

Frequency-domain-based Bayesian-optimized 1-D CNN feature learning for integrated bearing fault diagnosis and remaining useful life prediction

Md Hedayetul Islam Chy

Old Dominion University

Charles Lowe

Old Dominion University

Sujatha Alla

Old Dominion University

Hicham Chaoui

Old Dominion University

Abdullah Al Mamun

a1mamun@odu.edu

Old Dominion University <https://orcid.org/0000-0002-9199-8974>

Research Article

Keywords: Bayesian optimization, feature extraction, frequency-domain, Gaussian Process Regression, remaining useful life

Posted Date: September 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11056679/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Frequency-domain-based Bayesian-optimized 1-D CNN feature learning for integrated bearing fault diagnosis and remaining useful life prediction

Md Hedayetul Islam Chy¹, Charles Lowe¹, Sujatha Alla¹, Hicham Chaoui¹, Abdullah Al Mamun^{1*}

¹Batten College of Engineering and Technology, Old Dominion University
Norfolk, VA – 23529, U.S.A.

*Corresponding author(s). E-mail(s): al1mamun@odu.edu;

Contributing authors: mchy002@odu.edu; clowe@odu.edu; salla@odu.edu; hchaoui@odu.edu

Abstract

Bearing faults involve complex contact dynamics that influence vibration signatures generated by localized defects. However, limited run-to-failure datasets and reliance on empirically selected neural-network hyperparameters challenge the development of robust data-driven models for bearing health assessment. This study develops a unified frequency-domain, Bayesian-optimized one-dimensional convolutional neural network (1-D CNN) feature-learning framework for bearing fault diagnosis, degradation assessment, and remaining useful life (RUL) prediction. The framework is evaluated using three NASA Intelligent Maintenance Systems (IMS) run-to-failure datasets encompassing inner-race, rolling-element, and outer-race failures. Raw vibration signals are transformed into frequency-domain representations using the Fast Fourier Transform (FFT). An objective-function-driven Bayesian optimization strategy integrated with parallel computation systematically explores a broad multidimensional hyperparameter space encompassing initial learning rates, L-2 regularization coefficients, and adaptive numbers of filters across convolutional layers. Guided by model performance, the optimization identifies dataset-specific CNN configurations, while parallel computation accelerates candidate evaluations and reduces computational burden. The optimized deep features are employed for Gaussian-kernel Support Vector Machine (SVM)-based fault diagnosis and fused at the bearing level to construct degradation-sensitive health indicators for Bayesian-optimized Gaussian Process Regression (GPR)-based RUL prediction. The proposed method achieves diagnostic accuracies of 99.85%, 100.00%, and 99.98% for IMS Datasets #1, #2, and #3, respectively, compared with 99.69%, 96.69%, and 99.86% using the CNN-LSTM benchmark. The proposed framework also generates more progressive degradation-sensitive health trajectories than the benchmark. These results demonstrate a unified data-driven framework for robust fault diagnosis, degradation assessment, and RUL prediction, supporting intelligent predictive maintenance of rotating machinery.

Keywords: Bayesian optimization, feature extraction, frequency-domain, Gaussian Process Regression, remaining useful life

1 Introduction

1.1 Background

Rotating machinery commonly integrates various mechanical components, including bearings, shafts, gearboxes, pulleys, and motor-driven systems. Among these components, rolling-element bearings play a critical role in supporting rotating shafts, transmitting mechanical loads, and minimizing friction between moving surfaces [1–3]. Bearing faults are governed by complex elastohydrodynamic contact dynamics, where elastic deformation, lubricant-film behavior, contact pressure, and rolling-element-raceway interactions influence vibration signatures generated by localized defects [4]. Characterizing these dynamic responses throughout the bearing life cycle is essential for reliable

health assessment and prognostics. Consequently, any defects or degradation in rolling-element bearings can adversely affect the structural integrity, operational performance, and overall health of rotating machinery. The primary components of a rolling-element ball bearing include the rolling elements (balls), inner-ring raceway, and outer-ring raceway [5] as illustrated in Figure 1. The bearing balls roll along the inner and outer raceways, minimizing friction while efficiently transmitting applied loads. The inner-ring raceway serves as the contact interface between the inner ring and the rolling elements, where localized

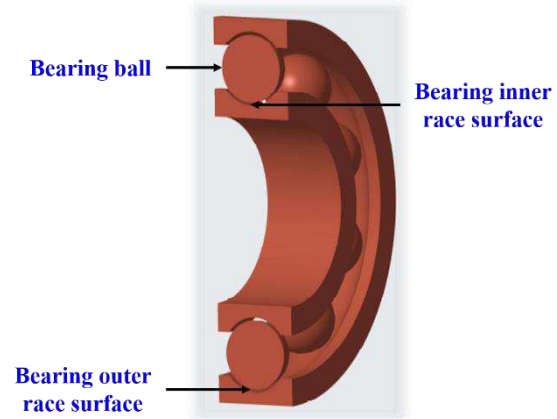


Figure 1. Rolling elements of a ball bearing structure.

surface defects produce periodic dynamic responses corresponding to characteristic inner-race fault frequencies. Similarly, localized defects on the outer-ring raceway produce repetitive impacts as the rolling elements pass over the damaged region, resulting in characteristic outer-race fault frequencies [6, 7], illustrated in Figure 2. The health condition of rolling-element bearings is commonly assessed using acoustic and vibration signatures, which can reveal abnormal operating conditions through unusual machine noise and changes in structural vibration characteristics.

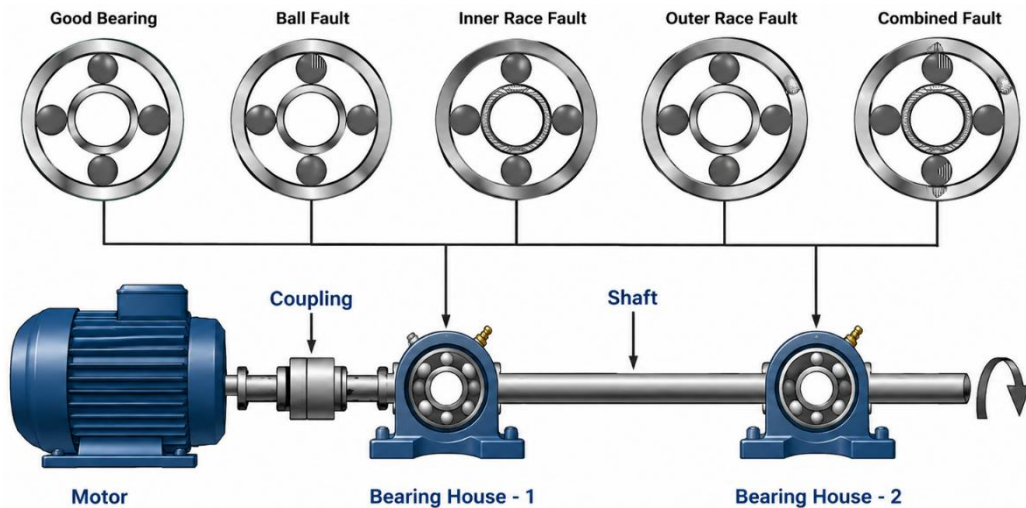


Figure 2. Different bearing health conditions of a rotating machinery.

However, subtle or incipient bearing faults may produce weak signatures that are difficult to detect through conventional observation alone, necessitating the integration of appropriate sensing technologies. In industrial condition-monitoring applications, acoustic sensors, such as microphones, are used to capture sound signatures, while accelerometers are widely employed to measure vibration responses generated by bearing dynamics [8, 9]. Analysis of these sensor signals can provide valuable information for detecting, diagnosing, and monitoring the progression of bearing faults. Bearing fault diagnosis based on vibration analysis is commonly performed under predefined fault conditions by leveraging various artificial intelligence (AI) techniques, including machine learning, deep learning, transfer learning, reinforcement learning, and, more recently, agentic AI approaches. However, many existing studies rely on artificially induced or predefined fault conditions, which may not fully represent the progressive degradation behavior of bearings during actual operation. In particular, the limited availability of run-to-failure bearing datasets presents a significant challenge for understanding how vibration signatures evolve from a healthy operating state through progressive degradation to eventual bearing failure. To address this limitation, this study utilizes the NASA publicly available bearing dataset, which provides run-to-failure vibration measurements. The dataset enables

systematic investigation of bearing health degradation and the evolution of vibration signatures from healthy operation to fault development and final failure, providing a realistic foundation for bearing fault diagnosis and RUL assessment. The bearing data was generated by the Center for Intelligent Maintenance Systems (IMS) and are widely known as the IMS bearing run-to-failure datasets [10], which is illustrated in Figure 3. To effectively diagnose bearing faults and predict RUL, a wide range of machine learning and deep learning approaches have been developed by integrating signal processing, feature extraction, and classification techniques [11, 12]. Among these approaches, CNNs have been extensively

