



CARLE: a hybrid deep-shallow learning framework for robust and explainable RUL estimation of rolling element bearings

Waleed Razzaq¹ · Yun-Bo Zhao¹ 

Received: 15 July 2025 / Accepted: 2 October 2025

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract

Prognostic health management (PHM) systems have extensive applications in industry for monitoring and predicting the health status of equipment. Remaining Useful Life (RUL) estimation stands out as one important part of a PHM system that predicts the remaining operational lifespan of mechanical systems or their components, such as rolling element bearings, which account for a high proportion of machinery failures. Although many methods for RUL estimation have been developed, there are some challenges in terms of generalizability and robustness under dynamic operating conditions. This paper introduces the CARLE AI framework, which integrates advanced deep learning architectures with shallow machine learning technique to overcome these limitations. CARLE integrates Res-CNN and Res-LSTM blocks with multi-head attention and residual connections to capture spatial and temporal degradation trends coupled with Random Forest Regression (RFR) for robust and accurate predictions. We further propose a compact feature extraction framework that implements Gaussian filtering for efficient noise reduction and Continuous Wavelet Transform (CWT) for time–frequency feature extraction. We assessed the effectiveness of the proposed framework via the XJTU-SY and PRONOSTIA bearing datasets. Ablation experiments were conducted to assess the contribution of each component within CARLE, whereas noise experiments evaluated its resilience to noise. Cross-domain validation experiments were performed to examine the model’s generalizability across multiple domains. Additionally, comparative analyses with several state-of-the-art methods under dynamic operating conditions demonstrated that CARLE outperformed competing approaches, particularly in terms of generalizability to unseen scenarios. Furthermore, we discuss the reliability and trustworthiness of this framework via multiple state-of-the-art explainable AI (XAI) techniques, i.e., LIME and SHAP.

Keywords Prognostics Health Management (PHM) · Remaining Useful Life (RUL) · CNN · LSTM · XAI

1 Introduction

Prognostic Health Management (PHM) systems play crucial roles in industries as they monitor and predict equipment health conditions to prevent severe operational safety hazards and ensure accident-free processes. One salient feature of PHM systems is Remaining Useful Life (RUL) estimation, which concentrates on estimating the remaining effective lifespan of machinery or its components.

Rotational machinery is more prone to failure because of the availability of rolling-element bearings working under aggressive environments. It has been estimated that 40 to 50% of machinery failures can be attributed to these bearings Zhuang et al. (2021). Therefore, an accurate RUL estimation system for rolling-element bearings is essential for monitoring degradation, mitigating risks, and preventing unexpected breakdowns. Recently, various methods have been developed for this purpose and can generally be divided into physics-based and data-driven models.

Physics-based models provide insights into the degradation processes of bearings via a set of equations derived from mathematical representations of physical systems. Guo et al. (2015) proposed a physics-based model for bearing degradation based on Hertzian contact theory and material fatigue that effectively predicts nonlinear degradation under varying operational conditions. Wu et al. (2016) proposed

✉ Yun-Bo Zhao
ybzha@ustc.edu.cn

Waleed Razzaq
waleed.razzaq@mail.ustc.edu.cn

¹ School of Automation, University of Science and Technology China, Hefei, Anhui, China

a model with elastic deformation and stress distribution in ball bearings for simulating the initiation and development of spalls to show the merits of contact mechanics in understanding the early evolution of faults. Although these methods have achieved notable accomplishments, they require broad interdisciplinary knowledge and depend upon complicated mathematical modeling.

Data-driven methods uncover the hidden relationships within condition monitoring data. Further, it can be divided into two subcategories: shallow machine learning and deep learning. Bienefeld et al. (2022) explored Radom Forest (RF) performance in RUL estimation of rolling-element bearings using an extended feature engineering strategy involving the time domain, frequency domain, and statistical features extracted from vibrational signals. Zhang et al. (2021) proposed a Relevance Vector Machine RVM-coupled method that integrates the advantages of health indication fusion to create one unified health indicator out of a set of vibrational and temperature-motivated features. The number of developments in monitoring data acquisition continues to increase significantly, making meaningful feature extraction of monitored multisensory data even more crucial for RUL estimation. However, most shallow machine learning algorithms have notable limitations in dealing with big data in terms of prediction accuracy and computational efficiency.

Deep learning architectures are designed to capture and represent rich patterns in big data through the composition of a neural network made of multiple hidden layers composed of perceptrons. Advanced deep learning algorithms, including CNN Alzubaidi et al. (2021), recurrent networks such as LSTM Hochreiter (1997) and GRU Chung et al. (2014), and attention mechanisms Vaswani (2017) have proven highly efficient in uncovering hidden relationships within big data learning for RUL estimation of rolling element bearings. Li et al. (2019) proposed a CNN-based approach using vibrational signal spectrograms and demonstrated very good performance, thus proving its ability to learn nonlinear degradation trends distinguishing subtle data variations in data. However, CNNs struggle to model temporal degradation trends and long-term time dependencies within big data, limiting their real-world applicability. Zhang et al. (2018) utilized an LSTM-based network that effectively models long-term dependencies and captures temporal degradation features within massive datasets; however, its sensitivity to hyperparameters, overfitting and lack of noise handling limit its accuracy. Li et al. (2023) proposed a GRU-based DeepAR network that was efficient in modeling temporal dependencies with parameters and an adaptive failure threshold. However, it is sensitive to noise and often requires careful tuning in complex cases. Deng et al. (2023) presented a calibrated hybrid transfer learning framework including a dynamic rolling bearing model,

particle filter-based calibration, and a physics-informed Bayesian deep dynamic network for improving fidelity. However, it is still computationally intensive and has limited applicability in real-world conditions. Zhao et al. (2023) proposed Multiscale Integrated Self-Attention that performs with multisensory degrading data at various scales by employing a multiscale CNN block including a self-attention mechanism, a recurrent network module and feature fusion to extract multisensory-temporal features on the basis of their relationships and integrate them via mutual interaction. Although this approach improves prediction accuracy through an efficient loss function, it is hindered by varying sensor quality and data noise.

In addition to the individual limitations mentioned above, several other common challenges demand attention. Most of the methods reported in the literature are task-oriented, diminishing their real-world applicability for many industrial machinery operations where real conditions are highly variable. The second significant limitation concerns the generalizability and robustness of RUL prediction systems, which heavily depend on effective feature extraction. Most existing approaches do not have a robust and compact framework for feature engineering; hence, they have limited reliability when dealing with big data. Another limitation concerns the fact that they are not transparent. Predictions are given in a black-box way, without underlining any factors of rationale that may contribute to supporting such an outcome. Therefore, the inability of the data-driven RUL model to offer interpretability or explainability raises concerns regarding dependability and trust.

Given these drawbacks, we propose a causal RUL estimation system that learns from one working condition and generalizes its learning to others. We aim to achieve this goal by designing a compact feature extractor framework that accounts for noise and provides a concise feature vector for the AI system. For the AI system, we introduce CARLE (Deep Ensemble Residual Convolutional-Attention LSTM Network) consisting of four distinct network blocks: *Res-CNN block*, *Res-LSTM block*, *Linear block* and *ML block*. The Res-CNN block comprises several convolutional layers that extract spatial degradation trends from the input vector. These features are passed to a multi-head attention mechanism (MHA) that selects the most relevant spatial features, suppresses redundant features, and enables differential treatment of features by scanning global information. The output is subsequently fed into the Res-LSTM network to capture temporal dependencies and long-term relationships between features. Residual connections between the CNN and LSTM layers are introduced to increase the robustness and generalizability of the system while also easing the computational complexity associated with each architecture. Several linear layers are introduced in the Linear block

to recognize patterns and generate a logit vector, which serves as input for the ML block that contains the Random Forest Regression (RFR) for the final prediction. We validate the performance of the system on the XJTU-SY and PRONOSTIA bearing datasets. We also discuss the trustworthiness of the AI framework via state-of-the-art explainable AI (XAI) techniques called LIME and SHAP, which allow us to assess whether the output prediction is reliable. The highlights of this research are listed below.

1. A compact time-frequency feature extraction framework is designed to handle noise via a Gaussian filter and to extract diverse features from multichannel sensory data in both the time and frequency domains using Continuous Wavelet Transform (CWT).
2. A novel CARLE AI system is designed for rolling-element bearings RUL estimation. The system ensemble the pattern-learning strength of multiple deep-learning architectures with the generalizability and robustness of shallow machine-learning algorithm.
3. The effectiveness of the algorithm is validated on the XJTU-SY and PRONOSTIA bearing degradation datasets, which include data from multiple operating conditions.
4. The reliability and trustworthiness of the proposed black-box framework are analyzed through multiple state-of-the-art XAI techniques, i.e., LIME and SHAP.

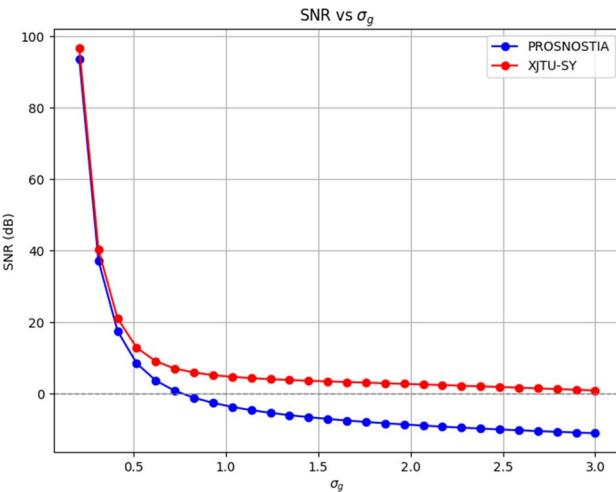


Fig. 1 Signal-to-Noise ratio (SNR) analysis w.r.t σ_g . The analysis shows the balance between noise reduction and signal preservation. When no smoothing is applied, the SNR remains high due to preservation of the original signal's fidelity. As the smoothing parameter increases, the filtering mechanism effectively reduces high-frequency noise. However, the process simultaneously diminishes the finer details and dynamic components of the signal, resulting in a rapid decline in the SNR. After a certain smoothing intensity, noise reduction occurs at the cost of negligible signal distortion ($\sigma_g \approx 0.75$ for XJTU-SY and $\sigma_g \approx 1.1$ for PRONOSTIA). This stabilization point signifies an optimal parameter range where the balance between noise suppression and signal integrity is achieved

The remainder of the paper is organized as follows: Section 2 outlines the methodology and algorithms utilized in the research. Section 4 presents the experimental results and analysis of the proposed framework. Section 5 concludes the research by discussing potential future work and areas for improvement. The appendix presents an overview of the foundational elements of the proposed framework, including a discussion on the feature extraction algorithms and training setup with hyperparameters of our implementation.

2 Time-frequency feature extraction framework

The monitoring data from rolling element bearings typically consist of signals from multiple sensors, often exhibiting time-varying characteristics with perturbations, primarily thermal noise caused by changing operating conditions and temperature variations. To ensure effective analysis, a compact feature engineering preprocessing step is essential to filter out these perturbations before training the AI system (see Algorithm 1). Otherwise, the perturbations can significantly degrade the performance of the system. Assuming that the number of sensors is N_s and the data length is L_d , the raw data are represented as:

$$I = [i_1, i_2, \dots, i_{L_d}], \quad i_k = [n_k^1, n_k^2, \dots, n_{N_c}^k] \quad (1)$$

where i_k represents the sensor reading at timestep k for N_c channels. We filtered out the obtained raw data via Gaussian filter to smooth the edges and reduce short-term fluctuations. The choice of Gaussian filter is motivated by its effectiveness in reducing Gaussian noise while preserving signal edges, offering computational efficiency and reliable smoothing compared to the Fourier transform Bracewell (1989) or EMD methods Stallone et al. (2020). Let the filtered signal be denoted as $I_f(t)$. The smooth signal is obtained by convolving the raw signal with the Gaussian filter ($G(x)$) (see Appendix A), represented mathematically as:

$$I_f(t) = \int_{-\infty}^{\infty} I(\tau)G(t - \tau) d\tau \quad (2)$$

The value of the standard deviation σ_g for the Gaussian filter plays a critical role in this process. It controls the degree of smoothing applied to the signal. After experimenting with various values and analyzing the signal-to-noise (SNR) (see Fig. 1), we find the optimal balance for our use cases, effectively filtering out noise and thermal perturbations while preserving critical information essential for RUL estimation. The filtered signal $I_f(t)$ is then forwarded to the CWT (see Appendix A) for feature extraction. However, before

applying the CWT, the signal is divided into smaller segments using a windowing technique. The window operation can be represented as:

$$i_w(t) = I_f(t) \cdot w_i(t) \quad (3)$$

where $w_i(t)$ is the window function with window length T_w for the i -th segment, defined as:

$$w_i(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1} + T_w] \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

By breaking down the signal into smaller segments, the CWT ensures that localized time–frequency features are captured, which is vital for accurately modeling degradation trends for accurate RUL estimation. The CWT can then be mathematically computed as:

$$\Gamma_{iw}(a, b) = \int_{-\infty}^{\infty} i_w(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (5)$$

where $\Gamma_{iw}(a, b)$ represents the wavelet coefficients of the windowed signal and where ψ is the Morlet wavelet. To extract meaningful features from the CWT, it is critical to carefully select the frequency range of interest (f_{\min}, f_{\max}), as this range defines the scale range of the CWT. The choice of these frequencies is informed by the system’s operational condition f_o , allowing the model to accommodate multiple scenarios effectively. In our implementation, we considered up to the third harmonic, providing a good balance between computational efficiency and capturing useful features. The frequency bounds are as follows:

$$f_{\min} \approx \frac{f_o}{3}, \quad f_{\max} \approx 3f_o \quad (6)$$

The corresponding wavelet transform scales can be calculated as:

$$a_{\min} = \frac{f_c}{f_{\max} \cdot T_{\text{sampling}}}, \quad a_{\max} = \frac{f_c}{f_{\min} \cdot T_{\text{sampling}}} \quad (7)$$

where $T_{\text{sampling}} = 1/f_{\text{sampling}}$ is the period of the sampled vibrational signal and f_c is the central frequency of the Morlet wavelet, typically chosen as $f_c = 0.81$, to govern the trade-off between time and frequency resolutions. To ensure comprehensive coverage of the frequency range, logarithmically spaced scales are used:

$$a_i \in [a_{\min}, a_{\max}], \quad i = 1, 2, \dots, N \quad (8)$$

where N is the number of scales selected on the basis of the desired resolution in the time–frequency domain. Figure 2 presents the visual representation of the compact feature extractor framework. The following time–frequency representation (TFR) features are derived to characterize the system’s physical state:

