

Article

An Adaptive Framework for Remaining Useful Life Prediction Integrating Attention Mechanism and Deep Reinforcement Learning

Yanhui Bai ^{1,2} , Jiajia Du ¹, Honghui Li ^{1,2,*} , Xintao Bao ¹, Linjun Li ³, Chun Zhang ^{1,2} , Jiahe Yan ¹, Renliang Wang ¹  and Yi Xu ¹

- ¹ School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China; baiyh@bjtu.edu.cn (Y.B.); 22281062@bjtu.edu.cn (J.D.); 22281119@bjtu.edu.cn (X.B.); chzhang1@bjtu.edu.cn (C.Z.); 20112038@bjtu.edu.cn (J.Y.); 21111099@bjtu.edu.cn (R.W.); 23111101@bjtu.edu.cn (Y.X.)
- ² Engineering Research Center of Network Management Technology for High Speed Railway, Ministry of Education, Beijing 100816, China
- ³ China National Energy Co., Ltd., Beijing 100048, China; 11591129@ceic.com
- * Correspondence: hhli@bjtu.edu.cn

Abstract

The prediction of Remaining Useful Life (RUL) constitutes a vital aspect of Prognostics and Health Management (PHM), providing capabilities for the assessment of mechanical component health status and prediction of failure instances. Recent studies on feature extraction, time-series modeling, and multi-task learning have shown remarkable advancements. However, most deep learning (DL) techniques predominantly focus on unimodal data or static feature extraction techniques, resulting in a lack of RUL prediction methods that can effectively capture the individual differences among heterogeneous sensors and failure modes under complex operational conditions. To overcome these limitations, an adaptive RUL prediction framework named ADAPT-RULNet is proposed for mechanical components, integrating the feature extraction capabilities of attention-enhanced deep learning (DL) and the decision-making abilities of deep reinforcement learning (DRL) to achieve end-to-end optimization from raw data to accurate RUL prediction. Initially, Functional Alignment Resampling (FAR) is employed to generate high-quality functional signals; then, attention-enhanced Dynamic Time Warping (DTW) is leveraged to obtain individual degradation stages. Subsequently, an attention-enhanced of hybrid multi-scale RUL prediction network is constructed to extract both local and global features from multi-format data. Furthermore, the network achieves optimal feature representation by adaptively fusing multi-source features through Bayesian methods. Finally, we innovatively introduce a Deep Deterministic Policy Gradient (DDPG) strategy from DRL to adaptively optimize key parameters in the construction of individual degradation stages and achieve a global balance between model complexity and prediction accuracy. The proposed model was evaluated on aircraft engines and railway freight car wheels. The results indicate that it achieves a lower average Root Mean Square Error (RMSE) and higher accuracy in comparison with current approaches. Moreover, the method shows strong potential for improving prediction accuracy and robustness in varied industrial applications.

Keywords: remaining useful life (RUL) prediction; attention mechanism; Deep Deterministic Policy Gradient (DDPG); Functional Alignment Resampling (FAR)



Academic Editor: Theodore E. Matikas

Received: 1 September 2025

Revised: 30 September 2025

Accepted: 3 October 2025

Published: 14 October 2025

Citation: Bai, Y.; Du, J.; Li, H.; Bao, X.; Li, L.; Zhang, C.; Yan, J.; Wang, R.; Xu, Y. An Adaptive Framework for Remaining Useful Life Prediction Integrating Attention Mechanism and Deep Reinforcement Learning. *Sensors* **2025**, *25*, 6354. <https://doi.org/10.3390/s25206354>

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/>).

1. Introduction

With the rapid digital and intelligent transformation of manufacturing, key equipment components are increasingly evolving toward greater scale and integration [1]. When mechanical equipment components operate under complex conditions, degradation rates and degradation patterns exhibit significant individual variability. Based on daily inspection and regular maintenance, planned preventive maintenance in traditional maintenance strategies results in serious delays in fault monitoring and weak preventive abilities [2]. The development and application of Prognostics and Health Management (PHM) technology, which relies on the real-time health status of equipment, are dedicated to avoiding excessive maintenance, reducing false-alarm rates, and ensuring safe equipment operation by dynamically scheduling maintenance. As an essential aspect of PHM, Remaining Useful Life (RUL) prediction focuses on continuously monitoring equipment conditions and predicting failure times to extend the operational duration of the equipment and improve economic efficiency. Typical RUL prediction approaches are generally categorized into physics-based approaches and data-driven approaches [3].

Physics-based approaches represent the degradation process by establishing physical models of a complex system. For instance, Zhang et al. [4] developed a capacity-cycling degradation model to estimate the RUL of online lithium-ion batteries. In their study, they estimated the core temperature, state of charge, and battery capacity by leveraging thermal and Coulomb SOC models. Similarly, Shutin et al. [5] proposed a degradation model integrating tribological theory with the physical wear mechanisms of rolling bearings to predict the RUL of hydrodynamic bearings. In another study, Protopapadakis et al. [6] implemented an understandable AI-assisted RUL estimation method for turbine engines by leveraging a degradation model derived from aerothermodynamics and analyzing measurement data. Although physics-based methods are easily interpretable, their reliance on in-depth information about the principle of equipment failure mechanisms and domain-specific knowledge limits their generalization capability for unknown complex systems.

The widespread adoption of intelligent sensors and advances in big data technology are rapidly increasing the volume of monitoring data available to industries. Extracting value from these multi-source, heterogeneous datasets enhances Remaining Useful Life (RUL) prediction. Thus, data-driven RUL prediction is becoming popular across industrial and academic domains. Currently, data-driven remaining useful life prediction mainly utilizes three prevalent approaches: statistical methods, machine learning (ML), and deep learning (DL). Statistical methods such as Hidden Markov Models (HMMs), Kalman filters, and Wiener processes analyze statistical distributions, trends, and patterns in historical data to model equipment degradation and predict the RUL. Zhang et al. [7] successfully addressed challenges related to nonlinearity, state transitions, and stochasticity in predicting the remaining useful life of lithium-ion batteries. Their approach combines a nonlinear drift-driven Wiener process, a Markov chain-switching model, and a fuzzy system. Furthermore, they significantly improved the model's reliability in terms of predictive precision by introducing adaptive filtering techniques in dynamic environments. Although statistical models are computationally efficient, they have limited ability in modeling nonlinear relationships. As a result, they are frequently combined with ML approaches, including Support Vector Machines (SVMs), Random Forests (RFs), and decision trees, to derive valuable information from the Probability Density Functions (PDFs) of datasets [8]. For instance, Alfarizi et al. [9] constructed a two-stage model to forecast the remaining useful life of experimental bearings. In the first stage, the input signals were decomposed into different frequency bands using empirical mode decomposition, eliminating irrelevant

frequencies and highlighting fault characteristics. In the second stage, they combined a random forest model with Bayesian hyperparameter tuning to enhance the accuracy of RUL prediction. Although ML methods show strong nonlinear modeling capabilities, the abovementioned component-level RUL prediction approaches grounded in ML methodologies demand an index that reflects degradation levels and heavily rely on feature engineering, rendering them unsuitable for RUL prediction using multidimensional time-series data (MTSD).

As a subfield of ML, DL provides substantial technological advantages for component RUL prediction, owing to its strengths in modeling intricate nonlinear relationships, processing high-dimensional data, and automating feature engineering. Recurrent Neural Network (RNNs), which are capable of capturing temporal degradation patterns, are extensively used for estimating RUL [10]. Nevertheless, for extended time-series data, they are prone to issues such as gradient explosion and reduced computational efficiency, limiting their practical deployment in industrial settings [11]. As enhanced successors of RNNs, the use of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRUs) strengthens the ability to model long-term dependencies and mitigate gradient vanishing and explosion via gating mechanisms and gradient optimization strategies, establishing them as dominant approaches in remaining useful life estimation. Chui et al. [12] presented a Non-dominated Sorting Genetic Algorithm II (NSGA-II)-based RUL prediction model that combines the short-term prediction strengths of RNNs with the long-term prediction capabilities of LSTM, ultimately efficiently addressing challenges related to machine downtime and redundant maintenance during the running to failure and preventive maintenance of turbofan engines. Beyond long-term temporal dependencies, it is essential to consider the local features of MTSD. To combine global temporal patterns and local and global information for the RUL prediction of equipment, Cao et al. [13] proposed a parallel RUL prediction architecture combining a multi-scale CNN (MSCNN) and multi-scale LSTM (MSLSTM) to extract multi-dimensional health indicators, effectively reducing the local fluctuation caused by the CNN. Then, they used a dynamic time warping (DTW)-based similarity-matching algorithm to identify historical training samples with degradation trends similar to the test sequence. Finally, they achieved accurate prediction for railway freight car wheels. Duan et al. [14] treated mechanical monitoring data as “natural language sequences of machines” and input them into a Transformer. This method begins by utilizing its attention layers to highlight core time-step information and compute the output in parallel, followed by the integration of two attention mechanisms in the Transformer structure with LSTM in the encoder to extract both local and long-term temporal dependency information of the degradation process. Finally, it employs a nonlinear Wiener process (NWP) to calculate the PDF of the RUL. Song et al. [15] addressed the impact of prediction outcomes of the complex interactions among high-dimensional variables in MTSD by constructing multi-dimensional feature-correlated spatiotemporal (MFCST) graphs to implement feature extraction for data in different formats while employing a stacked long short-term memory (ST-LSTM) network to comprehensively explore local and global temporal pattern of MTSD. Then, they strategically weighted spatial and temporal patterns to enhance the model’s generalization ability and spatial perception of high-dimensional variable feature structures. Currently, DL-based methods for RUL prediction frequently adopt averaging or predefined weights to address issues such as high-frequency noise, local fluctuations, and missing values in high-frequency sensor data. Additionally, models trained using randomly sampled training data cannot adequately leverage the intrinsic correlations among sensors and their different levels of contribution to the degradation process, which limits the model’s predictive capability on real-time data. To resolve

these difficulties, regularization techniques are typically used, but they require manual parameter tuning, increasing the cost of human intervention.

In contrast, deep reinforcement learning (DRL) stands out as an ML technique, providing significant support for the capture of temporal dependencies in equipment RUL prediction due to its remarkable ability to learn optimal strategies through interaction with the environment, achieving stronger exploration and generalization capabilities [16]. In particular, DRL is divided into two approaches: the value-based Deep Q-Network (DQN) algorithm and the Deterministic Policy Gradient (DPG) algorithm. DQN uses deep neural networks to approximate the Q-value function and improves stability through experience replay and target networks. On this basis, Yao et al. [17] proposed a Deep Transfer Reinforcement Learning (DTRL) network based on LSTM, which utilizes novel Q-function updates and transfer strategies to estimate the RUL of machinery operating under similar tool and cutting conditions. DQN demonstrates exceptional performance in discrete action spaces; however, it exhibits significant limitations when applied to continuous action spaces. In contrast, DPG directly optimizes policy parameters through gradient ascent to maximize expected rewards. Despite its theoretical elegance, DPG lacks scalability in deep learning frameworks [18]. To address this, Actor–Critic algorithm was developed as an enhancement of DPG, combining value-function approximation with policy gradient methods. By leveraging the advantage function, it effectively reduces variance in gradient estimation [19]. Nevertheless, the Actor–Critic model’s training remains unstable due to high variance. Building upon this, the Deep Deterministic Policy Gradient (DDPG) algorithm extends the Actor–Critic strategy and is specifically designed for continuous action spaces. DDPG incorporates deterministic policies and target networks, successfully scaling DPG to high-dimensional, continuous action spaces and improving training stability. Zheng et al. [20] designed a DRL model trained based on the Twin Delayed DDPG (TD3) algorithm. The model leverages the powerful feature representation of DL while maintaining temporal dependencies across samples through RL, enabling accurate RUL estimation for rolling bearings. Hu et al. [21] integrated DRL with a Markov Decision Process (MDP) framework to learn to derive an optimal strategy for RUL prediction. Although current deep fusion techniques mitigate the shortcomings of conventional DL in RUL prediction and improve model robustness and adaptability, the use of DRL to dynamically adjust time-scale parameters and prioritize critical degradation phases remains an unresolved challenge.

To overcome the above limitations, this study presents ADAPT-RULNet, an adaptive RUL prediction framework that integrates attention mechanisms with DRL based on a hybrid network. The proposed framework enhances the quality of input signals by utilizing Functional Alignment Resampling (FAR) for MTSD preprocessing and using DTW to construct personalized datasets with similar degradation stages. The framework extracts both local and long-term degraded information from MTSD through an attention-enhanced multi-scale CNN and LSTM. These features are then integrated through an adaptive Bayesian fusion layer, thereby achieving better prediction performance. Furthermore, the framework introduces the DDPG to adaptively adjust critical time-scale parameters to obtain an optimal global balance between prediction accuracy and model complexity. To validate the effectiveness of ADAPT-RULNet, extensive experiments were carried out on two datasets in comparison with the advanced method. The experimental results suggest that ADAPT-RULNet outperforms all other approaches, on average.

The main contributions of this work are outlined below:

1. We propose a robust and precise data preprocessing framework for mechanical equipment RUL prediction. By employing the novel FAR method for data signal optimization, we effectively address the issues of noise, heterogeneity, and inconsistent

time-series lengths in raw sensor data, thereby providing high-quality input signals for subsequent feature extraction. Furthermore, leveraging the attention-based DTW-enhanced model leads to the selection of degradation stages with highly similar processes across devices to construct a personalized dataset with high-quality and consistent degradation.