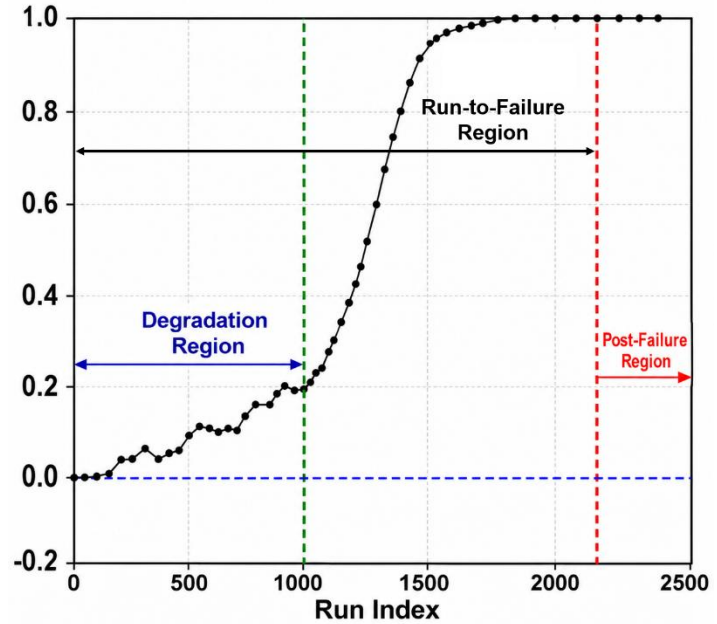


Figure 3. Bearing health evolution from degradation to run-to-failure and post-failure regions based on the normalized health indicator.

implemented because of their capability to automatically extract distinctive features from vibration signals. However, CNN execution is highly determined on the applicable selection of hyperparameters. To address this challenge, Bayesian optimization (BO)[13] with a Gaussian process (GP) surrogate model can be integrated with CNNs to systematically optimize influential hyperparameters, including the learning rate, L-2 regularization factor, and convolutional network depth [14–16]. This data-adaptive optimization framework can improve model performance while reducing the computational burden associated with exhaustive hyperparameter searches, thereby supporting accurate bearing fault diagnosis and RUL estimation. In this study, a frequency-domain-based *Bayesian-optimized 1-D CNN feature-learning* framework was employed for integrated bearing fault diagnosis and RUL prediction using

the IMS run-to-failure bearing datasets. To characterize the evolution of bearing health, vibration signals were extracted from different stages of each bearing's operational lifetime, including the first 10% and last 10% of the recorded signals, representing early-stage and late-stage operating conditions, respectively. Integrating vibration signatures across these stages enables the framework to capture the progressive changes in signal characteristics associated with bearing degradation, thereby providing valuable information for *fault diagnosis* and *RUL prediction under run-to-failure conditions*. The methodological foundation of this study integrates time-to-frequency-domain signal transformation, Bayesian-optimized 1-D CNN feature learning, Gaussian-kernel support vector machine (SVM) classification [17], and Gaussian process regression (GPR)[13, 18, 19]. First, the raw vibration signals are converted from the time domain to the frequency domain to capture frequency-specific characteristics associated with bearing degradation. A Bayesian-optimized 1-D CNN is then employed to automatically extract discriminative features, with Bayesian optimization used to determine the optimal CNN hyperparameters. The extracted features are subsequently utilized by a Gaussian-kernel SVM for bearing fault detection. In parallel, the learned features are incorporated into a GPR model for RUL prediction, providing an integrated framework for bearing fault diagnosis and prognostic assessment.

1.2 Technical contribution

Based on the above methodological foundation, the proposed framework provides the following key technical contributions for effective bearing fault diagnosis and RUL prediction under run-to-failure conditions:

- Adaptive learning-rate optimization: Bayesian optimization systematically searches a learning-rate range of to, enabling the 1-D CNN to adapt its learning behavior according to the objective function and improve model convergence.
- Improved generalization through L-2 regularization: An L-2 regularization range of is incorporated into the Bayesian optimization process to control model complexity, reduce overfitting, and improve generalization across different bearing conditions.
- Computationally efficient hyperparameter optimization: Bayesian optimization, integrated with parallel computing, efficiently identifies optimal CNN hyperparameters, including the adaptive selection of convolutional filter numbers, while reducing the computational burden associated with exhaustive parameter searches.
- High fault-diagnosis performance: The Bayesian-optimized 1-D CNN feature-learning framework achieves approximately 99.94% bearing fault classification accuracy, outperforming the benchmark CNN-LSTM-based feature extraction method, which achieves an average accuracy of approximately 98.74%.
- Integrated diagnosis and prognosis: The optimized CNN features are utilized for both GPR-based RUL prediction, providing a unified framework for assessing predicting future degradation.

2 Related work

2.1 Hyperparameters optimization methods

An effective optimization algorithm enhances the reproducibility and performance of CNN-based feature extraction [20, 21], enabling fair comparisons among different architectures and facilitating the selection of the most appropriate model under a consistent optimization strategy [22, 23]. Various hyperparameter optimization methods have been leveraged for CNN, including manual tuning (gradient/greedy search), grid search (GS), random search (RS), gradient descent (GD), and Bayesian optimization (BO) [24, 25]. A brief overview of these methods is presented below. *(i)* Manual Tuning (Gradient/Greedy Search): Manual hyperparameter tuning represents one of the earliest approaches for optimizing CNN models and primarily relies on trial-and-error experimentation. Although simple to implement, this approach becomes increasingly inefficient as model complexity, nonlinear interactions among hyperparameters, and the size of the hyperparameter space increase [26–28]. These limitations have motivated the development of automated hyperparameter optimization methods. *(ii)* Grid Search (GS): Grid search is a widely used exhaustive optimization strategy that evaluates every possible combination of predefined hyperparameter values [29–31]. While this approach guarantees comprehensive search within the specified parameter space, its computational cost increases exponentially with the number of hyperparameters, making it impractical for high-dimensional optimization problems. *(iii)* Random Search (RS): Random search was introduced to address the computational limitations of grid search by randomly sampling hyperparameter combinations within predefined search ranges. The selected candidates are evaluated until a specified optimization objective is achieved [32–34]. Although more computationally efficient than grid search, random search does not exploit information from previously evaluated configurations, which may result in redundant evaluations and slower convergence toward the optimal solution. *(iv)* Stochastic Gradient Descent (SGD): Stochastic Gradient Descent is a classical optimization algorithm that iteratively updates parameters in the direction of the negative gradient to minimize an objective function. When gradients of the hyperparameters can be computed, gradient descent provides efficient convergence toward a local optimum [35–37]. However, for non-convex or non-differentiable optimization problems commonly encountered in CNN hyperparameter tuning, gradient descent may converge to a local minimum rather than the global optimum. *(v)* Bayesian Optimization (BO): Bayesian optimization is an efficient automated hyperparameter optimization technique that identifies the optimal hyperparameter configuration by minimizing an objective function [14, 38, 39]. Unlike grid search and random search, Bayesian optimization exploits information obtained from previously evaluated hyperparameter combinations and constructs a probabilistic surrogate model, typically based on a Gaussian Process (GP) [40, 41], to guide subsequent evaluations. This sequential learning strategy substantially reduces the number of expensive objective function evaluations while improving optimization efficiency, making Bayesian optimization particularly suitable for computationally intensive deep learning models.

2.2 Optimizing hyperparameters

2.2.1 Learning rate

Several hyperparameters play a fundamental role in determining the learning capability, convergence behavior, and predictive performance of CNN [24]. Among these, the learning rate, L-2 regularization, and convolutional layer depth

are considered the most influential because they directly affect the optimization process, model generalization, and feature extraction capability. Learning rate governs the step size used to update network weights during the optimization process. An excessively high learning rate can lead to unstable training by causing the optimization process to overshoot the optimal solution, while a very low learning rate can result in slow convergence and increased computational cost [42, 43]. Therefore, selecting an appropriate learning rate is essential for achieving rapid convergence while maintaining stable and accurate model training. Recent advances in bearing fault diagnosis have demonstrated the widespread application of CNN for automatic feature learning and real-time condition monitoring. Wan et al. [44] developed a 1-D CNN using vibro-acoustic signals and carefully selected network architecture parameters, although neither the optimization algorithm nor the learning rate selection strategy was discussed. Several subsequent studies also relied on fixed learning rates during CNN training. Jiang et al. [45] employed the gradient descent-based ADAM optimizer for bearing fault diagnosis with a fixed learning rate of 0.0001. To improve CNN performance, several researchers have incorporated optimization techniques and hyperparameter tuning. Li et al. [46] utilized transfer learning based on the Stochastic Gradient Descent (SGD) optimization method together with learning rate of 0.0005 for bearing fault diagnosis. Despite achieving excellent classification accuracy, determining the optimal learning rate required extensive trial-and-error experiments. Ni et al. [47] proposed a wavelet-based CNN framework that incorporated fundamental network parameters, including the ReLU activation function, batch size, dropout rate, and training epochs, while maintaining a fixed learning rate of 0.01. Zheng et al. [48] manually searched for an appropriate learning rate by evaluating multiple candidate values of 0.01 and 0.0001, resulting in a computationally intensive optimization process. Although these studies reported promising diagnostic performance, they provided limited discussion regarding the optimization algorithms and tunable hyperparameters used during CNN training.