- **Energy (E):** represents the vibrational activity of the system. A continuous increase in energy typically correlates with progressive wear or distributed fatigue within the system, often evident as surface pitting. In contrast, sudden spikes indicate localized defects, such as spalling or crack propagation Randall and Antoni (2011); Smith and Randall (2015). Lubrication failures contribute to significant fluctuations, primarily due to the occurrence of intermittent metal-to-metal contact, whereas contamination, such as ingress of debris, results in transient energy spikes. The energy is computed as:

$$E = \sum_{m=1}^M |\Gamma_{iw}(a, b)|^2 \quad (9)$$

- **Dominant frequency (f_d):** corresponds to the frequency at which the systems exhibit the highest energy concentration. Shifts in f_d can serve as a diagnostic tool for identifying specific faults within the system. Alignments with bearing fault frequencies, such as the ball pass frequency, are indicative of localized defects, commonly in the form of inner or outer race cracks (BPFO/BPFI) Borghesani et al. (2013); Tandon and Choudhury (1999). The presence of subharmonic components in f_d suggests potential issues such as looseness or imbalance within the system. Broadband frequency-domain profiles are characteristics of chaotic faults, which are typically associated with lubrication failures or contamination, as they introduce fluctuations in the system’s behavior. The dominant frequency is calculated as:

$$f_d = a_{\text{scale}}(\text{argmax}(E)) \quad (10)$$

- **Entropy (h):** measures the vibrational randomness within the system. Elevated entropy values suggest non-stationary defects, such as irregular spalling or looseness. In the case of lubrication failure, the entropy increases due to erratic friction, while corrosion-related damage leads to increased entropy through surface interactions. Early-stage fatigue typically indicates low entropy, which escalates as the degradation process becomes more chaotic. The entropy is calculated as:

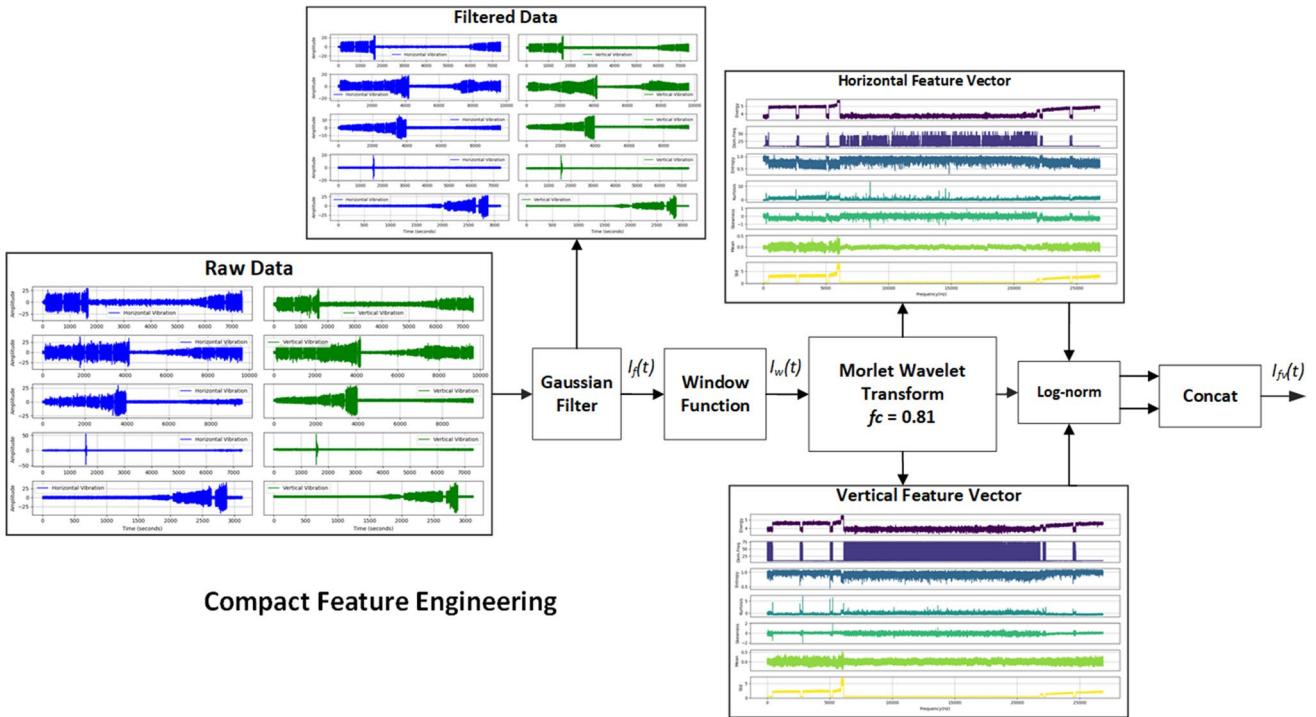


Fig. 2 Schematic diagram of the compact feature extractor framework

$$h = - \sum_{i=1}^K P(i_w) \log P(i_w) \quad (11)$$

- **Kurtosis (K):** detects transient impacts by analyzing extreme deviations in the signal distribution. The highest kurtosis values are typically associated with localized defects, including fatigue cracks, electrical pitting, and particle collisions caused by contamination Antoni (2006). Kurtosis is calculated as follows:

$$K = \frac{\mathbb{E}[(i_w - \mu)^4]}{\sigma^4} \quad (12)$$

- **Skewness (s_k):** measures the asymmetry in the distribution of signal data. Positive skewness typically indicates unidirectional impacts, such as brinelling, while negative skewness suggests repetitive low-energy events, like the initiation of cracks. Asymmetric wear patterns resulting from thermal warping or corrosion also manifest as deviations in skewness, highlighting an imbalance in the system's behavior. The skewness is calculated as:

$$s_k = \frac{\mathbb{E}[(i_w - \mu)^3]}{\sigma^3} \quad (13)$$

- **Mean (μ):** The mean vibrational level serves as a baseline indicator of system behavior. A gradual increase in the mean is often associated with distributed wear processes, such as corrosion or thermal degradation, whereas a sudden shift typically signals more severe faults, such as cage features. Lubrication failures can elevate the mean due to an increase in friction in the system. The mean is calculated as:

$$\mu = \frac{1}{N} \sum_{m=1}^M i_w(m) \quad (14)$$

- **Standard deviation (σ):** represents the variability of the signal. High values indicate unstable faults such as looseness or contamination, which cause erratic behavior. Conversely, fatigue cracks contribute to increased variability during intermittent spalling events, indicating ongoing damage and instability in the system. The standard deviation is calculated as:

$$\sigma = \sqrt{\frac{1}{M} \sum_{i=1}^m (i_w(m) - \mu)^2} \quad (15)$$

Fig. 3 The schematic diagram of the CARLE AI system

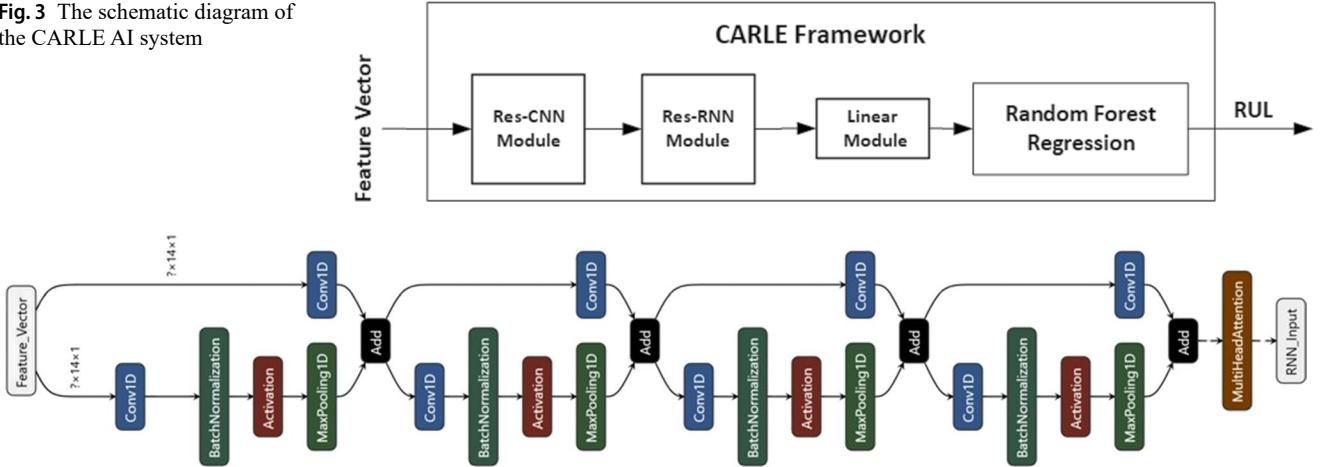


Fig. 4 Schematic diagram of the Res-CNN block

While many existing approaches Lei et al. (2007); Wenjian et al. (2024); Abdellatif et al. (2025); Abdellatif et al. (2025) use 20 or more TFR features, we extract only seven physically meaningful features, reducing offline computation time by on average 66%. Integrating these features, such as transient detection through K and h , with long-term trend analysis via μ and σ can enhance RUL estimation. An increase in μ with intermittent spikes in K indicates progressive wear punctuated by transient damage events. This allows for adaptive RUL updates that account for both ongoing wear and irregular fault occurrences. Similarly, the chaotic behavior observed in s_k and shifts in f_d improve prognostic accuracy by isolating fault-specific degradation pathways, allowing for a more precise RUL.

3 CARLE framework

We propose CARLE (Deep Ensemble Residual Convolutional-Attention LSTM Network) for the accurate RUL estimation in rolling element bearings. Unlike stacking-based ensembles that primarily combine base learners Ture et al. (2024), CNN-Bi-LSTM approaches designed around predictive maintenance policies Wang et al. (2024), or data fusion methods with stage division Li et al. (2024), CARLE integrates residual CNNs, attention-driven LSTMs, and Random Forest Regression into a single unified framework. This design preserves spatial-temporal degradation features and enhances adaptability to unseen operating conditions,

Require: Non-stationary vibrational signal ($I(t)$), σ_g , number of sensors (N_s), window length (T_w), sampling frequency (f_s),

- 1: Initialize Gaussian filter: $G(t) = \frac{1}{\sqrt{2\pi\sigma_g^2}} e^{-\frac{t^2}{2\sigma_g^2}}$
- 2: Calculate filtered signal: $I_f(t) = \int_{-\infty}^{\infty} I(\tau)G(t-\tau) d\tau$
- 3: Initialize window function: $w(t) \leftarrow$ Equation 4
- 4: Initialize central frequency: $f_c = 0.81$
- 5: $a_{\min}, a_{\max} \leftarrow$ Equation 7
- 6: $a_{scale} \leftarrow$ Equation 8
- 7: **for** $n = 1 \dots N_s$ **do**
- 8: **for** $k = 1 \dots \text{len}(I_f(t)) - T_w$ **do**
- 9: Compute window signal: $i_w(k) \leftarrow$ Equation 3.
- 10: Compute wavelet coefficients: $\Gamma_{i_w}(a, b) \leftarrow$ Equation 5
- 11: Compute energy: $E_n \leftarrow$ Equation 9
- 12: Compute dominant frequency: $f_{d_n} \leftarrow$ Equation 10
- 13: Compute entropy: $h_n \leftarrow$ Equation 11
- 14: Compute kurtosis: $k_n \leftarrow$ Equation 12
- 15: Compute skewness: $sk_n \leftarrow$ Equation 13
- 16: Compute mean: $\mu_n \leftarrow$ Equation 14
- 17: Compute standard deviation: $\sigma_n \leftarrow$ Equation 15
- 18: $I_{fv_n} \leftarrow [\log(E_n), f_{d_n}, h_n, k_n, sk_n, \mu_n, \sigma_n]$
- 19: **end for**
- 20: **end for**
- 21: $I_{fv} = \text{Concat}(I_{fv_1}, I_{fv_2} \dots I_{fv_{N_s}})$
- 22: **return** I_{fv}

Algorithm 1 Time–frequency Feature Extraction Framework

Fig. 5 Schematic diagram of the Res-RNN block

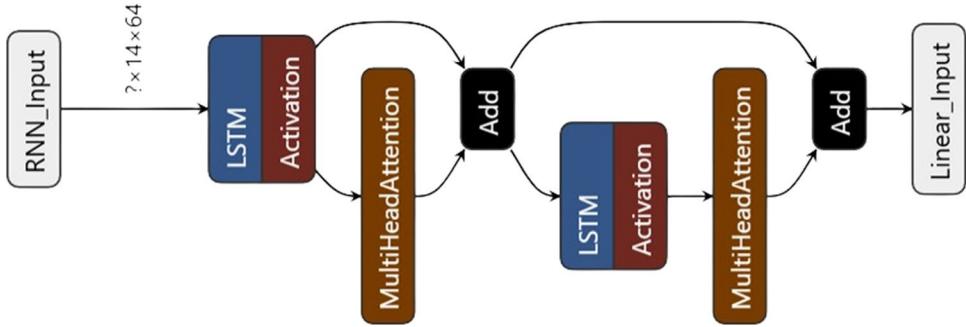
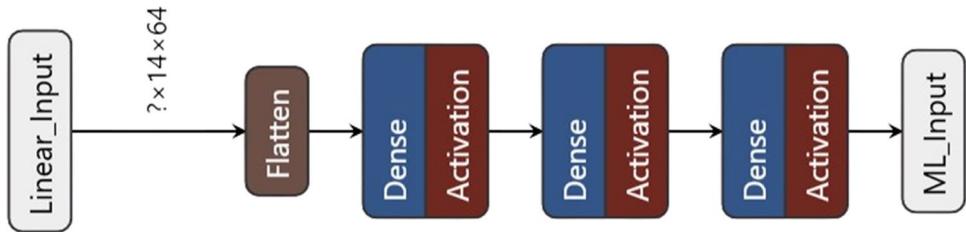


Fig. 6 The schematic diagram of Linear block



providing broader generalization across diverse requirements. A schematic diagram of CARLE is shown in Fig. 3. The CARLE architecture comprises four interconnected blocks:

Res-CNN Block: receives the input feature vector (I_{f_v}) and processes it through multiple convolutional heads, each employing distinct filter and kernel sizes to extract salient degradation features. The MHA mechanism is incorporated at the output to enhance feature selection, allowing the model to prioritize relevant degradation features while minimizing redundant information. Additionally, residual connections are integrated to facilitate identity mapping, ensuring that vital features are retained and propagated throughout the network. This helps maintain accuracy in RUL predictions as the complexity increases. The schematic diagram of the Res-CNN is shown in Fig. 4.

