Article

Stage-Based Remaining Useful Life Prediction for Bearings Using GNN and Correlation-Driven Feature Extraction

Guangzhong Huang [1], Wenping Lei [1,*], Xinmin Dong [1], Dongliang Zou [2], Shijin Chen [2] and Xing Dong [1]

[1] School of Mechanical and Power Engineering, Zhengzhou University, Zhengzhou 450001, China; hgz@gs.zzu.edu.cn (G.H.); dongxing@gs.zzu.edu.cn (X.D.)

[2] MCC5 Group Shanghai Corporation Limited, Shanghai 200400, China; dlzou2006@163.com (D.Z.); 18301759989@163.com (S.C.)

[*] Correspondence: lwp@zzu.edu.cn

Abstract: Bearings are critical components in mechanical systems, and their degradation process typically exhibits distinct stages, making stage-based remaining useful life (RUL) prediction highly valuable. This paper presents a model that combines correlation analysis feature extraction with a Graph Neural Network (GNN)-based approach for bearing degradation stage classification and RUL prediction, aiming to achieve accurate bearing life prediction. First, the proposed Pearson–Spearman correlation metric, along with Kernel Principal Component Analysis (KPCA) and autoencoders, is used to group and fuse health indicators (HIs), thereby obtaining a health indicator (HI) that effectively reflects the bearing degradation process. Then, a model combining Graph Convolutional Network (GCN) and Long Short-Term Memory (LSTM) networks is proposed for bearing degradation stage classification. Based on the classification results, the Adaptive Attention GraphSAGE–LSTM (AAGL) model, also introduced in this study, is employed to precisely predict the bearing's remaining useful life.

Keywords: rolling bearing; RUL prediction; KPCA; autoencoder; stage classification; GNN; LSTM

1. Introduction

Bearings are critical components in mechanical systems, directly influencing operational efficiency and lifespan [1]. As bearings degrade, their performance gradually declines, eventually leading to failure. Therefore, accurately predicting the remaining useful life (RUL) of bearings is essential for minimizing downtime and reducing maintenance costs. Effective bearing life prediction enables preventive maintenance, helps avoid unexpected failures, and enhances the reliability and safety of mechanical systems. During the bearing degradation process, distinct stages of degradation typically emerge [2]. Initially, the bearing remains in a healthy state, with the health indicators remaining stable. As degradation progresses, the health indicators begin to rise gradually. Eventually, the bearing enters a severe degradation phase, marked by a sharp increase in the health indicators, signaling imminent failure.

HIs play a significant role in mechanical predictions. Suitable HIs can simplify modeling and yield accurate predictions. Common HIs include time-domain, frequency-domain, and time-frequency domain indicators [3]. Various physics-based HIs (PHIs) have been used for bearing life prediction. For instance, Zhang et al. [4] used kurtosis values extracted from band-pass filtered vibration signals for RUL prediction. Gasperin et al. [5] extracted the power density of gear meshing frequencies from the envelope spectrum, and

Soualhi et al. [6] utilized the Hilbert–Huang transform to analyze vibration signals and extract defect frequencies as PHIs. Recently, virtual HIs (VHIs) that fuse multiple PHIs to construct features reflecting degradation trends have been increasingly used. Fusion methods such as principal component analysis (PCA) [7], KPCA [8], linear discriminant analysis (LDA) [9], t-distributed stochastic neighbor embedding (t-SNE) [10], and locality preserving projection (LPP) [11] are common, with PCA and its derivative KPCA being the most prevalent. Akpudo et al. [12] employed a second-order KPCA algorithm to map multiple features into a new feature space, generating a comprehensive degradation index and using a deep long short-term memory (DLSTM) model for RUL prediction. Wang et al. [13] calculated the Pearson correlation among features, grouped them, and used KPCA within each group to obtain new feature sets, combining them with an LSTM-COX model for bearing life prediction. However, this approach only considers linear correlations, potentially overlooking nonlinear relationships.

As faults progress, the HI of machinery usually exhibits different degradation trends as shown in Figure 1. Before predicting RUL, the degradation process should be divided into different health stages (HSs) based on HI trends. Previous studies often divided the degradation process into two main stages: healthy and degraded. For example, Wang et al. [14] used the Mahalanobis distance to fuse fourteen statistics into a new feature, employing an interval as an alarm threshold to determine the first prediction time (FPT) and assess bearing degradation. Ni et al. [15] introduced techniques based on the rule to divide the healthy stage and used a gated recurrent unit network to predict RUL. However, these two-stage models might oversimplify the complex degradation process in practical applications where bearings may exhibit multiple degradation trends due to healing effects or varying operational conditions.
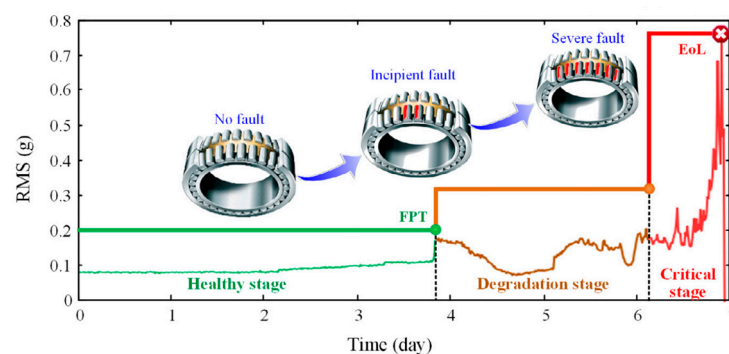

Figure 1. Bearing degradation process.

To address these limitations, some researchers proposed multi-stage degradation models. El-Thalji [16] developed a five-stage dynamic model based on accelerated test results, dividing the degradation process into run-in, steady operation, defect initiation, defect propagation, and damage growth. Similarly, Zhao et al. [17] identified three degradation stages through curve fitting and power spectral density analysis, while JB Ali et al. [18] proposed the RMS Entropy Estimator (RMSEE) feature and classified bearing health states using Simplified Fuzzy Adaptive Resonance Theory Maps (SFAMs). Based on the stage division, the RUL was obtained. Wang et al. [19] proposed an adaptive staged RUL prediction method that uses Gath-Geva fuzzy clustering for stage division and applies tailored prediction models for each degradation stage. However, these methods often depend on linear assumptions, specific statistical thresholds, or predefined clustering criteria, potentially overlooking the nonlinear relationships, feature dependencies, and the dynamic nature of bearing degradation in real-world conditions. Similarly, RUL prediction methods for CFRP structures face similar challenges, such as complex physical properties, limited labeled data,

and nonlinear feature dependencies, which are also present in bearing degradation [20]. These challenges highlight the importance of adopting advanced methodologies to improve the adaptability and generalizability of RUL prediction models.

Graph Neural Networks (GNNs) have emerged as a promising deep learning approach for bearing lifespan prediction. GNNs are particularly effective in modeling graph-structured data, especially when handling multi-dimensional and spatiotemporal data. However, bearing degradation is inherently sequential, with past operating conditions influencing future performance. To address this, LSTM networks [13], a type of recurrent neural network (RNN), are widely used for modeling time-series data due to their ability to capture long-term dependencies. LSTMs are especially suitable for bearing degradation prediction, where historical data plays a crucial role in forecasting future wear and failure. This paper proposes a hybrid model that combines the strengths of GNNs in spatial modeling and LSTMs in temporal prediction, aiming to more accurately predict bearing RUL in complex operating environments. Yang et al. [21] transformed bearing time series data into a graph structure and combined Graph Convolutional Networks (GCNs) with Gated Recurrent Units (GRUs) to predict the RUL from both spatial and temporal perspectives. Wen et al. [22] proposed a method that converts bearing time-frequency recursive graphs into graph structures, integrating GCN with Long Short-Term Memory (LSTM) networks to evaluate bearing degradation and predict RUL, overcoming the limitations of deep learning methods when dealing with non-Euclidean space data. Cui et al. [23] introduced a graph domain adaptation method driven by digital twins, where a dynamic twin model of the bearing's entire lifecycle generates rich twin data. This method, combined with a multi-layer cross-domain gated graph convolution network (MGGCN), addresses the limitations of traditional domain adaptation methods in processing non-Euclidean space data, enabling effective RUL prediction under limited real-world data conditions. Yang et al. [24] proposed a Spatiotemporal Multi-Scale Graph Convolutional Network (STMS-GCN) framework, which incorporates features that capture dynamic changes in vibration energy and uses a sliding window method to automatically detect fault occurrence times, significantly improving the accuracy of RUL predictions. Although significant progress has been made in the application of GNNs for bearing lifespan prediction, most existing models overlook the segmentation of degradation stages or rely on simple two-stage classification, which fails to capture the complex multi-stage degradation patterns observed in real-world bearing performance. While some studies have attempted to explore multi-stage RUL prediction, these approaches still suffer from limitations in accurately reflecting the full degradation process of bearings. Specifically, many GNN-based models do not effectively model the gradual transitions between degradation stages or fail to integrate spatial and temporal features that are essential for capturing the dynamic nature of the degradation process. This is a critical issue, as accurate RUL prediction requires not only understanding the final failure stage but also forecasting the intermediate degradation states that occur throughout the bearing's lifecycle. To address these gaps, this paper proposes an innovative multi-stage prediction approach based on GNNs, which leverages both spatial and temporal features to model the full spectrum of bearing degradation. By capturing the non-linear relationships within the degradation process, the proposed method demonstrates robust performance and significantly improves prediction accuracy across various degradation stages.

To address these issues, this paper proposes an improved method. First, a Pearson–Spearman correlation coefficient-based approach is introduced to group bearing degradation HIs, considering both the linear and nonlinear relationships between the health indicators. Next, KPCA and autoencoders are employed to extract health indicators that accurately reflect the bearing degradation trend. Then, a bearing degradation stage classi-

fication model based on GCN–LSTM is developed, and a global attention mechanism is applied to precisely segment the degradation stages. Finally, based on the classification results, an adjacency matrix is constructed and used, along with the feature matrix, as input to the proposed AAGL network for RUL prediction, significantly improving the accuracy and robustness of the predictions.

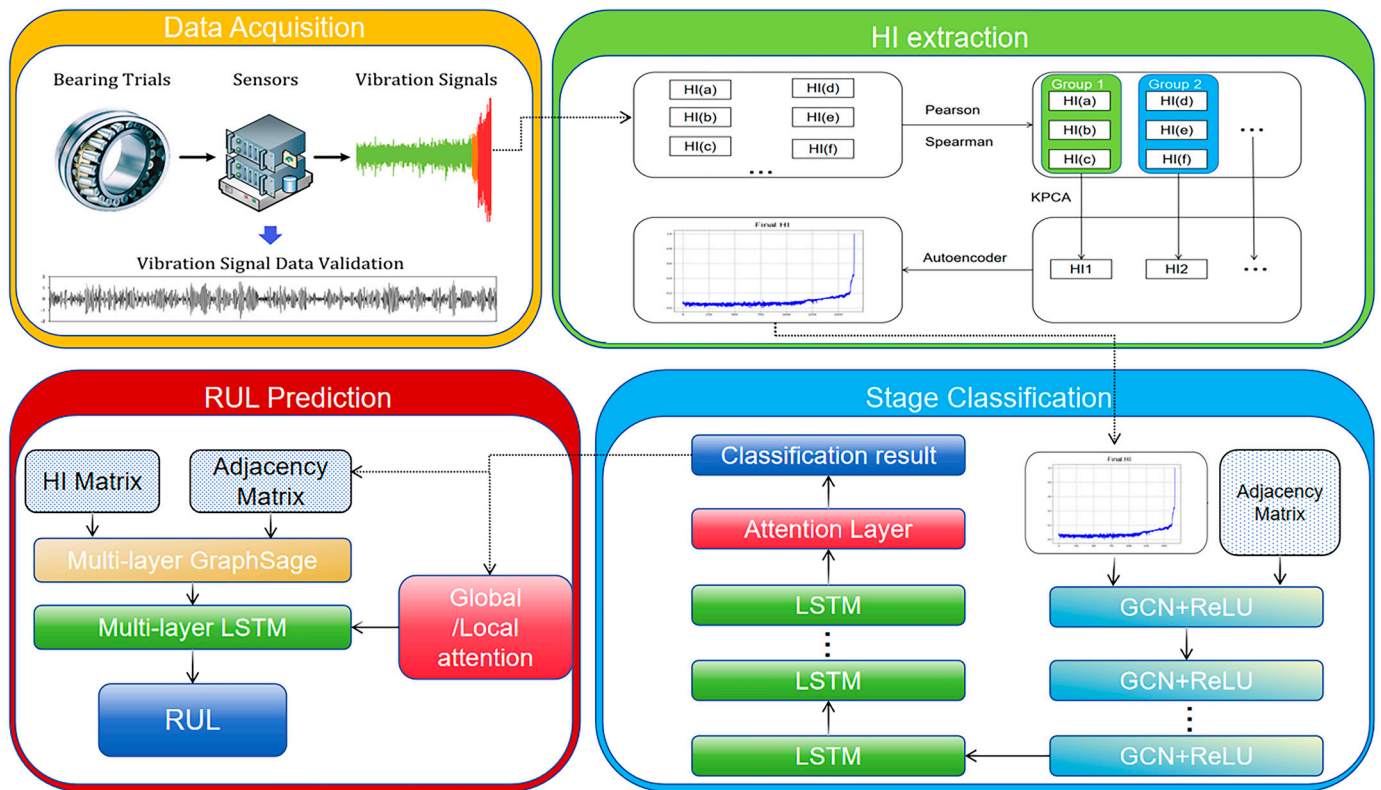The overall process of this study is illustrated in Figure 2.