2.2.2 L-2 regularization

L-2 regularization is a widely adopted technique for improving the generalization performance of CNNs by penalizing large model weights during training. As network complexity increases, the likelihood of overfitting also grows, reducing the model's ability to generalize to unseen data. By constraining the magnitude of the network parameters, L-2 regularization mitigates overfitting, enhances model robustness, and improves classification reliability. Alam et al. [49] employed the gradient descent-based adaptive moment estimation (ADAM) optimizer with a fixed learning rate of 0.001 and applied L-2 regularization with a regularization coefficient of 0.01 to improve the generalization performance of the CNN model. Liu et al. [50] incorporated L-2 regularization and a learning rate into their CNN architecture; however, they did not specify the hyperparameter optimization strategy used to determine these values. Li et al. [51] employed a multi-deep learning framework incorporating L-2 regularization and the gradient descent-based ADAM optimizer with a fixed learning rate of 0.001; however, no hyperparameter optimization strategy was reported. Iqbal et al. [52] proposed an intelligent fault diagnosis approach combining deep transfer learning and LSTM networks with L-2 regularization and the gradient descent-based ADAM optimizer using a fixed learning rate of 0.001. However, the study did not specify the L-2 regularization coefficient or introduce a systematic hyperparameter optimization strategy for selecting the learning rate or other tunable parameters. Su et al. [53] developed a LeNet-5 CNN model using the gradient descent-based ADAM optimizer with predefined values for the learning rate and L-2 regularization. Nevertheless, no automated hyperparameter optimization method was employed to determine these

parameters. Song et al.[54] proposed a bearing fault diagnosis method based on elastic net transfer learning and LSTM networks, incorporating L-2 regularization and a predefined learning rate. However, the study neither reported the specific values of these hyperparameters nor employed a systematic hyperparameter optimization strategy. Although CNN-based methods have demonstrated remarkable performance in bearing fault diagnosis, most studies rely on manually predefined or fixed values for critical hyperparameters, particularly the learning rate and L-2 regularization coefficient. Such empirical tuning strategies require considerable trial-and-error experimentation, increase computational cost, and may result in suboptimal model performance. Therefore, an ***automated hyperparameter optimization framework*** is essential for systematically identifying optimal training configurations, improving model generalization, and reducing computational complexity.

2.3 Summary of research gaps

Table 1 Summary of existing CNN hyperparameter optimization studies and identified research gaps.

CNN hyperparameter studies	Advantages	Research gaps
Learning rate optimization	<ul style="list-style-type: none"> Learning rate significantly influences CNN convergence, training stability, and classification accuracy. Adaptive optimization methods (e.g., SGD and ADAM) improve training efficiency. 	<ul style="list-style-type: none"> Most studies employ fixed or manually predefined learning rates [45–47]. Learning rate selection is typically based on trial-and-error experimentation [48]. Optimization strategies are often not explicitly reported [44].
L-2 regularization optimization	<ul style="list-style-type: none"> Reduces overfitting and improves model generalization. Widely incorporated into CNN-based diagnosis. 	<ul style="list-style-type: none"> Regularization coefficients are generally fixed or manually selected. Many studies do not report on the selected coefficient [50, 52, 53]. Automated optimization is rarely considered.
Integrated hyperparameter optimization	<ul style="list-style-type: none"> Automated optimization can improve convergence, generalization, and diagnostic accuracy. 	<ul style="list-style-type: none"> Existing studies typically optimize only one hyperparameter. Learning rate, L-2 regularization, filter numbers, feature dimension, mini-batch size, and network depth are seldom optimized simultaneously. A unified Bayesian optimization framework remains largely unexplored.

3 Overview of the proposed method

The proposed framework presents a unified three-stage diagnostics and prognostics approach for intelligent bearing condition monitoring, which is illustrated in Figure 4. In stage # 1, multi-channel vibration signals are transformed into frequency-domain representations using the FFT. The first and last 10% of the bearing life are selected as early- and late-stage training samples (total 20 %) to train a Bayesian-optimized 1-D CNN and remaining 80 % used for feature extraction. Bayesian optimization automatically determines the optimal CNN hyperparameters, including the learning rate, mini-batch size, regularization coefficient, feature dimension, and convolutional filters, enabling robust frequency-domain feature learning. The optimized CNN is subsequently employed to extract discriminative deep

feature vectors from all remaining bearing runs. In stage # 2, the extracted deep feature vectors are utilized for bearing fault diagnosis using a Bayesian-optimized Gaussian kernel multiclass support vector machine (SVM). Bayesian optimization determines the optimal SVM hyperparameters, including the box constraint and kernel scale, while an Error Correcting Output Code (ECOC) framework performs multiclass classification. Diagnostic performance is evaluated using accuracy, precision, recall, and F1-score over multiple shuffled runs to ensure robustness and generalization. In stage # 3, the learned deep features from channels

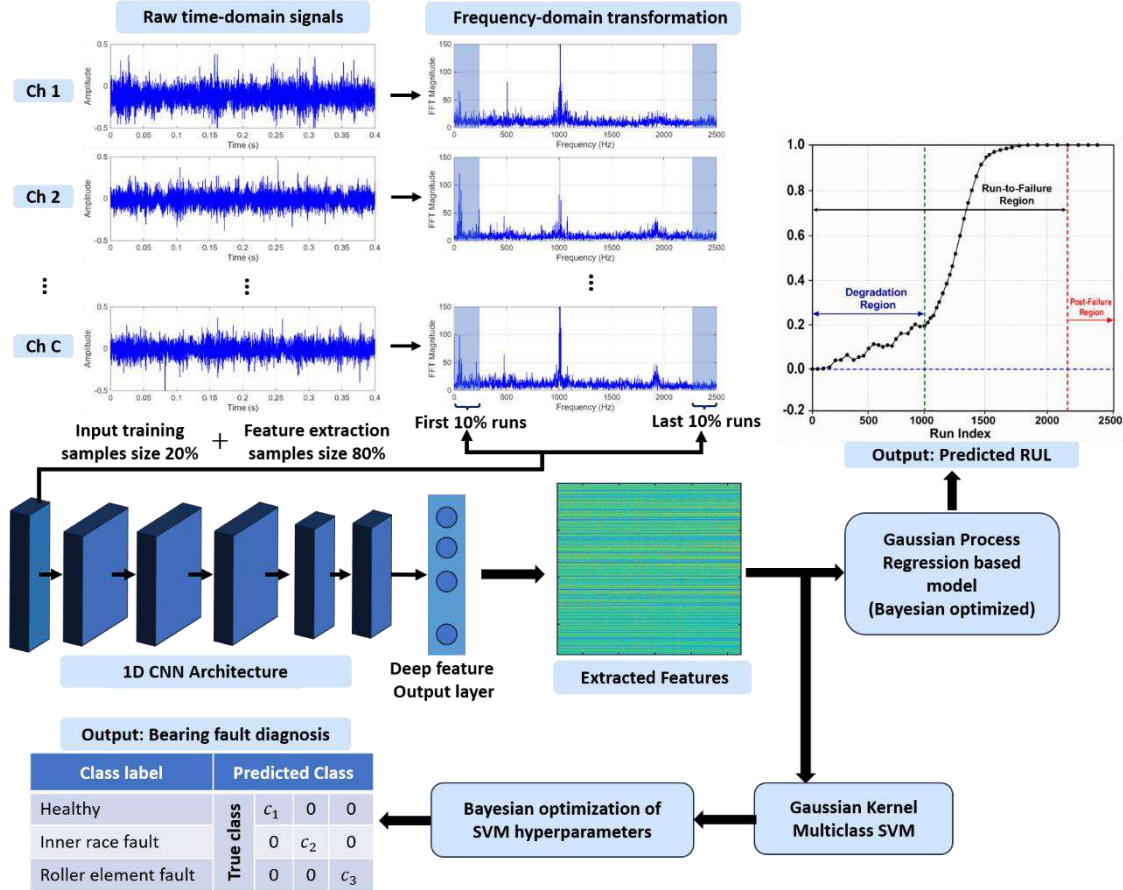


Figure 4. proposed framework presents a unified three-stage diagnostics and prognostics approach for intelligent bearing condition monitoring.

belonging to the same bearing are fused to construct a bearing-wise health representation. A Relative Squared Error (RSE)-based health indicator is generated, smoothed, normalized, and used to determine an adaptive failure threshold. A Bayesian-optimized GPR model is then trained using the healthy degradation trajectory to forecast the future health indicator together with its uncertainty. Finally, the predicted failure point is identified when the health indicator reaches the adaptive threshold, enabling estimation of the bearing's RUL. This integrated framework provides a comprehensive. solution for both accurate bearing fault diagnosis and reliable prognostics within a single frequency-domain deep learning architecture.