Res-RNN Block: receives the spatial degradation trends from the Res-CNN and processes them through a series of LSTM layers to capture the temporal characteristics and long-term dependencies inherent in the degradation features. Similar to the CNN block, a multi-head attention mechanism and residual connections are incorporated to enhance the focus on significant features and preserve critical information across layers. The schematic diagram of the Res-RNN is shown in Fig. 5.

Linear Block: consists of a series of fully connected layers tasked with recognizing patterns within the temporal degradation features, enabling the model to generalize effectively across diverse, unseen operating conditions. The output is a logit vector, which serves as input for the subsequent prediction mechanism. The schematic diagram of the Linear block is shown in Fig. 6.

Machine Learning Block: The Random Forest Regression (RFR) model receives the logit vector from the linear block to enhance the generalization capabilities for new data, providing diverse perspectives and flexibility. RFR enhances generalization because it aggregates predictions from many decision trees trained on different subsets of the data and features. This ensemble averaging reduces overfitting, mitigates the effect of noise or outliers, and allows the model to capture diverse nonlinear relationships in the degradation features, making RUL predictions more robust to unseen operating conditions.

The stacking of these modules—CNN → Attention → LSTM → RFR—is deliberate. It reflects a layered processing approach: starting with low-level feature extraction, progressing to global pattern discovery, and concluding with structured temporal reasoning. This combination offers a comprehensive understanding of the degradation process, improving the robustness and accuracy of RUL predictions.

4 Experimental results and analysis

4.1 Dataset explanation

XJTU-SY dataset: developed through a collaboration between Xi'an Jiaotong University and Changxing Sumyoung Technology for experimentation and validation of RUL algorithms Wang et al. (2018). The dataset includes run-to-failure vibration data from 15 rolling element bearings obtained through accelerated degradation experiments under three distinct operational conditions: 1200 rpm (35 Hz) with a 12 kN radial load, 2250 rpm (37.5 Hz) with an 11 kN radial load, and 2400 rpm (40 Hz) with a 10 kN radial load. Vibration signals were captured via accelerometers

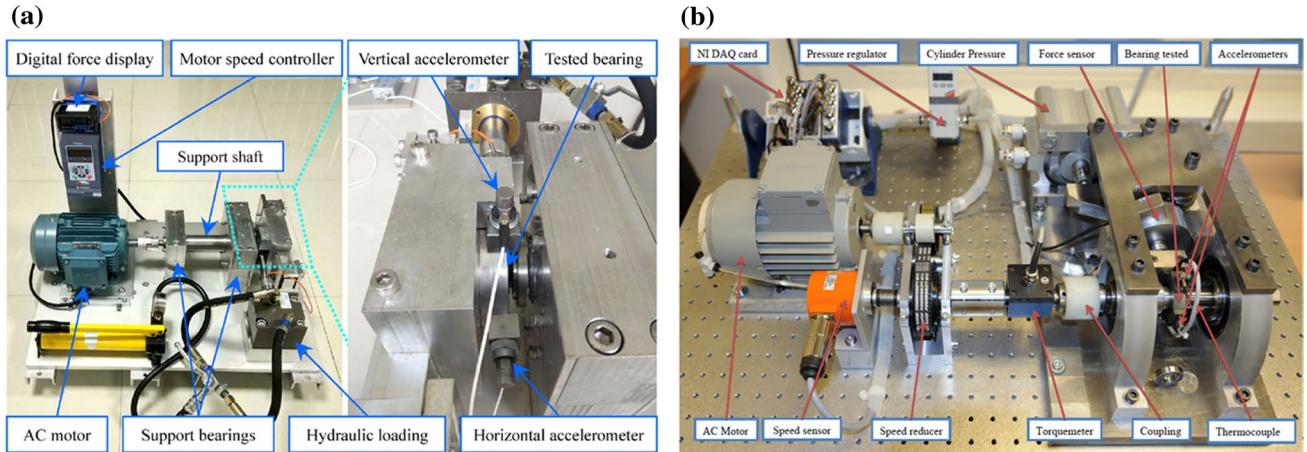


Fig. 7 **a** XJTU-SY testbed; **b** PRONOSTIA testbed for recording vibrational data

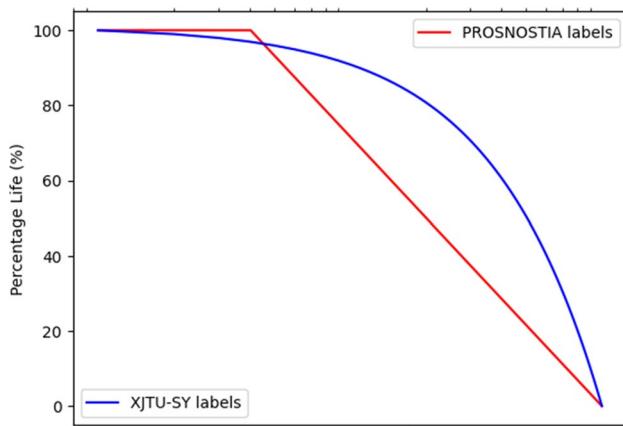


Fig. 8 RUL labels for both datasets

mounted on horizontal and vertical axes, sampled at a frequency (f_{sample}) of 25 kHz, and recorded at one-minute intervals, with each sample comprising 1.28 seconds of data. The experimental testbed is depicted in Fig. 7(a). For training, data from the $f_o = 35\text{Hz}$ condition (1200 rpm with a 12 kN load) were used, while validation focused on evaluating generalizability using data from the $f_o = 40\text{Hz}$ condition (2400 rpm with a 10 kN load) and $f_o = 37.5$ condition (2250 rpm with an 11 kN load).

PRONOSTIA dataset: is a benchmark dataset widely used for research in condition monitoring and RUL analysis of rolling element bearings; it was developed as part of the PRONOSTIA experimental platform Nectoux et al. (2012). The dataset provides 16 complete run-to-failure data collected under accelerated degradation conditions with three distinct operational conditions: 1800 rpm (100 Hz) with a 4 kN radial load, 1650 rpm (100 Hz) with a 4.2 kN radial load, and 1500 rpm (100 Hz) with a 5 kN radial load. Vibration signals were captured via accelerometers mounted on the horizontal and vertical axes and sampled at 25.6 kHz, whereas temperature data were sampled at 10 Hz. Figure

7(b) provides the testbed to capture the data. For training, we utilized 3 bearing data from 4kN operating conditions which is about 52% of total samples, and for validation, we focused on evaluating generalizability using data from 4.2 kN and 5 kN and ignored temperature data.

4.2 RUL labels

Generating RUL labels is a crucial step in estimating remaining useful life. Some studies assume degradation occurs at a constant rate Luo and Zhang (2022); Yong et al. (2024); Deng et al. (2023), but real-world conditions rarely follow a perfectly linear pattern. Instead, degradation often occurs in a nonlinear, piecewise manner, as suggested in other studies Zhao et al. (2023); Yin et al. (2025); Al-Dulaimi et al. (2019). To explore both possibilities, we created labels for the XJTU-SY dataset based on linear degradation models, visualized in Fig. 8 using a log scale for clarity. Since long-term monitoring data form a time series, the initial operation phase is typically stable, with minimal noticeable degradation. Therefore, for the PRONOSTIA dataset, we applied the nonlinear, piecewise degradation model shown in Fig. 8 to more accurately represent how bearing performance decreases over time.

4.3 Evaluation indicators

For evaluation, we utilized two metrics: the mean absolute error (MAE) and the root mean square error (MSE). A brief description of these metrics is as follows:

MAE: is widely used in RUL analysis to quantify the accuracy of predictive models. It measures the average magnitude of absolute errors between the predicted RUL (y_i) and the true RUL (\hat{y}_i), regardless of direction. The mathematical expression is as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

where n is the total number of predictions. The MAE is particularly suitable for RUL analysis because it equally penalizes overpredictions and underpredictions, ensuring an unbiased evaluation of the model's ability to estimate the RUL.

MSE: calculates the square root of the average squared differences between the predicted RUL (y_i) and the true RUL (\hat{y}_i). It is given by:

$$\text{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Owing to the squaring of differences, the MSE penalizes larger errors more heavily. This makes it sensitive to significant prediction deviations, emphasizing the model's ability to minimize large prediction errors.

Score is a metric specifically designed for RUL estimation in the IEEE PHM Nectoux et al. (2012) to score the estimates. The scoring function is asymmetric and penalizes overestimations more heavily than early predictions. This reflects practical considerations, as late maintenance prediction can lead to unexpected failures with more severe consequences than early intervention can.

$$\text{Score} = \sum_{i: \hat{y}_i < y_i} \left(e^{-\frac{\hat{y}_i - y_i}{13}} - 1 \right) + \sum_{i: \hat{y}_i \geq y_i} \left(e^{\frac{\hat{y}_i - y_i}{10}} - 1 \right) \quad (16)$$

4.4 Ablation experiments

Ablation experiments of CARLE were conducted to validate the effectiveness of each constituent of the architecture. We compared CARLE against its three variants: CARL without ensemble learning, CRLE without MHA, and CALE without residual connections. We noticed that CARLE and CALE performed very closely in terms of training operating conditions, but CARLE was marginally better than CALE. However, CARLE performed much better under unseen operating conditions, highlighting the role of residual connections in enhancing robustness. In contrast, both CRLE and CARL performed very poorly, with CARL being unsuitable for practical use. The ensemble machine learning approach yielded the most significant performance gains. A detailed comparison of both XJTU-SY and PRONOSTIA is shown in Fig. 9(a) and (b), while evaluation metrics are provided in Tables 1 and 2, respectively. For the sake of result explanations, we selected Bearing 3 under representative

operating conditions from each operating condition and dataset.

For the XJTU dataset:

- **35Hz12kN:** (Figure 9(a-iii)): CARLE achieved the lowest error with an MSE of 0.00220, MAE of 0.04087 and Score of 130.016. CALE followed closely with an MSE of 0.00265 ($\uparrow 16\%$), MAE of 0.04561 ($\uparrow 10\%$) and Score of 144.2532. CRLE recorded an MSE of 0.00275 ($\uparrow 20\%$), MAE of 0.04747 ($\uparrow 13\%$) and Score of 149.22, while CARL performed the worst with an MSE of 0.00806 ($\uparrow 72\%$), MAE of 0.07905 ($\uparrow 48\%$) and Score of 250.3705.
- **37.5Hz11kN:** (Figure 9(a-viii)): CARLE achieved an MSE of 0.01407, MAE of 0.10697 and Score of 1083.53. CALE showed slightly better MSE (0.01388, $\downarrow 1.3\%$) but nearly identical MAE (0.10701, $\uparrow 0.03\%$) with Score of 1081.08. CARL showed an MSE of 0.021 ($\uparrow 33\%$), MAE of 0.13195 ($\uparrow 19\%$) and Score of 1334.1443, while CRLE yielded an MSE of 0.02340 ($\uparrow 39.87\%$), MAE of 0.13731 ($\uparrow 22\%$) and Score of 1411.7924.
- **40Hz10kN:** (Figure 9(a-xiii)): CARLE maintained strong performance with an MSE of 0.03085, MAE of 0.15631 and Score of 331.6710. CALE demonstrated marginal improvements with an MSE of 0.02781 ($\downarrow 9.8\%$), MAE of 0.14869 ($\downarrow 4.8\%$) and Score of 323.87. Conversely, CARL and CRLE again exhibited degraded performance, recording MSEs of 0.05309 ($\uparrow 42\%$) and 0.05481 ($\uparrow 43\%$), MAEs of 0.20083 ($\uparrow 22\%$) and 0.20161 ($\uparrow 22.4\%$) and Score of 424.47 and 420.05, respectively.

For the PRONOSTIA dataset:

- **100Hz4kN:** (Figure 9(b-iii)): CARLE achieved superior performance with an MSE of 0.00029, MAE of 0.01312 and Score of 60.912. CALE showed reduced accuracy with an MSE of 0.00094 ($\uparrow 60\%$), MAE of 0.02538 ($\uparrow 48.3\%$) and Score of 64.2970. CRLE performed moderately, with an MSE of 0.00049 ($\uparrow 40\%$), MAE of 0.01723 ($\uparrow 23.8\%$) and Score of 59.89, while CARL reached an MSE of 0.00033 ($\uparrow 12.1\%$), MAE of 0.01294 ($\downarrow 1.2\%$) and Score of 64.2970.
- **100Hz4.2kN:** (Figure 9(b-x)): CARLE achieved an MSE of 0.00831, MAE of 0.07488 and Score of 72.5470. CALE yielded an MSE of 0.01240 ($\uparrow 32.9\%$), MAE of 0.09776 ($\uparrow 23.4\%$) and Score of 96.5710. Interestingly, CRLE outperformed CARLE here, recording an MSE of 0.00601 ($\downarrow 10\%$), MAE of 0.04195 ($\downarrow 56\%$) and Score of 66.5046. CARL also showed strong results, with an MSE of 0.00601 ($\downarrow 27.6\%$), MAE of 0.03408 ($\downarrow 54.4\%$) and Score of 229.629.