2. We construct an attention-enhanced multi-scale parallel feature extraction model. The proposed method leverages a multi-scale CNN with spatial–temporal attention mechanisms to extract temporal features and local degradation patterns from multi-modal sensor data. Simultaneously, multi-scale LSTM with multi-head attention mechanisms is employed to capture multi-modal temporal features and global degradation patterns. The multi-dimensional features are adaptively fused using Bayesian probability to enhance the accuracy of RUL prediction.
3. We introduce a complexity–efficiency balancing strategy based on DDPG. This approach reformulates the parameter optimization process in RUL prediction as an MDP. This strategy leverage the experience replay mechanism and target-network soft update technique within the DDPG framework to adaptively optimize the time-window size and the number of similar samples of key parameters during the construction of the feature extraction dataset. This ensures a global balance between prediction performance and model complexity.
4. With respect to practical application prospects, the approach was validated on public and industrial datasets, confirming its effectiveness. The prediction results surpass those of existing CNN-LSTM models, demonstrating strong potential for intelligent maintenance in real industrial applications.

The structure of this study is outlined as follows: Section 2 reviews the related research and technical methodologies. Section 3 describes the proposed adaptive RUL prediction framework. Section 4 analyzes the experimental results, while Section 5 discusses the conclusions and outlines future research directions.

2. Theoretical Basis

2.1. Multi-Head Self-Attention Mechanism

The attention mechanism is inspired by the human visual cognitive process. Its fundamental principle is to rapidly scan the global image, identify and focus on the target regions that require special attention, and allocate more attentional resources to these regions, thereby obtaining richer detailed information and suppressing irrelevant interference. The self-attention mechanism is a specialized form of the attention mechanism primarily used to capture dependencies within the same sequence. However, a single attention head often fails to fully capture the diverse features in the input data. To improve the expressive power of their model, Vaswani et al. [22] proposed the Multi-Head Self-Attention (MHSA) mechanism, as represented in Figure 1.

The MHSA mechanism combines multiple scaled dot-product attention modules in parallel, allowing different “heads” to attend to distinct portions of the input data. Specifically, each self-attention module independently computes its output, and these outputs are integrated through a weighted summation to produce the final result of the MHSA mechanism. This design not only enables the capture of more diverse features and patterns but also significantly reduces computational costs due to its parallel computation characteristics. Compared to a single self-attention mechanism, the MHSA mechanism can more stably capture complex dependencies and structural information while demonstrating stronger robustness.

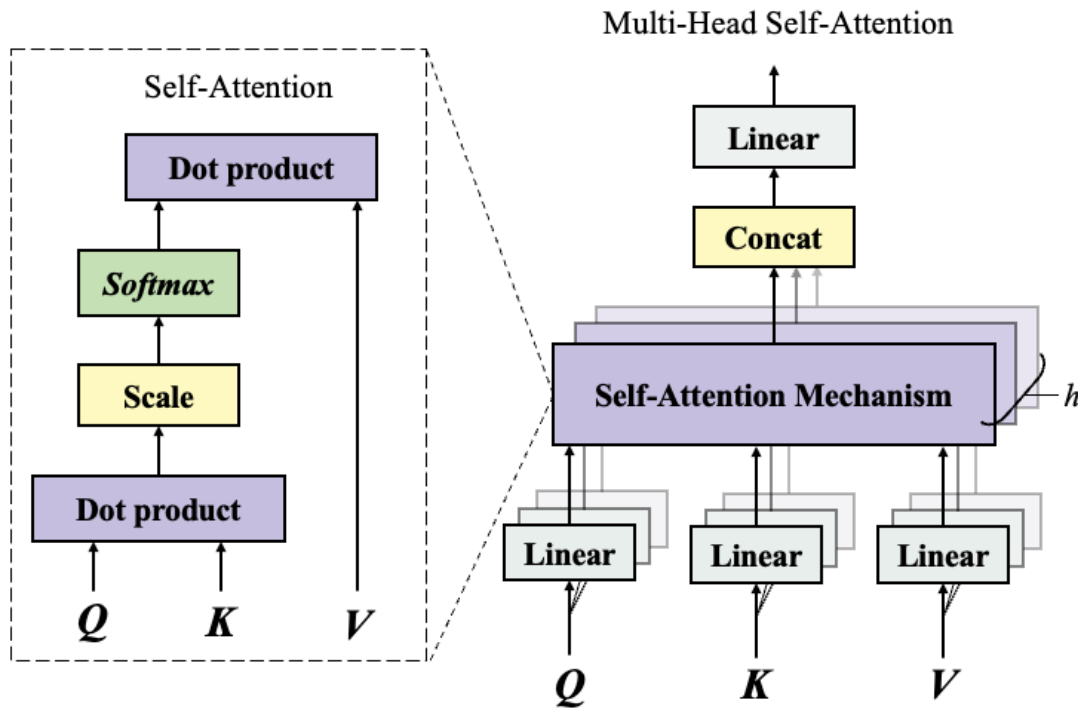


Figure 1. The architecture of the multi-head self-attention mechanism.

In the MHSA mechanism, the input data is first transformed to generate the query (Q), key (K), and value (V) vectors through distinct linear projections. Subsequently, each attention head independently computes its corresponding attention weights and produces the output of that head via a weighted summation process. Finally, the outputs from all attention heads are concatenated and passed through a final linear transformation. The combined result is expressed in Equation (1).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(h_1, \dots, h_H)W^O \quad (1)$$

where h_i denotes the output of a single head, as shown in Equation (2):

$$h_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i, \quad i = 1, 2, \dots, H$$

$$Q_i = XW_q^i, \quad K_i = XW_k^i, \quad V_i = XW_v^i \quad (2)$$

where W_q^i , W_k^i , W_v^i , and W^O denote learnable weight matrices; $\text{Concat}(\cdot)$ represents the concatenation operation; and H signifies the number of self-attention heads. The attention distribution coefficient matrix is computed based on the query matrix (Q) and the key matrix (K) and is subsequently normalized using the softmax function applied to the scaled dot-product results.

2.2. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were first proposed by LeCun et al. in 1989 [23], representing a specialized class of feedforward neural networks inspired by the biological visual system. By emulating three structural properties of visual cortex cells—local receptive fields, shared weights, and subsampling—CNNs preserve invariance to translation, scaling, and distortion, thereby enabling the extraction of topological features from raw input data. Additionally, a CNN can directly process 2D array inputs, avoiding the complex feature extraction and data reconstruction steps typically required by traditional recognition algorithms.

The convolutional layer consists of convolution and activation operations, performing convolution on the local receptive field using a filter (also referred to as a weight) window. Because of the weight-sharing property inherent in the convolutional layers of a CNN, the model complexity is significantly reduced, while the risk of overfitting caused by an excessive number of network parameters is effectively mitigated [24]. Given a 2D input (X) to the convolutional layer of a CNN, the output is computed as shown in Equation (3):

$$C_n = \sigma \left(\sum_{k=1}^K W_{n,k} \otimes X_k + b_n \right) \quad (3)$$

where K represents the input-channel count and n denotes the number of convolutional kernels. C_n signifies the n -th hidden feature. $W_{n,k}$ corresponds to the weight parameters between the n -th convolutional kernel and the k -th input channel, while b_n represents the bias term associated with the n -th convolutional kernel. The $\sigma(\cdot)$ function denotes the function of activation, and $\otimes$ symbolizes the convolution operation.

The pooling layer, also known as the downsampling layer, primarily serves to introduce translation invariance, reduce the dimensionality of the feature maps output by the convolutional layer, mitigate the risk of overfitting, and decrease computational complexity. Using degradation data of critical mechanical equipment as an example, determining whether a component has reached its end-of-life failure is more critical than identifying which specific part of the data represents the degradation phase. Common pooling operations include max pooling and average pooling. In this study, max pooling is employed, which extracts the maximum value within a rectangular neighborhood, as shown in Equation (4):

$$P(i, j) = \max_{(x, y) \in R} C(x, y) \quad (4)$$

where (x, y) denotes the size of the pooling window, $P(i, j)$ represents the output feature value after pooling, and $\max C(\cdot)$ indicates the maximum value within the pooling region.

2.3. Attention-Enhanced Depthwise Separable Convolution

Depthwise Separable Convolution (DSC) [25] enhances model efficiency by decomposing standard convolution into depthwise convolution and pointwise convolution, which collectively generate the final feature map. Depthwise convolution operates independently on each input channel, and its expression is shown in Equation (5):

$$\text{Depthwise Conv: } Y_{i,j,k} = \sum_{m,n} K_{m,n,k} \cdot X_{i+m,j+n,k} \quad (5)$$

On the other hand, pointwise convolution integrates channel information through a 1×1 convolutional kernel, as shown in Equation (6):

$$\text{Pointwise Conv: } Z_{i,j,l} = \sum_k W_{k,l} \cdot Y_{i,j,k} \quad (6)$$

where X represents the input feature map, K denotes the depthwise convolution kernel, W is the pointwise convolution kernel, Y corresponds to the output of the depthwise convolution, and Z signifies the output of the pointwise convolution.

To augment the model's capability in identifying crucial features, a Channel Attention Module (CAM) and Spatial Attention Module (SAM) are integrated. Specifically, the CAM dynamically assigns weights to each channel through a combination of Global Average Pooling (GAP) and fully connected layers, thereby emphasizing the channel features that contribute most significantly to the task, as shown in Equation (7):

$$M_c = \sigma(W_1(W_0(\text{GAP}(X)) + b_1) \quad (7)$$

where $\text{GAP}(X)$ denotes the global average pooling operation, W_0 and W_1 denote the weights of the fully connected layers, b_1 represents the bias, and σ signifies the Sigmoid activation function.

The SAM dynamically allocates weights to each spatial location by concatenating max pooling and average pooling results, followed by a convolutional operation, thereby emphasizing the spatial regions that contribute most significantly to the task, as shown in Equation (8):

$$M_s = \sigma(\text{Conv}_{c \times c}([\text{MaxPool}(X); \text{AvgPool}(X)])) \quad (8)$$

where $\text{Conv}_{c \times c}$ denotes the $c \times c$ convolutional operation and $[\text{MaxPool}(X); \text{AvgPool}(X)]$ represents the concatenation of the results from max pooling and average pooling. The weighting expression for channel attention is shown in (9):

$$X_{att} = M_c \bullet M_s \bullet X \quad (9)$$

where X_{att} represents the feature map after attention weighting.

2.4. LSTM Network

The LSTM network [26] is a specialized variant of the Recurrent Neural Network (RNN). By incorporating a gating mechanism, the LSTM network dynamically adjusts the memory state. This mechanism facilitates the effective transfer of information from previous time steps to subsequent units, resolving the issue of long-term dependencies in sequential data. Furthermore, LSTM effectively mitigates the common problems of vanishing and exploding gradients in RNNs, providing robust support for the modeling of long sequential data. As shown in Figure 2, the framework of LSTM primarily consists of three gating units: the forget gate (f_t), the input gate (i_t), and the output gate (o_t).

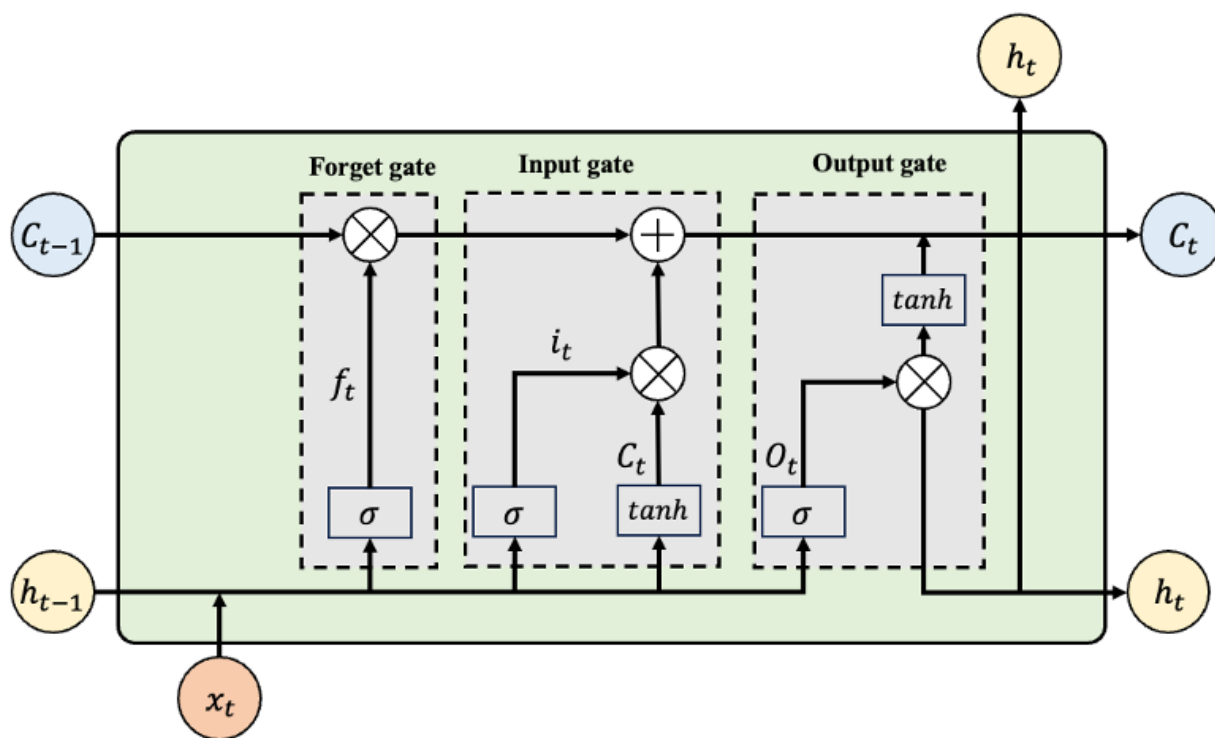


Figure 2. Structure of the LSTM network.

the agent randomly samples a batch of data to update the parameters of both the actor and critic networks.

