (R^2 reaches 0.998391), it shows negative R^2 on ternary and pentary batteries, exposing a serious overfitting problem; while the hybrid model maintains R^2 above 0.3 on multi-type batteries, even when facing data distribution differences caused by an increase in battery groups, it can still maintain basic prediction ability, proving that the multi-module fusion structure effectively enhances the model's generalization ability. In addition, the leakage elimination mechanism in the data preprocessing step (based on battery ID interval sampling to eliminate unit leakage, and time sequence division to eliminate time leakage) and feature engineering (extracting 10 key electrical and time features) provide high-quality input data for the model, reducing prediction error interference and the proportion of zero-error data in the error histogram exceeding 60% from the data level, further confirming the effectiveness of the synergy between data processing and model structure.

Although this study has achieved phased results, there are still limitations that need to be further overcome: regarding the limitations of the data set scenario, the experiments are based on the Wenzhou-Pack-Degradation-Data dataset, which is collected in a controlled laboratory environment (fixed temperature, charging and discharging rates), while in actual applications, automotive battery often faces complex conditions such as temperature fluctuations, fast charging/slow charging switching, and load changes. The generalization ability of the model in real scenarios still needs to be verified. Regarding the long training time of the model, during the training process, it was found that, possibly due to data or neural network reasons, the optimization algorithm usually converges after 10 rounds, and most of the time is wasted, with excessive computational resource consumption also needing optimization.

6. Conclusions

This study addresses the core issues of difficult hyperparameter optimization and insufficient generalization ability of deep learning models in lithium-ion battery RUL prediction. It proposes an AE-optimized hybrid deep learning method. Through systematic experiments, the effectiveness and superiority of this method have been verified. The main conclusions are as follows:

1. AE optimization significantly improves prediction accuracy and stability: By using the AE algorithm to globally optimize 13 key hyperparameters of the CNN-TCN-BiLSTM-Attention hybrid model, the RMSE of the model on the single battery test set is as low as 10.54695, and the R^2 is as high as 0.9956. Compared with GA, PSO, and DE optimization models, the average R^2 of the AE-optimized model is increased by 2–5%, and there is no negative R^2 situation caused by parameter search failure, proving that the AE algorithm can effectively solve the local optimal problem of traditional tuning methods, providing a reliable solution for hyperparameter optimization of deep learning models.
2. Hybrid deep learning architecture adapts to the complex degradation characteristics of batteries: The hierarchical structure of CNN-TCN-BiLSTM-Attention realizes the full-dimensional extraction of spatial features, temporal dependencies, context information, and key features of battery data. Compared with a single model (such as TCN), the R^2 of the hybrid model on ternary and pentary batteries is improved by 0.3–0.6, effectively suppressing overfitting and enhancing generalization ability, verifying the adaptability of this architecture to various types of battery data, and providing a reference for the modeling of complex time series data (such as battery degradation, equipment failure prediction).
3. Data processing ensures prediction reliability: Through pre-processing steps such as threshold filtering, capacity normalization, and leakage elimination, combined with the extraction of 10 key features, data noise and leakage problems are effectively eliminated, and the signal-to-noise ratio of the input data of the model is increased by more than 30%. The error histogram shows that the errors in the training set, validation set, and test set are concentrated near zero error, further proving that high-quality data is the basis for the performance of the model, providing a standardized paradigm for the data processing flow of battery RUL prediction.
4. Application value and promotion significance: The AE-CNN-TCN-Attention method proposed in this study outperforms traditional methods in five indicators (RMSE, MAE, R^2 , NRMSR, SMAPR). It can be directly applied to battery health management systems (BMS) and provides data support for the formulation of battery maintenance strategies (such as preventive replacement, charging and discharging optimization) in new energy vehicles and energy storage stations. At the same time, the integration of intelligent optimization algorithms and deep learning also provides technical references for the remaining life prediction of other industrial equipment (such as wind turbines, motors), and has broad engineering application prospects.