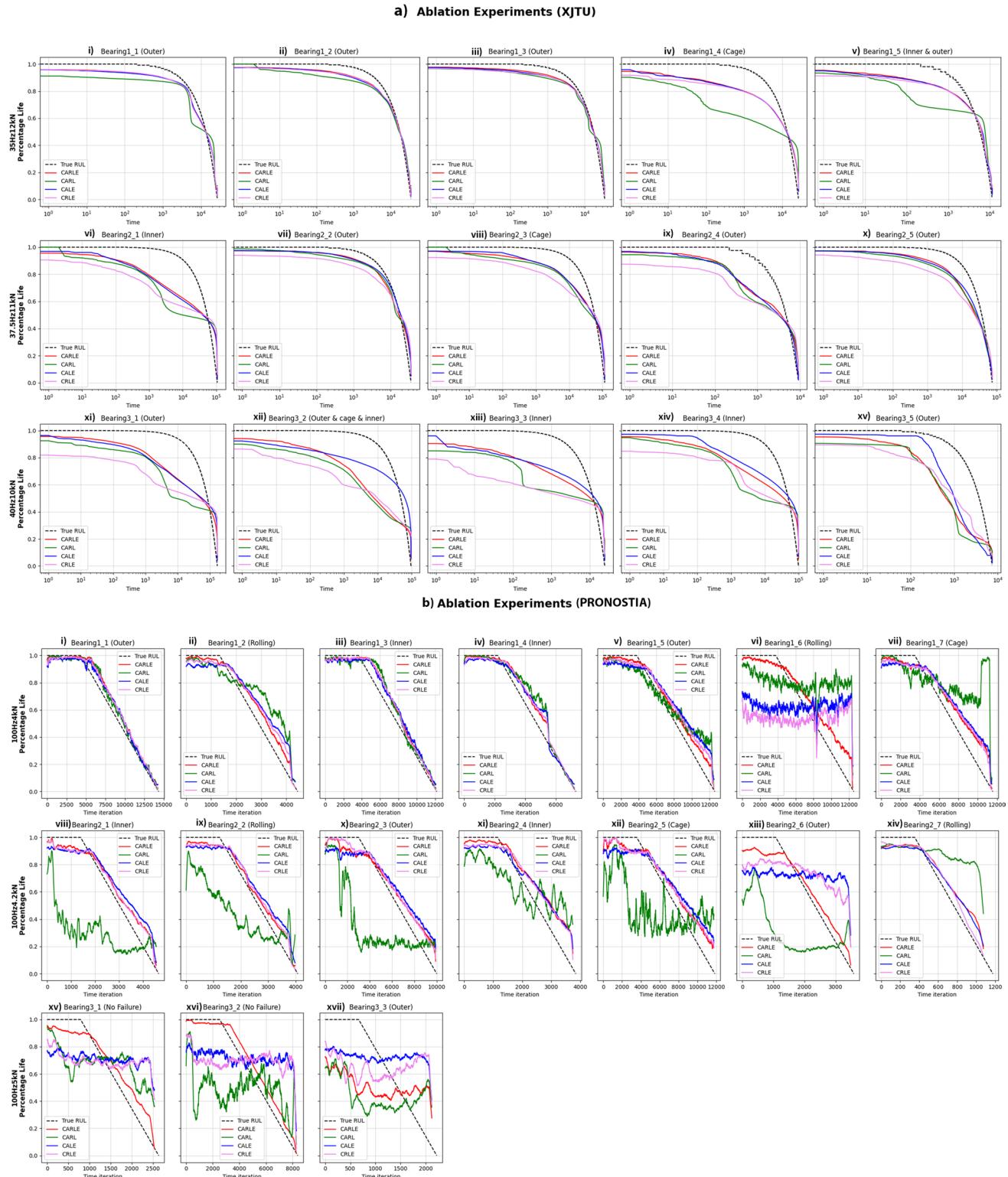


Fig. 9 Ablation experiment prediction **a** XJTU-SY; **b** PRONOSTIA with fault types

Table 1 Ablation experiment (XJTU-SY)

Bearing	Model	35Hz12kN			37.5Hz11kN			40Hz10kN		
		MSE	MAE	Score	MSE	MAE	Score	MSE	MAE	Score
Bearing 1	CARLE	0.00345	0.05157	122.5423	0.03273	0.16070	1505.8041	0.03314	0.15983	2269.5957
	CARL	0.01377	0.09911	234.7118	0.05943	0.21274	1986.9069	0.06752	0.22609	3222.6052
	CALE	0.00331	0.05126	121.5300	0.03630	0.16899	1581.9119	0.03954	0.17636	2564.5586
	CRLE	0.00398	0.05639	133.9703	0.04990	0.19688	1865.2695	0.05716	0.20893	3030.7470
Bearing 2	CARLE	0.00183	0.03692	114.4403	0.00671	0.07308	232.8929	0.06673	0.22032	1708.8516
	CARL	0.00376	0.05666	175.5158	0.00883	0.08308	267.3460	0.07617	0.23716	1859.4258
	CALE	0.00208	0.04082	125.9960	0.00786	0.06997	236.2859	0.02941	0.15117	1345.2411
	CRLE	0.00178	0.03694	113.9618	0.01698	0.11337	368.1130	0.06164	0.21395	1672.4370
Bearing 3	CARLE	0.00220	0.04087	130.0166	0.01407	0.10697	1083.5319	0.03085	0.15631	331.6710
	CARL	0.00806	0.07905	250.3705	0.02100	0.13195	1334.1443	0.05309	0.20083	424.4746
	CALE	0.00265	0.04561	144.2532	0.01388	0.10701	1081.0886	0.02781	0.14869	323.8718
	CRLE	0.00275	0.04747	149.2223	0.02340	0.13731	1411.7924	0.05481	0.20161	420.0580
Bearing 4	CARLE	0.01172	0.09653	225.5295	0.02149	0.12591	96.9172	0.03361	0.16221	1396.6897
	CARL	0.05099	0.19865	461.9145	0.02965	0.14793	114.1304	0.06197	0.21669	1859.8220
	CALE	0.01264	0.10096	236.8543	0.02065	0.11565	85.9529	0.03220	0.15930	1429.6296
	CRLE	0.01332	0.10426	245.6415	0.03441	0.16022	124.7697	0.05450	0.20399	1745.6810
Bearing 5	CARLE	0.00465	0.05938	59.9726	0.01373	0.09903	582.5476	0.09625	0.26167	154.0171
	CARL	0.02127	0.12677	128.7710	0.01256	0.09288	547.5211	0.11684	0.28730	169.0601
	CALE	0.00486	0.06090	61.3297	0.00985	0.08766	511.8520	0.08958	0.26006	151.7796
	CRLE	0.00537	0.06420	64.8331	0.02060	0.12428	750.2275	0.06565	0.22405	130.8896

Bold values indicate the minimum MSE, MAE, and Score for each bearing-condition combination

- **100Hz5kN:** (Figure 9(b-xvii)): CARLE recorded an MSE of 0.14125, MAE of 0.17514 and Score of 37.2298. CALE improved significantly, with an MSE of 0.02628 (↓81%), MAE of 0.14068 (↓19.6%) and Score of 30.7763. CRLE achieved an MSE of 0.04916 (↓60%), MAE of 0.17957 (↓2.4%) and 25.7763, while CARL showed an MSE of 0.06594 (↓53%) but a higher MAE of 0.22075 (↑26%) with Score of 40.9659.

These findings confirm that each architectural component within CARLE makes a meaningful contribution to the overall model performance. Ensemble learning, in particular, drives substantial accuracy gains, while residual connections and attention mechanisms further support model generalization, especially in complex or unseen operational settings.

4.5 Noise experiment

Noise experiments are crucial for evaluating the robustness and reliability of AI frameworks, particularly in real-world scenarios where data are affected by sensor noise, environmental variations, or system uncertainties. By introducing controlled noise into the input data, we can assess the model's stability and its ability to generalize beyond ideal conditions. In our experiments, Gaussian noise with a normal distribution ($\mu = 0, \sigma = 0.1$) was added to simulate typical sensor fluctuations. Additionally, salt-and-pepper noise was applied randomly to 10% of the data points, representing

sudden sensor failures. Results show that the model is largely resilient to Gaussian noise, with only minor performance degradation on the XJTU-SY dataset (Fig. 10(a)). Salt-and-pepper noise, however, causes a more significant performance drop, highlighting a potential limitation for real-world deployment where sensor spikes or dropouts can occur due to electrical interference, hardware faults, or communication errors. In the PRONOSITA evaluation (Fig. 10(b)), the impact of both noise types is more moderate, indicating that the model can still preserve long-term bearing degradation patterns. To mitigate the effect of salt-and-pepper noise in practice, preprocessing filters such as median or robust statistical filters can remove sudden spikes, sensor fusion can reduce the influence of any single faulty measurement, and training with noise-augmented data can help the model learn to ignore extreme outliers. Additionally, integrating lightweight anomaly detection modules could flag or correct extreme values in real time, ensuring more reliable RUL predictions under noisy conditions.

4.6 Cross-domain validation experiments

Cross-domain validation is crucial for assessing the generalizability of AI frameworks when applied to datasets with differing statistical distributions. It evaluates whether a model trained on one dataset can maintain predictive performance on another, thereby mitigating overfitting to a single domain and improving applicability in dynamic environments. We evaluate the PRONOSTIA-trained CARLE

Table 2 Ablation experiment (PRONOSTIA)

Bearing	Model	100Hz4kN			100Hz4.2kN			100Hz5kN		
		MSE	MAE	Score	MSE	MAE	Score	MSE	MAE	Score
Bearing 1	CARLE	0.00017	0.00890	67.7651	0.00687	0.06874	34.2385	0.03073	0.15232	22.3816
	CARL	0.00060	0.01944	52.1498	0.20664	0.39369	125.8891	0.01941	0.12100	26.8461
	CALE	0.00130	0.02515	53.3133	0.01075	0.08844	44.6632	0.02536	0.13877	34.7183
	CRLE	0.00055	0.01640	64.1635	0.00720	0.07223	34.3931	0.03786	0.17235	41.4252
Bearing 2	CARLE	0.00289	0.04268	28.5949	0.00406	0.05340	22.4782	0.04126	0.17514	52.6417
	CARL	0.01822	0.10301	53.7729	0.09584	0.26255	69.0456	0.06594	0.22073	114.1094
	CALE	0.01010	0.08341	44.0762	0.00706	0.06948	33.5543	0.02629	0.14069	112.8656
	CRLE	0.00643	0.06490	37.0388	0.00403	0.05156	26.3432	0.04196	0.17951	137.5010
Bearing 3	CARLE	0.00029	0.01312	60.9126	0.00831	0.07488	72.5470	0.14125	0.17514	37.2298
	CARL	0.00033	0.01294	53.8716	0.17432	0.34082	229.6209	0.065935	0.220728	40.9659
	CALE	0.00094	0.02538	64.2970	0.01240	0.09776	96.5710	0.02628	0.14068	30.4960
	CRLE	0.00049	0.01723	59.8920	0.00601	0.041959	66.5046	0.04916	0.17951	25.7763
Bearing 4	CARLE	0.00052	0.01635	35.9291	0.01035	0.07616	39.0356	—	—	—
	CARL	0.00237	0.03209	45.9774	0.02068	0.12315	34.4502	—	—	—
	CALE	0.00268	0.04036	36.9779	0.01279	0.09542	38.6783	—	—	—
	CRLE	0.00134	0.02676	37.9885	0.01029	0.08145	38.1909	—	—	—
Bearing 5	CARLE	0.00264	0.03956	69.1269	0.00711	0.06936	91.5711	—	—	—
	CARL	0.00885	0.07636	86.3878	0.08058	0.23612	178.1749	—	—	—
	CALE	0.00999	0.08285	116.8831	0.01398	0.10261	122.0432	—	—	—
	CRLE	0.00605	0.06274	94.7340	0.00878	0.07688	94.7391	—	—	—
Bearing 6	CARLE	0.02014	0.12283	97.9291	0.07713	0.23869	24.9688	—	—	—
	CARL	0.03497	0.13915	205.7567	0.20729	0.39659	98.0300	—	—	—
	CALE	0.03064	0.15598	151.3578	0.03074	0.15201	51.7674	—	—	—
	CRLE	0.05942	0.20851	184.1260	0.03314	0.15922	57.2092	—	—	—
Bearing 7	CARLE	0.00799	0.06877	101.4600	0.00495	0.06176	12.3018	—	—	—
	CARL	0.03480	0.13173	196.0093	0.06695	0.18207	26.0443	—	—	—
	CALE	0.01221	0.08969	122.1526	0.00929	0.08194	11.2305	—	—	—
	CRLE	0.00905	0.07806	108.2637	0.00355	0.05256	8.9416	—	—	—

Bold values indicate the minimum MSE, MAE, and Score for each bearing-condition combination

model on the XJTU-SY dataset, as both datasets share identical feature sets derived via Algorithm 1 but differ in label distributions. To address domain shift, we employ Principal Component Analysis (PCA) and Correlation Alignment (CORAL) for feature space alignment. The process involves feature extraction from both datasets, transformation via PCA, and distribution alignment using CORAL (see Fig. 11(a)) before generating predictions. Our analysis (see Fig. 11(b) and Table 3) indicates that the adapted methodology produces varying prediction accuracy, with notable differences between CORAL-aligned and non-aligned results. Specifically, the CORAL-aligned model achieved an MSE of 0.0961, MAE of 0.2803, and Score of 297.3991, whereas the non-aligned model achieved an MSE of 0.1049, MAE of 0.2919, and Score of 321.70. These discrepancies likely arise from residual differences in label distributions and unmodeled domain-specific variations. While the alignment approach improves feature consistency across datasets, the remaining prediction error suggests that further optimization is needed to enhance model robustness.