Figure 2. Schematic diagram of the research process.

2. Methodology

2.1. Methodological Approach to HI Extraction

The extraction of HIs plays a crucial role in monitoring bearing degradation, as it provides a quantitative representation of the bearing's operational condition. These indicators are essential for detecting early signs of degradation and tracking the progression of wear and damage. By accurately extracting and analyzing HIs, specific patterns or trends in bearing health can be identified and correlated with different stages of degradation. This not only enables more precise predictions of the RUL but also supports the implementation of proactive maintenance strategies. In the subsequent study, effective health indicators will be extracted by combining comprehensive correlation analysis with KPCA.

2.1.1. HIs

Time-domain signals play a crucial role in bearing life prediction, enabling effective monitoring and forecasting of bearing health. The commonly used time-domain features are listed below.

Kurtosis [25] describes the thickness of the tails of the signal's probability distribution, with high kurtosis indicating outliers and aiding in detecting wear or damage. The kurtosis indicator directly reflects changes in kurtosis, tracking abnormal peaks. The interquartile range (IQR) [26], robust to asymmetrical noise and outliers, reflects statistical dispersion. Mean absolute deviation (MAD) [27] indicates the average absolute deviation level, sensi-

tive to outliers. The margin indicator, reflecting the safety margin during operation, helps prevent failures from excessive wear. Modulus maximum [28], the signal's maximum absolute value, detects the strongest vibrations or impacts. The peak indicator monitors maximum peaks, crucial for early damage detection. Peak-to-peak measures the amplitude difference between the signal's maximum and minimum values. The pulse indicator identifies repeating pulse patterns, recognizing periodic faults. Root mean square (RMS) [29] represents the energy of the signal, assessing overall vibration levels and detecting wear-type faults. Signal entropy describes randomness or complexity, identifying irregular fault patterns. Skewness measures distribution asymmetry, with changes indicating bearing condition shifts. The Teager energy mean [30] estimates energy content, capturing dynamic load changes. Variance monitors signal variability to track operating condition changes. Lastly, the waveform indicator detects impacts and anomalies by comparing the peak value of the signal to its mean.

In Table 1, x_i represents the time series, $Q1$ and $Q3$ denote the first and third quartiles respectively, and $p(x_i)$ represents the probability of signal x occurring at a value of x_i.

Table 1. Time domain health indicators formula.

Feature	Formula	Feature	Formula
Interquartile range	$I_Q = Q3 - Q1$	Peak to peak	$X_{pp} = Max(x_i) - Min(x_i)$
Kurtosis	$\beta = \frac{1}{n}\sum_{i=1}^{n} x_i^4$	Root mean square	$x_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$
Kurtosis indicator	$K_r = \frac{\beta}{x_{rms}^4}$	Signal entropy	$H = -\sum_{i=1}^{k} p(x_i) \log p(x_i)$
Waveform indicator	$s_f = \frac{x_{rms}}{\lvert\bar{x}\rvert}$	RMSEE	$x_{rmsee} = \frac{1}{10}\sum_{i=1}^{10} -RMS(i) \cdot \log(RMS(i))$
Margin indicator	$X_{CLf} = \frac{x_{max}}{\left[\frac{1}{n}\sum_{i=1}^{n}\sqrt{\lvert x_i \rvert}\right]^2}$	Modulus maximum	$M_{mod} = Max(\lvert x_i \rvert)$
Skewness	$X_s = \frac{\frac{1}{n}\sum_{i=1}^{n} x_i^3}{X_{rms}}$	Teager energy mean	$Mean(\Phi[x]) = Mean((x(n)^2 - x(n-1) \cdot x(n+1)))$
Peak indicator	$X_{Cf} = \frac{x_{max}}{x_{rms}}$	Variance	$X_{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}\left[x_i - \frac{1}{n}\sum_{i=1}^{n} x_i\right]^2$
Pulse indicator	$I_f = \frac{x_{max}}{\lvert\bar{x}\rvert}$	Mean absolute deviation	$M = \frac{1}{N}\sum_{i=1}^{N}\lvert x_i - \bar{x}\rvert$

Frequency-domain features, derived from the analysis of vibration signal frequencies, enable us to identify frequency patterns specific to different fault types, which is crucial for accurately predicting bearing health and RUL. The commonly used frequency-domain features are listed below.

Center frequency [31] describes the centroid of the frequency spectrum, identifying the frequency region where the main energy of the vibration signal is concentrated. The frequency domain amplitude average (FDAA) [32], calculated by averaging the amplitude in the frequency domain, measures the overall energy level of the signal. Peak frequency refers to the frequency at which the maximum amplitude occurs in the vibration signal, often corresponding to specific bearing faults and serving as a key indicator for identifying fault types. Root mean square frequency (RMSF) [33] provides a measure of the average energy distribution in the frequency domain, useful for analyzing the overall energy characteristics of the vibration signal. Spectral energy represents the total energy of the signal in the frequency domain. Spectral entropy measures the randomness and complexity of the spectrum; high spectral entropy indicates the presence of multiple frequency components, aiding in the identification of complex or irregular fault patterns. Spectral flatness assesses the uniformity of the frequency components in the signal spectrum. Spectral kurtosis [34]

measures the thickness of the spectrum tails; high kurtosis indicates prominent peaks in the spectrum, often associated with mechanical faults. Spectral skewness describes the asymmetry of the spectrum; changes in skewness can reflect changes in bearing condition, especially in the early stages of damage development. Spectral spread represents the width of the spectral energy distribution, indicating the dispersion of frequencies in the signal. Standard deviation frequency (SDF) measures the dispersion of amplitude distribution in the frequency domain, used to analyze the variability and instability of the signal in the frequency domain.

In Table 2, $X(k)$ represents the characteristic frequency-domain amplitude spectrum, and f_k is the kth frequency component. p_k denotes the probability density of the kth spectral component. p_k and $\overline{P}$ represent the power of the kth frequency component in the frequency domain and the average power of all frequency components, respectively. F_c is the spectral centroid, and the formula of F_c is $F_c = \frac{\sum_{k=1}^{N} f_k P(k)}{\sum_{k=1}^{N} P(k)}$.

Table 2. Frequency domain health indicators formula.

Feature	Formula
Center frequency	$F_1 = \frac{1}{\sum_{k=1}^{N} X(k)} \sum_{k=1}^{N} f_k X(k)$
Frequency domain amplitude average	$F_2 = \frac{1}{N} \sum_{k=1}^{N} X(k)$
Root mean square frequency	$F_3 = \sqrt{\frac{1}{\sum_{k=1}^{N} X(k)} \sum_{k=1}^{N} f_k^2 X(k)}$
Standard deviation frequency	$F_4 = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (f_k - F_1)^2 X(k)}$
Peak frequency	$F_5 = \frac{\sum_{k=1}^{N} (f_k - F_1)^2 X(k)}{N F_4^4}$
Spectral energy	$F_6 = \sum_{k=1}^{N} \lvert X(k) \rvert^2$
Spectral entropy	$F_7 = -\sum_{k=1}^{N} p_k \cdot \log p_k$
Spectral flatness	$F_8 = \frac{\exp\left(\frac{1}{N} \sum_{k=1}^{N} \log X(k)\right)}{\frac{1}{N} \sum_{k=1}^{N} X(k)}$
Spectral kurtosis	$F_9 = \frac{N \sum_{k=1}^{N} (P(k) - \overline{P})^4}{\left(\sum_{k=1}^{N} (P(k) - \overline{P})^2\right)^2}$
Spectral skewness	$F_{10} = \frac{N \sum_{k=1}^{N} (P(k) - \overline{P})^3}{\left(\sum_{k=1}^{N} (P(k) - \overline{P})^2\right)^{\frac{3}{2}}}$
Spectral spread	$F_{11} = \sqrt{\frac{\sum_{k=1}^{N} (f_k - F_c)^2 P(k)}{\sum_{k=1}^{N} P(k)}}$

Following the analysis of time-domain and frequency-domain features, time-frequency domain feature analysis provides a more comprehensive perspective. It combines both time and frequency information, allowing us to explore the dynamic changes and complexity of signals in greater depth. We extract the energy features of the first five frequency bands based on Wavelet Packet Decomposition (WPD) and the energy features of the first five intrinsic mode function (IMF) components based on Empirical Mode Decomposition (EMD).

2.1.2. Feature Selection Based on Variance

Before conducting feature correlation analysis, this study employs the Variance Threshold method for feature selection. This method evaluates the variance of features within the sample, with lower variance indicating limited information and minimal contribution to the model. By eliminating low-variance features, this approach optimizes the input feature set for model training and reduces interference from irrelevant information.

2.1.3. Pearson–Spearman Correlation Analysis

Pearson correlation analysis is a statistical method used to evaluate the strength of the linear relationship between two variables [35], suitable for continuous data following a normal distribution. The Pearson correlation coefficient ranges from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no linear relationship. The calculation is based on the deviation of the observed values from their mean. The formula for calculating the Pearson correlation coefficient is as follows:

$$r_p = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}} \tag{1}$$

where X_i and Y_i are the ith observations of features X and Y, respectively, $\overline{X}$ and $\overline{Y}$ are the means of features X and Y, and n is the number of observations.

Spearman correlation analysis measures the monotonic relationship between variables without requiring a specific distribution, making it suitable for nonlinear relationships [36]. The Spearman correlation coefficient also ranges from -1 to 1, with larger absolute values indicating stronger relationships. It is calculated based on the rank differences of the data pairs:

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{2}$$

where $d_i = rank(X_i) - rank(P_i)$ represents the rank difference for the ith data pair, and n is the number of data pairs.

This study proposes a method combining Pearson and Spearman correlation analyses. By calculating the absolute mean of these two coefficients, a new comprehensive index, the Pearson–Spearman correlation coefficient r, is formed:

$$r = Corr(x, y) = \frac{|r_p| + |r_s|}{2} \tag{3}$$

where $Corr(x, y)$ represents the correlation between feature X and feature Y.

This index comprehensively considers both linear and nonlinear relationships in the data, aiming to provide a deeper insight into the interactions between feature signals and offer a broader analytical perspective.

2.1.4. HIs Fusion with KPCA and Autoencoders

Despite initial screening, data redundancy remains a significant challenge. Therefore, this study further reduces the dimensionality of HIs by combining KPCA and autoencoders. The dimensionality reduction process involves the following steps:

- Correlation analysis and grouping: First, perform correlation analysis on the features and set a threshold to group highly correlated features together.
- KPCA processing: Apply KPCA to each group of features, retaining principal components that explain 90% of the cumulative variance.
- Autoencoder optimization: Input the retained principal components and ungrouped features into an autoencoder, adjusting the learning weights according to the original number of features. The output from the bottleneck layer of the autoencoder is used as the final HI, representing the bearing's degradation state.

Technical background:

KPCA [37] is a nonlinear dimensionality reduction technique that maps data into a high-dimensional feature space and performs linear PCA, effectively capturing the data's nonlinear structures.

The autoencoder [38] is an unsupervised learning neural network that automatically learns low-dimensional representations of data while capturing nonlinear relationships. The network structure includes the following components:

Input layer: This layer receives the original high-dimensional data X.

Encoder: This is composed of several hidden layers, it gradually reduces the number of neurons to compress the input data into a low-dimensional latent representation.

Bottleneck layer: Located at the end of the encoder, this forms the low-dimensional representation of the data, capturing key features.

Decoder: Symmetrical to the encoder, this gradually increases the number of neurons to reconstruct the data back to its original high-dimensional space.

Output layer: Outputs the reconstructed high-dimensional data X'.