The critic network is updated by minimizing the mean squared error between the target Q-value and the current Q-value. The target value is defined in Equation (11):

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}; \boldsymbol{\varphi}'); \boldsymbol{\theta}') \quad (11)$$

where $\boldsymbol{\theta}$ denotes the parameters of the critic network, which are optimized to learn the Q-value function ($Q(s, a; \boldsymbol{\theta})$). Simultaneously, $\boldsymbol{\theta}'$ represents the parameters of the target critic network used to compute the target Q-value. The critic network minimizes the loss ($L(\boldsymbol{\theta})$) through Equation (12):

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N (y_i - Q(s_i, a_i; \boldsymbol{\theta}))^2 \quad (12)$$

Meanwhile, the actor network updates its parameters ($\boldsymbol{\varphi}$) via the policy gradient, as shown in Equation (13):

$$\nabla_{\boldsymbol{\varphi}} J(\boldsymbol{\varphi}) \approx \frac{1}{N} \sum_{i=1}^N \left(\nabla_a Q(s_i, a; \boldsymbol{\theta}) \Big|_{a=\mu(s_i; \boldsymbol{\varphi})} \cdot \nabla_{\boldsymbol{\varphi}} \mu(s_i; \boldsymbol{\varphi}) \right) \quad (13)$$

To prevent training instability caused by abrupt changes in the target network parameters, a “soft update” strategy is implemented, as formalized in Equation (14):

$$\begin{aligned} \boldsymbol{\varphi}' &\leftarrow \tau \boldsymbol{\varphi} + (1 - \tau) \boldsymbol{\varphi}' && \text{(update target Actor)} \\ \boldsymbol{\theta}' &\leftarrow \tau \boldsymbol{\theta} + (1 - \tau) \boldsymbol{\theta}' && \text{(update target Critic)} \end{aligned} \quad (14)$$

where τ represents the soft update coefficient, which governs the synchronization rate of the target network parameters towards the primary network.

3. Proposed Method

To efficiently integrate heterogeneous sensor data, mine the degradation patterns of mechanical equipment components under complex operating conditions, and accurately predict the RUL, this paper proposes an adaptive RUL prediction framework—ADAPT-RULNet, which integrates attention mechanisms and deep reinforcement learning, as illustrated in Figure 4.

The proposed framework consists of four tightly coupled modules:

1. **Data preprocessing** employs Functional Alignment Resampling (FAR) to optimize raw sensor signals by addressing noise, heterogeneity, and inconsistent time-series lengths.
2. **Personalized dataset construction** utilizes attention-enhanced Dynamic Time Warping (DTW) to build similarity-based degradation stages, ensuring that samples with highly similar degradation trajectories are grouped together.
3. **Hybrid network-based RUL prediction** constructs a hybrid architecture combining multi-scale CNN (MSCNN) and multi-scale LSTM (MSLSTM) with attention mechanisms and applies Bayesian fusion for adaptive feature integration.
4. **Reinforcement learning-based adaptive parameter tuning** introduces a Deep Deterministic Policy Gradient (DDPG) to adaptively adjust critical parameters such as time-window size and the number of selected similar samples, balancing prediction accuracy with model complexity.

These modules work synergistically to remarkably enhance the model's robustness and generalization capability under complex operating conditions.

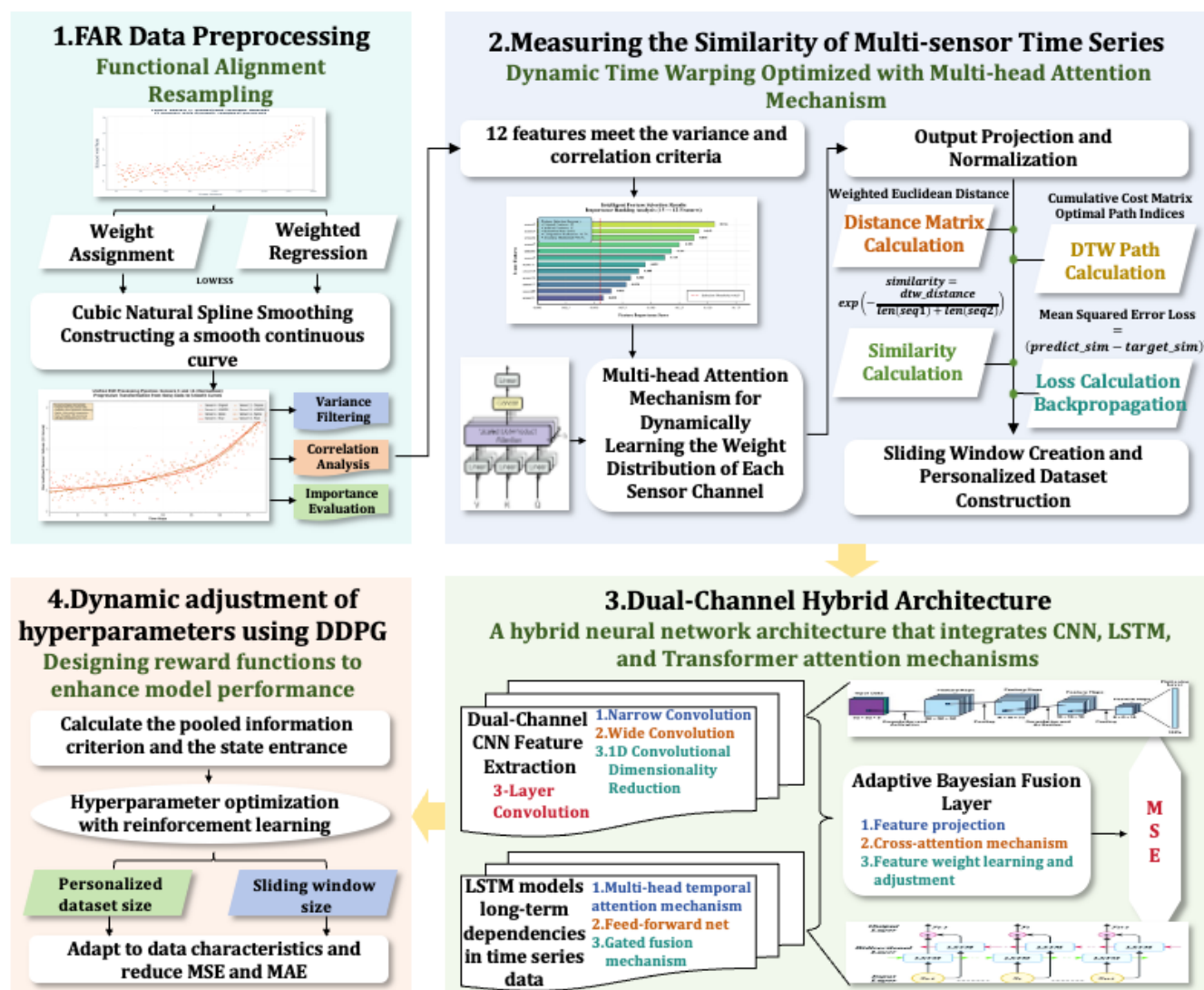


Figure 4. Framework of the proposed ADAPT-RULNet.

3.1. Data Preprocessing

In real-world industrial scenarios, the operational conditions of mechanical equipment exhibit high complexity, and their degradation processes demonstrate significant individual variability. Data collected from multi-source sensors often suffer from missing values, nonlinear characteristics, high levels of noise interference, unequal-length time series, and multi-source heterogeneity. Traditional data preprocessing methods encounter difficulty in capturing intricate degradation patterns. To tackle these issues, this paper introduces the Functional Adaptive Regression (FAR) approach, which reconstructs temporal continuity to better capture local or global degradation trends. By transforming time-series signals into functional signals, it facilitates the subsequent RUL prediction process.

Locally Weighted Scatterplot Smoothing (LOWESS) [28] and Cubic Natural Spline (CNS) smoothing serve as the core methodologies of FAR, enabling the transformation of raw time-series signals into functional signals. LOWESS employs localized smoothing techniques to effectively capture high-frequency noise and local fluctuations of the MTSD to adapt the local variations in the time series. Let $X = x(i, u)_t$, where $t = \{t_1, t_2, \dots, t_n\}$ represents the duration of the time series, $x = \{x_1, x_2, \dots, x_m\}$ represents the number

of devices, and $u = \{u_1, u_2, \dots, u_d\}$ signifies the number of sensors. The term $x(\bar{i}, u)$ represents the average value of all $x(i, u)_t$ within a local window (h), as illustrated in Equation (15):

$$x(\bar{i}, u) = \frac{1}{N} \sum_{j \in h} x(i, u)_j \quad (15)$$

The smoothed value ($\hat{x}(i, u)_t$) of LOWESS is computed through locally weighted regression, as shown in Equation (16):

$$\hat{x}(i, u)_t = \beta_0 + \beta_1 (x(i, u)_t - x(\bar{i}, u)) \quad (16)$$

where β_0 and β_1 are the regression coefficients obtained through weighted least squares and the weights (ω_{ij}) are defined as shown in Equation (17):

$$\omega_{ij} = \left(1 - \left| \frac{x(i, u)_j - x(i, u)_t}{h} \right|^3 \right)^3, \quad |x(i, u)_j - x(i, u)_t| \ll h$$

$$\min_{\beta_0, \beta_1} \sum_j \omega_{ij} \left(x_j^{(i, u)} - \left(\beta_0 + \beta_1 (x_j^{(i, u)} - x_t^{(i, u)}) \right) \right)^2 \quad (17)$$

where $x(i, u)_j$ is the observed value of time-series signal x_i at time step j .

The CNS interpolation method is applied to the multi-channel functional signals ($\hat{x}(i, u)_t$) to perform global fitting, generating continuous and smooth functional signal data. The fitting function ($g_m(X)$) is given by Equation (18):

$$g_m(X) = a_m + b_m(X - X_m) + c_m(X - X_m)^2 + d_m(X - X_m)^3 \quad (18)$$

where coefficients a_m , b_m , c_m , and d_m are obtained by minimizing the following objective function:

$$\min_{\beta_m} \sum_{i=1}^n (\hat{x}(i, u)_t - g_m(X))^2 + \lambda \int (g_m''(X))^2 dX \quad (19)$$

where λ is the smoothing parameter, controlling the smoothness of the fitted curve, and $g_m''(X)$ is the second derivative of the spline function, used to measure the curvature of the curve. Finally, by performing global fitting on the smoothed data, a continuous functional signal ($X_i(t)$) is generated, as shown in Equation (20):

$$X(t) = \sum_{m=1}^M \beta_m h_m(X) \quad (20)$$

3.2. Construction of Personalized Datasets with Similarity Degradation Stages

To completely capture the local suddenness and long-term trends of degradation behavior in mechanical equipment throughout the entire life cycle, this paper integrates a multi-head attention mechanism with DTW for personalized dataset construction (Attention-DTW). This approach overcomes the limitation of traditional DTW in terms of ignoring the differences among various sensors during the degradation process due to fixed weights. At the same time, it addresses the limitation of traditional Euclidean distance in effectively measuring the waveform similarity between two time series. By selecting the most relevant historical degradation samples for the test sequence, this method better captures the time-varying characteristics in dynamic uncertain environments and identifies similarities among operational signals, thereby providing reliable dataset support for the accuracy and robustness of RUL prediction.

Specifically, let the input MTSD be $\mathbf{X} = \{x_1, x_2, \dots, x_U\}$, where $x_u \in \mathbb{R}^T$ represents the time-series data of the u -th sensor. For the test sequence ($\mathbf{X}_v$) and the historical sequence ($\mathbf{X}_i$), the attention scores of each sensor channel are first calculated, as shown in Equation (21):

$$\alpha_u = \frac{\exp(\text{MLP}(x_v^u) \cdot \text{MLP}(x_i^u))}{\sum_{k=1}^U \exp(\text{MLP}(x_v^k) \cdot \text{MLP}(x_i^k))}, \quad (21)$$

where $\text{MLP}(\cdot)$ is a multi-layer perceptron that maps a single-channel sequence into a feature vector, while α_u reflects the importance weight of the u -th sensor in the similarity measurement. By incorporating the weight (α_u) into the calculation of the multi-channel DTW distance, we obtain Equation (22):

$$d_M(\mathbf{X}_v, \mathbf{X}_i) = \sum_{u=1}^U \alpha_u \cdot \min_{\pi} \sum_{(t,t') \in \pi} |x_{v,t}^u - x_{i,t'}^u|, \quad (22)$$

where this distance metric adaptively focuses on sensors sensitive to degradation while effectively mitigating noise interference.

During the system degradation process, the tail data typically best reflects the current state and the latest trends of the system. In the implementation, a fixed window of length L is used to extract the last L time points from each unlabeled sample ($\mathbf{X}_v$), resulting in $\mathbf{X}_{\tau t}$. Subsequently, a sliding window of length L is applied to the historical data ($\mathbf{X}_i$) to extract all candidate segments ($\{\mathbf{X}_{\tau t, j}\}$). Based on the Attention-DTW similarity calculation, the distance ($d_M(\mathbf{X}_v, \mathbf{X}_i)$) is computed according to Equation (22), and the M most similar segments are selected as shown in Equation (23):

$$\mathbf{X}_{\tau, v}^{i_m} = \arg \min_Y d(i, j), \quad v, \quad m = 1, 2, \dots, M \quad (23)$$

The predicted label (y_{i_m}) is derived based on Equation (24):

$$y_{i_m} = T_{i_m} - e_{i_m} + L, \quad m = 1, 2, \dots, M \quad (24)$$

and the final dataset is formulated as Equation (25):

$$\mathbf{D}_v = \left\{ \left(\mathbf{X}_{\tau, v}^{i_m}, y_{i_m} \right) \right\}_{m=1}^M \quad (25)$$

This process builds the complete training dataset ($\{\mathbf{D}_v, \mathbf{Y}_v | \mathbf{X}_v\}$), which will be used in the subsequent CNN feature extraction network.

3.3. Attention-Enhanced Multi-Scale Hybrid Network for Remaining Useful Life Prediction

The extraction of features from local and long-term degradation trends in the ADAPT-RULNet structure is mainly achieved through an attention-enhanced multi-scale hybrid RUL prediction network based on Bayesian fusion, as illustrated in Figure 5.