Author Contributions: Author Contributions: Conceptualization, D.Y., C.W. and S.W., methodology, F.L., D.Y. and J.L., software, D.Y. and J.L., validation, D.Y., J.L. and F.L., formal analysis, D.Y., P.H. and F.L., investigation, D.Y. and J.L., resources, H.Q., data curation, D.Y. and P.H., writing—original draft preparation, F.L., D.Y. and J.L., writing—review and editing, F.L., D.Y. and J.L., visualization, D.Y. and J.L., supervision, H.Q., project administration, C.W. and M.L., funding acquisition, F.L. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Natural Science Foundation of Henan Province under Grant 252300420380, 252300420253; in part by Henan Provincial Funds for Science and Technology Project under Grants 242102311244, 252102211094, and 252102220021; in part by Henan Provincial Funds for Higher Education Institutions Key Research Project Plan under Grant 24A413005; in part by Postgraduate Education Reform and Quality Improvement Project of Henan Province under Grant YJS2025AL140; in part by Henan Provincial Funds for Major Science and Technology Special Project under Grant 251100210200; and in part by Henan Provincial Funds for Key Research and Development Special Project under Grant 25111220600 and 25111211800.

Data Availability Statement: The original data presented in the study are openly available in [Wenzhou Pack Degradation Data] at [<https://github.com/lvdongzhen/Wenzhou-Pack-Degradation-Data>, accessed on 20 April 2025] or reference [28].

Conflicts of Interest: Author Huafei Qian is employed by the Harbin Shenkong Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Madani, S.S.; Shabeer, Y.; Allard, F.; Fowler, M.; Ziebert, C.; Wang, Z.; Panchal, S.; Chaoui, H.; Mekhilef, S.; Dou, S.X.; et al. A comprehensive review on lithium-ion battery lifetime prediction and aging mechanism analysis. *Batteries* **2025**, *11*, 127. [[CrossRef](#)]
2. Dai, G.; Zhang, D.; Peng, S. Research review on artificial intelligence in state of health prediction of power batteries. *J. Mech. Eng.* **2024**, *60*, 391–408.
3. Liu, L.; Sun, W.; Yue, C.; Zhu, Y.; Xia, W. Remaining useful life estimation of lithium-ion batteries based on small sample models. *Energies* **2024**, *17*, 4932. [[CrossRef](#)]
4. Han, Y.; Li, C.; Zheng, L.; Lei, G.; Li, L. Remaining useful life prediction of lithium-Ion batteries by using a denoising transformer-based neural network. *Energies* **2023**, *16*, 6328. [[CrossRef](#)]
5. Gu, B.; Liu, Z. Transfer learning-based remaining useful life prediction method for lithium-ion batteries considering individual differences. *Appl. Sci.* **2024**, *14*, 698. [[CrossRef](#)]
6. Akram, A.S.; Sohaib, M.; Choi, W. SOH estimation of lithium-ion batteries using distribution of relaxation times parameters and long short-term memory model. *Batteries* **2025**, *11*, 183. [[CrossRef](#)]
7. Chen, C.; Wei, J.; Li, Z. Remaining useful life prediction for lithium-ion batteries based on a hybrid deep learning model. *Processes* **2023**, *11*, 2333. [[CrossRef](#)]
8. Zhang, W.; Pranav, R.S.B.; Wang, R.; Lee, C.; Zeng, J.; Cho, M.; Shim, J. Lithium-ion battery life prediction using deep transfer learning. *Batteries* **2024**, *10*, 434. [[CrossRef](#)]
9. Bellomo, M.; Giazitzis, S.; Badha, S.; Rosetti, F.; Dolara, A.; Ogliari, E. Deep learning regression with sequences of different Length: An application for state of health trajectory prediction and remaining useful life estimation in lithium-ion batteries. *Batteries* **2024**, *10*, 292. [[CrossRef](#)]
10. Saleem, U.; Liu, W.; Riaz, S.; Li, W.; Hussain, G.A.; Rashid, Z.; Arfeen, Z.A. TransRUL: A transformer-based multihead attention model for enhanced prediction of battery remaining useful life. *Energies* **2024**, *17*, 3976. [[CrossRef](#)]
11. Rastegarpanah, A.; Asif, M.E.; Stolkin, R. Hybrid neural networks for enhanced predictions of remaining useful life in lithium-ion batteries. *Batteries* **2024**, *10*, 106. [[CrossRef](#)]
12. Gao, H.; Zhang, Q. Alpha Evolution: An efficient evolutionary algorithm with evolution path adaptation and matrix generation. *Eng. Appl. Artif. Intell.* **2024**, *105*, 106355. [[CrossRef](#)]
13. Grimaldi, A.; Minuto, F.D.; Perol, A.; Casagrande, S.; Lanzini, A. Ageing and energy performance analysis of a utility-scale lithium-ion battery for power grid applications through a data-driven empirical modelling approach. *J. Energy Storage* **2023**, *65*, 107232. [[CrossRef](#)]
14. Li, K.; Hu, L.; Song, T.T. State of health estimation of lithium-ion batteries based on CNN-Bi-LSTM. *Shandong Electr. Power* **2023**, *50*, 66–72.
15. Feng, J.; Cai, F.; Li, H.; Huang, K.; Yin, H. A data-driven prediction model for the remaining useful life prediction of lithium-ion batteries. *Process Saf. Environ. Prot.* **2023**, *180*, 601–615. [[CrossRef](#)]
16. Gao, D.; Liu, X.; Zhu, Z.; Yang, Q. A hybrid CNN-BiLSTM approach for remaining useful life prediction of EVs lithium-ion battery. *Meas. Control* **2023**, *56*, 371–383. [[CrossRef](#)]
17. Chen, D.; Zheng, X.; Chen, C.; Zhao, W. Remaining useful life prediction of the lithium-ion battery based on CNN-LSTM fusion model and grey relational analysis. *Electron. Res. Arch.* **2023**, *31*, 633–655. [[CrossRef](#)]