In this study, the output from the bottleneck layer of the autoencoder is used as the final HI to represent the bearing's degradation process. The schematic diagram of the autoencoder is shown in Figure 3.
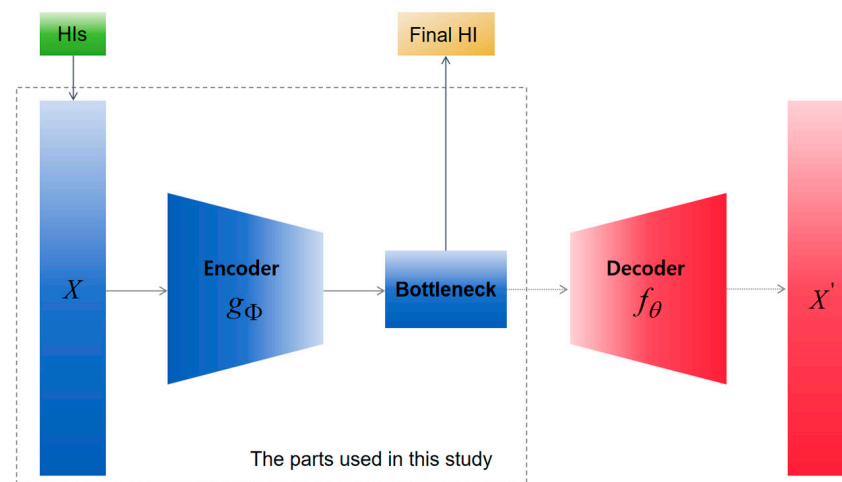

Figure 3. The structure of the autoencoder and the parts used in this study.

2.2. Bearing Degradation Stage Classification

2.2.1. Gaussian Mixture Model (GMM)

Before constructing the stage classification model, it is necessary to first obtain the stage labels. The GMM [39] is a probabilistic clustering algorithm that assumes the data are composed of multiple Gaussian distributions. Each Gaussian distribution is defined by its mean vector and covariance matrix, and these parameters are estimated using the Expectation–Maximization (EM) algorithm. The EM algorithm iteratively optimizes the parameters in two stages:

Expectation step (E-step): Compute the posterior probability, known as the responsibility $\gamma(z_{ik})$, that each data point belongs to each Gaussian distribution. The formula is as follows:

$$\gamma(z_{ik}) = \frac{\pi_k N(x_i|\mu_k, \sum_k)}{\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \sum_j)} \tag{4}$$

where π_k is the mixing coefficient of the k-th Gaussian distribution, and N is the Gaussian probability density function.

Maximization step (M-step): Update the parameters of each Gaussian distribution (mean, covariance, and mixing coefficient) to maximize the log-likelihood function. The update formulas include the following:

1. Mean vector μ_k:

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma(z_{ik}) x_i}{\sum_{i=1}^{N} \gamma(z_{ik})}$$

(5)

2. Covariance matrix $\sum_k$:

$$\sum_k = \frac{\sum_{i-1}^{N} \gamma(z_{ik})(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} \gamma(z_{ik})}$$

(6)

3. Mixing coefficient π_k:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma(z_{ik})$$

(7)

After training is completed, each data point is assigned to the Gaussian distribution with the highest posterior probability, achieving clustering as follows:

$$label(x_i) = \mathrm{argmax}_k \gamma(z_{ik})$$

(8)

In this study, the GMM is used to cluster the health indicators (HIs) extracted in Section 3.1, dividing them into three stages: healthy, slight degradation, and severe degradation. These stages provide labels for the subsequent stage classification model training.

2.2.2. Graph Convolutional Network

The Graph Convolutional Network (GCN) [40] is a specific type of GNN that excels in capturing the structural relationships between nodes. Unlike traditional deep learning methods, the GCN utilizes adjacency matrices to aggregate features from nodes and their neighbors, effectively preserving topological information during feature extraction. This property allows the GCN to demonstrate stronger expressive power when dealing with non-Euclidean data that has complex relationships, making it particularly well-suited for extracting HIs and classifying states in the bearing degradation process, especially when there are complex nonlinear relationships between features at different degradation stages.

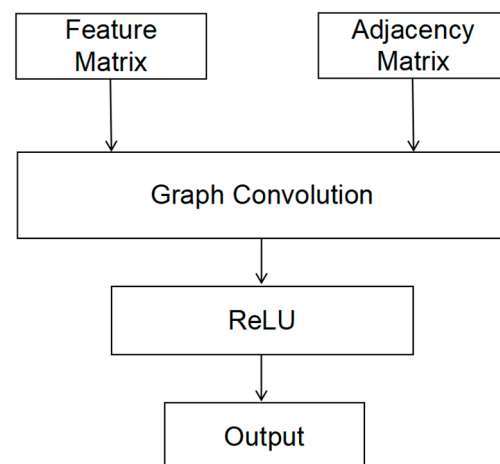The basic structure of the GCN is shown in Figure 4.

Figure 4. GCN structure.

The network input includes the feature matrix X, with dimensions $N \times F$, and the normalized adjacency matrix A, with dimensions $N \times N$. The graph convolutional layer updates the representation of each node by aggregating features from its neighbors as follows:

$$H^{(l+1)} = \sigma(\overline{A} H^{(l)} W^{(l)})$$

(9)

where $H^{(l)}$ represents the node feature representation at the l-th layer (for the 0-th layer, $H^{(0)} = X$, i.e., the initial feature matrix), and $W^{(l)}$ is the learnable weight matrix for the l-th layer. σ is a non-linear activation function, such as ReLU. $\overline{A}$ is the normalized adjacency matrix used to aggregate features from each node and its neighbors.

After each layer of GCN, a non-linear activation function like ReLU is applied. The ReLU function is defined as follows:

$$\sigma(x) = \max(0, x) \tag{10}$$

The activation function introduces non-linearity, allowing the model to learn more complex relationships.

2.2.3. GCN–LSTM

During the bearing degradation process, the health state is often associated with complex spatiotemporal dependencies, which traditional classification methods may struggle to fully capture. To address this, the present study proposes a model architecture that combines GCN and LSTM networks to improve the classification accuracy of degradation stages. The model integrates the spatial feature extraction capability of the GCN with the temporal modeling strengths of LSTM, thereby enhancing classification performance. Specifically, the GCN aggregates features from nodes and their neighbors to capture spatial dependencies in the degradation process, while LSTM models the dynamic evolution of these features over time [41]. This synergy enables the GCN–LSTM model to simultaneously capture the complex relationships between nodes during bearing degradation and the temporal trends in the sequence data. Furthermore, by incorporating a global attention mechanism, the GCN–LSTM model can automatically identify critical time steps in the degradation process, further improving classification accuracy and robustness.

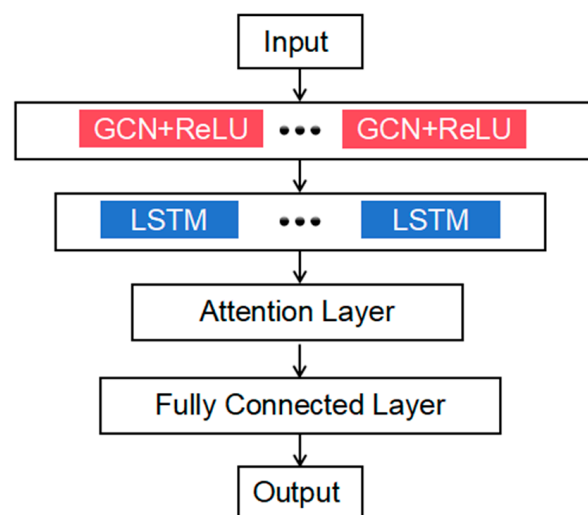The structure of the GCN–LSTM model is illustrated in Figure 5.

Figure 5. GCN–LSTM stage classification model structure.

The input data consist of two components: the feature matrix X and the adjacency matrix A. The feature matrix captures X, the characteristics of bearing nodes at each time step, while the adjacency matrix A represents the connectivity relationships between these nodes. The construction of the adjacency matrix is detailed as follows:

1. Euclidean distance calculation:

First, the Euclidean distance between each node and all other nodes is calculated, with smaller Euclidean distances indicating higher similarity between nodes. The Euclidean distance between nodes i and j is computed as follows:

$$d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \tag{11}$$

where $x_{i,k}$ represents the feature value of the k-th feature for node i.

2. Initial connections:

Each node is initially connected to itself via a self-loop, ensuring that every node has at least one connection. In addition, each node is connected to its three preceding and three succeeding neighbors based on their time steps. Specifically, node i is connected to nodes at time steps $i-3$, $i-2$, $i-1$, $i+1$, $i+1$, $i+2$, and $i+3$, with periodic boundary conditions (i.e., connections wrap around at the beginning and end of the time sequence).

3. Additional neighbors selection:

In addition to the aforementioned connections, each node is connected to the three nodes with the smallest Euclidean distances (i.e., the closest neighbors in terms of feature similarity). This step ensures that the connectivity reflects not only the temporal order but also the feature-based similarity between nodes.

4. Weight distribution:

The weight of the self-loop is set to 0.1. For the other connections (both the preceding, succeeding, and nearest neighbors), the weights are inversely proportional to the Euclidean distances. Specifically, the weight $w(i,j)$ between two nodes i and j is calculated as follows:

$$w(i,j) = \frac{1}{d(i,j) + \varepsilon} \tag{12}$$

where ε is a small positive constant to avoid division by zero. To ensure that the sum of all weights for a given node equals 1, we normalize the weights:

$$w'(i,j) = \frac{0.9w(i,j)}{\sum_{j'} w(i,j')} \tag{13}$$

where the denominator is the sum of the weights of all connections for node i, excluding the self-loop. After normalization, the total weight of these connections equals 0.9, and together with the self-loop weight of 0.1, the sum equals 1.

The process of constructing the adjacency matrix, as described above, can be implemented through the following Algorithm 1.

To capture temporal dependencies, the data are divided into sequences of length, with each sequence containing T consecutive time steps, and the input shape is (batch_size, T, N, F), where N is the number of nodes and F is the feature dimension of each node. For each time step, the GCN processes the node feature matrix X_t and the adjacency matrix A_t. Through multiple layers of graph convolution, GCN effectively aggregates information from each node and its neighbors, thereby generating updated node representations. The GCN output H_t is used as input to the LSTM layer. The LSTM captures long-term dependencies and short-term variations in the temporal sequence. For each sequence, the input dimension to LSTM is (batch_size, T, F'), where F' is the node feature dimension generated by the GCN.

Algorithm 1. The pseudocode for constructing the adjacency matrix in the RUL prediction model.

Input: Feature matrix *X*, Node count *N*, Number of neighbors *k*, Self-loop weight W_s
Output: Adjacency matrix *A*

for *i* in range(1, *N*) **do**:
  for *j* in range(1, *N*) **do**:
    if *i* != *j*:
      distance(*i*, *j*) = *compute_distance*(*X*[*i*], *X*[*j*]);
    end if
  end for
  A[*i*][*i*] = *Ws*;
  for *offset* in range(−3, 4):
  neighbor_idx = (*i* + *offset*) % *N*;
  if *neighbor_idx* != *i*:
    A[*i*][*neighbor_idx*] = 1;
    end if
  end for
  sort *distances*[*i*] in ascending order;
  nearest_neighbors = *distances*[*i*][:*k*];
  for *neighbor* in *nearest_neighbors* **do**:
  A[*i*][*neighbor*] = 1/(*distance*(*i*, *neighbor*) + *epsilon*);
  end for
  row_sum = *sum*(*A*[*i*]) − *A*[*i*][*i*];
  for *j* in range(1, *N*) **do**:
  if *A*[*i*][*j*] != 0:
    A[*i*][*j*] = (1 − *Ws*)*A[*i*][*j*]/*row_sum*;
  end if
  end for
 return *A*;

To further enhance the model's classification accuracy, a global attention mechanism is introduced after the LSTM layer. The global attention mechanism weights each time step in the entire sequence, automatically identifying the most critical time steps for determining the bearing degradation stage, and assigning higher weights to these critical moments. This mechanism allows the model to focus on the key moments related to degradation while avoiding over-reliance on less relevant time steps, thereby improving classification accuracy. The output of the LSTM is further used for stage classification.

To achieve three-stage classification, this study employs two GCN–LSTM networks to classify between healthy/slight degradation and slight degradation/severe degradation.