The proposed method primarily consists of three components:

(a) Attention-Enhanced Multi-Scale Depthwise Separable Convolution (DSC).

This module is designed to accurately identify short-term dependencies across multiple scales while reducing computational complexity. Four convolutional kernels with distinct dimensions—(3, 1), (5, 1), (7, 1), and (9, 1)—are employed to capture multi-scale features from the input data. Residual connections are introduced to preserve and enhance low-level feature information, mitigating the risk of gradient vanishing in deep networks. To further optimize feature representation, a Combined Spatial and Channel Attention Module (CSAM) is integrated, which fuses the Channel Attention Module (CAM) and the

Spatial Attention Module (SAM) to dynamically weight key features and achieve efficient fusion of multi-level features.

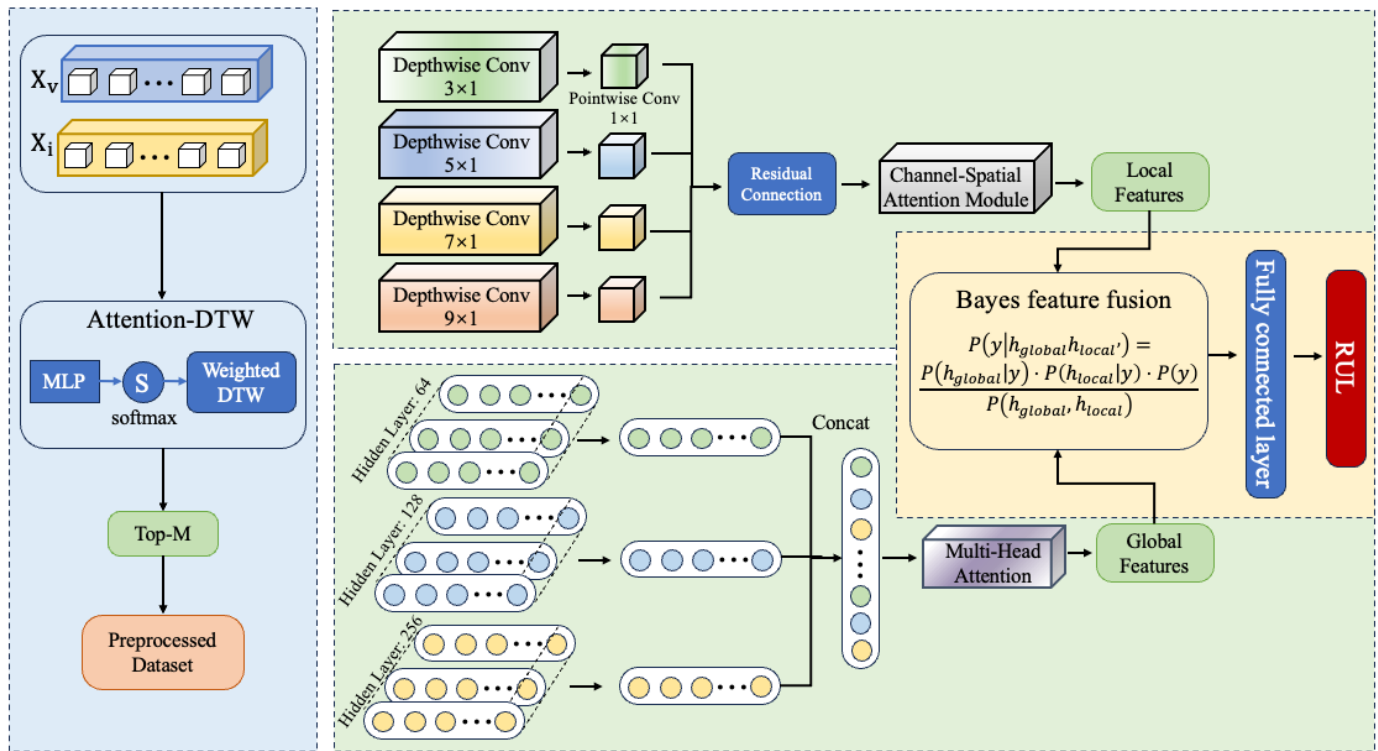


Figure 5. Hybrid RUL prediction network.

(b) Attention-Enhanced Multi-Scale LSTM Network. This network is proposed to enhance the extraction of long-term degradation trends. It employs three hidden layers with dimensions of 64, 128, and 256 to capture multi-scale temporal patterns while avoiding overfitting. To improve the identification of degradation characteristics, a Multi-Head Self-Attention (MHSA) mechanism is introduced, which dynamically allocates weights to focus on the most critical time-scale features for RUL prediction. Specifically, the attention mechanism computes weights for each time-scale feature, performs weighted fusion, and integrates the feature maps extracted by the LSTM network across the three scales to generate a comprehensive global feature representation.

(c) Bayesian Feature Fusion. A Bayesian probability-based feature fusion approach is designed to optimally combine local and global features while mitigating the uncertainty inherent in feature extraction from different networks. The mathematical formulation is given in Equation (26):

$$P(y | h_{global}, h_{local}) = \frac{P(h_{global} | y) \cdot P(h_{local} | y) \cdot P(y)}{P(h_{global}, h_{local})} \quad (26)$$

where $P(y | h_{global}, h_{local})$ represents the fused feature distribution; $P(h_{global} | y)$ and $P(h_{local} | y)$ denote the conditional probability distributions of the global and local features, respectively; and $P(y)$ is the prior distribution.

The fused feature representation is then obtained as follows:

$$h_{fused} = \arg \max_y P(y | h_{global}, h_{local}) \quad (27)$$

Finally, the features obtained through adaptive probabilistic fusion are fed into a multi-layer fully connected network to achieve precise prediction of the equipment's RUL. The training procedure of the Attention-Enhanced Multi-Scale Hybrid Network is presented in Algorithm 1.

Algorithm 1 Attention-Enhanced Multi-Scale Hybrid Network Training

Require:

- 1: *train_data*: Preprocessed training data
- 2: *train_labels*: Corresponding RUL labels
- 3: *val_data*: Validation data
- 4: *val_labels*: Corresponding RUL labels
- 5: *device*: Computational device ('cpu' or 'cuda')
- 6: *fusion_dim*: Dimension for feature fusion
- 7: *learning_rate*: Initial learning rate
- 8: *epochs*: Number of training epochs
- 9: *batch_size*: Batch size for training

Ensure:

- 10: *trained_model*: Trained neural network model
 - 11: **procedure** TRAINNEURALNETWORK(*train_data*, *train_labels*, *val_data*, *val_labels*, *device*, *fusion_dim*, *learning_rate*, *epochs*, *batch_size*)
 - 12: Initialize CNN-LSTM model with attention mechanisms
 - 13: Define loss function (e.g., MSE) and optimizer (e.g., AdamW)
 - 14: **for** epoch = 1 to epochs **do**
 - 15: **for** batch = 1 to len(*train_data*)/*batch_size* **do**
 - 16: Load batch data and labels
 - 17: Forward pass: compute model output
 - 18: Calculate loss
 - 19: Backward pass: compute gradients
 - 20: Update model parameters
 - 21: **end for**
 - 22: Validate model on validation set
 - 23: Compute validation loss and metrics
 - 24: Update learning rate scheduler if needed
 - 25: **end for** **return** *trained_model*
 - 26: **end procedure**
-

3.4. Strategy for Balancing Model Complexity and Efficiency

To achieve global balancing of model complexity and predictive performance and to overcome the lack of flexibility of traditional fixed-window approaches in capturing both long-term and short-term dependencies, this study innovatively introduces the DDPG algorithm into RUL prediction. By abstracting the adaptive parameter adjustment problem of the RUL prediction model into a DRL environment, we define the state space, action space, and reward function, thereby constructing a dynamic optimization framework.

State Space S: The state vector of the DDPG agent is composed of the performance metrics of the current RUL prediction model and the parameters of the personalized dataset. Specifically, the state vector includes the Mean Squared Error (MSE), Akaike Information Criterion (AIC), and DTW similarity, as well as the current time-window length (L) and

dataset size (M). These metrics comprehensively reflect the model's prediction accuracy, complexity, and dataset quality. The AIC value is defined as follows:

$$AIC = 2\phi - 2\ln(\hat{L}) = 2\phi + U \times L \times M [\ln(2\pi \cdot \text{MSE}) + 1] \quad (28)$$

where ϕ denotes the total number of model parameters, U represents the number of sensor channels, L is the sliding-window size, M is the number of similar segments, and $\hat{L}$ is the likelihood-function value of the maximum likelihood estimation.

Action Space A: The agent adjusts two continuous parameters— ΔL and ΔM , which represent the adjustment magnitudes of L and M , respectively. These values are normalized within $[-1, 1]$ and mapped to the valid ranges of L and M via linear scaling, ensuring rational and feasible parameter updates.

Reward Function R: The reward drives the optimization process of the DDPG agent. A composite reward is designed to dynamically adjust L and M , enabling better adaptability across different degradation stages and achieving global balancing between complexity and accuracy.

(i) *Base Reward:* Negatively correlated with MSE and AIC and positively correlated with DTW similarity:

$$R_{base} = -\alpha_1 \cdot \text{MSE} - \alpha_2 \cdot \text{AIC} + \alpha_3 \cdot \text{DTW}_{sim} \quad (29)$$

where α_1, α_2 , and α_3 are weighting coefficients.

(ii) *Improvement Reward:* Measures improvement compared to the previous step:

$$R_{improve} = \beta_1(\text{MSE}_{prev} - \text{MSE}_{curr}) + \beta_2(\text{AIC}_{prev} - \text{AIC}_{curr}) + \beta_3(\text{DTW}_{sim,curr} - \text{DTW}_{sim,prev}) \quad (30)$$

where β_1, β_2 , and β_3 are weighting coefficients.

(iii) *Stability Reward:* Penalizes unstable fluctuations in recent performance:

$$R_{stability} = -\gamma_1 \cdot \text{Std}(\text{MSE}_{recent}) \quad (31)$$

where γ_1 is a weighting coefficient.

(iv) *Total Reward:* The overall reward is defined as follows:

$$R_{total} = R_{base} + R_{improve} + R_{stability} \quad (32)$$

During DDPG training, the agent iteratively interacts with the RUL prediction environment. At each time step, the actor network selects an action, and the environment updates L and M and recalculates the MSE, AIC, and DTW similarity. The corresponding reward and new state are returned, which are stored in the replay buffer. The agent periodically samples from the buffer to update actor and critic parameters, while target networks are updated via a soft-update strategy. Through iterative training, the agent progressively learns an optimal policy that adaptively adjusts L and M to balance model complexity and predictive performance. The process of DDPG for hyperparameter optimization is presented in Algorithm 2.

Algorithm 2 DDPG for Hyperparameter Optimization**Require:**

- 1: *env*: Environment for RUL prediction
- 2: *state_dim*: Dimension of state space
- 3: *action_dim*: Dimension of action space
- 4: *action_range*: Range of actions
- 5: *memory_capacity*: Capacity of replay memory
- 6: *batch_size*: Batch size for training DDPG
- 7: *gamma*: Discount factor
- 8: *tau*: Soft update coefficient
- 9: *actor_lr*: Learning rate for actor network
- 10: *critic_lr*: Learning rate for critic network

Ensure:

- 11: *ddpg_agent*: Trained DDPG agent
- 12: **procedure** DDPG(*env*, *state_dim*, *action_dim*, *action_range*, *memory_capacity*, *batch_size*, *gamma*, *tau*, *actor_lr*, *critic_lr*)
- 13: Initialize actor network μ and critic network Q
- 14: Initialize target networks μ' and Q'
- 15: Initialize replay memory
- 16: Initialize actor and critic optimizers
- 17: **for** each training step **do**
- 18: Obtain current state from environment
- 19: Select action using actor network
- 20: Execute action in environment, obtain reward and next state
- 21: Store transition in replay memory
- 22: Sample random batch from replay memory
- 23: Update critic network using sampled batch
- 24: Update actor network using sampled batch
- 25: Soft update target networks
- 26: **end for** **return** *ddpg_agent*
- 27: **end procedure**

4. Experiments**4.1. Evaluation Metrics**

The model was evaluated with the Root Mean Square Error (RMSE), accuracy, and Score function (Score). Experiments were conducted on datasets from engines and railway freight car wheels to assess the algorithm's adaptability. Lower values of RMSE and Score indicate higher accuracy and quality of the algorithm.

(1) As a widely adopted regression metric, RMSE provides an intuitive measure of the model's performance in predicting target values. Its definition is given by Equation (33):

$$d_i = \widehat{RUL}_i - RUL_i, \quad RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n d_i^2} \quad (33)$$

Here, $\widehat{RUL}$ and RUL are the predicted and actual RUL values. d_i is the prediction error.

(2) Accuracy is used to evaluate the percentage of predictions within the correct range in the model's output. If the prediction error is $e_j \in [-13, 10]$ and $\text{Cor}(e_j) = 1$, the prediction is considered reasonable; otherwise, $\text{Cor}(e_j) = 0$. The mathematical formulation is provided in Equation (34):

$$\text{Accuracy} = \frac{100}{N} \sum_{j=1}^N \text{Cor}(e_j) \quad (34)$$

(3) Score is a well-known RUL statistical metric and indicates better RUL estimation performance when its value is smaller. Score penalizes late RUL predictions, whereas

RMSE assigns equal weights to premature and delayed predictions. Given that early fault detection holds greater significance for maintenance planning, its expression is given by Equation (35):

$$\text{Score} = \begin{cases} \sum_{i=1}^M \left(e^{-\frac{d_n}{13}} - 1 \right), & \text{if } d_n < 0 \\ \sum_{i=1}^M \left(e^{\frac{d_n}{10}} - 1 \right), & \text{if } d_n > 0 \end{cases} \quad (35)$$

where d_n is the difference between $\widehat{RUL}_i$ and RUL_i and M denotes the total size of the dataset.