3.1 Frequency-domain-based 1-D CNN feature extraction

3.1.1 Time-to-frequency domain formulation

Let the raw vibration signal from run i and channel c can be $x_{i,c}(n)$, $n = 0, 1, \dots, L - 1$, where L is the number of time-domain samples. The discrete Fourier transform is presented in equation (1). The two-sided amplitude spectrum is shown in equation (2), where the single-sided FFT spectrum is shown in equation (3).

$$X_{i,c}(k) = \sum_{n=0}^{L-1} x_{i,c}(n) e^{-j2\pi kn/L}, k = 0, 1, \dots, L - 1 \quad (1)$$

$$A_{i,c}^{(2)}(k) = \frac{|X_{i,c}(k)|}{L} \quad (2)$$

$$A_{i,c}(k) = \begin{cases} A_{i,c}^{(2)}(k), & k = 0 \text{ or } k = L/2 \\ 2A_{i,c}^{(2)}(k), & 1 \leq k \leq \frac{L}{2} - 1 \end{cases} \quad (3)$$

Thus, each signal is transformed into an FFT feature vector is presented in equation (4), where $N_f = \frac{L}{2} + 1$

$$\mathbf{P}_{i,c} = [A_{i,c}(k), A_{i,c}(1), \dots, A_{i,c}(L/2)]^T \in \mathbb{R}^{N_f} \quad (4)$$

Regarding the early-late training dataset: let the full dataset contain N runs and C sensor channels. The first 10 % of runs are labelled as early-stage condition: $I_E = \{1, 2, 3, \dots, N_p\}$ and the last 10 % of runs are labelled as last-state condition: $I_L = \{N - N_p + 1, \dots, N\}$, where $N_p = \text{round}(0.10N)$. For each selected run and each channel, the FFT vector is used as one CNN input sample: $\mathbf{X}_{i,c} = \mathbf{P}_{i,c}$ with label as shown in equation (5). Therefore, the total number of training samples is shown in equation (6).

$$y_{i,c} = \begin{cases} \text{Early}, & i \in I_E \\ \text{Late}, & i \in I_L \end{cases} \quad (5)$$

$$N_{train} = C(|I_E| + |I_L|) \quad (6)$$

The CNN receives each FFT vector as $\mathbf{X}_{i,c} \in \mathbb{R}^{N_f \times 1 \times 1}$. The convolution operation at layer ℓ is presented in equation (7), where $*$ denotes convolution, $\mathbf{W}^{(\ell)}$ is the convolution kernel, and $\mathbf{b}^{(\ell)}$ is the bias. Batch normalization is applied as in equation (8) and followed by ReLU activation shown in equation (9). Max pooling reduces the feature dimension and shows in equation (10).

$$\mathbf{Z}^{(\ell)} = \mathbf{W}^{(\ell)} * \mathbf{H}^{(\ell-1)} + \mathbf{b}^{(\ell)} \quad (7)$$

$$\hat{\mathbf{Z}}^{(\ell)} = \frac{\mathbf{Z}^{(\ell)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (8)$$

$$h^{(\ell)} = \max(0, \hat{\mathbf{Z}}^{(\ell)}) \quad (9)$$

$$h_p^{(\ell)} = \max_{\tau \in \Omega} h^{(\ell)}(\tau) \quad (10)$$

After convolution, pooling, and global average pooling, CNN maps the FFT input to a learned feature vector: $\mathbf{f}_{i,c} = \phi(\mathbf{P}_{i,c}; \boldsymbol{\Theta}) \in \mathbb{R}^d$, where d is the optimized feature dimension from Bayesian optimization. The final classification probability is computed using Softmax in the equation (11).

$$\hat{p}(y = q | \mathbf{P}_{i,c}) = \frac{\exp(s_q)}{\sum_{r=1}^2 \exp(s_r)}, \text{ where } q \in \{Early, Late\} \quad (11)$$

3.1.2 Bayesian optimized -1D CNN training

CNN is trained by minimizing cross-entropy loss with L_2 regularization and shows in equation (12), where M is the number of training samples, λ is the L_2 regularization coefficient, and $\boldsymbol{\Theta}$ represents CNN trainable parameters.

$$L(\boldsymbol{\Theta}) = -\frac{1}{M} \sum_{m=1}^M \sum_{q=1}^2 y_{m,q} \log(\hat{p}_{m,q}) + \lambda \|\boldsymbol{\Theta}\|_2^2 \quad (12)$$

The Bayesian optimization formulation and hyperparameter vector is $\alpha = [\eta, B, \lambda, d, F_1, F_2, F_3]$, where η is the learning rate, B is the mini-batch size, λ is L_2 regularization, d is the feature dimension, and F_1, F_2, F_3 are the numbers of filters in the first three convolution layers. For each candidate α , the CNN is trained on a holdout training set and evaluated on validation data. Validation accuracy is shown in equation (13).

$$Acc_{val}(\alpha) = \frac{1}{M_{val}} \sum_{m=1}^{M_{val}} I(\hat{y}_m = y_m) \quad (13)$$

The Bayesian optimization objective is $J(\alpha) = 1 - Acc_{val}(\alpha)$ and the optimal hyperparameters are obtained as $\alpha^* = \arg \min_{\alpha \in A} J(\alpha)$, where A is the bounded hyperparameter search space.

3.1.3 Feature extraction

After Bayesian optimization, the final CNN is trained using the best hyperparameters, which is shown in equation (14).

$$\boldsymbol{\Theta}^* = \arg \min_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}; \alpha^*) \quad (14)$$

For each remaining run $i \in I_F$, where $I_F = \{1, \dots, N\} \setminus \{I_E \cup I_F\}$ and the CNN feature vector is extracted from the layer $\mathbf{f}_{i,c} = \phi(\mathbf{P}_{i,c}; \boldsymbol{\Theta}^*) \in \mathbb{R}^d$ and for all channels, the final combined feature vector is shown in equation (15).

$$\mathbf{F}_i = [\mathbf{f}_{i,1}, \mathbf{f}_{i,2}, \dots, \mathbf{f}_{i,C}] \in \mathbb{R}^{C \times d} \quad (15)$$

- **Algorithm 1: frequency-domain-based training data preparation**

Table 2. FFT-based training data preparation.

Input: Raw run data R , early early index set I_E , late index set I_L , number of FFT points N_f
Output: CNN input tensor $\mathbf{X}_{train}$, labels $\mathbf{Y}_{train}$
1. Initialize empty training tensors and label vector.
2. For each run $i \in I_E$: Compute FFT spectrum $\mathbf{P}_{i,c}$ for each channel c . Assign label “Early”
3. For each run $i \in I_L$: Compute FFT spectrum $\mathbf{P}_{i,c}$ for each channel c . Assign label “Late”
4. Return $\mathbf{X}_{train}$ and $\mathbf{Y}_{train}$

- **Algorithm 2: Bayesian-optimized -1D-CNN training**

Table 3. Bayesian-optimized 1-D CNN training

Input: $\mathbf{X}_{train}$ and $\mathbf{Y}_{train}$, hyperparameter search space $\mathcal{A}$
Output: Optimal hyperparameters α^* , trained CNN model
1. Partition the training dataset into separate training and validation subsets using a holdout validation approach.
2. For each Bayesian optimization iteration: Select candidate hyperparameters α Construct CNN architecture using α Train CNN using ADAM optimizer Predict validation labels Compute validation accuracy Return objective value $J = 1 - Acc_{val}$
3. Select the best hyperparameters α^*
4. Train the final CNN using α^*
5. Return the optimized CNN model.

- **Algorithm 3: Feature extraction**

Table 4. Bayesian-optimized 1-D CNN for feature extraction.

Input: Optimized CNN, feature extraction runs I_F , FFT vectors
Output: Channel-wise CNN features and combined CNN feature matrix
<ol style="list-style-type: none"> 1. For each channel $c = 1, \dots, C$ <ul style="list-style-type: none"> For each feature extraction run $i \in I_F$ Compute FFT vector $\mathbf{P}_{i,c}$ Feed $\mathbf{P}_{i,c}$ into the optimized CNN Extract feature vector from the CNN last layer 2. Store channel-wise features $\mathbf{f}_{i,c}$ 3. Concatenate all channel features: $\mathbf{F}_i = [\mathbf{f}_{i,1}, \mathbf{f}_{i,2}, \dots, \mathbf{f}_{i,C}]$ 4. Return to the final feature matrix.

3.2 Bearing fault diagnostics

3.2.1 Multiclass Gaussian SVM classification

The extracted Bayesian optimized FFT-1D CNN feature vectors are used as inputs to a Gaussian kernel multiclass SVM classifier. Let the CNN feature vector extracted from run i and channel c be $\mathbf{f}_{i,c} = [\mathbf{f}_{i,c,1}, \mathbf{f}_{i,c,2}, \dots, \mathbf{f}_{i,c,d}]^T \in \mathbb{R}^d$, where d denotes the optimized CNN feature dimension. The complete classification dataset is represented as $D_{SVM} = \{(\mathbf{f}_{i,c}, z_{i,c})\}$, where $z_{i,c}$ denotes the fault class label. For example, the IMS 1st Test dataset, the class assignments are presented in equation (16).

$$z_{i,c} = \begin{cases} 1, c \in \{1, 2, 3, 4\} \\ 2, c \in \{5, 6\} \\ 3, c \in \{7, 8\} \end{cases}, \quad (16)$$

corresponding to {Healthy, Inner Race Fault, Roller Element Fault}, respectively.