2.3. Bearing Remaining Useful Life Prediction

2.3.1. GraphSAGE

GraphSAGE [42] is an efficient GNN model designed to address the challenge of learning from large-scale graph-structured data. Unlike traditional GCN, GraphSAGE utilizes a sampling strategy that aggregates information from a fixed number of neighboring nodes to generate node embeddings. This aggregation mechanism enhances the model's expressive power while significantly reducing computational complexity, making it particularly suitable for processing large-scale graph data.

One of the primary advantages of GraphSAGE is its inductive learning capability, which enables the model to make predictions on unseen nodes, making it highly effective in dynamic graph environments. Specifically, GraphSAGE learns to aggregate local neighborhood features to create a global representation, making it particularly effective for tasks such as node classification and prediction.

The basic structure of GraphSAGE consists of the following steps:

Neighbor sampling: A fixed number of neighboring nodes are randomly sampled for each node, reducing computational costs and improving training efficiency.

Feature aggregation: Features from the sampled neighbors are aggregated using techniques such as mean aggregation, max-pooling, or LSTM-based aggregation.

Node embedding update: The aggregated neighbor features are concatenated with the node's own features and transformed using a learnable linear transformation to generate the updated node embedding.

This flexible sampling and aggregation mechanism enables GraphSAGE to effectively capture local relationships within the graph while maintaining scalability and memory efficiency. When applied to bearing degradation prediction, GraphSAGE can extract both local and global features, providing high-quality feature representations for subsequent RUL prediction tasks.

The architecture of GraphSAGE follows these key steps: first, the model samples local neighborhood information for each node to ensure that the node's embedding incorporates its surrounding graph structure. Next, the aggregation strategy combines these neighboring node features to form a new feature representation. These aggregated features are concatenated with the node's own features and transformed using learnable weight matrices to generate the updated node embeddings. This process can be repeated across multiple layers to refine the node representations.

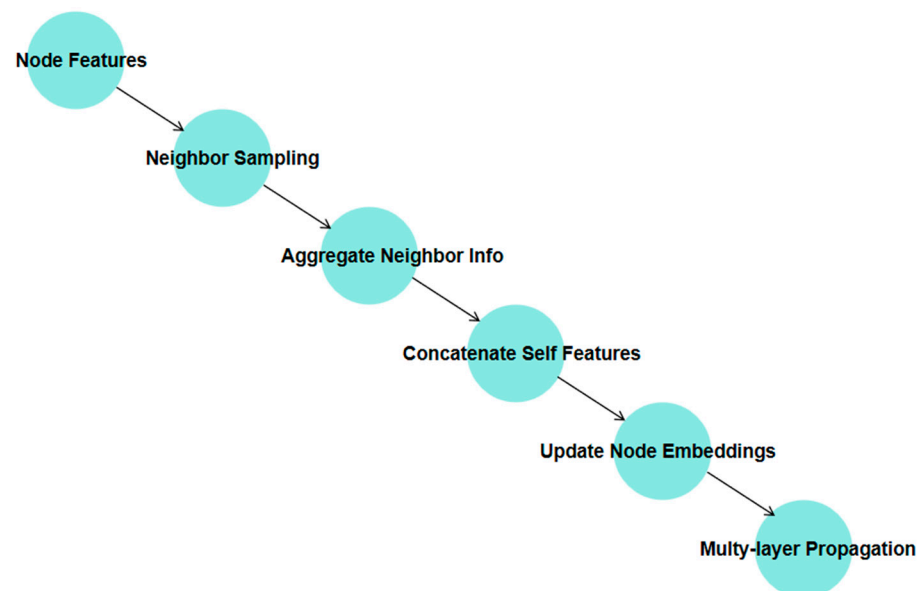The network architecture is illustrated in Figure 6.

Figure 6. GraphSAGE structure schematic.

GraphSAGE's flexibility and efficiency make it well-suited for handling large-scale data in bearing degradation prediction, particularly in scenarios where capturing and modeling degradation features at different stages is crucial. By effectively aggregating local node information, GraphSAGE provides reliable input for the AAGL model, thereby improving the accuracy and robustness of RUL predictions for bearings.

2.3.2. Adaptive Attention GraphSAGE–LSTM

This study proposes the Adaptive Attention GraphSAGE–LSTM (AAGL) model, which integrates the spatial feature extraction capabilities of GraphSAGE with the temporal sequence modeling power of LSTM and incorporates an adaptive attention mechanism. Additionally, the adjacency matrix in the model's input is constructed based on the degradation stage, in order to better address the complex characteristics of the bearing degradation process. The input to the model includes both the feature matrix and the adjacency matrix. The construction of the adjacency matrix in this model differs from the classification model, as it is adjusted according to the bearing's degradation stages to better capture the varying feature changes over time.

The overall structure of the model consists of the following components, as shown in Figure 7:

1. GraphSAGE feature extractor: First, GraphSAGE is used to extract both local and global spatial features from bearing data. These features reflect the bearing's state during both healthy and degraded phases, providing input for subsequent time-series modeling. The multi-layer structure of GraphSAGE allows it to iteratively aggregate information from neighboring nodes, capturing high-order dependencies. This is especially well-suited for bearing degradation data with complex topologies.

2. LSTM temporal dynamics modeling: The extracted spatial features are fed into an LSTM network to capture the temporal dynamics of the bearing degradation process. LSTM effectively models long-term dependencies over time, helping to identify trends and changes across different degradation phases. Given that degradation process features often exhibit significant temporal dependencies, LSTM's memory mechanism is well-equipped to model these, providing more accurate predictions of degradation states.

3. Adaptive attention mechanism: The model incorporates different attention mechanisms at various stages of degradation to better capture critical features. In the early stages of degradation, where it is important to detect overall trends, the model uses a global attention mechanism to identify key features across the entire sequence, aiding in the early detection of degradation trends. In the severe degradation stage, where short-term features become more significant, the model applies a local attention mechanism to focus on rapid changes over short periods, providing a more detailed description of severe degradation.
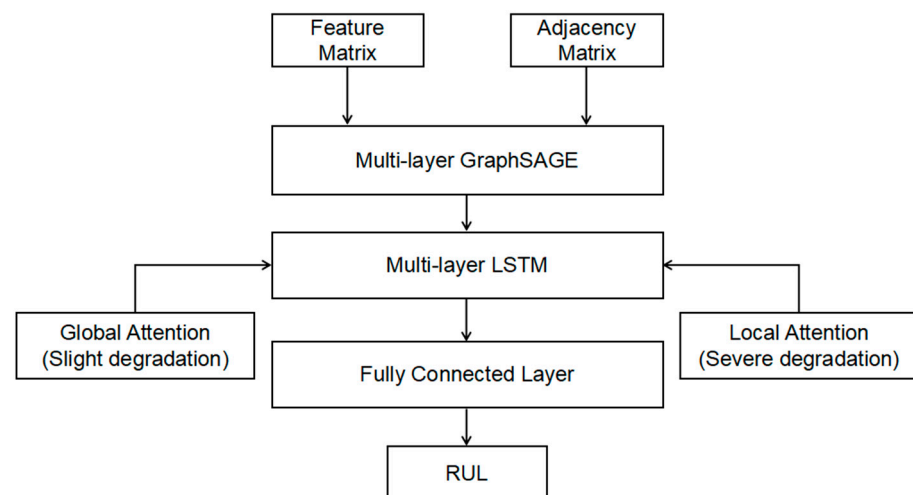

Figure 7. RUL prediction model structure.

The construction of the adjacency matrix is adjusted according to the different degradation stages of the bearing, as described in the following logic:

- Healthy stage: Since feature changes are minimal, the adjacency matrix construction remains simple. Each node is connected to itself, as well as to its preceding and following neighbors, with the weights evenly distributed (each edge weight set to 0.33). This ensures basic connectivity while avoiding overfitting by irrelevant information.
- Slight degradation stage: As feature changes begin to manifest, each node is connected to the two preceding and two following neighbors, as well as to the two other nodes with the smallest Euclidean distances. This results in a total of seven connected nodes, including itself. The self-loop weight is set to 0.1, while the weights for the other connections are assigned based on the inverse of the Euclidean distances, as described in Equations (12) and (13), ensuring that more similar nodes receive higher weights.
- Severe degradation stage: As feature changes become more pronounced, each node is connected to the two preceding and two following neighbors, along with the four other nodes with the smallest Euclidean distances. This forms connections to a total of nine nodes, including itself. The self-loop weight remains at 0.1, and the weights for the other nodes are distributed according to the inverse of the Euclidean distances, as described in Equations (12) and (13).

The adjacency matrix construction allows the model to dynamically adjust its feature extraction and attention mechanism based on the bearing's degradation stage, enhancing the model's adaptability and improving the accuracy of the RUL prediction. The pseudocode for the construction of the adjacency matrix is shown in Algorithm 2.

The model's output is the prediction of the remaining useful life (RUL), which is estimated using both the spatial and temporal features extracted from the degradation process. The model's performance is evaluated by comparing predicted values with actual RUL values from the test data, showcasing its effectiveness in predicting the lifespan of bearings under different operational conditions.

Algorithm 2. The pseudocode for constructing the adjacency matrix in the RUL prediction model.

Input: Feature matrix X, Node count N, Stage type (Healthy, Slight Degradation, Severe Degradation), Self-loop weight Ws
Output: Adjacency matrix A

```
for i in range(1, N) do:
    for j in range(1, N) do:
        if i != j:
            distance(i, j) = compute_distance(X[i], X[j]);
end if
    end for

    if stage_type == "Healthy":
        A[i][i] = Ws;
        for offset in range(−1, 2):
            neighbor_idx = (i + offset) % N;
            if neighbor_idx != i:
                A[i][neighbor_idx] = 0.33;
            end if
        end for
```

Algorithm 2. *Cont.*

```
        else if stage_type == "Slight Degradation":
            A[i][i] = Ws;
            for offset in range(−2, 3):
                neighbor_idx = (i + offset) % N;
                if neighbor_idx != i:
                    A[i][neighbor_idx] = 1;
                end if
            end for
            sort distances[i] in ascending order;
            nearest_neighbors = distances[i][:2];
            for neighbor in nearest_neighbors do:
                A[i][neighbor] = 1/(distance(i, neighbor) + epsilon);
              end for
      else if stage_type == "Severe Degradation":
            A[i][i] = Ws;
            for offset in range(−2, 3):
                neighbor_idx = (i + offset) % N;
                if neighbor_idx != i:
                    A[i][neighbor_idx] = 1;
                end if
            end for
            sort distances[i] in ascending order;
            nearest_neighbors = distances[i][:4];
            for neighbor in nearest_neighbors do:
                A[i][neighbor] = 1/(distance(i, neighbor) + epsilon);
            end for
        end if

      if stage_type != "Healthy":
          row_sum = sum(A[i]) − A[i][i];
          for j in range(1, N) do:
              if A[i][j] != 0:
                  A[i][j] = (1 − Ws)*A[i][j]/row_sum;
              end if
          end for
      end if
  end for

  return A;
```

---

3. Experimental Results

3.1. HI Extraction

This study utilized the IMS bearing dataset provided by the Intelligent Maintenance System Center at the University of Cincinnati. During testing, four Rexnord ZA-2115 double-row bearings were mounted on a shaft, with accelerometers attached to the bearing housings to monitor vibrations. Figure 8 illustrates the test setup, which includes an oil circulation system for lubrication and a magnetic plug on the oil feedback pipe to collect debris. When the debris accumulation exceeds a predefined threshold, an electrical switch

automatically stops the test. After testing, the bearings were inspected to record their failure modes in detail.
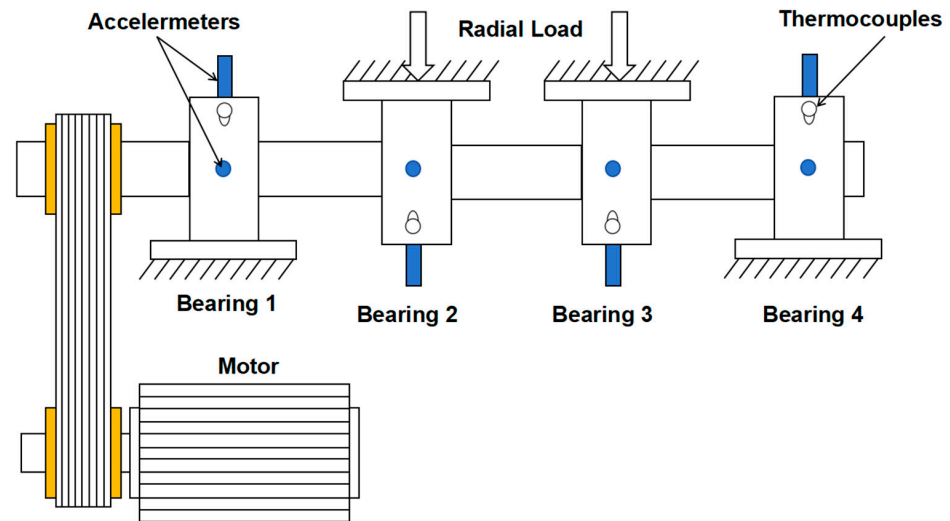

Figure 8. Schematic diagram of the test rig.