4.2. Engine Dataset Prediction

The dataset for a turbofan engine was generated by the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) platform [29]. The dataset includes four subsets, as shown in Table 1. The progression of complexity from FD001 to FD004 stems from diverse operational scenarios and failure types. Each subset is divided into training and testing components. Training components feature comprehensive run-to-failure time series from multiple engine units, while testing components present incomplete operational records preceding engine failure events. Each operational record includes 26 variables: variable 1 is the engine ID, variable 2 is the operational cycle, variables 3 to 5 are operational setting parameters that notably influence engine performance, and the remaining 21 variables are noisy sensor readings.

Table 1. Overview of the C-MAPSS datasets.

Dataset	FD001	FD002	FD003	FD004
Training Units	100	260	100	249
Testing Units	100	259	100	248
Operating Conditions (OCs)	1	6	1	6
Fault Modes (FMs)	1	1	2	2

4.2.1. Data Preprocessing

Following the methodology proposed by Ragab et al. [30], the data was preprocessed, and fourteen sensors exhibiting degradation trends were selected, with the engine's RUL label serving as the final input for the ADAPT-RULNet model. Taking SENSOR2, SENSOR4, SENSOR7, and SENSOR11 as representative cases, Figure 6 presents a comparative analysis between the raw sensor data and the data processed using the FAR method. The FAR technique demonstrates its efficacy by effectively eliminating high-frequency noise, mitigating localized fluctuations, and addressing inconsistencies in time-series length.

Finally, Z-score normalization was applied to all data that had undergone smoothing and interpolation processes. This step was essential to eliminate the discrepancies in scale and numerical range across different sensors, thereby accelerating the convergence of model training and avoiding dominance by high-magnitude features.

4.2.2. Construction of Personalized Datasets

To achieve efficient and precise similarity measurement and sample selection, this paper proposes a strategy for personalized dataset construction with similarity degradation stages by Attention-DTW. This strategy dynamically learns the weight distribution of each sensor channel, enabling the DTW alignment process to adaptively focus on key modal information that is more indicative of degradation trends.

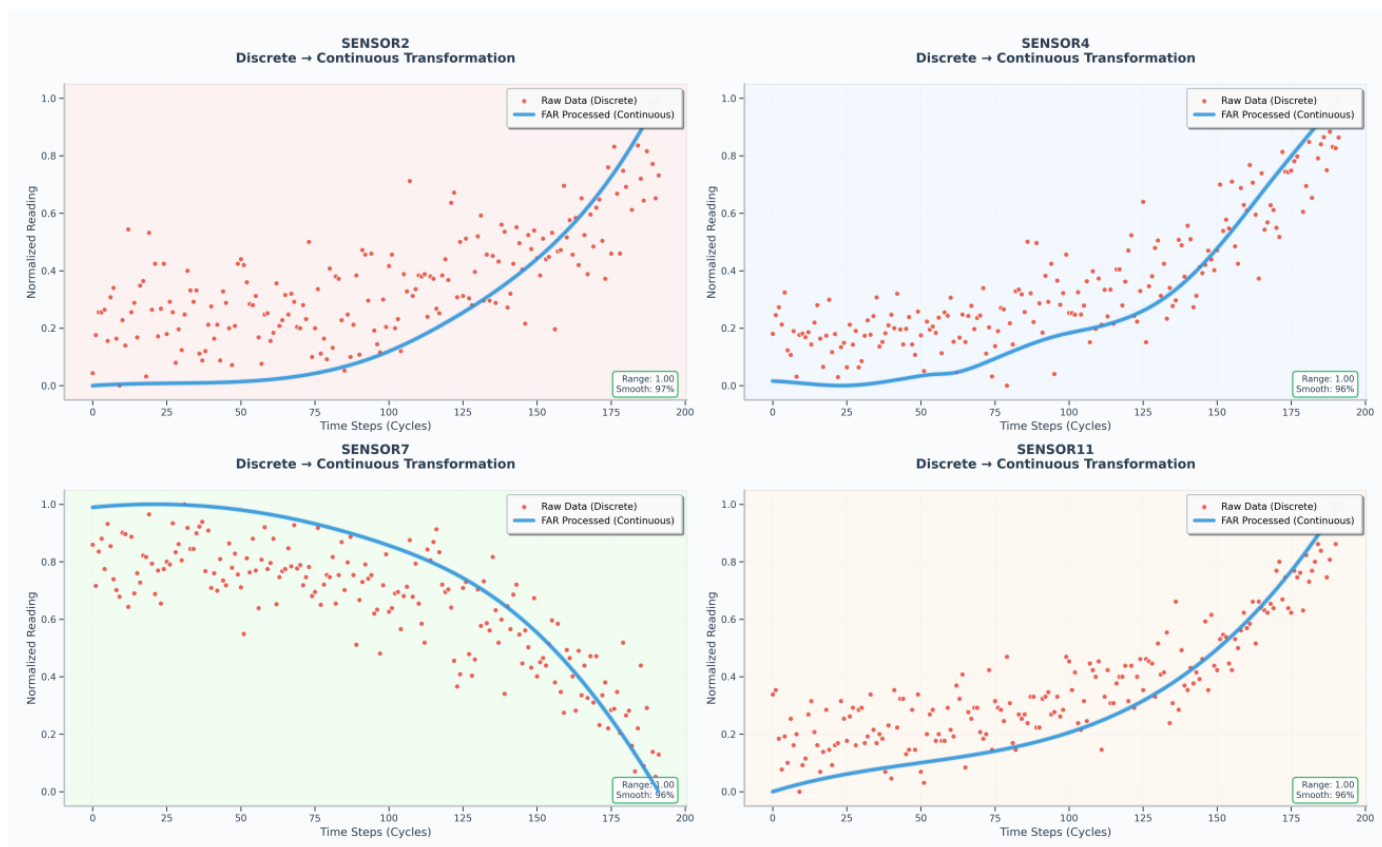


Figure 6. Comparative diagrams for preprocessing of sensor data.

As illustrated in Figure 7, the construction process of the personalized dataset includes the following steps: Step 1—raw sequences are extracted from the general training set and mapped to a two-dimensional feature space for visualization; Step 2—a test sequence of length M is input as the query object; Step 3—the candidate windows most similar to the test sequence are screened from the training set through Attention-DTW similarity calculation; Step —the top M samples are selected based on similarity ranking to form the personalized dataset, thereby providing a more reliable data foundation for degradation trend analysis.

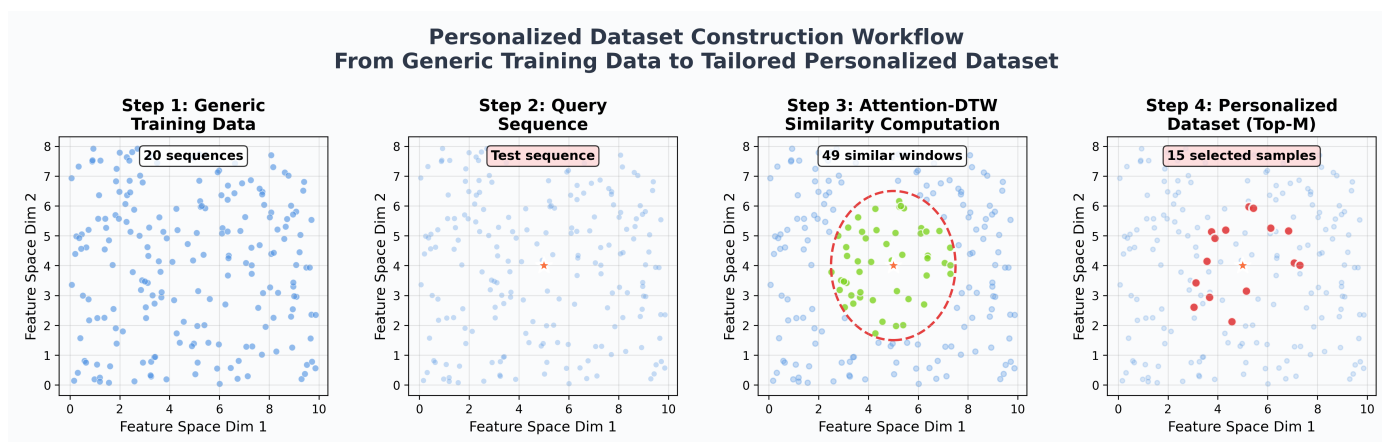


Figure 7. The construction process of the personalized datasets.

4.2.3. Parameter Selection

The parameter configuration of the proposed ADAPT-RULNet model is listed in Table 2.

Table 2. The parameters of the ADAPT-RULNet model.

Parameter	Value
input_dim	14
cnn_filters	[64, 128, 256, 512]
cnn_kernels	[3, 5, 7, 9]
lstm_hidden	256
lstm_layers	3
fusion_dim	512
Learning rate	0.001
Training epochs	200
Window length	50
Loss function	NSE Loss
Batch size	64
Optimizer	AdamW

For model optimization, the AdamW optimizer is employed to update the parameters, initialized with a learning rate of 0.001. To further enhance the training performance, the learning rate is dynamically adjusted using the Cosine Annealing Warm Restarts scheduler, where the initial cycle (T_0) is set to 10, the cycle multiplier (T_{mult}) is set to 2, and the minimum learning rate is 1×10^{-6} . Additionally, early stopping prevents overfitting by terminating training when validation loss plateaus for 20 epochs.

During hyperparameter selection, comparative experiments were conducted on the learning rate, the number of LSTM layers, the kernel sizes of the CNN, and the number of filters. Multiple experiments were performed to evaluate their impact on RUL prediction accuracy, and the most suitable hyperparameters were identified accordingly. The comparative results are shown in Figure 8, where the violin-shaped regions represent kernel density estimates, the black lines indicate the upper and lower bounds after removing outliers, the horizontal lines denote the medians, and the white dots represent outliers. As illustrated in the figure, the highest prediction accuracy is achieved when the learning rate is 0.001, the number of LSTM layers is 3, the CNN kernel sizes are [3, 5, 7, 9], and the CNN filter counts are [64, 128, 256, 512].

4.2.4. Reinforcement Learning-Driven Adaptive Parameter Tuning

The parameter adjustment problem is formulated as a sequential decision-making task within the DDPG framework, which adaptively optimizes the key parameters in personalized dataset construction through interaction with the prediction environment to learn the most effective policy. Specifically, the state space of the agent is designed as a five-dimensional vector comprising the performance metrics of the current RUL estimation model and the parameters of the personalized datasets, including the window length (L), the dataset size (M), MSE, AIC value, and DTW similarity.

The RUL prediction environment is responsible for updating the L and M parameters based on the actions of the DDPG agent and providing corresponding performance feedback. Figure 9 illustrates the evolution trends of the L and M parameters during DDPG training. The horizontal axis denotes the training episodes, the left vertical axis represents L, and the right vertical axis represents M. The light-blue and dark-blue curves represent the raw values and the 10-episode moving average of L, respectively, while the light-red and dark-red curves represent the raw values and the 10-episode moving average of M, respectively. Moving average processing provides a clearer depiction of the overall trend of parameter changes.

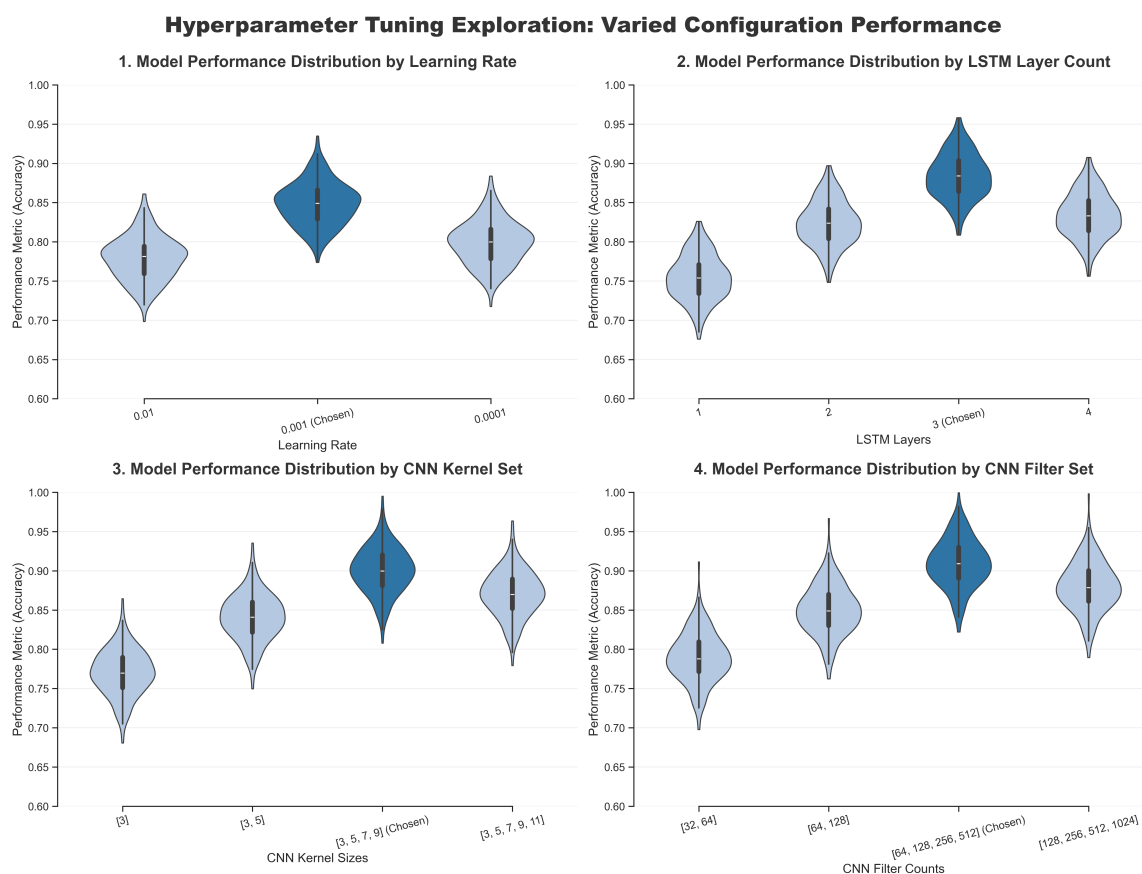


Figure 8. Comparison of accuracy of different parameters.