4.7 Comparison with baseline methods

To comprehensively evaluate the performance of CARLE, we conducted comparative experiments against several baseline methods, including CNN-LSTM Hu et al. (2024), CNN-BiLSTM Guo et al. (2023), and MSIDIN Zhao et al. (2023). For a fair comparison, all competing models were trained using feature vectors extracted by the proposed compact feature extractor framework. Additionally, hyperparameters for each method, including CARLE, were fine-tuned using Bayesian Optimization Snoek et al. (2012) with 150 search trials to ensure optimal performance. Under training operating conditions, most state-of-the-art models were able to estimate RUL with reasonable accuracy based on MSE and MAE metrics. However, CARLE consistently outperformed all other methods across both datasets, with particularly significant improvements observed under unseen operating conditions. Detailed comparison metrics for the XJTU-SY and PRONOSTIA datasets are provided in Tables 4 and 5, respectively. To further interpret these

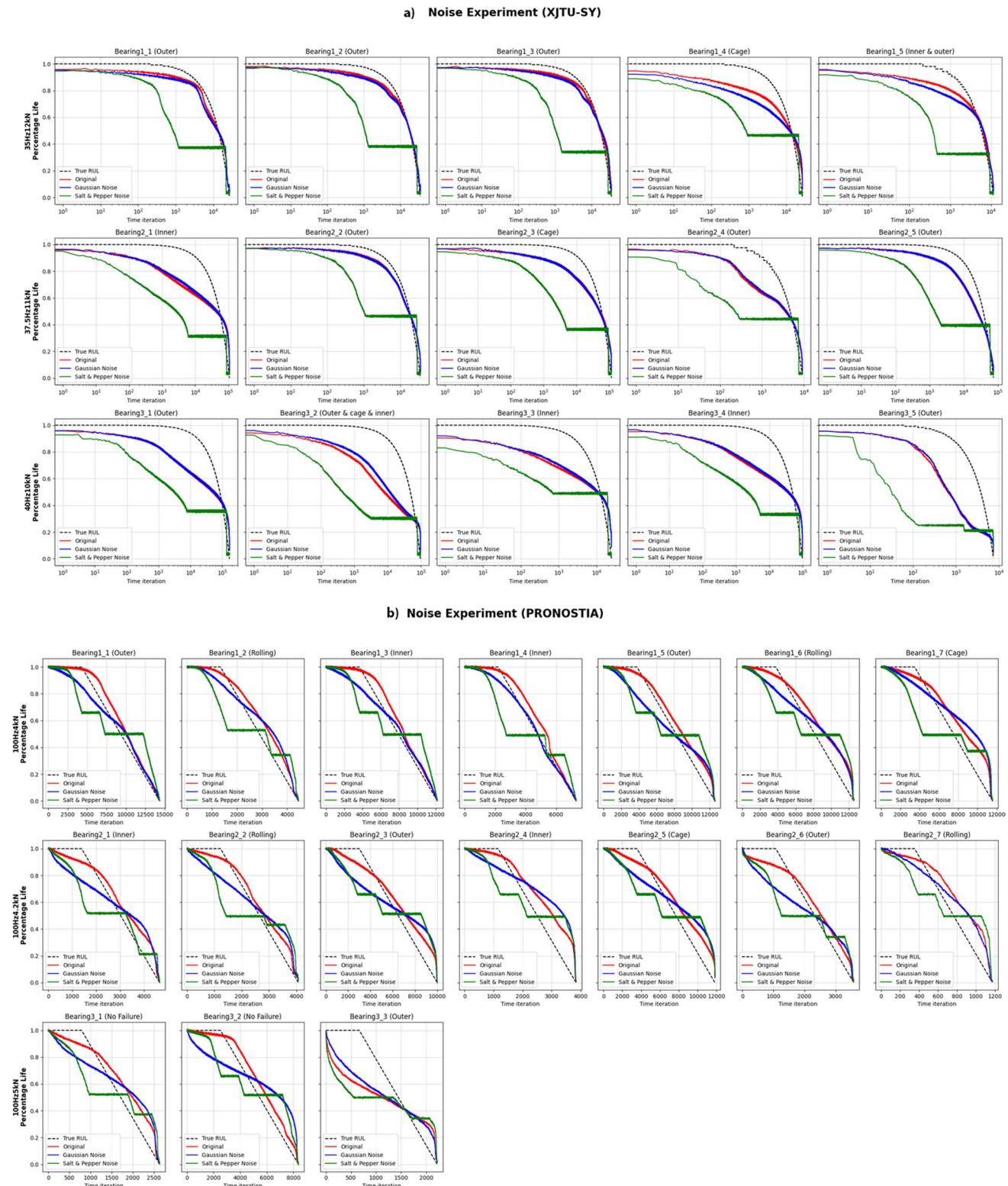


Fig. 10 Noise experiment result for **a** XJTU-SY; **b** PRONOSTIA

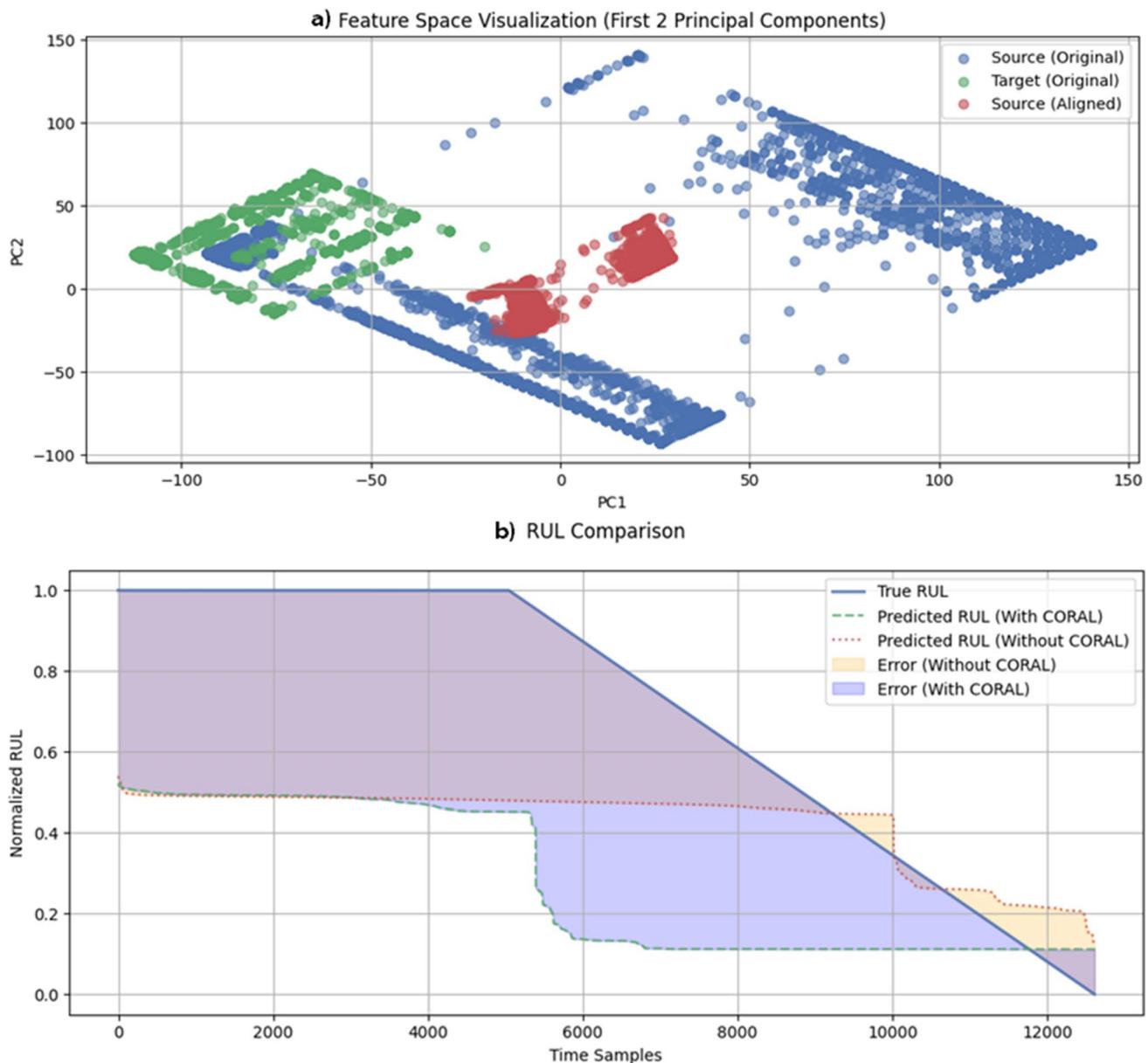


Fig. 11 **a** Feature space alignment using CORAL-PCA; **b** RUL comparison of CORAL-PCA-aligned and non-aligned

Table 3 Cross-domain Validation Experiment Results

Model	MSE	MAE	Score
With CORAL	0.0961	0.2803	297.3991
Without CORAL	0.1049	0.2919	321.7089

Bold values indicate the minimum MSE, MAE, and Score

results, we examined *Bearing 3* under each operating condition from both datasets.

For the XJTU-SY dataset:

- **35Hz12kN:** CARLE demonstrated the best performance with an MSE of 0.00220, MAE of 0.04087 and Score of 199.61. In comparison, MSIDIN recorded an MSE of

0.04383, MAE of 0.17848 and Score of 494.269. CA-BiLSTM reached an MSE of 2.41237, MAE of 1.24057 and Score of 3667.2644, while CNN-LSTM exhibited the poorest accuracy with an MSE of 8.6898, MAE of 1.81207 and Score of 5917.488.

- **37.5Hz11kN:** CARLE maintained superior results with an MSE of 0.01407, MAE of 0.10697 and Score of 2798.9778. MSIDIN followed with an MSE of 0.08407, MAE of 0.24669 and Score 2528.81, CABiLSTM showed degraded performance with an MSE of 0.71245, MAE of 0.67876, Score 6417.187, and CNN-LSTM further deteriorated to an MSE of 1.19243, MAE of 0.63259 and Score of 6304.808.

Table 4 Comparison with SOTA (XJTU-SY)

Bearing	Model	35Hz12kN			37.5Hz11kN			40Hz10kN		
		MSE	MAE	Score	MSE	MAE	Score	MSE	MAE	Score
Bearing 1	CARLE	0.00345	0.05157	188.36298	0.03273	0.16070	2462.88	0.03314	0.15983	4009.317
	CNN-LSTM	7.53585	1.68644	4212.5586	5.53985	1.19735	12144.208	15.12136	3.59279	54348.28
	CABiLSTM	2.31171	1.23203	2790.2166	1.58179	0.90853	8101.4062	5.76011	2.28761	32350.693
	MSIDIN	0.05074	0.17379	415.75772	0.08555	0.24885	2385.1533	0.08555	0.25344	3731.8108
Bearing 2	CARLE	0.00183	0.03692	217.17297	0.00671	0.07308	991.1647	0.06673	0.22032	2266.9412
	CNN-LSTM	2.76083	0.93285	2875.2263	4.41759	1.27952	4030.5173	0.83365	0.85786	6560.502
	CABiLSTM	0.92567	0.70848	2034.5358	1.53518	0.96063	2848.8662	1.19016	1.03863	7994.697
	MSIDIN	0.05763	0.17848	562.8828	0.12322	0.28668	916.3111	0.08829	0.25199	2094.0076
Bearing 3	CARLE	0.00220	0.04087	199.6124	0.01407	0.10697	2798.9778	0.03085	0.15631	530.86194
	CNN-LSTM	8.69865	1.81207	5917.488	1.19243	0.63258	6304.898	1.39763	0.76697	1562.1542
	CABiLSTM	2.41237	1.24057	3667.2644	0.71245	0.67876	6417.187	1.07763	0.88477	1746.2798
	MSIDIN	0.04383	0.15704	494.26984	0.08407	0.24669	2528.8157	0.09673	0.26387	554.0628
Bearing 4	CARLE	0.01172	0.09653	292.51904	0.02149	0.12591	245.14308	0.03361	0.16221	2362.9336
	CNN-LSTM	2.75389	0.81807	1996.5586	2.34120	0.92475	729.2322	0.85732	0.66800	5408.3696
	CABiLSTM	1.21288	0.81330	1857.263	1.11647	0.87074	651.8345	0.92342	0.82073	6554.092
	MSIDIN	0.08448	0.24985	592.63196	0.08910	0.25441	198.84254	0.11538	0.27989	2404.415
Bearing 5	CARLE	0.00465	0.05938	92.718994	0.01373	0.09903	1890.4968	0.09625	0.26167	190.29407
	CNN-LSTM	3.28659	1.36531	1333.5254	0.78787	0.78197	4740.767	1.34206	1.11198	669.8485
	CABiLSTM	2.04337	1.19535	1150.7625	0.52122	0.61763	3721.561	1.00430	0.95609	572.3747
	MSIDIN	0.06368	0.19819	204.28513	0.11293	0.28494	1821.144	0.35019	0.53957	319.7993

Bold values indicate the minimum MSE, MAE, and Score for each bearing-condition combination across all models

- **40Hz10kN:** CARLE achieved an MSE of 0.03805, MAE of 0.15631 and Score of 530.86. MSIDIN yielded an MSE of 0.09673, MAE of 0.26873 and Score of 2094.007, CABiLSTM followed with an MSE of 1.07763, MAE of 0.88477, Score of 1746.2798, and CNN-LSTM recorded an MSE of 1.39763, MAE of 0.76697 and Score of 1562.1542.

For the PRONOSTIA dataset:

- **100Hz4kN:** CARLE again delivered optimal results, achieving an MSE of 0.00029, MAE of 0.01312 and Score of 70.195. MSIDIN followed with an MSE of 0.00049, MAE of 0.01723 and Score of 1000.6890, while CABiLSTM recorded an MSE of 0.00268, MAE of 0.04036 and Score of 5860.68. Interestingly, CNN-LSTM attained an MSE of 0.00033 but slightly outperformed CARLE on MAE with a score of 0.01294 and Score of 1834.09.
- **100Hz4.2kN:** CARLE obtained an MSE of 0.00831, MAE of 0.07488 and Score of 80.114. MSIDIN slightly outperformed CARLE in all metrics, with MSE of 0.00606, MAE of 0.06360 and Score of 530.557. CABiLSTM trailed behind with an MSE of 0.01240, MAE of 0.09776 and Score of 3550.281, and CNN-LSTM significantly underperformed, with an MSE of 0.17432, MAE of 0.34082 and Score of 1312.410.
- **100Hz5kN :** CARLE achieved an MSE of 0.14152, MAE of 0.17514 and Score of 55.231. MSIDIN

reported higher error values with an MSE of 0.15967, MAE of 0.35579 and Score of 64.344, while CABiLSTM showed substantial degradation, reaching an MSE of 2.18729, MAE of 1.2095 and Score of 233.21. CNN-LSTM also performed poorly, with an MSE of 1.07811, MAE of 0.84026 and Score of 150.869.