The test results yielded 2156, 984, and 6324 data points for the three sets, respectively. Specific failures included inner race defects for Bearing 1-3 and rolling element defects for Bearing 1-4 in the first set. Outer race failures were observed for Bearing 2-1 in the second set and Bearing 3-3 in the third set. This paper analyzes the data starting from the 500th time step for Bearings 1-3 and 1-4, while using the full dataset for Bearings 2-1 and 3-3. For data acquisition, each bearing in the first set had two signal acquisition channels, whereas each bearing in the second and third sets had only one signal acquisition channel.

Forty-one features were extracted from the vibration data of each signal channel (see Table 3 for details). We calculated the variance of each feature for Bearings 1-3, 1-4, 2-1, and 3-3 and summed the variances of the same features. A threshold θ of 0.025 was set, and features below this threshold were eliminated. This threshold was chosen based on the distribution of feature variances (as shown in Figure 9). Specifically, the first five features exhibited significantly lower variances compared to the others, indicating minimal variation across samples and potentially limited contribution to the prediction task. By setting the threshold to 0.025, these low-variance features, including mean, WPE1, median, LSS, and mean square value, were effectively filtered out while retaining most high-variance, informative features.
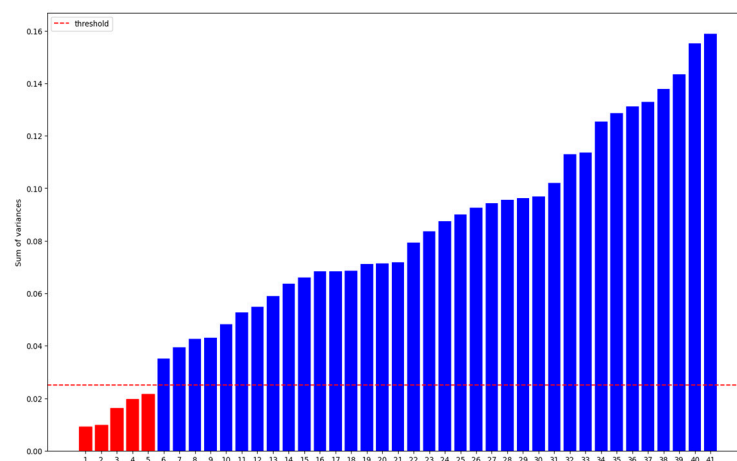

Figure 9. Ranking of variances and filtering of results.

Table 3. Features before and after initial selection.

Features Before Initial Selection		Features After Initial Selection	
IQR	FDAA	IQR	Spectral entropy
Kurtosis	Peak frequency	Kurtosis	Spectral flatness
Kurtosis indicator	RMSF	Kurtosis indicator	Spectral kurtosis
MAD	Spectral energy	MAD	Spectral skewness
Margin indicator	Spectral entropy	Margin indicator	Spectral spread
LSS	Spectral flatness	Modulus max	SDF
Mean	Spectral kurtosis	Peak indicator	EMD1
Mean square value	Spectral skewness	Peak to peak	EMD2
Median	Spectral spread	Pulse indicator	EMD3
Modulus max	SDF	RMSEE	EMD4
Peak indicator	EMD1	Root mean square	EMD5
Peak to peak	EMD2	Signal entropy	WPE2
Pulse indicator	EMD3	Skewness	WPE3
RMSEE	EMD4	Teager energy mean	WPE4
Root mean square	EMD5	Variance	WPE5
Signal entropy	WPE1	Waveform indicator	
Skewness	WPE2	Center frequency	
Teager energy mean	WPE3	FDAA	
Variance	WPE4	Peak frequency	
Waveform indicator	WPE5	RMSF	
Center frequency		Spectral energy	

The choice of 0.025 as the threshold is not arbitrary; it aligns with the observed data characteristics and is consistent with commonly accepted practices in feature engineering, where variance thresholds typically range between 0.01 and 0.05. The variance ranking and the changes in features before and after initial screening are shown in Figure 9 and Table 3.

To ensure consistency across features, the initially screened features were normalized. Pearson–Spearman correlation coefficients were then calculated between each feature and all other features, with a threshold set at 0.75. The selection of this threshold was based on the following considerations: Wang et al. [13] employed the Pearson correlation coefficient with a threshold of 0.8. However, in this study, the Spearman correlation coefficient was additionally introduced to account for both linear and nonlinear relationships, resulting in a Pearson–Spearman composite coefficient that tends to be slightly lower compared to the standalone Pearson coefficient. Experimental results showed that setting the threshold to 0.8 led to overly sparse correlation results, making subsequent feature grouping challenging. By slightly reducing the threshold to 0.75, more reasonable correlation results were obtained, which better aligned with the requirements of feature grouping. The correlation results for Bearings 1-3 and 1-4 are shown in Figures 10 and 11, where the red areas indicate significant correlations between features.

The highly correlated HIs were grouped together. Specifically, the grouping results for Bearings 1-3 and 1-4 are shown in Tables 4 and 5, respectively.

KPCA was applied to each group of HIs using an RBF kernel, with the gamma parameter set to 0.01, retaining principal components that explain over 90% of the variance. These principal components, along with the ungrouped HIs, were used as inputs to the autoencoder. Specifically, for Bearing 1-3, seven principal components were retained along with three ungrouped HIs, totaling ten HIs. For Bearing 1-4, three principal components and four ungrouped HIs were retained, totaling seven HIs.
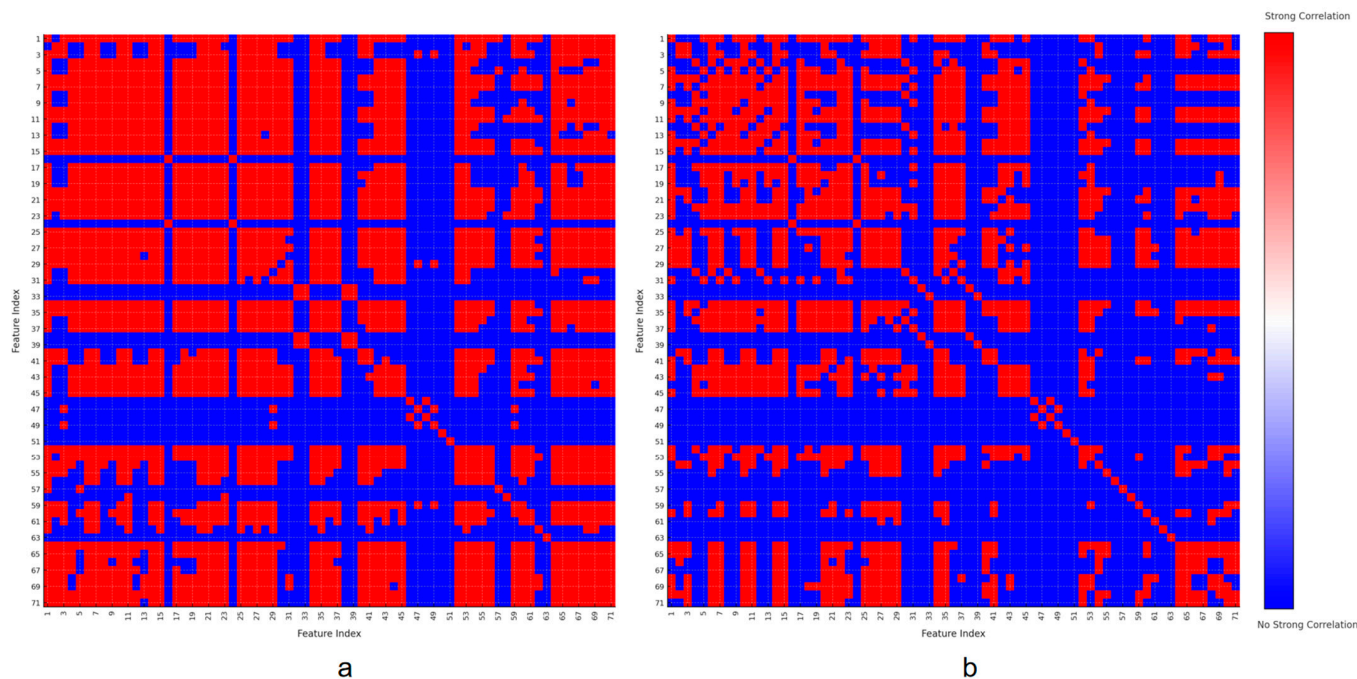
Figure 10. Comparison of feature correlation results for Bearings 1-3 under different Pearson–Spearman correlation thresholds: (**a**) threshold = 0.75 and (**b**) threshold = 0.8.
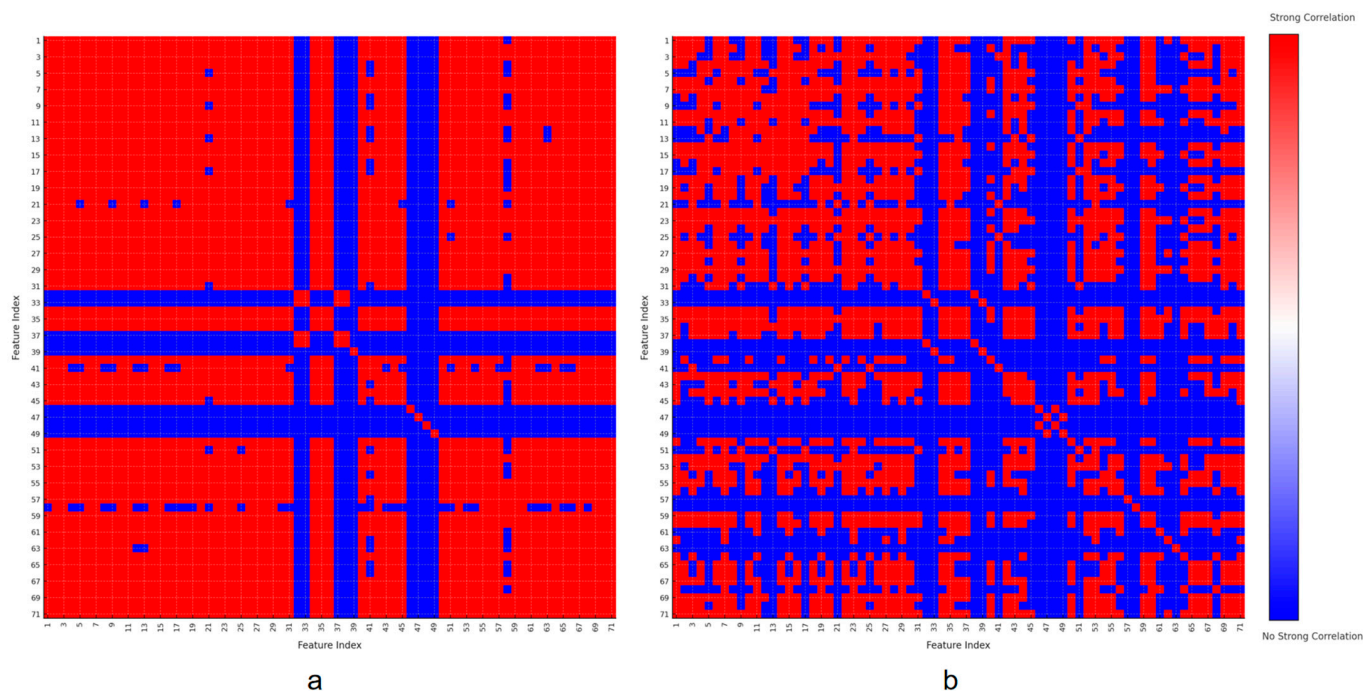

Figure 11. Comparison of feature correlation results for Bearings 1-4 under different Pearson–Spearman correlation thresholds: (**a**) threshold = 0.75 and (**b**) threshold = 0.8.

Table 4. Bearing 1-3 HI grouping results.

Group1	Group2	Group3	Group4
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 35, 36, 37, 38, 41, 42, 43, 44, 45, 46, 48, 50, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 65, 66, 67, 68, 69, 70, 71, 72	33, 34, 39, 40	17, 25	47, 49

Table 5. Bearing 1-4 HI grouping results.