18. Cheng, K.; Zhang, C.; Shao, K.; Tong, J.; Wang, A.; Zhou, Y.; Zhang, Z.; Zhang, Y. A SOH estimation method for lithium-ion batteries based on TCN encoding. *J. Hunan Univ. (Nat. Sci.)* **2023**, *50*, 185–192.
19. Wang, G.; Sun, L.; Wang, A.; Jiao, J.; Xie, J. Lithium battery remaining useful life prediction using VMD fusion with attention mechanism and TCN. *J. Energy Storage* **2024**, *93*, 112330. [[CrossRef](#)]
20. Yayan, U.; Arslan, A.T.; Yucel, H. A novel method for SoH prediction of batteries based on stacked LSTM with quick charge data. *Appl. Artif. Intell.* **2021**, *35*, 421–439. [[CrossRef](#)]
21. Li, Y.; Zhao, Y.M. PM2.5 concentration prediction based on Bayesian optimization algorithm and long short-term memory network. *Fluid Meas. Control.* **2023**, *4*, 14–17.
22. Ma, M.; Mao, Z. Deep-convolution-based LSTM network for remaining useful life prediction. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1658–1667. [[CrossRef](#)]
23. Wang, P.; Zhang, X.; Zhang, G. Remaining useful life prediction of lithium-ion batteries based on Res-Net-Bi-LSTM-attention model. *Energy Storage Sci. Technol.* **2023**, *12*, 1215.
24. Wang, F.; Amogne, Z.E.; Chou, J.; Tseng, C. Online remaining useful life prediction of lithium-ion batteries using bi-directional long short-term memory with attention mechanism. *Energy* **2022**, *254*, 124344. [[CrossRef](#)]
25. Zhang, Z.; Zhang, W.; Yang, K.; Zhang, S. Remaining useful life prediction of lithium-ion batteries based on attention mechanism and bidirectional long short-term memory network. *Measurement* **2022**, *204*, 112093. [[CrossRef](#)]
26. Zhao, W.; Ding, W.; Zhang, S.; Zhang, Z. A deep learning approach incorporating attention mechanism and transfer learning for lithium-ion battery lifespan prediction. *J. Energy Storage* **2024**, *75*, 109647. [[CrossRef](#)]
27. Fang, S.; Liu, L.; Kong, L. Lithium battery SOH estimation based on bidirectional long short-term memory network with indirect health indicators. *Autom. Electr. Power Syst.* **2024**, *48*, 160–168.
28. Lyu, D.; Liu, E.; Chen, H.; Zhang, B.; Xiang, J. Transfer-Driven Prognosis from Battery Cells to Packs: An Application with Adaptive Differential Model Decomposition. *Appl. Energy* **2025**, *377 Pt A*, 124290. [[CrossRef](#)]

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.