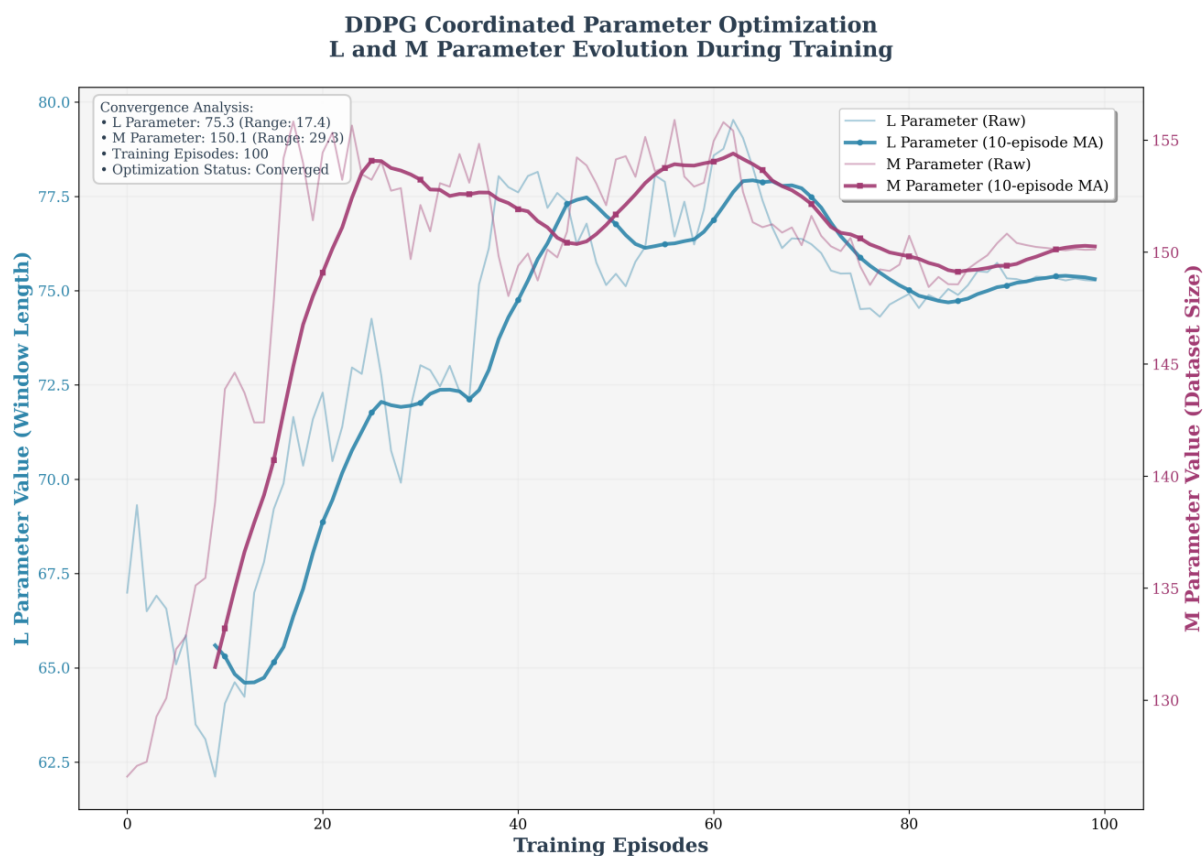


Figure 9. The evolution and convergence analysis of the L and M parameters during DDPG training.

The convergence analysis in the upper-left corner indicates that L ultimately converges to 75.3 and M converges to 150.1, with the optimization state marked as “Converged”. During initial training, the raw values of L and M exhibit significant fluctuations, reflecting the active exploration of parameter adjustments. As training progresses, the moving-average curves gradually smooth out and stabilize in the mid-to-late stages, validating the effective optimization and convergence process of the DDPG algorithm.

Figure 10 illustrates the evolution of the total reward of the RUL prediction model based on the DDPG algorithm over 100 training episodes. As shown in the figure, the agent’s learning process undergoes four distinct stages: an initial exploration phase characterized by low and highly fluctuating rewards; a subsequent learning and optimization phase with gradually increasing rewards; and, finally, a convergence phase where stable, high-performance rewards are achieved. This evolutionary trend confirms that the DDPG algorithm can effectively learn the optimal strategy for adaptively adjusting the time-window length (L) and the dataset size (M), thereby achieving a dynamic balance between model complexity and prediction accuracy.

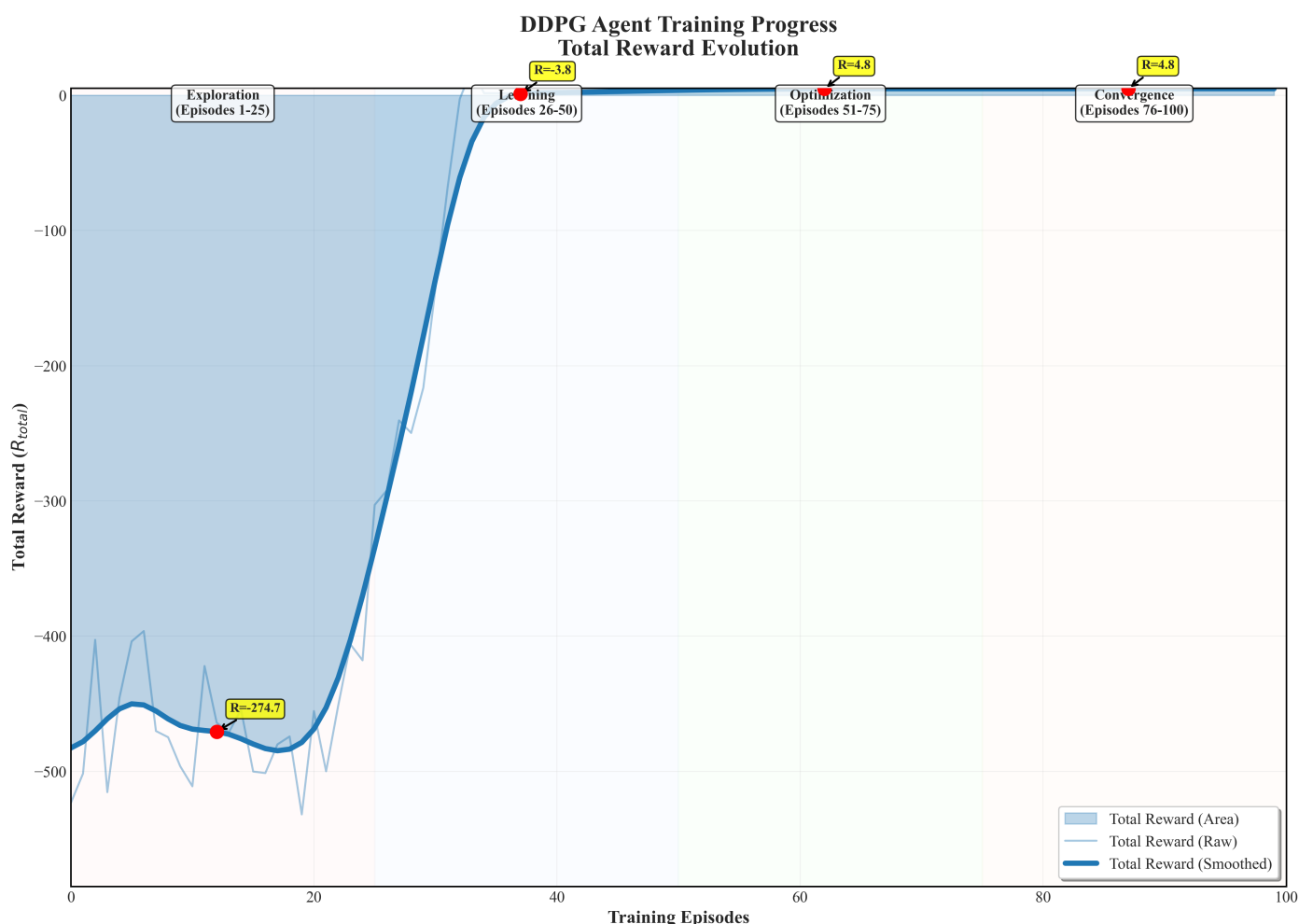


Figure 10. Evolution of total reward.

Figure 11 presents a contour analysis of model performance distribution based on the combinations of L and M, including four subplots for RMSE, R^2 Score, MAE, and comprehensive performance. In each subplot, the values of L are plotted along the horizontal axis, whereas the values of M are plotted along the vertical axis. The color gradient and contour lines visualize the performance under different parameter combinations, where

cool tones (e.g., blue) indicate better performance and warm tones (e.g., red) indicate poorer performance.

Performance Landscape Contour Analysis Parameter Optimization Surface with Optimal Point Identification

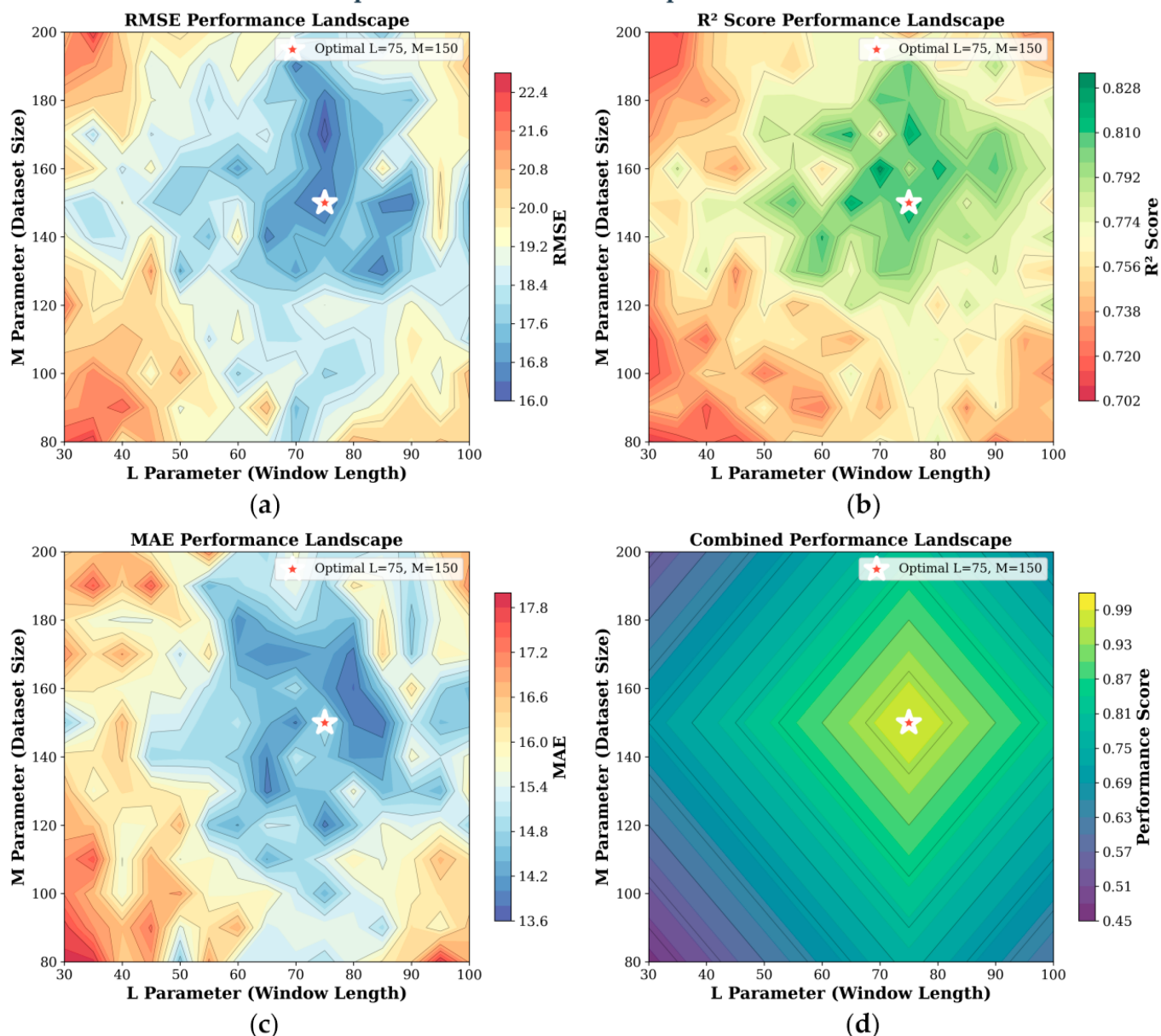


Figure 11. Contour analysis of model performance distribution with respect to the L and M parameters. Subplots show results for (a) RMSE, (b) R^2 Score, (c) MAE, and (d) comprehensive performance. The white star denotes the optimal parameter point (L = 75, M = 150).

Each subplot marks the optimal parameter point as “Optimal L = 75 and M = 150” (denoted by a white star), demonstrating that this combination achieves optimal performance in terms of RMSE (measuring the magnitude of prediction error, where lower values are better), R^2 Score (reflecting the model’s ability to explain data variation, where values closer to 1 are better), MAE (calculating the mean absolute error between predictions and true values, where lower values are better), and comprehensive performance. Such visualizations intuitively illustrate the impact of parameter interactions on model performance, identify universally optimal parameters, and provide critical insights for parameter tuning

and model performance validation, thereby aiding in understanding parameter sensitivity regions and optimization stability.

4.2.5. Experimental Results

This section presents the experimental results of the proposed adaptive RUL prediction framework on the C-MAPSS dataset and compares it with current mainstream approaches both quantitatively and qualitatively, in order to validate the effectiveness of the proposed method.