Group1	Group2
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 35, 36, 37, 40, 41, 42, 43, 44, 45, 46, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72	33, 34, 38, 39

In this study, only the encoder part of the autoencoder structure was utilized to reduce the dimensionality of the input HIs into a low-dimensional latent space representation, effectively capturing the degradation process of the bearings. The encoder consists of two fully connected (FC) layers: the first layer maps the input features from their original dimension to 64 dimensions with a ReLU activation function, while the second layer further compresses the features into a latent space of one dimension. During training, the autoencoder minimizes the mean squared error (MSE) between the reconstructed and input data to ensure the effectiveness of the latent representation. Although the decoder is involved in the training process to enhance the latent space's quality, only the output of the encoder (i.e., the bottleneck layer) is used for dimensionality reduction in practical applications. The hyperparameters for the encoder were set as follows: the input dimension is determined by the number of health indicators, the output dimension is fixed at one (latent space), the learning rate is 0.001, the optimizer is Adam, and the number of training epochs is 200 to ensure sufficient training and convergence. This design enables the fused HIs to effectively extract low-dimensional features, resulting in HIs that are suitable for bearing health assessment. The HI fusion process for Bearings 1-3 and 1-4 is illustrated in Figure 12.
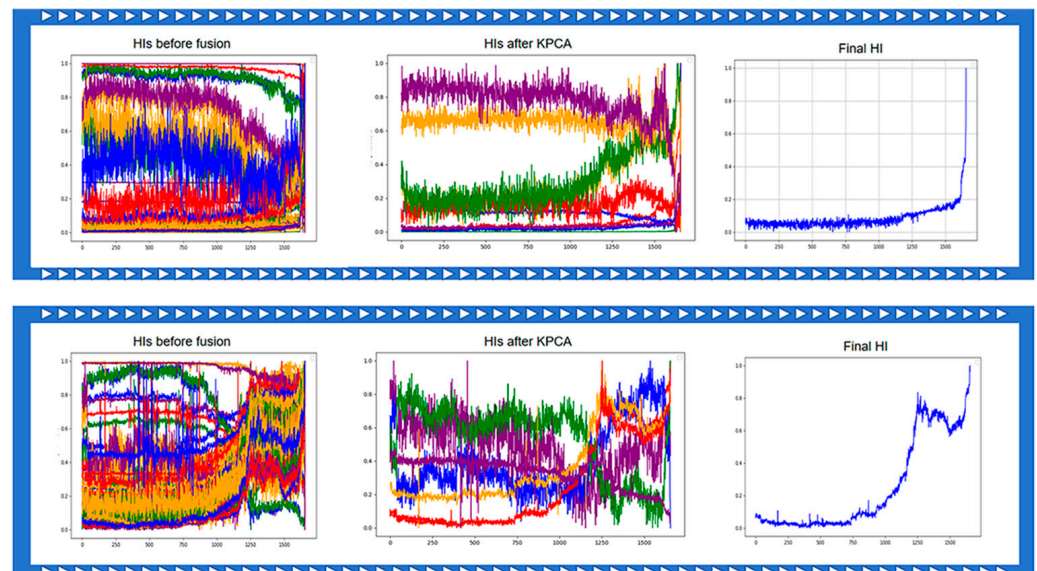
Figure 12. Bearing 1-3 (**top**) and Bearing 1-4 (**bottom**) HI fusion process.

KPCA and autoencoder dimensionality reduction were also performed on the HIs for Bearings 2-1 and 3-3, generating the final HI. To improve the accuracy of the stage classification model, the Savitzky–Golay filter was applied to smooth the HIs. The results are shown in Figure 13.
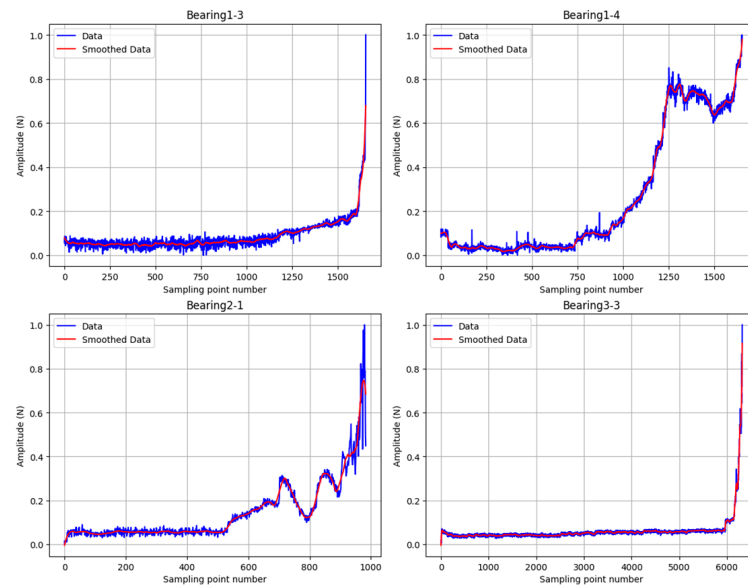
Figure 13. Result of HI smoothing.

The final HI is used for the subsequent construction of the stage classification model and the RUL prediction model.

3.2. Stage Classification

The GMM was used to perform clustering analysis on the HIs obtained for bearings 1-3, 1-4, 2-1, and 3-3 in Section 3.1. The clustering results served as stage labels for the data. The detailed clustering results are shown in Figure 14.
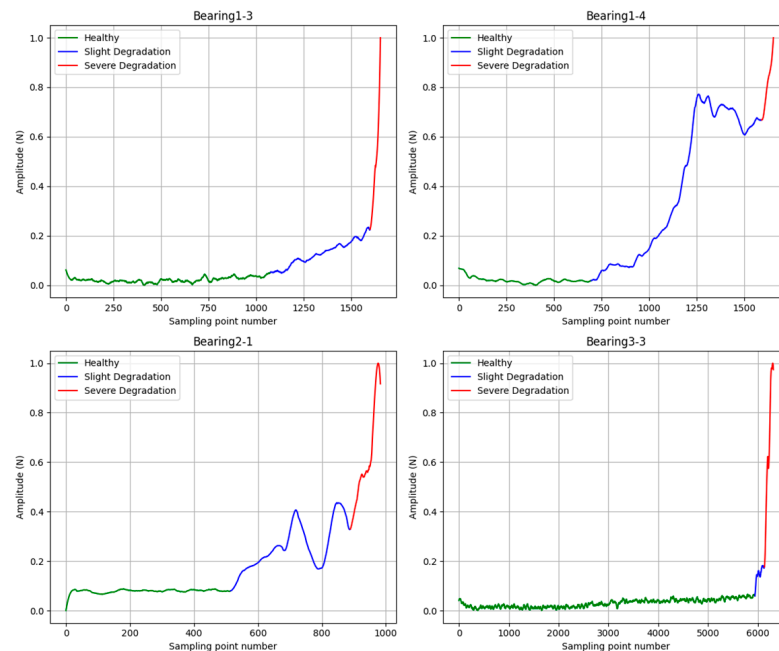

Figure 14. Clustering results.

This study employs two binary classification networks based on GCN–LSTM to classify the stages of bearing degradation data. Specifically, the two classification networks are as follows:

1. Healthy stage vs. slight degradation stage classification network: This network is used to distinguish between the healthy stage and the slight degradation stage of bearings.

2. Slight degradation stage vs. severe degradation stage classification network: This network is used to differentiate between the slight degradation and severe degradation stages.

The network parameters are detailed in Tables 6 and 7.

Table 6. Health/slight degradation classification model parameters.

Parameters	Parameter Value	Parameters	Parameter Value
GCN layers	3	Activation function	ReLU
GCN channels	32	Optimizer	NAdam
LSTM layers	6	Learning rate	0.0001
LSTM hidden layers	64	Loss function	CrossEntropyLoss

Table 7. Slight degradation/severe degradation classification model parameters.

Parameters	Parameter Value	Parameters	Parameter Value
GCN layers	3	Activation function	ReLU
GCN channels	64	Optimizer	NAdam
LSTM layers	8	Learning rate	0.00015
LSTM hidden layers	64	Loss function	CrossEntropyLoss

The inputs to these classification networks include the feature matrix and the adjacency matrix. The feature matrix is composed of previously obtained health indicators (HIs), while the adjacency matrix represents the relationships between nodes, as described in Section 2.2.3. The training set for the model consists of data from Bearing 1-3, which includes all three degradation stages—healthy, slight degradation, and severe degradation—allowing the model to comprehensively learn feature variations and degradation patterns. To evaluate the model's generalization capability, the complete datasets of Bearings 1-4, 2-1, and 3-3, which were not used during training, were designated as the test set.

Before being fed into the model, all HI data were normalized to the range [0, 1] to eliminate dimensional disparities and improve training efficiency. Additionally, the training labels were generated based on clustering results obtained using the Gaussian Mixture Model (GMM). The training process lasted for 300 epochs to ensure sufficient model convergence, with a batch size of 128 to balance training efficiency and computational resource usage. The experimental results, as shown in Figure 15, illustrate the classification outcomes for the healthy vs. slight degradation and slight degradation vs. severe degradation stages. Figure 15A1/A2, B1/B2, and C1/C2 present the classification results for Bearings 1-4, 2-1, and 3-3, respectively, validating the model's ability to accurately classify data from unseen bearings.

To demonstrate the superiority of this model over other models in stage classification accuracy, this study also compared the GCN–LSTM results with those from CNN, GCN, and CNN–LSTM models. Each model involved two binary classification tasks, and the classification accuracies are listed in Table 8:

Table 8. Comparison of the performance of stage classification models.

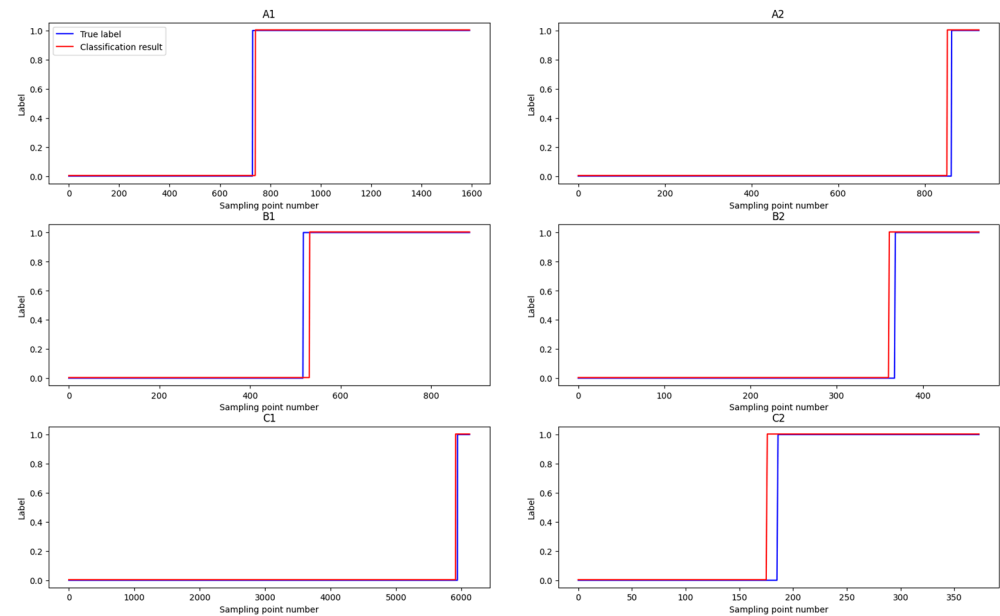| Models	Bearing 1-4	Bearing 2-1	Bearing 3-3
GCN–LSTM	99.31%/98.92%	98.42%/98.50%	99.53%/97.33%
CNN	82.27%/76.13%	79.81%/80.29%	72.45%/69.88%
GCN	88.40%/83.44%	84.62%/82.46%	86.91%/87.69%
CCN–LSTM	94.57%96.83%	89.45%/90.68%	93.40%/94.74%

Figure 15. Results of stage classification. Subfigure (**A1**,**A2**) represent the classification results for Dataset A, (**B1**,**B2**) for Dataset B, and (**C1**,**C2**) for Dataset C. The blue line indicates the true label, and the red line indicates the classification result.

The experimental results indicate that the GCN–LSTM model can accurately distinguish the bearing state across different degradation stages, providing a solid foundation for subsequent RUL prediction.

3.3. RUL Prediction

After completing the stage classification, this study employed an RUL prediction model to estimate the lifespan of bearings.

In this study, the training process includes pre-training during the healthy stage and formal training during the degradation stages. The inputs to the model consist of a feature matrix and an adjacency matrix. The feature matrix is composed of the HIs of the bearings, while the construction logic of the adjacency matrix is described in Section 2.3.2.