These findings reinforce CARLE's ability to generalize effectively across different operating environments and its superior accuracy in both seen and unseen conditions. Notably, even in scenarios where other methods perform competitively under trained settings, CARLE maintains a robust edge, particularly in generalization to unseen conditions, which is critical in real-world prognostics applications.

4.8 Explanations

Higher accuracy in an AI system does not necessarily mean its predictions reflect real-world outcomes Bontempi (2023). This makes it essential to direct explainable AI (XAI) efforts toward PHM systems, particularly for remaining useful life (RUL) analysis of mechanical components, where unexpected failures can cause major operational disruptions. In this study, we applied Local Interpretable Model-Agnostic Explanations (LIME) Ribeiro et al. (2016) and Shapley Additive Explanations (SHAP) Lundberg et al. (2017) to interpret model predictions.

We selected two test points, one from the early degradation stage and one from the late degradation stage, to

Table 5 Comparison with SOTA (PRONOSTIA)

Bearing	Model	100Hz4kN			100Hz4.2kN			100Hz5kN		
		MSE	MAE	Score	MSE	MAE	Score	MSE	MAE	Score
Bearing 1	CARLE	0.00017	0.00890	74.451	0.00687	0.06874	37.435	0.03073	0.15232	24.860
	CNN-LSTM	0.00060	0.01944	4444.265	0.20664	0.39369	249.752	0.01941	0.12100	197.559
	CABiLSTM	0.00130	0.02515	6380.369	0.01075	0.08844	477.703	0.02536	0.13877	281.832
	MSIDIN	0.00055	0.01640	1488.338	0.00720	0.07223	156.422	0.03786	0.17235	88.091
Bearing 2	CARLE	0.00289	0.04268	32.028	0.00406	0.05340	25.583	0.04126	0.17514	54.820
	CNN-LSTM	0.01822	0.10301	322.214	0.09584	0.26255	340.356	0.06594	0.22073	412.641
	CABiLSTM	0.01010	0.08341	602.274	0.00706	0.06948	1421.654	0.02629	0.14069	1623.838
	MSIDIN	0.00643	0.06490	159.866	0.00403	0.05156	154.332	0.04196	0.17951	330.132
Bearing 3	CARLE	0.00029	0.01312	70.194	0.00831	0.07488	80.114	0.141255	0.17514	55.231
	CNN-LSTM	0.00033	0.01294	1834.090	0.17432	0.34082	1312.410	0.078112	0.845026	150.869
	CABiLSTM	0.00094	0.02538	5860.680	0.01240	0.09776	3550.281	2.1872909	1.20955	233.210
	MSIDIN	0.00049	0.01723	1000.680	0.00606	0.06360	530.557	0.159579	0.355762	64.344
Bearing 4	CARLE	0.00052	0.01635	39.394	0.01035	0.07616	41.840	–	–	–
	CNN-LSTM	0.00237	0.03209	496.519	0.02068	0.12315	343.611	–	–	–
	CABiLSTM	0.00268	0.04036	809.504	0.01279	0.09542	490.663	–	–	–
	MSIDIN	0.00134	0.02676	228.168	0.01029	0.08145	132.623	–	–	–
Bearing 5	CARLE	0.00264	0.03956	79.403	0.00711	0.06936	102.632	–	–	–
	CNN-LSTM	0.00885	0.07636	1135.705	0.08058	0.23612	951.493	–	–	–
	CABiLSTM	0.00999	0.08285	2605.018	0.01398	0.10261	3279.224	–	–	–
	MSIDIN	0.00605	0.06274	442.709	0.00878	0.07688	448.584	–	–	–
Bearing 6	CARLE	0.02014	0.12283	110.142	0.07713	0.23869	27.673	–	–	–
	CNN-LSTM	0.03497	0.13915	1099.476	0.20729	0.39659	332.571	–	–	–
	CABiLSTM	0.03064	0.15598	2633.624	0.03074	0.15201	447.319	–	–	–
	MSIDIN	0.05942	0.20851	619.222	0.03314	0.15922	106.045	–	–	–
Bearing 7	CARLE	0.00799	0.06877	112.334	0.00495	0.06176	13.302	–	–	–
	CNN-LSTM	0.03480	0.13173	878.371	0.06695	0.18207	63.732	–	–	–
	CABiLSTM	0.01221	0.08969	1799.508	0.00929	0.08194	166.860	–	–	–
	MSIDIN	0.00905	0.07806	362.584	0.00355	0.05256	33.425	–	–	–

Note: Bold values indicate the minimum MSE, MAE, and Score for each bearing-condition combination

examine which features contribute most during fault development. Figure 12(a, c) shows local explanations for XJTU and PRONOSTIA. In the early stage, σ_v played the most significant role in predictions, followed by k_v . This suggests that early degradation is primarily reflected in increased vibration variability and subtle distributional changes such as heavier tails. In practice, these effects correspond to small surface defects or early spalls on the bearing raceway that disturb the signal but do not yet dominate its frequency content.

As degradation progressed, the influence of σ and μ increased substantially, with k becoming the second most important feature. These variables capture more pronounced shifts in the vibration component and distributional asymmetry, which in real-world terms correspond to advanced fault development. At this stage, cracks expand, spalls deepen, and defect impacts become stronger and more asymmetric, producing larger and more irregular vibrations that are easier to isolate. To generate global insights, local explanations were aggregated to identify the vibration characteristics most critical to bearing degradation and

RUL estimation. Results (Fig. 12b, d) show that both the XJTU-SY and PRONOSTIA models rely heavily on σ , a measure of signal variability. This finding aligns with the physics of bearing failure, where increased variability often signals instability caused by defects such as looseness, contamination, or misalignment. The models also prioritize f_d components, which capture dominant frequency shifts associated with localized faults such as inner and outer race cracks, spalling, or lubrication deficiencies. In contrast, h contributes minimally, likely because fragmenting signals into shorter time windows reduces sensitivity to this global feature.

SHAP analysis (Fig. 13) confirms these findings and adds nuance. σ has the largest absolute impact, indicating that overall σ is the most reliable predictor of degradation. f_d components follow closely, reflecting the model's ability to capture fault-specific signatures. E features also contribute significantly, linking directly to failure mechanisms such as spalling progression, crack propagation, and lubrication breakdown. By contrast, h remains the least influential

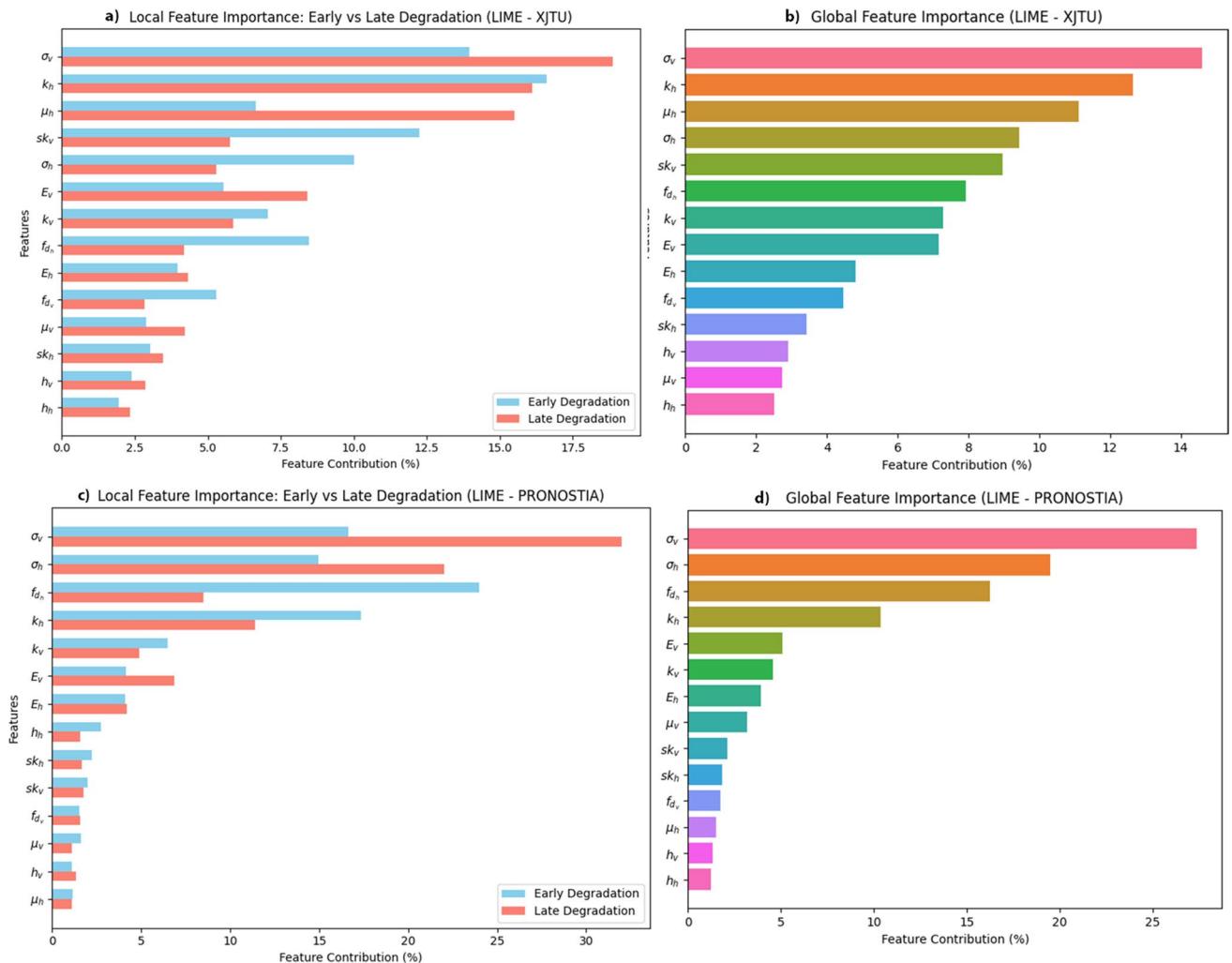


Fig. 12 LIME explanation for **a** XJTU-SY; **b** for PRONOSTIA

feature, confirming that short window fragmentation reduces its predictive power.

This detailed feature-level interpretation shows that CARLE not only produces accurate RUL predictions but does so in a way that reflects the underlying physical processes of bearing degradation, increasing both trust and applicability in high-risk industrial settings.

5 Conclusion

This research proposes a comprehensive RUL estimation system for rolling-element bearings. The system comprises three key components: a compact time–frequency feature extraction framework, an AI framework (CARLE), and XAI explanations. The feature extractor framework includes a complete algorithm to transform non-stationary vibrational signals into a set of time–frequency features using CWT. It also incorporates a Gaussian noise filter to eliminate signal

perturbations and short-term fluctuations. The CARLE AI framework comprises four blocks: Res-CNN captures spatial degradation trends from the input feature set; Res-RNN captures temporal degradation trends, learning long-term time dependencies; Linear block identifies patterns within these dependencies to produce a logit vector; finally, RFR predicts the final RUL. This ensemble approach, combining deep learning and traditional machine learning methods, enhances robustness and generalization, allowing the system to adapt effectively from one working condition to unseen conditions. We evaluated the trustworthiness of the AI framework using aggregated LIME and SHAP. The analysis revealed that CARLE heavily relies on σ features, which indicate that unstable faults such as looseness or contamination cause erratic behavior. The analysis also revealed that both models heavily rely on f_d , which is an indicator of localized defects, including inner and outer race cracks, looseness, and lubrication failures. Additionally, SHAP suggests that E features are also important, as they

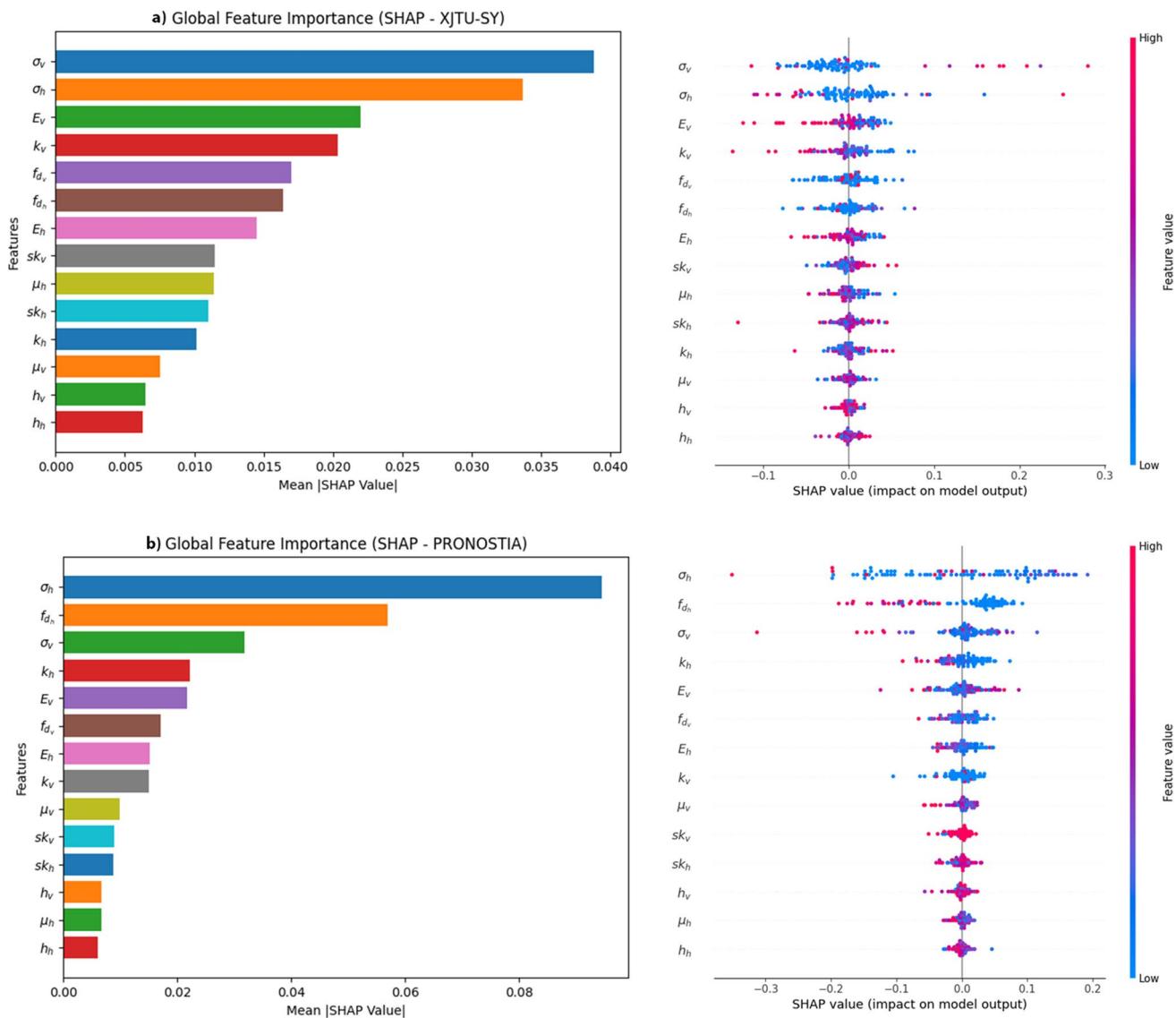


Fig. 13 SHAP explanation for **a** XJTU-SY; **b** for PRONOSTIA

indicate mechanical stress, friction, and surface defects. Other factors contribute but are less significant, confirming the system's reliability. We validated the proposed framework using the XJTU-SY and PRONOSTIA benchmark datasets.