The prediction results of ADAPT-RULNet on the C-MAPSS dataset are shown in Figure 12. The ADAPT-RULNet model demonstrates a better alignment between predicted and observed RUL curves across different subsets of the dataset. Notably, as the actual RUL diminishes—indicating the engine’s approach to failure—the prediction error is markedly reduced, with the majority of the predicted values falling within a reasonable error margin. This indicates the model’s proficiency in accurately identifying significant degradation features during the latter stages of equipment deterioration, particularly within the pivotal prediction window, thereby facilitating precise RUL forecasting.

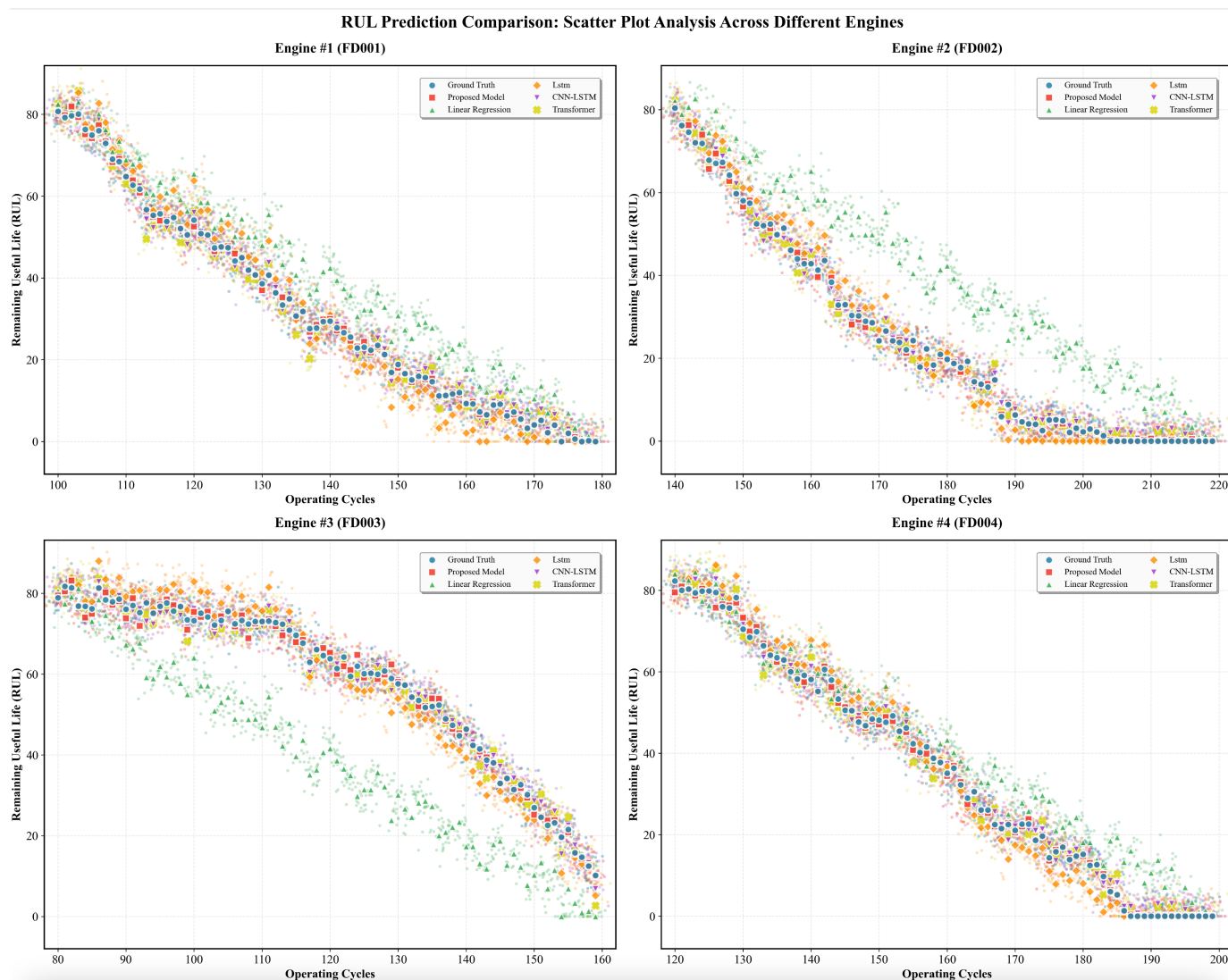


Figure 12. Prediction results of ADAPT-RULNet on the C-MAPSS dataset.

For the purpose of evaluating the performance of the proposed framework, we further compared ADAPT-RULNet with recent prediction methods. The comparative results are summarized in Table 3.

From a horizontal perspective, the evaluation metrics (RMSE, Score, and Accuracy) show a gradual decline across the FD001 to FD004 datasets, reflecting the increased complexity of the dataset under the diversity of operating conditions and fault modes, which brings greater challenges to the prediction task. In a vertical comparative analysis, the ADAPT-RULNet model proposed in this study demonstrates notable competitiveness across all evaluation metrics. Its average RMSE and average Accuracy achieve the best optimal values among existing methods, clearly indicating that the proposed framework surpasses previous solutions with regard to overall prediction accuracy. Furthermore, we investigated the computational performance of the models. As shown in Figure 13, by comparing the computational efficiency of advanced models listed in Table 3, including ADAPT-RULNet, Transformer-ED, BiLSTM-ED, BiGRU-AS, and LSTM-AON, we observed that the proposed method achieves an upper-middle level of efficiency. In addition, further analysis of the neural network component and the reinforcement learning training cost revealed that the efficiency bottleneck of the proposed method primarily lies in the reinforcement learning module. In future work, lightweight architectures and meta-learning strategies will be explored to further enhance the efficiency of the reinforcement learning component. Although further performance improvements might be attainable through a more extensive hyperparameter search and more efficient and accurate data preprocessing, the proposed and validated RUL prediction framework, which integrates attention mechanisms and DRL, leverages its adaptive advantages to effectively address the inherent limitations of traditional deep learning methods in handling heterogeneous sensor data and complex operating condition failure patterns. This not only provides an effective approach to reducing prediction inaccuracies but also offers practical references for researchers in hyperparameter optimization.

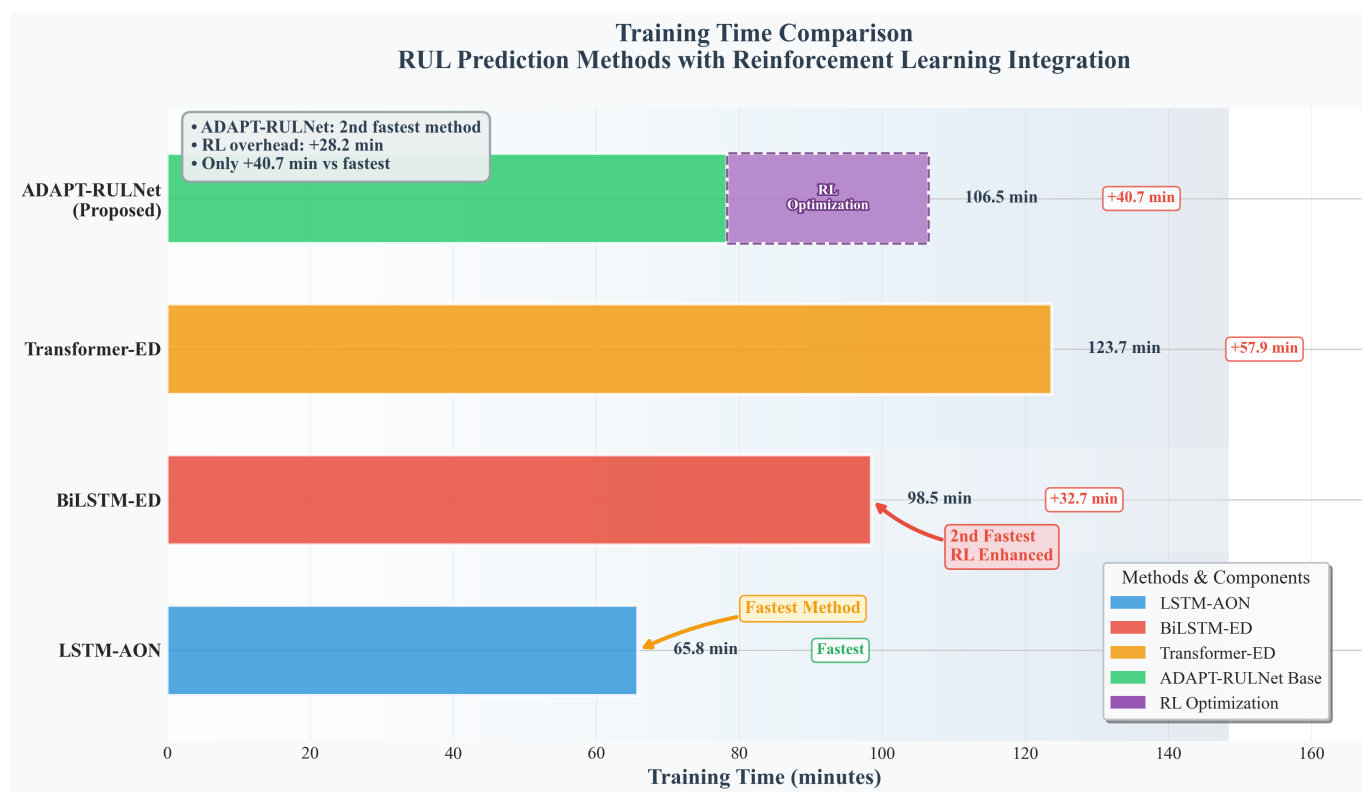


Figure 13. Comparison of computational performance of algorithms.

Table 3. Comparison of evaluation metrics.

Method	Metric	FD001	FD002	FD003	FD004	Average
LSTM-AON [31]	Score	284	2454	428	4708	1968.5
	Accuracy	64	62	57	34	54.25
	RMSE	13.68	20.81	<u>15.53</u>	27.31	19.332
BiLSTM-ED [32]	Score	273	3099	574	3202	1787
	Accuracy	57	49	42	40	47
	RMSE	14.74	22.07	17.48	23.49	19.445
KGHM [33]	Score	250.99	1131.03	333.44	3356.10	1267.89
	Accuracy	<u>67</u>	<u>46</u>	<u>59</u>	<u>45</u>	54.25
	RMSE	13.18	13.25	13.54	19.86	14.958
CNN-LSTM [34]	Score	256	–	–	–	–
	Accuracy	58	–	–	–	–
	RMSE	13.34	–	–	–	–
Attention CNN-RUL [35]	Score	217.02	789.32	216.79	1107.96	582.77
	Accuracy	59	–	–	–	–
	RMSE	10.43	11.02	10.03	16.23	11.93
Transformer-ED [36]	Score	286	2799	574	4655	2078.5
	Accuracy	64	61	57	<u>44</u>	<u>56.50</u>
	RMSE	13.87	<u>20.76</u>	17.42	24.41	19.115
ADAPT-RULNet	Score	275	2324	460	3155	1553.5
	Accuracy	68	65	60	47	60.0
	RMSE	12.48	19.45	14.92	22.97	17.45

4.3. Prediction of Railway Freight Car Wheels Dataset

To additionally confirm the generalization capability of the ADAPT-RULNet framework in engineering practice, we migrated it to the non-public TWDS railway freight car wheel dimension monitoring dataset under real-world railway operation conditions for experimental verification.

(1) Freight Car Wheel Dataset

The wheel dataset comprises 400 sets of wheel inspection data, each with an operational cycle exceeding two years, as detailed in [13]. The dataset is structured with 17 fields, among which fields C1 to C6 contain wheel positioning information utilized for the precise identification and tracking of each wheel's location. The actual deterioration information is documented in fields C7 to C16, which meticulously reflect the wear and performance evolution of the wheels under actual operating conditions.

(2) Data preprocessing

Because of the effects of wheel vibration and sensor data acquisition errors, the raw data exhibit significant fluctuations. Additionally, during wireless transmission, network latency and external environmental interference can cause data loss, noise, and anomalies.

Therefore, it is necessary to perform Functional Adaptive Regression (FAR) initialization on the raw data. Subsequently, six sensors that reflect the wear evolution patterns were identified, including left wheel thickness (lyhd), left wheel tread-wear depth (lycz), rim wear (tmyz), wheel-axle diameter parameter (lwhd), mileage data (lj), and inner measurement value (ncj). These parameters comprehensively characterize the wear evolution of wheel-axle components under actual operating conditions. The preprocessing results of the wheel dataset are illustrated in Figure 14, which shows that FAR effectively eliminates high-frequency noise, mitigates localized fluctuations, and enhances the consistency of time-series signals.



Figure 14. Data trend after preprocessing.

The pre-trained weights of the ADAPT-RULNet model were used on the railway freight car wheel dataset for RUL prediction. Experimental results presented in Figure 15 demonstrate that the proposed model achieves stable predictive performance in the quantitative analysis on the TWDS dataset in comparison with the baseline LSTM and CNN-LSTM models. As a result of heterogeneous data sources and operational contexts, its absolute performance metrics exhibit some differences relative to its performance on the C-MAPSS dataset. However, the model demonstrates better transfer learning capabilities, effectively capturing degradation trends in non-public data. Specifically, on the TWDS dataset, ADAPT-RULNet maintains an average RMSE within an acceptable range, and its prediction curves exhibit high consistency with the actual RUL trends. These findings further validate the practical value of our proposed framework in handling unknown or complex real-world data, laying the foundation for future industrial deployment.