The Gaussian kernel function is defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$, where σ denotes the kernel scale. The multiclass SVM classifier is implemented using an Error Correcting Output Code (ECOC) framework with a one-versus-one coding strategy. For K classes, the total number of binary SVM classifiers is $N_{SVM} = \frac{K(K-1)}{2}$, for the IMS 1st Test dataset $K = 3$ and therefore $N_{SVM} = 3$ binary Gaussian SVM classifiers are trained. The final class prediction is obtained through majority voting: $\hat{z} = \arg \max_k V_k$, where V_k denotes the vote count for class k . Bayesian Optimization of Gaussian SVM, the SVM hyperparameter vector is $\beta = [C_s, \sigma]$, where C_s is the SVM box constraint, and σ is the Gaussian kernel scale. For each candidate hyperparameter vector, β , a five-fold cross-validation procedure is performed. The cross-validation accuracy is defined in the equation (17). The Bayesian optimization objective is shown in equation (18). The optimal hyperparameters are obtained as in equation (19), where B denotes the bounded SVM hyperparameter search space. The optimized classifier is therefore shown in equation (20).

$$Acc_{CV}(\beta) = \frac{1}{N_{CV}} \sum_{m=1}^{N_{CV}} I(\hat{z}_m - z_m) \quad (17)$$

$$J_{SVM}(\beta) = 1 - Acc_{CV}(\beta) \quad (18)$$

$$\beta^* = \arg \min_{\beta \in B} J_{SVM}(\beta) \quad (19)$$

$$g^*(\mathbf{f}) = g(\mathbf{f}; \beta^*) \quad (20)$$

3.2.2 Classification performance evaluation

The classification performance of the proposed and benchmark methods is evaluated using accuracy, precision, and recall, as defined in equations (21) to (23), respectively. Accuracy measures the overall fraction of samples that are correctly classified, whereas precision indicates the proportion of correct positive classifications among all samples predicted as positive. Recall evaluates the ability of the classifier to correctly identify positive samples from all actual positive instances. For multiple shuffled runs, the mean classification accuracy is calculated using equation (24) to provide a robust estimate of the overall classification performance. For R independent shuffled runs, the mean classification accuracy is calculated as equation (24), where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. $Accuracy_r$ denotes the classification accuracy obtained from the r-th shuffled run, and R denotes the total number of shuffled runs.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (21)$$

$$Precision = \frac{TP}{TP + FP} \quad (22)$$

$$Recall = \frac{TP}{TP + FN} \quad (23)$$

$$Mean\ accuracy = \frac{1}{R} \sum_{r=1}^R Accuracy_r \quad (24)$$

3.3 Bearing remaining useful life prediction

3.3.1 Bearing-wise feature fusion for GPR-RUL

After the optimized CNN extracts channel-wise features, $\mathbf{f}_{i,c} = \phi(\mathbf{P}_{i,c}; \Theta^*) \in \mathbb{R}^d$, the features belonging to the same bearing are fused. For bearing b , let the channel set be $\mathcal{C}_b = \{c_1, c_2, \dots, c_m\}$ and the bearing-wise feature vector is $\mathbf{F}_{i,b} = [\mathbf{f}_{i,c_1}, \mathbf{f}_{i,c_2}, \dots, \mathbf{f}_{i,c_m}] \in \mathbb{R}^{md}$ and $\mathbf{X}_b = [F_{1,b}, F_{2,b}, \dots, F_{N_b,b}]^T \in \mathbb{R}^{N_b \times md}$. The healthy baseline length is shown in equation (25). The baseline feature matrix is calculated in equation (26). The baseline mean and standard deviation are shown in equations (27) and (28).

$$N_{base} = \max(5, \text{round}(\rho_{base} N_b)) \quad (25)$$

$$X_{base} = X_{base}(1:N_{base},:) \quad (26)$$

$$\mu_{base,j} = \frac{1}{N_{base}} \sum_{i=1}^{N_{base}} X_b(i,j) \quad (27)$$

$$\sigma_{base,j} = \sqrt{\frac{1}{N_{base} - 1} \sum_{i=1}^{N_{base}} (X_b(i,j) - \mu_{base,j})^2} \quad (28)$$

RSE-Based Health Indicator: The relative error is $e_{i,j} = \frac{X_b(i,j) - \mu_{base,j}}{\sigma_{base,j}}$ and the relative squared error is $RSE_i = \sqrt{\frac{1}{D_b} \sum_{j=1}^{D_b} e_{i,j}^2}$, where $D_b = md$. The smooth health indicator is denoted in equation (29). The normalized health indicator is shown in equation (30).

$$HI_i = \frac{1}{w} \sum_{r=i-\lfloor \frac{w}{2} \rfloor}^{i+\lfloor \frac{w}{2} \rfloor} RSE_r \quad (29)$$

$$HI_i^{norm} = \frac{HI_i - \min(HI)}{\max(HI) - \min(HI) + \epsilon} \quad (30)$$

Adaptive Failure Threshold: The baseline threshold is $T_{base} = \mu(HI_{base}^{norm}) + K\sigma(HI_{base}^{norm})$ and the late-stage threshold is $T_{late} = \mu(HI_{late}^{norm}) - 0.5\sigma(HI_{late}^{norm})$ and the percentile threshold is $T_p = P_{90}(HI^{norm})$. The final adaptive failure threshold is $T_f = \max(T_{base}, T_{late}, T_p)$, with bounded adjustment: $T_f = \min(T_f, 0.98)$ and $T_f = \min(T_f, T_{base})$.

Bayesian Optimized GPR Prediction: The time index is $t_i = i$, $i = 1, 2, \dots, N_b$. The GPR training dataset is $D_{GPR} = \{(t_i, HI_i^{norm})\}_{i=1}^{N_{GPR}}$, where $N_{GPR} = \text{round}(\rho_{GPR} N_b)$. The GPR model is $HI^{norm}(t) = g(t) + \varepsilon$, where $g(t) = GP(m(t), k(t, t'))$ and $\varepsilon \sim N(0, \sigma_n^2)$. Using the squared exponential kernel, which shown in equation (31).

$$k(t, t') = \sigma_f^2 \exp\left(-\frac{(t - t')^2}{2l^2}\right) \quad (31)$$

The GPR hyperparameter vector is $\gamma = [l, \sigma_n]$ and Bayesian optimization estimates $\gamma^* = \arg \min_{\gamma} J_{GPR}(\gamma)$.

3.3.2 Failure point and RUL estimation

The future prediction range is $t^* = 1, 2, \dots, N_b + H$ and the predicted HI is $\widehat{HI}(t^*) = GPR(t^*, \gamma^*)$. The predicted failure point is $t_f = \min\{t^*: \widehat{HI}(t^*) \geq T_f\}$. If no crossing occurs, $t_f = N_b + H$ and the current point is $t_c = N_b$. The predicted RUL in samples is $RUL_{samples} = \max(t_f - t_c, 0)$ and the predicted RUL in hours is $RUL_{hours} = \frac{RUL_{samples} \times (\Delta t)}{60}$; The predicted RUL in days is $RUL_{days} = \frac{RUL_{hours}}{24}$.

3.4 Benchmark method

To assess the effectiveness of the proposed deep feature-learning framework, a benchmark prognostic approach integrating FFT with a CNN–LSTM architecture is employed for comparative evaluation. [55]. Initially, frequency-domain features extracted using CNN-LSTM from multiple vibration channels belonging to the same bearing were fused to construct a bearing-wise feature representation. Across IMS Bearing Datasets 1–3, the first 10% and the last 10% of the data were used for CNN training and validation, while the remaining 80% were reserved exclusively for feature extraction. The same data-allocation strategy was also applied to benchmark method to ensure a consistent and fair comparison of feature-extraction performance and bearing fault diagnosis conditions. The Relative Squared Error (RSE) between each bearing feature vector and the corresponding healthy baseline was subsequently computed to quantify the progression of bearing degradation over time. The resulting RSE sequence was smoothed using a moving-average filter and normalized to generate a bearing health indicator (HI). An adaptive failure threshold was subsequently determined by combining statistical information from the healthy baseline, late-stage degradation, and the 90th percentile of the normalized HI. Finally, a Bayesian-optimized Gaussian Process Regression model employing a squared exponential kernel was trained using the healthy degradation trajectory, where Bayesian optimization automatically determined the optimal kernel scale and noise variance. The trained GPR model predicted

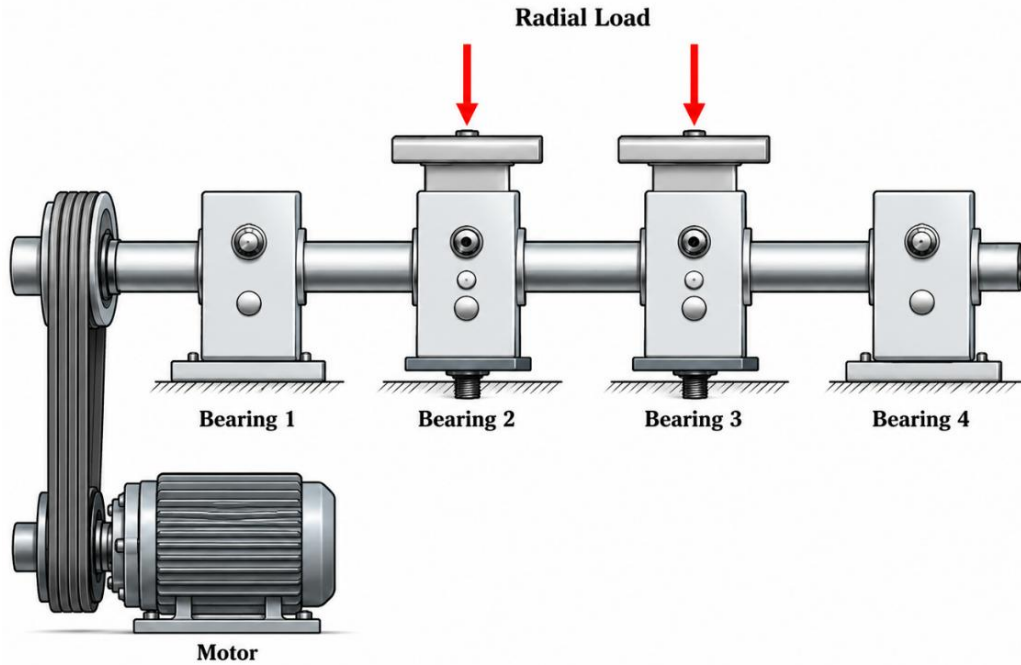


Figure 5. Bearing loading setting in Intelligent Maintenance Systems (IMS).

the future health degradation trajectory together with its confidence intervals, and the RUL was estimated as the first predicted time at which the health indicator reached the adaptive failure threshold. This benchmark provides a representative probabilistic prognostics approach based on handcrafted frequency-domain features for comparison with the proposed deep learning-based framework.