During the healthy stage, pre-training was performed using data from Bearings 1-3, enabling the model to fully learn the baseline characteristics of bearings under normal operating conditions. This served as a robust starting point for the subsequent modeling of the degradation stages. The GraphSAGE module was employed to extract the spatial features of the nodes, with feature dimensions set to (8, 16), (16, 32), (32, 64), and (64, 128) across four layers. The extracted spatial features were fed into a multi-layer LSTM module, which consists of five layers with hidden units set to (128, 50), (50, 100), (100, 150), (150, 200), and (200, 100), respectively. The model's parameters were optimized using the mean squared error (MSE) loss function, with AdamW as the optimizer. The learning rate was set to 0.0015, and the pre-training process lasted for 52 epochs.

During the degradation stages, formal training was conducted using data from Bearings 1-3 during the slight and severe degradation stages. The GraphSAGE module continued to extract spatial features from the degradation data, which were then processed by the LSTM module for multi-layer time series modeling. Attention mechanisms were employed to enhance the model's capability to adapt to different degradation stages:

For the slight degradation stage, a global attention mechanism was utilized to capture critical features across the entire time sequence.

For the severe degradation stage, a local attention mechanism was applied to focus on rapid changes within short time windows, improving sensitivity to local features.

During formal training, the learning rate was set to 0.008, and the model was trained for 400 epochs using the AdamW optimizer. To prevent overfitting, regularization techniques such as Dropout and early stopping were incorporated.

The validation process utilized data from Bearings 1-4, 2-1, and 3-3, which were not included in the training set, to assess the model's generalization ability. The prediction results, shown in Figure 16. The performance of the model was comprehensively evaluated using multiple metrics, including root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), coefficient of determination (R^2), and adjusted R^2, as shown in Table 9. These metrics provide complementary perspectives on the model's prediction accuracy and generalization ability. RMSE and MAE measure absolute errors, with RMSE emphasizing larger deviations, while MAPE evaluates relative errors as percentages. R^2 and adjusted R^2 assess the proportion of variance in the actual RUL explained by the model, with adjusted R^2 accounting for the complexity of the model.
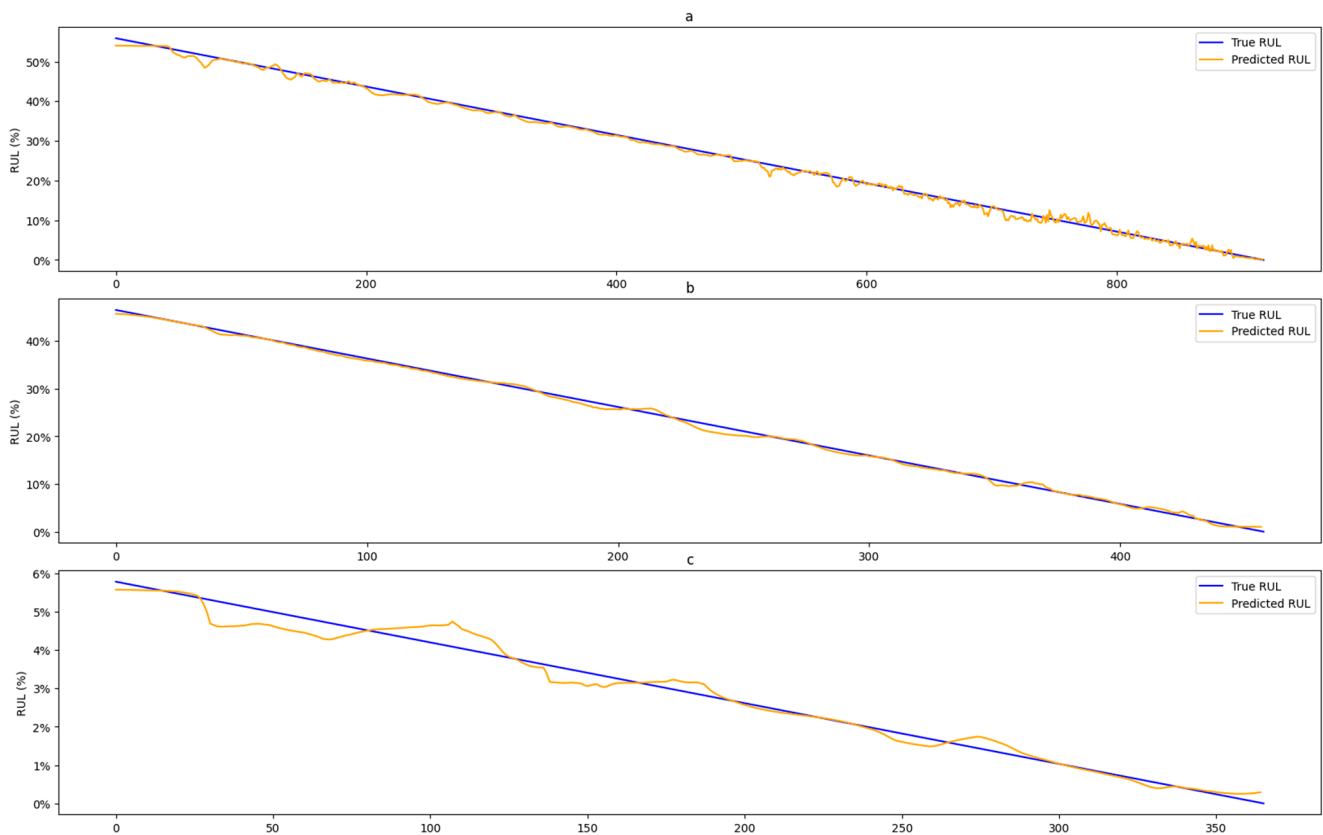

Figure 16. Result of RUL prediction. (**a**) Bearing 1-4 prediction. (**b**) Bearing 2-1 prediction. (**c**) Bearing 3-3 prediction.

Table 9. RMSE, MAE, MAPE, and R^2 evaluation for bearing lifespan prediction.

Bearings	RMSE	MAE	MAPE	R^2	Adjusted R^2
Bearing 1-4	0.00706	0.00599	0.04994	0.997555	0.997552
Bearing 2-1	0.00494	0.00403	0.07415	0.998545	0.998542
Bearing 3-3	0.00265	0.00180	0.17506	0.978568	0.978509

From Table 9, it can be observed that the proposed model achieves consistently low RMSE, MAE, and MAPE values across Bearings 1-4, 2-1, and 3-3, demonstrating high prediction accuracy and robustness. The R^2 and adjusted R^2 values, all exceeding 0.97, confirm the model's ability to effectively capture the variance in the actual RUL. The prediction results were further compared with the bearing lifetime predictions from three

other studies using the IMS bearing dataset, as shown in Table 10. This comparison highlights the superior performance of the proposed model in predicting RUL across different degradation stages. By comparing the predicted values with the actual ones, it is evident that the model accurately captures degradation patterns with minimal prediction errors, underscoring its reliability and generalization capability.

Table 10. RMSE comparison for bearing life prediction across different studies.

Models	1-4	2-1	3-3
AAGL	0.00706	0.00494	0.00265
[43]	0.00772	0.00525	0.00732
[44]	0.1739	-	-
[45]	0.0691	0.1071	0.0822

To prevent overfitting during training, regularization techniques such as early stopping and dropout were implemented. The validation results further confirm that the model can make accurate predictions on bearings it has not encountered during training, highlighting its robustness and generalization ability.

This study compares the proposed AAGL model with three baseline models: LSTM, CNN–LSTM, and GCN–LSTM. All models incorporate an adaptive attention mechanism, and both GCN–LSTM and AAGL share the same phase-based adjacency matrix construction logic. Table 11 presents the RMSE results of RUL prediction across these models.

Table 11. RMSE comparison for bearing life prediction across different models.

Models	1-4	2-1	3-3
LSTM	0.02732	0.01492	0.02346
CNN–LSTM	0.01613	0.00963	0.01372
GCN–LSTM	0.00967	0.007670	0.00663
AAGL	0.00706	0.00494	0.00265

Using the LSTM model as the baseline, the CNN–LSTM model demonstrates a significant improvement by leveraging the CNN module's capability to extract local features, resulting in an overall RMSE reduction of 39.91%. The GCN–LSTM model further enhances prediction accuracy by utilizing graph neural networks to model spatial dependencies and leveraging the phase-based adjacency matrix to capture relationships between degradation phases, achieving a 63.52% overall RMSE improvement compared to LSTM.

In comparison, the AAGL model outperforms all other models by introducing the GraphSAGE module, which employs neighborhood sampling and aggregation mechanisms to efficiently capture local features in the graph structure. Combined with the LSTM module's ability to model temporal dependencies, AAGL excels at capturing the complex spatiotemporal relationships inherent in the bearing degradation process. Consequently, AAGL achieves the best prediction performance, with an overall RMSE reduction of 77.72% compared to LSTM, demonstrating its superiority in the RUL prediction task.

3.4. Parameter Analysis

Different parameter settings can have varying impacts on the predictive performance of the model. In this experiment, different combinations of pre-training and formal training learning rates were used, and the mean root mean square error (RMSE) of the prediction results for different bearings from the test set was compared. The pre-training learning rates L1, L2, L3, and L4 were set to 0.015, 0.01, 0.0015, and 0.00015, respectively, while

the formal training learning rates L1, L2, L3, and L4 were set to 0.02, 0.01, 0.015, and 0.008, respectively. The experimental results, shown in Figure 17, reveal that as both the pre-training and formal training learning rates decrease, the RMSE also decreases, with the optimal combination of learning rates being 0.0015 for pre-training and 0.008 for formal training.
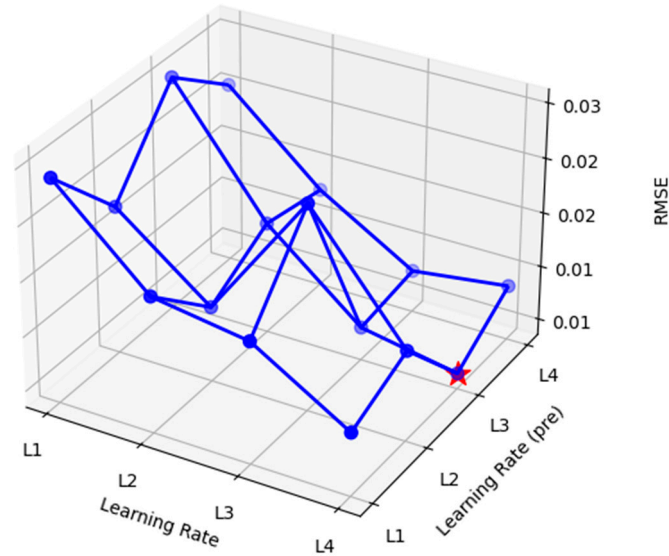

Figure 17. Trials for learning rate groups.

Additionally, the number of connections each node has with other nodes in the adjacency matrix at different degradation stages significantly influences the model's performance. In this study, different combinations of node connections were tested, with each node connected to four, five, six, and seven nodes, respectively, during the slight degradation stage, and to seven, eight, nine, and ten nodes, respectively, during the severe degradation stage. By comparing the mean RMSE of the prediction results for different bearings on the test set under these different combinations, the optimal adjacency matrix construction strategy was identified. This strategy was then used to optimize the model's performance. The comparison results of different combinations are shown in Figure 18.
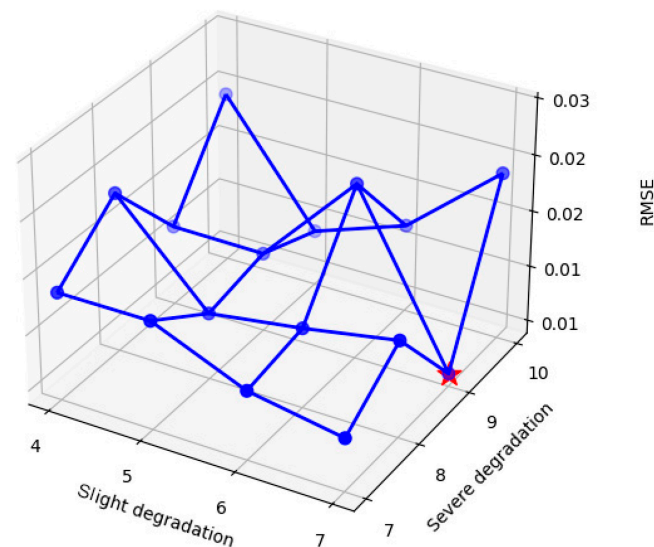

Figure 18. Trials for different combinations.