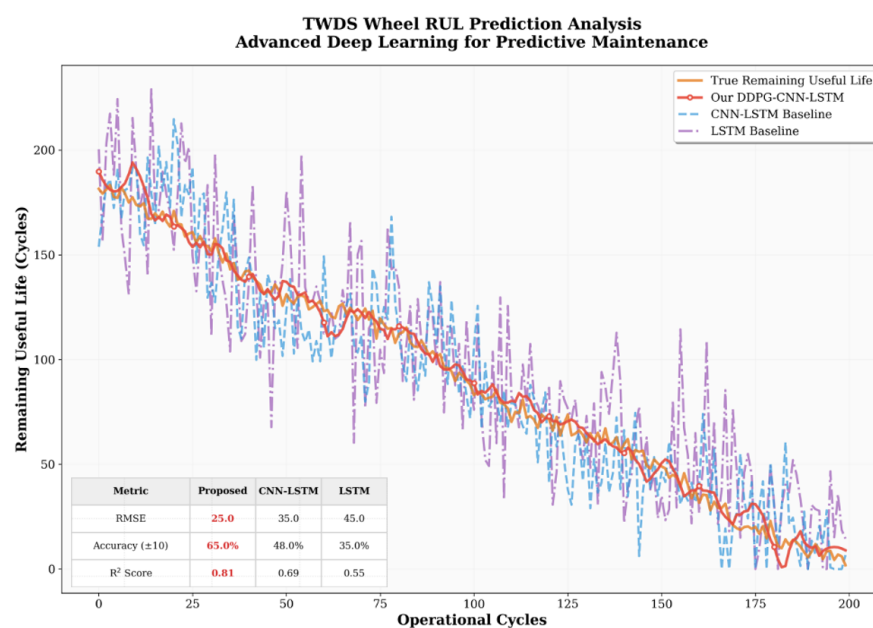


Figure 15. The RUL prediction results of ADAPT-RULNet on the TWDS dataset.

5. Conclusions

Accurate prediction of RUL is of paramount importance in optimizing maintenance strategies, reducing operational expenses, and ensuring the operational safety of equipment. This study introduces an adaptive framework for RUL prediction that integrates attention mechanisms and RL. The primary objective is to address challenges related to the adaptability of the prediction process, the accuracy of prediction results, and the generalization ability of the prediction model. Additionally, the method employs the FAR approach for data preprocessing and utilizes attention mechanism-based DTW to construct personalized datasets, ensuring high-quality signal input while improving the efficiency of feature extraction. The proposed attention-enhanced CNN-LSTM hybrid network architecture achieves the fusion of local temporal and global dependency features, enhancing the accuracy of RUL prediction, particularly providing high-precision predictions during the later stages of equipment degradation. Finally, to balance model complexity and prediction performance, parameter tuning is transformed into an MDP model, and techniques including experience replay and target-network soft updates of the DDPG algorithm are adopted to adaptively adjust key parameters in personalized dataset construction. The method's efficacy was validated on datasets covering two distinct components. Comparative experiments with various current DL methods were conducted. The findings indicate that the new approach achieves the highest accuracy and average metrics.

However, our proposed method also has limitations, especially in terms of model performance and interpretability. In upcoming research, we plan to continuously optimize hyperparameters and the reinforcement learning reward function, as well as further improve the computational efficiency of the algorithm. In addition, we will explore physics-informed deep learning by integrating physical models with neural networks to gain deeper insights into the degradation mechanisms of mechanical components, thereby enhancing both interpretability and predictive accuracy. We also aim to incorporate Bayesian neural networks for uncertainty estimation, providing more reliable confidence intervals for RUL predictions and enabling decision-makers to assess risks more scientifically. Furthermore, the integration of the RUL prediction model with digital twin technology will be pursued to achieve real-time monitoring and online updating of equipment status, significantly improving prediction accuracy and timeliness. Through these forward-looking studies, we expect to advance the field of RUL prediction and provide stronger technical support for equipment maintenance and operational management. In future work, we plan to continuously optimize hyperparameters and the reinforcement learning reward function, as well as further improve the computational efficiency of the algorithm. In addition, we will explore physics-informed deep learning by integrating physical models with neural networks to gain deeper insights into the degradation mechanisms of mechanical components, thereby enhancing both interpretability and predictive accuracy. We also aim to incorporate Bayesian neural networks for uncertainty estimation, providing more reliable confidence intervals for RUL predictions and enabling decision-makers to assess risks more scientifically. Furthermore, the integration of the RUL prediction model with digital-twin technology will be pursued to achieve real-time monitoring and online updating of equipment status, significantly improving prediction accuracy and timeliness. Through these forward-looking studies, we expect to advance the field of RUL prediction and provide stronger technical support for equipment maintenance and operational management.

Author Contributions: The specific contributions of each author to this study are outlined as follows: Y.B. was responsible for the experimental design and execution, participated in data analysis and interpretation of the results, and wrote the entire paper. J.D. provided research ideas and theoretical guidance and made modifications and improvements to the overall structure and content of the paper. H.L. collected and organized experimental data, played a crucial role in experimental design and data analysis, provided professional statistical analysis support, and participated in the interpretation of conclusions and experimental results in the paper. X.B. participated in the research of the basic methods of the paper and revised and polished the language and details of the paper. L.L. and C.Z. participated in the design of research ideas and data interpretation for this paper and contributed to the interpretation of the conclusions and experimental results. J.Y., R.W., and Y.X. contributed to the review of the usability of the paper's design methods. Each author's contribution to the research was indispensable, jointly promoting the progress and achievement of the research. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Scientific and Technological Development Program Project of China State Railway Group Co., Ltd. (Grant No. P2023T002), and the Scientific and Technological Innovation Project of China National Energy Co., Ltd. (Grant No. SHGF-17-56).

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

Conflicts of Interest: Author Linjun Li was employed by the company China National Energy Co. 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. Lin, T.; Wang, H.; Guo, X.; Wang, P.; Song, L. A novel prediction network for remaining useful life of rotating machinery. *Int. J. Adv. Manuf. Technol.* **2023**, *124*, 4009–4018. [\[CrossRef\]](#)
2. Li, H.; Wang, H.; Xie, Z.; He, M. Fault diagnosis of railway freight car wheelset based on deep belief network and cuckoo search algorithm. *Proc. Inst. Mech. Eng. F J. Rail Rapid Transit* **2022**, *236*, 501–510. [\[CrossRef\]](#)
3. Rezamand, M.; Kordestani, M.; Cariveau, R.; Ting, D.S.K.; Orchard, M.E.; Saif, M. Critical wind turbine components prognostics: A comprehensive review. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9306–9328. [\[CrossRef\]](#)
4. Zhang, D.; Dey, S.; Perez, H.E.; Moura, S.J. Remaining useful life estimation of lithium-ion batteries based on thermal dynamics. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 4042–4047.
5. Shutin, D.; Bondarenko, M.; Polyakov, R.; Stebakov, I.; Savin, L. Prediction of remaining useful life of passive and adjustable fluid film bearings using physics-based models. In Proceedings of the 11th IFToMM International Conference on Rotordynamics, Beijing, China, 18–21 September 2023; International Conference on Rotor Dynamics. Springer: Cham, Switzerland, 2023; pp. 211–223.
6. Protopapadakis, G.; Apostolidis, A.; Kalfas, A.I. Explainable and interpretable AI-assisted remaining useful life estimation for aeroengines. In *Turbo Expo: Power for Land, Sea, and Air*; ASME: Rotterdam, The Netherlands, 2022; Paper V002T05A002.
7. Zhang, Y.; Feng, F.; Wang, S.; Meng, J.; Xie, J.; Ling, R.; Yi, H.; Zhang, K.; Chai, Y. Joint nonlinear-drift-driven Wiener process-Markov chain degradation switching model for adaptive online predicting lithium-ion battery remaining useful life. *Appl. Energy* **2023**, *341*, 121043. [\[CrossRef\]](#)
8. Zhang, Y.; Feng, K.; Ji, J.C.; Yu, K.; Ren, Z.; Liu, Z. Dynamic model-assisted bearing remaining useful life prediction using the cross-domain transformer network. *IEEE/ASME Trans. Mechatron.* **2022**, *28*, 1070–1080. [\[CrossRef\]](#)
9. Alfarizi, M.G.; Tajiani, B.; Vatn, J.; Yin, S. Optimized random forest model for remaining useful life prediction of experimental bearings. *IEEE Trans. Ind. Inf.* **2022**, *19*, 7771–7779. [\[CrossRef\]](#)
10. Guo, L.; Li, N.; Jia, F.; Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **2017**, *240*, 98–109. [\[CrossRef\]](#)
11. Schmidt, R.M. Recurrent neural networks (RNNs): A gentle introduction and overview. *arXiv* **2019**, arXiv:1912.05911. [\[CrossRef\]](#)
12. Chui, K.T.; Gupta, B.B.; Vasant, P. A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine. *Electronics* **2021**, *10*, 285. [\[CrossRef\]](#)
13. Cao, D.; Li, H.; Zhang, N.; Zhang, C. A health indicator construction method based on unsupervised parallel multiscale neural networks. In Proceedings of the 2022 Global Reliability and PHM (PHM-Yantai), Yantai, China, 13–16 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–7.

14. Duan, Y.; Liu, Z.; Li, H.; Zhang, C.; Zhang, N. A hybrid-driven remaining useful life prediction method combining asymmetric dual-channel autoencoder and nonlinear Wiener process. *Appl. Intell.* **2023**, *53*, 25490–25510. [\[CrossRef\]](#)
15. Song, L.; Jin, Y.; Lin, T.; Zhao, S.; Wei, Z.; Wang, H. Remaining useful life prediction method based on the spatiotemporal graph and GCN nested parallel route model. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 3511912. [\[CrossRef\]](#)
16. Namdari, A.; Samani, M.A.; Durrani, T.S. Lithium-ion battery prognostics through reinforcement learning based on entropy measures. *Algorithms* **2022**, *15*, 393. [\[CrossRef\]](#)
17. Yao, J.; Lu, B.; Zhang, J. Tool remaining useful life prediction using deep transfer reinforcement learning based on long short-term memory networks. *Int. J. Adv. Manuf. Technol.* **2022**, *118*, 1077–1086. [\[CrossRef\]](#)
18. Demkowicz, L.; Gopalakrishnan, J. Discontinuous Petrov–Galerkin (DPG) method. In *Encyclopedia of Computational Mechanics*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2017; pp. 777–792.
19. Hao, S.; Zheng, J.; Yang, J.; Sun, H.; Zhang, Q.; Zhang, L.; Jiang, N.; Li, Y. Deep reinforcement learning for joint optimization of condition-based maintenance and spare ordering. *Inf. Sci.* **2023**, *634*, 85–100. [\[CrossRef\]](#)
20. Zheng, G.; Li, Y.; Zhou, Z.; Yan, R. A remaining useful life prediction method of rolling bearings based on deep reinforcement learning. *IEEE Internet Things J.* **2024**, *11*, 22938–22949. [\[CrossRef\]](#)
21. Hu, Q.; Zhao, Y.; Wang, Y.; et al. Remaining useful life estimation in prognostics using deep reinforcement learning. *IEEE Access* **2023**, *11*, 32919–32934. [\[CrossRef\]](#)
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*; MIT Press: Long Beach, CA, USA, 2017; Volume 30.
23. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [\[CrossRef\]](#)
24. Borst, N.; Verhagen, W.J.C. Introducing CNN-LSTM network adaptations to improve remaining useful life prediction of complex systems. *Aeronaut. J.* **2023**, *127*, 2143–2153. [\[CrossRef\]](#)
25. Deng, F.; Chen, Z.; Hao, R. Research on bearing remaining useful life prediction based on an MsTCN-Transformer model. *J. Vib. Shock* **2024**, *43*, 279–287.
26. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries. *IEEE Trans. Ind. Electron.* **2017**, *65*, 6730–6739. [\[CrossRef\]](#)
27. Francois-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An introduction to deep reinforcement learning. *Found. Trends Mach. Learn.* **2018**, *11*, 219–354. [\[CrossRef\]](#)
28. Ma, T.; Xu, J.; Li, R.; Yao, N.; Yang, Y. Online short-term remaining useful life prediction of fuel cell vehicles based on cloud system. *Energies* **2021**, *14*, 2806. [\[CrossRef\]](#)
29. Zha, W.; Ye, Y. An aero-engine remaining useful life prediction model based on feature selection and the improved TCN. *Frankl. Open* **2024**, *6*, 100083. [\[CrossRef\]](#)
30. Ragab, M.; Chen, Z.; Wu, M.; Foo, C.S.; Kwok, C.K.; Yan, R.; Li, X. Contrastive adversarial domain adaptation for machine remaining useful life prediction. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5239–5249. [\[CrossRef\]](#)
31. Li, S.; Zhang, C.; Liu, L.; Zhang, X. Gated transient fluctuation dual attention unit network for long-term remaining useful life prediction of rotating machinery using IIoT. *IEEE Internet Things J.* **2024**, *11*, 18593–18604. [\[CrossRef\]](#)
32. Yu, W.; Kim, I.I.Y.; Mechefske, C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mech. Syst. Signal Process.* **2019**, *129*, 764–780. [\[CrossRef\]](#)
33. Li, Y.; Chen, Y.; Hu, Z.; Zhang, H. Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. *Reliab. Eng. Syst. Saf.* **2023**, *229*, 108869. [\[CrossRef\]](#)
34. Philip, J. CNN-LSTM Hybrid Deep Learning Model for Remaining Useful Life Estimation. *arXiv* **2024**, arXiv:2412.15998. [\[CrossRef\]](#)
35. Wahid, A.; Breslin, J.G.; Intizar, M.A. TCRSCANet: Harnessing temporal convolutions and recurrent skip component for enhanced RUL estimation in mechanical systems. *Hum.-Centric Intell. Syst.* **2024**, *4*, 1–24. [\[CrossRef\]](#)
36. Duan, Y.; Li, H.; Zhang, N. Mechanical health indicator construction and similarity remaining useful life prediction based on natural language processing model. *Meas. Sci. Technol.* **2022**, *33*, 094008. [\[CrossRef\]](#)

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.