4 Testbed- dataset

The bearing data was generated by NASA: IMS and is widely known as the IMS bearing run-to-failure datasets, which is demonstrated in Figure 5. Table 5 provides IMS dataset with different run-time and channels. Set 1 contains 827.56 h of eight-channel vibration data, with two channels assigned to each bearing, and resulted in an inner-race defect in Bearing 3 and a rolling-element defect in Bearing 4. Sets 2 and 3 contain 163.83 h and 753.57 h of four-channel data, respectively, with one channel assigned to each bearing. Set 2 resulted in an outer-race failure in Bearing 1, whereas Set 3 exhibited an outer-race failure in Bearing 3. All datasets were recorded at 10-minute intervals, providing run-to-failure vibration measurements suitable for investigating bearing degradation and failure progression. Across IMS Bearing Datasets 1–3, the first 10% and the last 10% of the data were used for CNN training and validation, while the remaining 80% were reserved exclusively for feature extraction. The same data-allocation strategy was also applied to benchmark method to ensure a consistent and fair comparison of feature-extraction performance.

Table 5. IMS datasets with different run-time and channels.

Dataset	Recording duration (hours)	No. of channels	Channel arrangement	Recording interval (minutes)	Failure description
IMS Bearing dataset # set 1	827.56	8	Bearing 1: Ch 1–2 Bearing 2: Ch 3–4 Bearing 3: Ch 5–6 Bearing 4: Ch 7–8	10	Inner race defect in bearing 3 and roller element defect in bearing 4
IMS Bearing dataset # set 2	163.83	4	Bearing 1: Ch 1 Bearing 2: Ch 2 Bearing 3: Ch 3 Bearing 4: Ch 4	10	Outer race failure in bearing 1
IMS Bearing dataset # set 3	753.57	4	Bearing 1: Ch 1 Bearing 2: Ch 2 Bearing 3: Ch 3 Bearing 4: Ch 4	10	Outer race failure in bearing 3

4.1 Results and discussion

4.1.1 Hyperparameter selection

The proposed 1-D CNN architecture was developed using a Bayesian optimization framework to systematically determine the optimal network and training hyperparameters. The numbers of convolutional filters in the first, second, and third convolutional layers were optimized within the ranges of 8 – 32, 16 – 64, and 32 – 128, respectively. The network employed Rectified Linear Unit (ReLU) activation functions within the convolutional layers, while a Softmax

activation function was used in the output layer for classification. A dropout rate of 0.20 was incorporated to mitigate overfitting and improve the generalization capability of the CNN model. The network was trained using the ADAM (Adaptive Moment Estimation) optimizer. During Bayesian optimization, the initial learning rate was searched over the logarithmic range of $[10^{-5} \sim 10^{-2}]$, while the L-2 regularization coefficient was optimized between $[10^{-6} \sim 10^{-2}]$. The mini-batch size was optimized within the range of 16–128, and the CNN feature dimension was optimized between 32 and 128. An 80%/20% training–validation partition was employed during hyperparameter optimization. Each candidate CNN architecture was trained for 10 epochs, while Bayesian optimization was conducted for up to 30 objective-function evaluations using the expected-improvement-plus acquisition function. Validation accuracy was used to evaluate the candidate configurations, with the optimization objective defined as: *objective function* = $1 - Accuracy_{val}$. Following identification of the optimal hyperparameter configuration, the final CNN model was trained for 30 epochs and subsequently employed for FFT-based feature extraction. All computational experiments were executed on a workstation equipped with an Intel® Core™ i7-13700 processor and an NVIDIA GeForce RTX 4060 GPU.

4.1.2 Bayesian optimization computation cost

Bayesian optimization was independently employed for the three bearing datasets to identify dataset-specific hyperparameter configurations for the proposed 1-D CNN-based feature learning framework. For the first bearing dataset, the optimal configuration consisted of an initial learning rate of 0.0010794, a mini-batch size of 24, an L-2 regularization coefficient of 1.2515×10^{-5} , a feature dimension of 34, and 19, 48, and 97 filters in the first, second, and third convolutional layers, respectively. For the second bearing dataset, the corresponding optimized parameters were 0.0016086, 17, 9.1205×10^{-5} , 97, and 18, 27, and 100 filters, respectively. For the third bearing dataset, Bayesian optimization selected an initial learning rate of 1.1123×10^{-5} , a mini-batch size of 84, an L-2 regularization coefficient of 1.1135×10^{-6} , a feature dimension of 91, and 8, 48, and 52 filters, respectively. These results demonstrate that Bayesian optimization produced distinct CNN configurations for the three bearing datasets rather than imposing a single fixed hyperparameter configuration across datasets, which is shown in Table 6.

Table 6. Dataset-specific hyperparameters of the Bayesian-optimized 1-D CNN for feature learning.

Optimized parameter	1 st bearing dataset	2 nd bearing dataset	3 rd bearing dataset
Initial learning rate	0.0010794	0.0016086	1.1123×10^{-5}
Mini-batch size	24	17	84
L-2 regularization	1.2515×10^{-5}	9.1205×10^{-5}	1.1135×10^{-6}
Feature dimension	34	97	91
No. of filters – 1 st convolutional layer	19	18	8
No. of filters – 2 nd convolutional layer	48	27	48
No. of filters – 3 rd convolutional layer	97	100	52
Estimated objective function value	0.0027152	-3.621×10^{-7}	-0.00015599
Estimated function evaluation time (s)	16.3743	5.9717	31.0187

4.1.3 Comparative results for bearing conditions diagnostics

Tables 7 to 9 compare the bearing fault diagnostic performance of the proposed frequency-domain Bayesian-optimized 1-D CNN feature-learning method and the benchmark CNN-LSTM method without Bayesian optimization across the three IMS run-to-failure datasets. For IMS Dataset #1, the proposed method achieved an overall classification accuracy of 99.85%, compared with 99.69% for the benchmark method. For healthy bearings, the proposed method obtained average precision and recall values of 0.9992 ± 0.0007 and 0.9983 ± 0.0008 , respectively, whereas the benchmark method achieved 0.9962 ± 0.0011 precision and 0.9994 ± 0.0004 recall. For inner-race faults, the proposed method achieved 0.9963 ± 0.0014 precision and a perfect recall of 1.0000 ± 0.0000 , compared with 0.9994 ± 0.0005 precision and 0.9977 ± 0.0008 recall for the benchmark. Similarly, for roller-element faults, the proposed method achieved 0.9996 ± 0.0008 precision and 0.9975 ± 0.0019 recall, while the benchmark produced 0.9969 ± 0.0015 precision and 0.9921 ± 0.0019 recall. Although both methods demonstrated excellent diagnostic performance, the proposed method provided higher overall accuracy and particularly improved recall for the inner-race and roller-element fault classes. The consistently low standard deviations further indicate stable diagnostic performance across the repeated evaluations. For IMS Dataset #2, the performance advantage of the proposed method was considerably more pronounced. The proposed method achieved 100.00% overall classification accuracy, with both precision and recall reaching 1.0000 ± 0.0000 for the healthy and outer-race fault conditions. In comparison, the benchmark CNN-LSTM method achieved an overall accuracy of 96.69%. For healthy bearings, the benchmark achieved 0.9698 ± 0.0057 precision and 0.9955 ± 0.0016 recall, while for the outer-race fault it obtained 0.9955 ± 0.0016 precision but only 0.9068 ± 0.0182 recall. The lower outer-race fault recall indicates that the benchmark method missed a noticeably greater proportion of fault instances. Consequently, the proposed method improved the overall diagnostic accuracy by 3.31 percentage points and demonstrated substantially stronger sensitivity to outer-race degradation. These findings indicate that Bayesian-optimized 1-D CNN feature learning provides a more discriminative representation of the healthy and degraded bearing conditions in IMS Dataset #2. For IMS Dataset #3, both approaches again demonstrated very high classification performance, although the proposed method maintained a slight advantage. The proposed method achieved 99.98% overall accuracy, compared with 99.86% for the benchmark method. For healthy bearings, the proposed method achieved precision and recall values of 0.9999 ± 0.0002 and 0.9999 ± 0.0001 , respectively, whereas the benchmark achieved 0.9993 ± 0.0006 precision and 0.9976 ± 0.0015 recall. For outer-race faults, the proposed method obtained 0.9997 ± 0.0003 precision and 0.9996 ± 0.0005 recall, compared with 0.9992 ± 0.0005 precision and 0.9980 ± 0.0018 recall for the benchmark. Thus, while the difference in overall accuracy was relatively small for IMS Dataset #3, the proposed method consistently maintained slightly higher class-wise precision and recall together with lower variability. Overall, the results across IMS Datasets #1 to #3 demonstrate that the proposed Bayesian-optimized 1-D CNN feature-learning framework provides consistently high bearing fault diagnostic performance compared with the fixed-hyperparameter CNN-LSTM benchmark. The improvements in overall accuracy were 0.16 percentage points for Dataset #1, 3.31 percentage points for Dataset #2, and 0.12 percentage points for Dataset #3. The most substantial improvement was observed for IMS Dataset #2, where the proposed method achieved perfect classification and substantially increased outer-race fault recall. These results suggest that the proposed frequency-domain feature-learning framework provides improved class separability and robust fault-