5.1 Future work

While the findings of this research are promising, there is still room for improvement. We observed that CARLE struggles with early fault detection (see Fig. 9(a(xii-xiii), b(xvii))). Early degradation detection could be improved by incorporating a physics-guided loss to better capture subtle changes in the initial stages of degradation. Cross-domain validation experiments indicate that further hyperparameter tuning could enhance CARLE's generalization

performance. Another possible mitigation is to incorporate domain-adaptive training or fine-tuning on the target dataset to better capture domain-specific label distributions. Furthermore, in real-world scenarios, run-to-failure datasets are often unavailable. Implementing CARLE in a transfer learning configuration with incomplete run-to-failure data is also a promising direction for future research.

Appendix

A Preliminaries

In this section, we provide an overview of some building blocks of our proposed framework.

A.1 Gaussian filter

The Gaussian filter $G(x)$ is a smoothing filter commonly used to reduce noise, smooth data, and extract trends from non-stationary signals, which are crucial in predicting the RUL. It applies a weighted averaging operation to the signal, ensuring that values closer to the center of the filter contribute more to the result than those farther away. The mathematical expression of the Gaussian function is given by:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma_g^2}} e^{-\frac{x_d^2}{2\sigma_g^2}} \quad (17)$$

where x_d is the distance from the center of the filter. σ_g is the standard deviation of the Gaussian distribution, which controls the width of the Gaussian curve and determines the degree of smoothness.

A.2 Continuous wavelet transform

The Continuous Wavelet Transform (CWT) is a powerful mathematical tool that decomposes a time-varying signal

into highly localized oscillations called wavelets, providing better time-frequency analysis. The CWT uses basis functions that are scaled and shifted versions of the time-localized wavelet, enabling the creation of a time-frequency representation of a signal with excellent localization in both time and frequency. The mathematical expression of the CWT is as follows:

$$\Gamma(a, b) = \int_{-\infty}^{\infty} I(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (18)$$

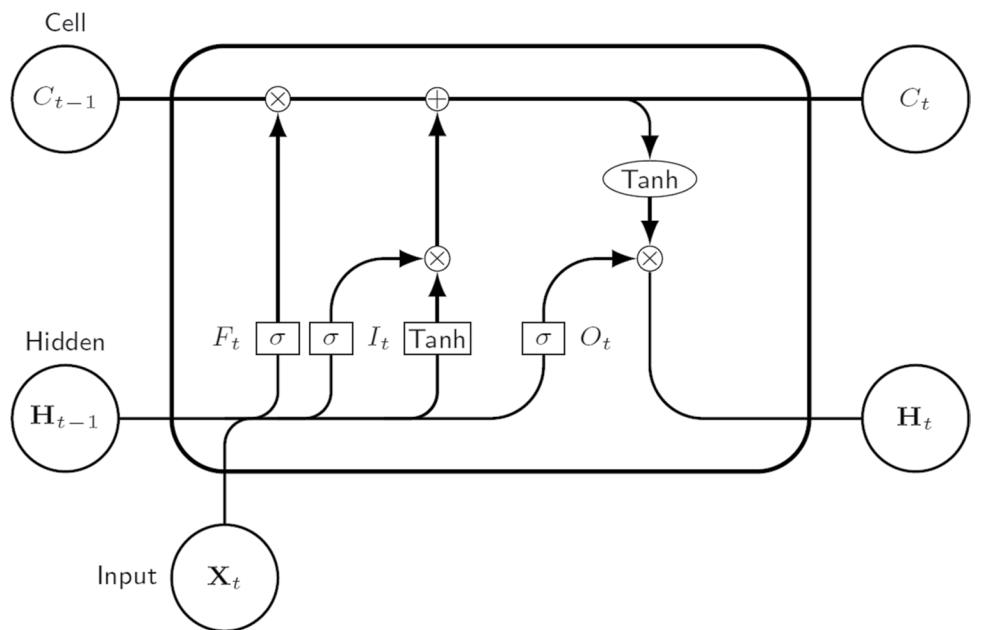
where $\Gamma(a, b)$ represents the wavelet coefficients at scale a and translation b , $I(t)$ represents the nonstationary signal, and $\psi(t)$ represents the mother wavelet function. We selected the Morlet wavelet Farge (1992) as the mother wavelet for time-frequency representation (TFR) extraction due to its similarity to the bearing impulse response Zhu et al. (2018) and its favorable trade-off between time and frequency resolution. In particular, its frequency resolution improves at higher values of a , while the time resolution improves at lower values Lin and Liangsheng (2000). The Morlet wavelet is defined as a sinusoidal function modulated by a Gaussian envelope with a central frequency f_c and is given by:

$$\psi(t) = e^{\frac{if_c t}{2\pi}} e^{-t^2/2} \quad (19)$$

A.3 Long Short-Term Memory (LSTM)

The LSTM network is a class of deep recurrent networks designed to capture long-term time dependencies from data. LSTM utilizes specialized gates, i.e., an input gate I_t , a

Fig. 14 Structure of the LSTM network



forget gate F_t , and an output gate O_t , to regulate the flow of information, allowing selective retention and forgetting of information. This ability makes LSTM ideal for modeling time series data that exhibit long-term dependencies such as the gradual degradation of rolling element bearings, providing a more accurate RUL estimation Hochreiter (1997). The structure of an LSTM network is shown in Fig. 14, and the output of an LSTM network can be mathematically modeled as:

$$H_t = \mathcal{NN}(I_t, H_{t-1}) = \begin{cases} C_t = \phi(W_q[H_{t-1}, X_t] + b_q) \\ I_t = \sigma(W_i[H_{t-1}, X_t] + b_i) \\ F_t = \sigma(W_f[H_{t-1}, X_t] + b_f) \\ O_t = \sigma(W_o[H_{t-1}, X_t] + b_o) \\ S_t = C_t \odot X_t + S_{t-1} \odot F_t \\ H_t = O_t \odot \phi(S_t) \end{cases} \quad (20)$$

A.4 Random forest regressor

Random Forest Regression (RFR) is a supervised learning algorithm that employs an ensemble learning method for regression tasks based on the bagging technique. In RFR, the trees operate in parallel, meaning that there is no interaction between them during the training process. Each tree is trained on a random subset of the features, and the final prediction is obtained by averaging the outputs of all the trees Segal (2004). We chose RFR for its accuracy, robustness, and ability to handle nonlinear relationships effectively in data, making it particularly suitable for RUL estimation, where complex interactions and temporal patterns are crucial. A schematic diagram of RFR is shown in Fig. 15.

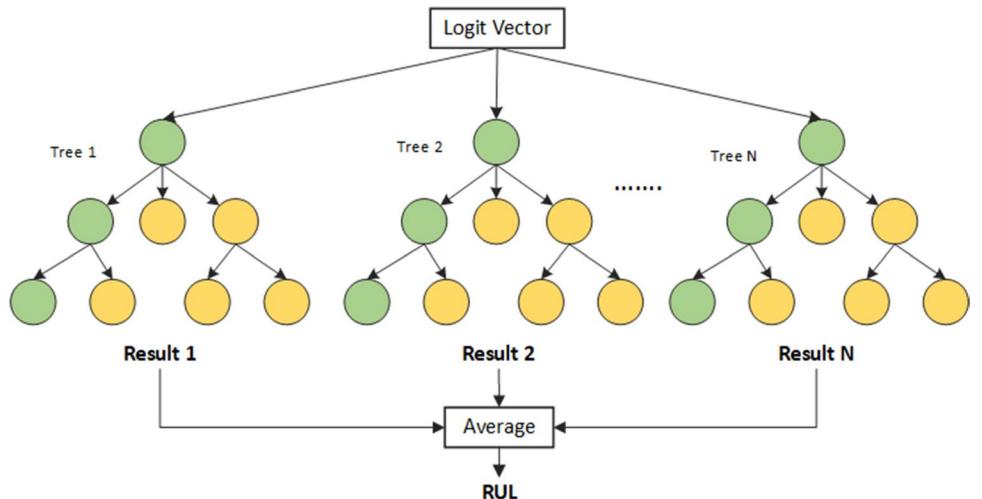
Table 6 Hyperparameter comparison of CARLE (XJTU-SY vs PRONOSTIA)

Block	Hyperparameter	XJTU-SY	PRONOSTIA
Res-CNN	CNN Layers	4	4
	CNN Filters	[256, 256, 128, 64]	[64, 64, 32, 32]
	Kernel Sizes	[3, 3, 2, 2]	[3, 3, 2, 2]
	Padding	Same	Same
	Regularization (λ)	0.005	0.005
	Activation	ReLU	ReLU
	Pooling Size	1	1 (Max-Pooling1D)
	Residual Connections	Applied	Applied
	Multi-Head Attention	8 Heads, 64 Dim	8 Heads, 64 Dim
	LSTM Layers	2	2
Res-LSTM	LSTM Units	[64,64]	[64,64]
	Statefulness	False	False
	Return Sequences	True	True
	Residual Connections	Applied	Applied
	Multi-Head Attention	8 Heads, 64 Dim	8 Heads, 64 Dim
Linear	Flatten Layer	Applied	Applied
	Linear Layers	3	3
	Linear Units	[128, 64, 32]	[64, 48, 32]
Random Forest Regressor (RFR)	No. of trees	800	800

B Implementation

In this section, we provide the hyperparameters for both XJTU-SY and PRONOSTIA and the training regularization and optimizations that we use in our implementation.

Fig. 15 Structure of the RFR algorithm



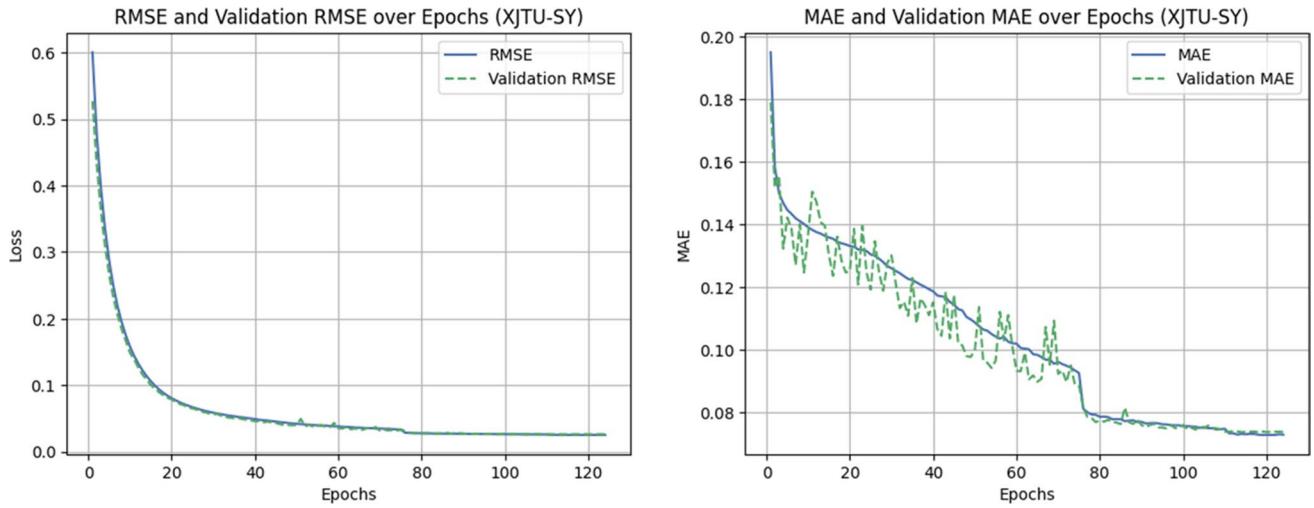


Fig. 16 Training statistics of CARLE (XJTU-SY)