4. Discussion

In this study, we proposed an innovative hybrid model combining GNNs and LSTM networks for predicting the RUL of bearings. The results demonstrate that the proposed model outperforms traditional machine learning and deep learning approaches, particularly in capturing the complex multi-stage degradation patterns of bearings under varying operational conditions. The GNN component excels at modeling spatial dependencies, while the LSTM network captures temporal dynamics, both of which are crucial for accurate RUL prediction. This hybrid approach offers a comprehensive solution by addressing both spatial and temporal aspects simultaneously, overcoming the limitations of existing methods.

The implications of this research are significant, especially in the field of predictive maintenance for industrial systems. By accurately predicting the RUL of bearings, this model can help optimize maintenance schedules, reduce downtime, and extend the operational life of machinery. Moreover, the proposed method can be extended to other machinery components or systems exhibiting similar degradation patterns.

However, several limitations remain. First, the model's performance could be further enhanced by incorporating additional sensor data or integrating advanced techniques such as reinforcement learning or transfer learning to address data scarcity and improve generalization. Additionally, while the hybrid model has shown promising results in controlled environments, further validation in real-world industrial settings is necessary to assess its robustness under diverse operating conditions.

Future research will focus on leveraging transfer learning to improve the model's generalization across different operational environments. A key part of this future work will be the development of a new full-lifecycle dataset, which will be used to evaluate the model's performance in more complex and varied conditions. This new dataset will also allow for further validation of the model's practical applicability and help to refine its predictive capabilities in real-world scenarios.

Author Contributions: Conceptualization, G.H. and W.L.; methodology, G.H.; software, X.D. (Xing Dong); validation, G.H., D.Z. and S.C.; formal analysis, G.H.; investigation, X.D. (Xinmin Dong); resources, W.L.; data curation, X.D. (Xinmin Dong); writing—original draft preparation, G.H.; writing—review and editing, W.L.; visualization, S.C.; supervision, X.D. (Xinmin Dong); project administration, D.Z.; funding acquisition, W.L. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Dongliang Zou and Shijin Chen were employed by the company MCC5 Group Shanghai Corporation Limited. 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. Du, X.; Jia, W.; Yu, P.; Shi, Y.; Gong, B. RUL prediction based on GAM–CNN for rotating machinery. *J. Braz. Soc. Mech. Sci. Eng.* **2023**, *45*, 142. [CrossRef]
2. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [CrossRef]
3. Pan, C.; Shang, Z.; Liu, F.; Li, W.; Gao, M. Optimization of rolling bearing dynamic model based on improved golden jackal optimization algorithm and sensitive feature fusion. *Mech. Syst. Signal Process.* **2023**, *204*, 110845. [CrossRef]
4. Zhang, Z.X.; Si, X.S.; Hu, C.H. An age-and state-dependent nonlinear prognostic model for degrading systems. *IEEE Trans. Reliab.* **2015**, *64*, 1214–1228. [CrossRef]

5.   Gašperin, M.; Juričić, Đ.; Boškoski, P.; Vižintin, J. Model-based prognostics of gear health using stochastic dynamical models. *Mech. Syst. Signal Process.* **2011**, *25*, 537–548. [CrossRef]

6.   Soualhi, A.; Medjaher, K.; Zerhouni, N. Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression. *IEEE Trans. Instrum. Meas.* **2014**, *64*, 52–62. [CrossRef]

7.   Tayade, A.; Patil, S.; Phalle, V.; Kazi, F.; Powar, S. Remaining useful life (RUL) prediction of bearing by using regression model and principal component analysis (PCA) technique. *Vibroeng. Procedia* **2019**, *23*, 30–36. [CrossRef]

8.   Li, Z.; Jiang, W.; Zhang, S.; Xue, D.; Zhang, S. Research on prediction method of hydraulic pump remaining useful life based on KPCA and JITL. *Appl. Sci.* **2021**, *11*, 9389. [CrossRef]

9.   Zhao, M.; Tang, B.; Tan, Q. Bearing remaining useful life estimation based on time–frequency representation and supervised dimensionality reduction. *Measurement* **2016**, *86*, 41–55. [CrossRef]

10.   Dong, S.; Wu, W.; He, K.; Mou, X. Rolling bearing performance degradation assessment based on improved convolutional neural network with anti-interference. *Measurement* **2020**, *151*, 107219. [CrossRef]

11.   Yang, D.; Lv, Y.; Yuan, R.; Yang, K.; Zhong, H. A novel vibro-acoustic fault diagnosis method of rolling bearings via entropy-weighted nuisance attribute projection and orthogonal locality preserving projections under various operating conditions. *Appl. Acoust.* **2022**, *196*, 108889. [CrossRef]

12.   Akpudo, U.E.; Hur, J.W. A feature fusion-based prognostics approach for rolling element bearings. *J. Mech. Sci. Technol.* **2020**, *34*, 4025–4035. [CrossRef]

13.   Wang, Y.; Zhao, J.; Yang, C.; Xu, D.; Ge, J. Remaining useful life prediction of rolling bearings based on Pearson correlation-KPCA multi-feature fusion. *Measurement* **2022**, *201*, 111572. [CrossRef]

14.   Wang, Y.; Peng, Y.; Zi, Y.; Jin, X.; Tsui, K.-L. A two-stage data-driven-based prognostic approach for bearing degradation problem. *IEEE Trans. Ind. Inform.* **2016**, *12*, 924–932. [CrossRef]

15.   Ni, Q.; Ji, J.C.; Feng, K. Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1301–1311. [CrossRef]

16.   El-Thalji, I.; Jantunen, E. A descriptive model of wear evolution in rolling bearings. *Eng. Fail. Anal.* **2014**, *45*, 204–224. [CrossRef]

17.   Zhao, H.; Liu, H.; Jin, Y.; Dang, X.; Deng, W. Feature extraction for data-driven remaining useful life prediction of rolling bearings. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–10. [CrossRef]

18.   Ben Ali, J.; Chebel-Morello, B.; Saidi, L.; Malinowski, S.; Fnaiech, F. Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network. *Mech. Syst. Signal Process.* **2015**, *56*, 150–172. [CrossRef]

19.   Wang, Z.; Zhao, W.; Li, Y.; Dong, L.; Wang, J.; Du, W.; Jiang, X. Adaptive staged RUL prediction of rolling bearing. *Measurement* **2023**, *222*, 113478. [CrossRef]

20.   Liu, C.; Chen, Y.; Xu, X. Fatigue life prognosis of composite structures using a transferable deep reinforcement learning-based approach. *Compos. Struct.* **2025**, *353*, 118727. [CrossRef]

21.   Yang, X.; Zheng, Y.; Zhang, Y.; Wong, D.S.-H.; Yang, W. Bearing remaining useful life prediction based on regression shapalet and graph neural network. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]

22.   Wen, G.; Lei, Z.; Chen, X.; Huang, X. Remaining Life Assessment of Rolling Bearing Based on Graph Neural Network. In *New Generation Artificial Intelligence-Driven Diagnosis and Maintenance Techniques: Advanced Machine Learning Models, Methods and Applications*; Springer Nature: Singapore, 2024; pp. 281–298.

23.   Cui, L.; Xiao, Y.; Liu, D.; Han, H. Digital twin-driven graph domain adaptation neural network for remaining useful life prediction of rolling bearing. *Reliab. Eng. Syst. Saf.* **2024**, *245*, 109991. [CrossRef]

24.   Yang, X.; Li, X.; Zheng, Y.; Zhang, Y.; Wong, D.S.-H. Bearing remaining useful life prediction using spatial-temporal multiscale graph convolutional neural network. *Meas. Sci. Technol.* **2023**, *34*, 085009. [CrossRef]

25.   Ran, B.; Peng, Y.; Wang, Y. Bearing degradation prediction based on deep latent variable state space model with differential transformation. *Mech. Syst. Signal Process.* **2024**, *220*, 111636. [CrossRef]

26.   Cho, I.; Park, S.; Kim, J. A fire risk assessment method for high-capacity battery packs using interquartile range filter. *J. Energy Storage* **2022**, *50*, 104663. [CrossRef]

27.   Sehgal, R.; Jagadesh, P. Data-driven robust portfolio optimization with semi mean absolute deviation via support vector clustering. *Expert Syst. Appl.* **2023**, *224*, 120000. [CrossRef]

28.   Vatanshenas, A.; Länsivaara, T.T. Estimating maximum shear modulus (G0) using adaptive neuro-fuzzy inference system (ANFIS). *Soil Dyn. Earthq. Eng.* **2022**, *153*, 107105. [CrossRef]

29.   Hou, D.; Chen, J.; Cheng, R.; Hu, X.; Shi, P. A bearing remaining life prediction method under variable operating conditions based on cross-transformer fusioning segmented data cleaning. *Reliab. Eng. Syst. Saf.* **2024**, *245*, 110021. [CrossRef]

30.   Zhang, X.; Wan, S.; He, Y.; Wang, X.; Dou, L. Teager energy spectral kurtosis of wavelet packet transform and its application in locating the sound source of fault bearing of belt conveyor. *Measurement* **2021**, *173*, 108367. [CrossRef]

31.   Jiang, X.; Shen, C.; Shi, J.; Zhu, Z. Initial center frequency-guided VMD for fault diagnosis of rotating machines. *J. Sound Vib.* **2018**, *435*, 36–55. [CrossRef]

32. Chen, P.; He, A.; Zhang, T.; Dong, X. Weak vibration signal detection based on frequency domain cumulative averaging with DVS system. *Opt. Fiber Technol.* **2024**, *88*, 103834. [CrossRef]

33. Mahapatra, A.G.; Horio, K. Classification of ictal and interictal EEG using RMS frequency, dominant frequency, root mean instantaneous frequency square and their parameters ratio. *Biomed. Signal Process. Control* **2018**, *44*, 168–180. [CrossRef]

34. Hashim, S.; Shakya, P. A spectral kurtosis based blind deconvolution approach for spur gear fault diagnosis. *ISA Trans.* **2023**, *142*, 492–500. [CrossRef]

35. Han, S.; Li, D.; Li, K.; Wu, H.; Gao, Y.; Zhang, Y.; Yuan, R. Analysis and study of transmission line icing based on grey correlation Pearson combinatorial optimization support vector machine. *Measurement* **2024**, *236*, 115086. [CrossRef]

36. Jiang, J.; Zhang, X.; Yuan, Z. Feature selection for classification with Spearman's rank correlation coefficient-based self-information in divergence-based fuzzy rough sets. *Expert Syst. Appl.* **2024**, *249*, 123633. [CrossRef]

37. Zhang, Z.; Tang, X.; Liu, C.; Li, X.; Ren, S. Multiple ultrasonic partial discharge DOA estimation performance of KPCA Pseudo-Whitening mnc-FastICA. *Measurement* **2024**, *231*, 114596. [CrossRef]

38. Zhang, M.; Zhong, J.; Zhou, C.; Jia, X.; Zhu, X.; Huang, B. Deep learning-driven pavement crack analysis: Autoencoder-enhanced crack feature extraction and structure classification. *Eng. Appl. Artif. Intell.* **2024**, *132*, 107949. [CrossRef]

39. Chaleshtori, A.E.; Aghaie, A. A novel bearing fault diagnosis approach using the Gaussian mixture model and the weighted principal component analysis. *Reliab. Eng. Syst. Saf.* **2024**, *242*, 109720. [CrossRef]

40. 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*, 1–12. [CrossRef]

41. Kamat, P.; Kumar, S.; Sugandhi, R. Vibration-based anomaly pattern mining for remaining useful life (RUL) prediction in bearings. *J. Braz. Soc. Mech. Sci. Eng.* **2024**, *46*, 290. [CrossRef]

42. Tao, L.; Wu, H.; Zheng, X. Remaining Useful Life Prediction of Lithium-ion Batteries Based on Multi-graph-network model. In Proceedings of the 2024 43rd Chinese Control Conference (CCC), Kunming, China, 28–31 July 2024; pp. 8477–8482.

43. Ding, H.; Yang, L.; Cheng, Z.; Yang, Z. A remaining useful life prediction method for bearing based on deep neural networks. *Measurement* **2021**, *172*, 108878. [CrossRef]

44. Yang, C.; Ma, J.; Wang, X.; Li, X.; Li, Z.; Luo, T. A novel based-performance degradation indicator RUL prediction model and its application in rolling bearing. *ISA Trans.* **2022**, *121*, 349–364. [CrossRef] [PubMed]

45. Ding, G.; Wang, W.; Zhao, J. Prediction of remaining useful life of rolling bearing based on fractal dimension and convolutional neural network. *Meas. Control* **2022**, *55*, 79–93. [CrossRef]