sensitive representations across different run-to-failure operating conditions. Table 10 further presents a representative confusion-matrix comparison between the proposed and benchmark methods for bearing fault diagnosis using IMS Dataset #1, considering healthy, inner-race fault, and roller-element fault conditions. The proposed method achieved an accuracy of approximately 99.79%, correctly classifying 1,918 healthy, 960 inner-race fault, and 954 roller-element fault samples. Only eight samples were misclassified: two healthy samples were identified as inner-race faults, four roller-element fault samples were classified as healthy, and two roller-element fault samples were classified as inner-race faults. Notably, all inner-race fault samples were correctly identified. Based on the confusion-matrix counts provided for the benchmark method, it correctly classified 918 healthy, 958 inner-race fault, and 953 roller-element fault samples, with 11 total misclassifications, corresponding to an accuracy of approximately 99.61%. The benchmark errors included two healthy samples classified as roller-element faults, two inner-race fault samples misclassified, and seven roller-element fault samples assigned to other conditions. Overall, the confusion-matrix results further indicate that the proposed method produces slightly stronger class discrimination and fewer misclassifications than the CNN-LSTM benchmark for IMS Dataset #1. One important issue to check before publication: in Table 10, the proposed confusion matrix contains 3,840 samples (1920+960+960), whereas the benchmark confusion matrix contains only 2,840 samples (920+960+960). If both methods were evaluated using the same train/test physical-run split, those totals should normally be identical. I would verify whether the benchmark healthy row should be 1,918/1,920-range rather than 918/920-range before finalizing Table 10.

Table 7. Comparative results of bearing fault diagnosis in IMS dataset # 1.

Class label	Metric	Bearing fault diagnostics # 1st dataset			
		Proposed method		Benchmark method	
		Avg.	Std.	Avg.	Std.
Healthy bearing	Precision	0.9992	0.0007	0.9962	0.0011
	Recall	0.9983	0.0008	0.9994	0.0004
Inner race fault	Precision	0.9963	0.0014	0.9994	0.0005
	Recall	1.0000	0.0000	0.9977	0.0008
Roller element fault	Precision	0.9996	0.0008	0.9969	0.0015
	Recall	0.9975	0.0019	0.9921	0.0019
Total accuracy		99.85%		99.69%	

Table 8. Comparative results of bearing fault diagnosis in IMS dataset # 2.

Class label	Metric	Bearing fault diagnostics # 2nd dataset			
		Proposed method		Benchmark method	
		Avg.	Std.	Avg.	Std.
Healthy bearing	Precision	1.0000	0.0000	0.9698	0.0057
	Recall	1.0000	0.0000	0.9955	0.0016
Outer race fault	Precision	1.0000	0.0000	0.9955	0.0016
	Recall	1.0000	0.0000	0.9068	0.0182

Total accuracy	100.00%	96.69%
-----------------------	----------------	---------------

Table 9. Comparative results of bearing fault diagnosis in IMS dataset # 3.

Class label	Metric	Bearing fault diagnostics # 3rd dataset			
		Proposed method		Benchmark method	
		Avg.	Std.	Avg.	Std.
Healthy bearing	Precision	0.9999	0.0002	0.9993	0.0006
	Recall	0.9999	0.0001	0.9976	0.0015
Outer race fault	Precision	0.9997	0.0003	0.9992	0.0005
	Recall	0.9996	0.0005	0.9980	0.0018
Total accuracy		99.98%		99.86%	

Table 10. Bearing fault diagnosis accuracy on IMS dataset #1.

Bearing conditions	Proposed method				Benchmark method			
	True class	Predicted class			True class	Predicted class		
Healthy		1918	2	0		918	0	2
Inner race fault		0	960	0		1	958	1
Roller element fault		4	2	954		6	1	953

4.1.4 Comparative results for RUL predictions

Figures 6 to 9 present the comparative RUL prediction results of the proposed method and the benchmark method for the failed bearings across the three IMS run-to-failure datasets. The normalized health indicator (HI) represents the progressive health condition of each bearing, while GPR is used to extrapolate the degradation trajectory from the selected prediction point toward the predefined failure threshold. The corresponding 95% confidence interval (CI) illustrates the uncertainty associated with the RUL prediction. The intersection of the predicted HI trajectory with the failure threshold determines the predicted failure point and, consequently, the estimated RUL. For IMS Dataset #1, Figures 6 and 7 show the RUL prediction results for Bearing 3 with an inner-race defect and Bearing 4 with a rolling-element defect, respectively. The proposed method produces an HI trajectory that more clearly represents the progressive evolution of bearing degradation prior to failure. Following the prediction point, the observed HI exhibits an increasing degradation tendency and eventually approaches or exceeds the failure threshold. This progressive behavior provides a physically meaningful basis for extrapolating the degradation trajectory and estimating the remaining operating period. In comparison, the benchmark method exhibits less representative degradation behavior in several regions, including rapid changes and saturation of the HI near its upper limit, which can reduce the reliability of long-horizon degradation extrapolation. Figure 8, corresponding to Bearing 1 with an outer-race failure in IMS Dataset #2, further demonstrates the difference between the two approaches. The proposed method captures a gradual evolution of the HI before the onset of accelerated degradation. Although uncertainty increases as the prediction horizon extends, the GPR prediction maintains a progressive trend toward the failure threshold. In contrast, the benchmark HI reaches a relatively high normalized level considerably earlier and remains close to saturation over a

substantial portion of the bearing lifetime. Such early saturation makes it more difficult to distinguish different stages of degradation and can adversely affect the interpretability of the resulting RUL prediction. A similar observation can be made from Figure 9 for Bearing 3 with an outer-race failure in IMS Dataset #3. The proposed method preserves the gradual degradation characteristics of the bearing and provides a clearer transition from the relatively healthy condition to accelerated degradation. The observed HI increases substantially after the selected prediction point and ultimately reaches the failure region. Conversely, the benchmark method demonstrates a comparatively abrupt increase in HI, followed by values concentrated near the upper normalization limit. Consequently, benchmark representation provides less information regarding the progressive evolution of degradation as the bearing approaches failure. Overall, the comparative results in Figures 6–9 indicate that the primary advantage of the proposed method is its ability to generate a more progressive and degradation-sensitive health representation across different bearing components, fault types, and IMS datasets. Rather than prematurely saturating the HI, the proposed approach preserves changes in degradation severity throughout the run-to-failure process. This behavior is particularly important for RUL prognosis because GPR-based extrapolation depends strongly on the quality and temporal progression of the underlying HI. The results therefore demonstrate the potential of the proposed framework to provide a more interpretable basis for bearing degradation assessment and RUL prediction under different failure modes.

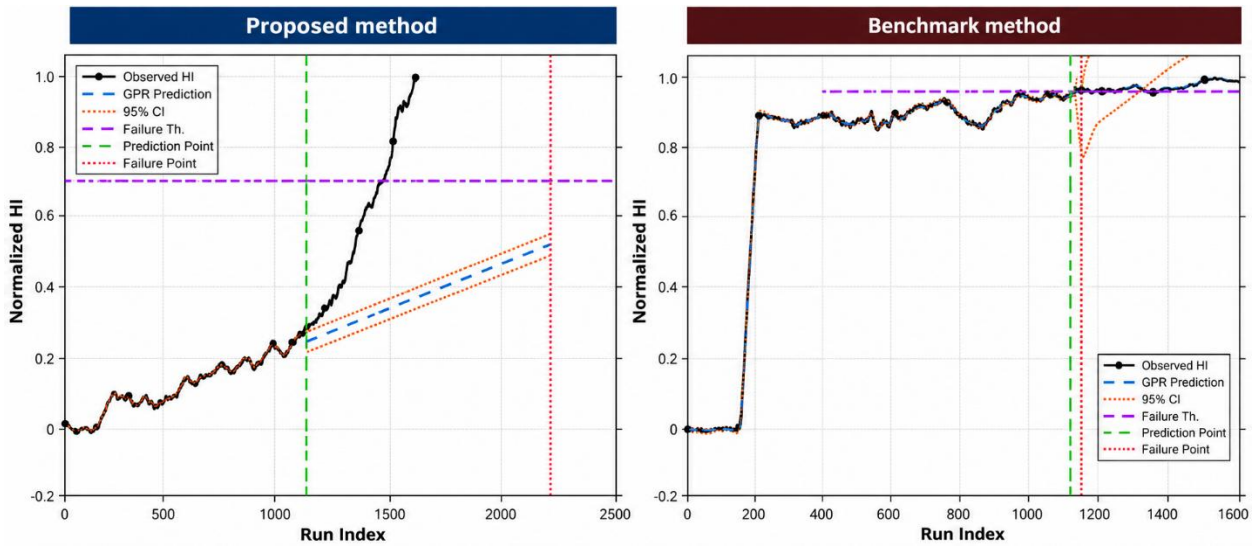


Figure 6. RUL comparison among bearing 3 (Inner race defect in bearing 3) at IMS dataset # 1.