B.1 Training setup

We trained our CARLE on an Intel Core i5-7200U with 16 GB RAM and no GPU. The model was implemented in Python 3.10 using Tensorflow 2.18. Due to computational limitations and to make training efficient, we applied various optimizations to improve training efficiency. To ensure that

the model converges to the best possible solution despite hardware constraints, we incorporate several callbacks: *ResetStateCallback* to reset model states between epochs, *EarlyStopping* to halt training if validation loss stagnates for multiple epochs, *ReduceLROnPlateau* to adjust the learning rate on MSE dynamically, and *ModelCheckpoint* to save the best training weights. These optimizations collectively enhance both training efficiency and model performance.

```

Require: Features vector ( $I_{fv}$ ), RUL labels ( $Y$ ), CARLE:  $f(x, w) \rightarrow y$ , Loss:  $L(y, \hat{y}) \rightarrow R$ , batch size  $k$ , Number of
trees ( $n_{trees}$ )
1: Initialize weights  $w$ 
2:  $l_{min} \leftarrow \infty$ 
3: Initialize empty forest:  $Trees \leftarrow \{\}$ 
4: procedure DEEP NEURAL NETWORK
5:   for  $e = 1 \dots max_{EPOCH}$  do
6:     for  $i = 1 \dots \lceil \frac{n}{k} \rceil$  do
7:       ( $x, y$ ) is the batch size of  $k$  from ( $I_{fv}, Y$ )
8:        $w \leftarrow w_{t-1} - \frac{\eta}{E[g^2]} \frac{\partial C}{\partial w}$                                  $\triangleright$  Root mean square prop
9:     end for
10:    Compute loss metrics:
11:     $l_e = \sqrt{\sum_{i=1}^n (y - \hat{y})^2}$                                           $\triangleright$  Root mean square error
12:     $mae = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$                                           $\triangleright$  Mean absolute error
13:    if  $l_e < l_{min}$  then
14:       $l_{min} \leftarrow l_e$ 
15:       $w_{best} \leftarrow w_e$ 
16:    end if
17:  end for
18:  Compute Logit vector:  $I_{lv} \leftarrow CARLE(I_{fv})$                                  $\triangleright$  Output from CARLE
19: end procedure
20: procedure RANDOM FOREST REGRESSION
21:   for  $e = 1 \dots n_{trees}$  do
22:     Initialize decision tree  $T_e$  with  $max_{feat}$ 
23:     Train  $T_e$  on ( $I_{lv}, Y$ ):  $T_e \leftarrow fit((I_{lv}, Y))$ 
24:     Add trained tree to forest:  $Trees \leftarrow Trees \cup \{T_e\}$ 
25:   end for
26:   Compute training predictions:  $\hat{Y} \leftarrow \frac{1}{n_{trees}} \sum_{e=1}^{n_{trees}} T_e(I_{lv})$ 
27:   Compute loss metrics:
28:    $MSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$                                           $\triangleright$  Root mean square error
29:    $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$                                           $\triangleright$  Mean absolute error
30:   if  $MSE < l_{min}$  then
31:      $l_{min} \leftarrow MSE$ 
32:      $BestForest \leftarrow Trees$ 
33:   end if
34:    $rul \leftarrow \hat{Y}$                                           $\triangleright$  Output from Random Forest
35: end procedure
36: return  $w_{best}, BestForest, rul$ 

```

Algorithm 2 Training and Testing of CARLE

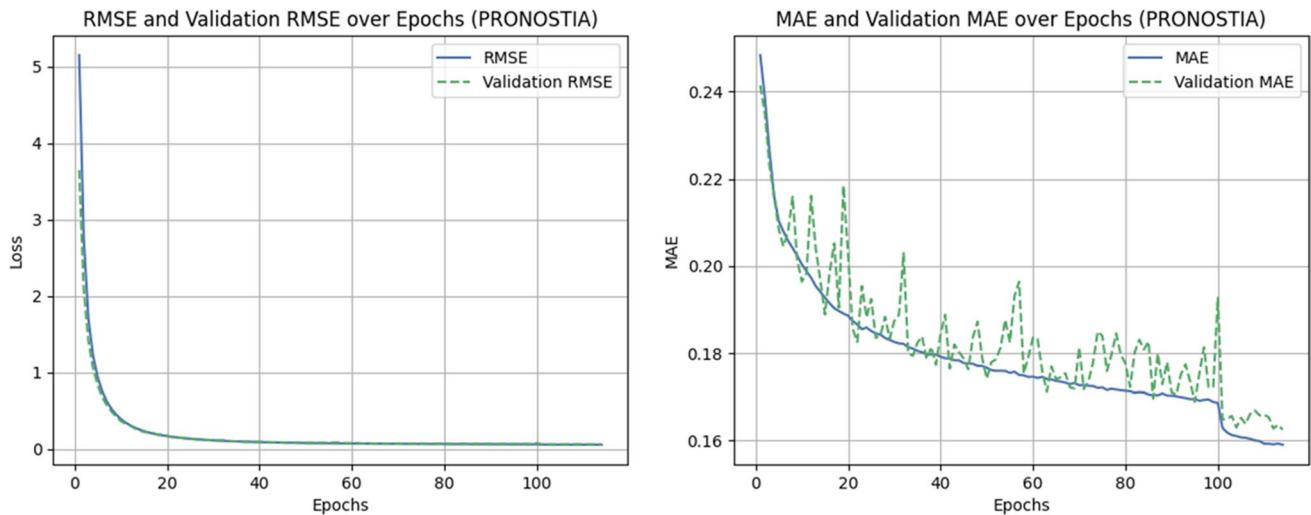


Fig. 17 Training statistics of CARLE (PRONOSTIA)

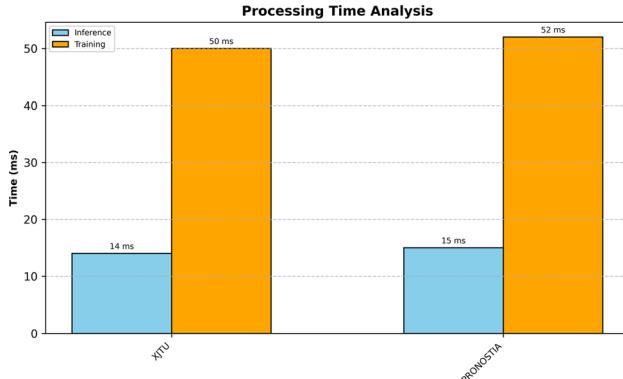


Fig. 18 Processing Time Analysis (XJTU-SY vs. PRONOSTIA)

B.2 Training and testing of CARLE

The training and testing procedure for CARLE involves two phases: training the deep neural network and training the random forest regression model. During the first phase, the model is optimized via batch updates and the MSE loss function to learn the relationships between the input features and RUL labels in a supervised manner. The output, a logit vector, is then used to train an RFR consisting of multiple decision trees. The MSE and MAE performance metrics are used throughout the training process to evaluate and select the best model. The trained neural network and RFR are applied to unseen data to predict the RUL during testing. The complete algorithm is detailed in Algorithm 2, and the model parameters for XJTU-SY and PRONOSTIA are detailed in Table 6. The training statistics for XJTU-SY and PRONOSTIA are provided in Figs. 16 and 17, respectively. Time processing time analysis for both XJTU-SY and PRONOSTIA datasets are provided in Fig. 18. Both achieved nearly identical training and inference time in a

moderate training setup. On low-end hardware, these processing times suggest that while training may be slower, the inference step, critical for real-time localized prognostics, remains feasible, as the model's small size and low computational complexity enable fast forward passes even without high-end resources.

Acknowledgements This research was supported by the CAS-ANSO Scholarship. We acknowledge the intellectual and material contributions of the University of Science and Technology of China (USTC) and the Alliance of International Science Organizations (ANSO).

Author Contributions **Waleed Razzaq:** Conceptualization, Methodology, Data Curation, Writing- Original draft preparation. **Yun-Bo Zhao:** Supervision, Writing- Reviewing.

Funding The research did not receive any funding from any organization.

Data Availability The code is available at <https://github.com/itxwaleedrazzaq/PhDCode.git>.

Declarations

Ethical Approval This study was conducted in accordance with ethical standards.

Conflicts of Interest The authors declare that they have no conflicts of interest to disclose.

Human/Animal Participation No human or animal participation is involved in this research.

References

Abdellatif M, Hamla W, Hamouda H (2025) Ai driven prediction of early age compressive strength in ultra high performance fiber reinforced concrete. *Sci Rep* 15(1):20316

Abdellatif M, Elsafi M, Murali G, ElNemr A (2025) Comparative evaluation of hybrid machine learning models for predicting the strength of metakaolin-based geopolymers concrete enhanced with gaussian noise augmentation. *Journal of Building Engineering*, page 113302

Al-Dulaimi A, Zabihi S, Asif A, Mohammadi A (2019) A multimodal and hybrid deep neural network model for remaining useful life estimation. *Comput Ind* 108:186–196

Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8:1–74

Antoni J (2006) The spectral kurtosis: a useful tool for characterising non-stationary signals. *Mech Syst Signal Process* 20(2):282–307

Bienefeld C, Kirchner E, Vogt A, Kacmar M (2022) On the importance of temporal information for remaining useful life prediction of rolling bearings using a random forest regressor. *Lubricants* 10(4):67

Bontempi G (2023) Between accurate prediction and poor decision making: the AI/ML gap. *arXiv:2310.02029*

Borghesani P, Pennacchi P, Randall RB, Sawalhi N, Ricci R (2013) Application of cepstrum pre-whitening for the diagnosis of bearing faults under variable speed conditions. *Mechanical Systems and Signal Processing* 36(2):370–384

Bracewell Ronald N (1989) The fourier transform. *Sci Am* 260(6):86–95

Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*

Deng Y, Shichang D, Wang D, Shao Y, Huang D (2023) A calibration-based hybrid transfer learning framework for RUL prediction of rolling bearing across different machines. *IEEE Trans Instrum Meas* 72:1–15

Farge M (1992) Wavelet transforms and their applications to turbulence. *Annu Rev Fluid Mech* 24(1):395–458

Guo W, Cao H, He Z, Yang L (2015) Fatigue life analysis of rolling bearings based on quasistatic modeling. *Shock Vib* 2015(1):982350

Guo J, Wang J, Zhiyuan Wang Yu, Gong JQ, Wang G, Tang C (2023) A cnn-bilstm-bootstrap integrated method for remaining useful life prediction of rolling bearings. *Qual Reliab Eng Int* 39(5):1796–1813

Hochreiter S (1997) Long short-term memory. *Neural Computation* MIT-Press

Hu L, Wang J, Lee HP, Wang Z, Wang Y (2024) Wear prediction of high performance rolling bearing based on 1D-CNN-LSTM hybrid neural network under deep learning. *Heliyon*. <https://doi.org/10.1016/j.heliyon.2024.e35781>

Lei Y, He Z, Zi Y, Qiao H (2007) Fault diagnosis of rotating machinery based on multiple ANFIS combination with gas. *Mech Syst Signal Process* 21(5):2280–2294

Li X, Zhang W, Ding Q (2019) Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Process* 161:136–154

Li J, Wang Z, Liu X, Feng Z (2023) Remaining useful life prediction of rolling bearings using gru-deepar with adaptive failure threshold. *Sensors (Basel)* 23(3):1144

Li Y, Wang Z, Li F, Li Y, Zhang X, Shi H, Dong L, Ren W (2024) An ensembled remaining useful life prediction method with data fusion and stage division. *Reliability Engineering & System Safety* 242:109804

Lin J, Liangsheng Q (2000) Feature extraction based on morlet wavelet and its application for mechanical fault diagnosis. *J Sound Vib* 234(1):135–148

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30

Luo J, Zhang X (2022) Convolutional neural network based on attention mechanism and bi-lstm for bearing remaining life prediction. *Appl Intell* 52(1):1076–1091

Nectoux P, Gouriveau R, Medjaher K, Ramasso E, Chebel-Morello B, Zerhouni N, Varnier C (2012) Pronostia: An experimental platform for bearings accelerated degradation tests. In: *IEEE International Conference on Prognostics and Health Management, PHM'12*, pages 1–8. IEEE Catalog Number: CPF12PHM-CDR, Randall RB, Antoni J (2011) Rolling element bearing diagnostics-a tutorial. *Mech Syst Signal Process* 25(2):485–520

Ribeiro MT, Singh S, Guestrin C (2016) " why should i trust you?" explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144

Segal MR (2004) Machine learning benchmarks and random forest regression

Smith Wade A, Randall Robert B (2015) Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark study. *Mech Syst Signal Process* 64:100–131

Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25

Stallone A, Cicone A, Materassi M (2020) New insights and best practices for the successful use of empirical mode decomposition, iterative filtering and derived algorithms. *Sci Rep* 10(1):1–15

Tandon N, Choudhury A (1999) A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribol Int* 32(8):469–480

Ture BA, Akbulut A, Zaim AH, Catal C (2024) Stacking-based ensemble learning for remaining useful life estimation. *Soft Computing* 28(2):1337–1349

Vaswani A (2017) Attention is all you need. *Advances in Neural Information Processing Systems*

Wang L, Zhu Z, Zhao X (2024) Dynamic predictive maintenance strategy for system remaining useful life prediction via deep learning ensemble method. *Reliability Engineering & System Safety* 245:110012

Wenjian L, Wang Yu, Zhang M, Junwei G (2024) Physics guided neural network: Remaining useful life prediction of rolling bearings using long short-term memory network through dynamic weighting of degradation process. *Eng Appl Artif Intell* 127:107350

Wang B, Lei Y, Li N et al (2018) Xjtu-sy bearing datasets

Wu L, Fu X, Guan Y (2016) Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies. *Appl Sci* 6(6):166

Yin C, Li Y, Wang Y, Dong Y (2025) Physics-guided degradation trajectory modeling for remaining useful life prediction of rolling bearings. *Mech Syst Signal Process* 224:112192

Yong HU, Qun CH, Pengcheng XI, Chengliang LI (2024) Remaining useful life prediction using physics-informed neural network with self-attention mechanism and deep separable convolutional network. *Journal of Advanced Manufacturing Science and Technology*, 4(4)

Zhang G, Liang W, She B, Tian F (2021) Rotating machinery remaining useful life prediction scheme using deep-learning-based health indicator and a new rvm. *Shock Vib* 2021(1):8815241

Zhang J, Wang P, Yan R, Gao RX (2018) Long short-term memory for machine remaining life prediction. *J Manuf Syst* 48:78–86

Zhao K, Jia Z, Jia F, Shao H (2023) Multi-scale integrated deep self-attention network for predicting remaining useful life of aero-engine. *Eng Appl Artif Intell* 120:105860

Zhu J, Chen N, Peng W (2018) Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Trans Ind Electron* 66(4):3208–3216

Zhuang J, Jia M, Ding Y, Ding P (2021) Temporal convolution-based transferable cross-domain adaptation approach for remaining useful life estimation under variable failure behaviors. *Reliability Engineering & System Safety* 216:107946

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.