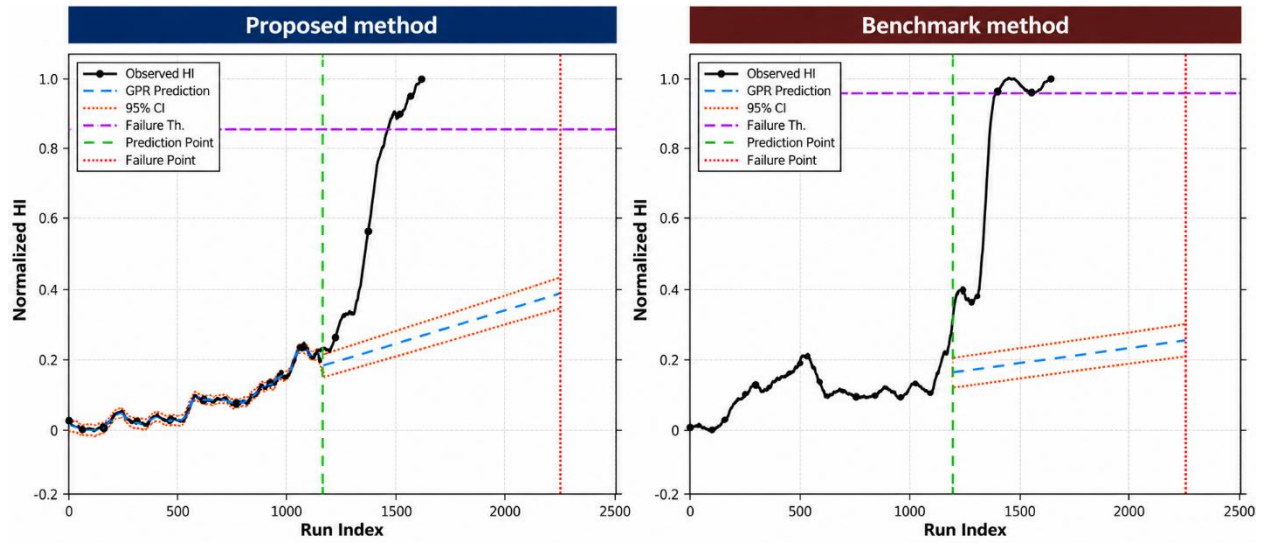


Figure 7. RUL comparison among bearing 4 (roller element defect in bearing 4) at IMS dataset # 1.

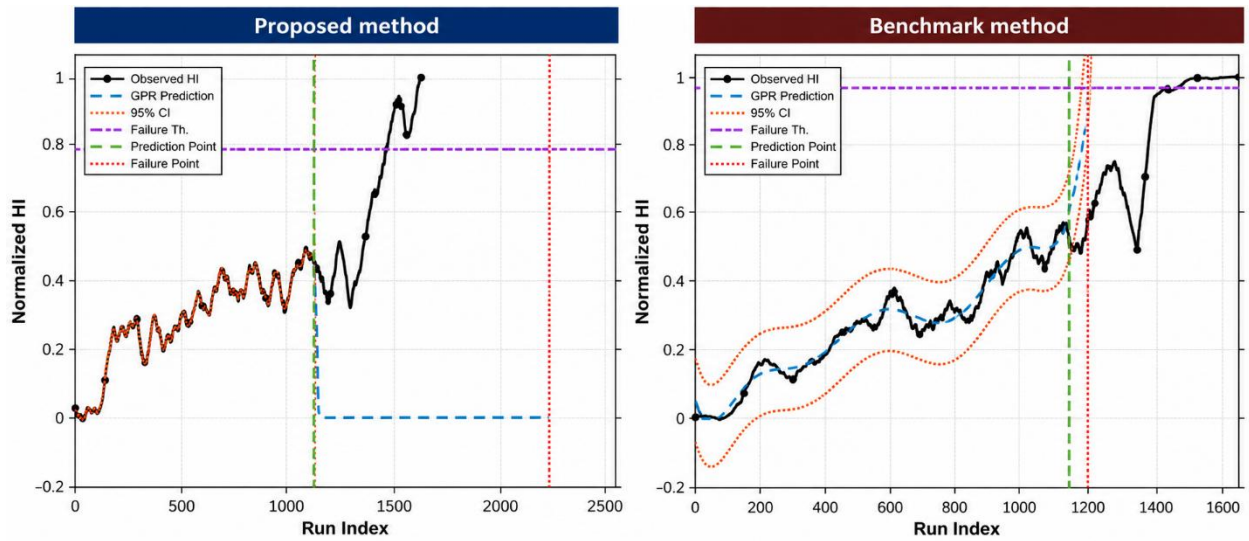


Figure 8. RUL comparison among bearing 1 (Outer race failure in bearing 1) at IMS dataset # 2.

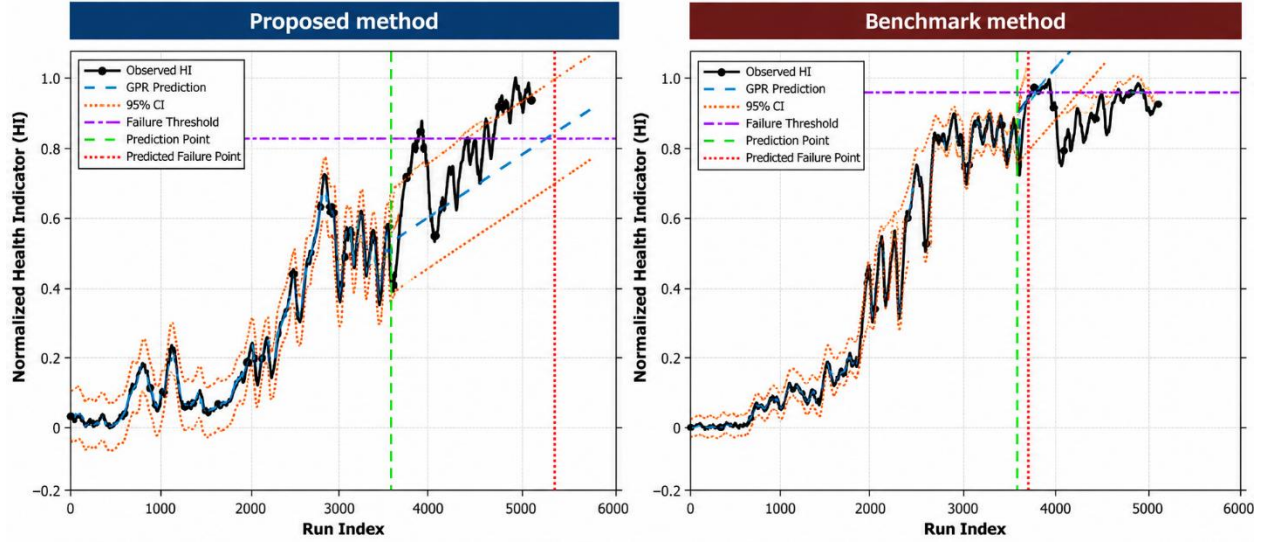


Figure 9. RUL comparison among bearing 3 (Outer race failure in bearing 3) at IMS dataset # 3.

4.1.5 Features extraction-based comparative results

Figure 10 compares the t-SNE feature distributions obtained from the proposed Bayesian-optimized FFT-1D-CNN method and the benchmark FFT-CNN-LSTM method for IMS Dataset #2. In the proposed method, the healthy and outer-race-fault samples form relatively distinct and compact clusters with only limited overlap between the two classes. The outer-race-fault features are concentrated in several localized regions that are well separated from most of the healthy-bearing clusters, indicating stronger class discrimination in the learned latent feature space. In contrast, the benchmark CNN-LSTM representation exhibits considerably greater overlap between the healthy and outer-race-fault samples, particularly in the central portion of the t-SNE space. Although several class-specific clusters remain distinguishable, a substantial number of outer-race-fault samples are embedded within or adjacent to the healthy-bearing clusters, suggesting weaker feature separability. Overall, the visualization indicates that the proposed Bayesian-optimized FFT-1D-CNN learns a more discriminative feature representation for distinguishing healthy and outer-race-fault bearing conditions than the CNN-LSTM benchmark. This observation is consistent with the quantitative diagnostic results for IMS Dataset #2, where the proposed method achieved 100.00% classification accuracy, compared with 96.69% for the benchmark method. The improved visual separation of the outer-race-fault

samples is also consistent with the higher fault-detection sensitivity of the proposed method, particularly compared with the lower outer-race-fault recall of 0.9068 ± 0.0182 obtained by the benchmark method.

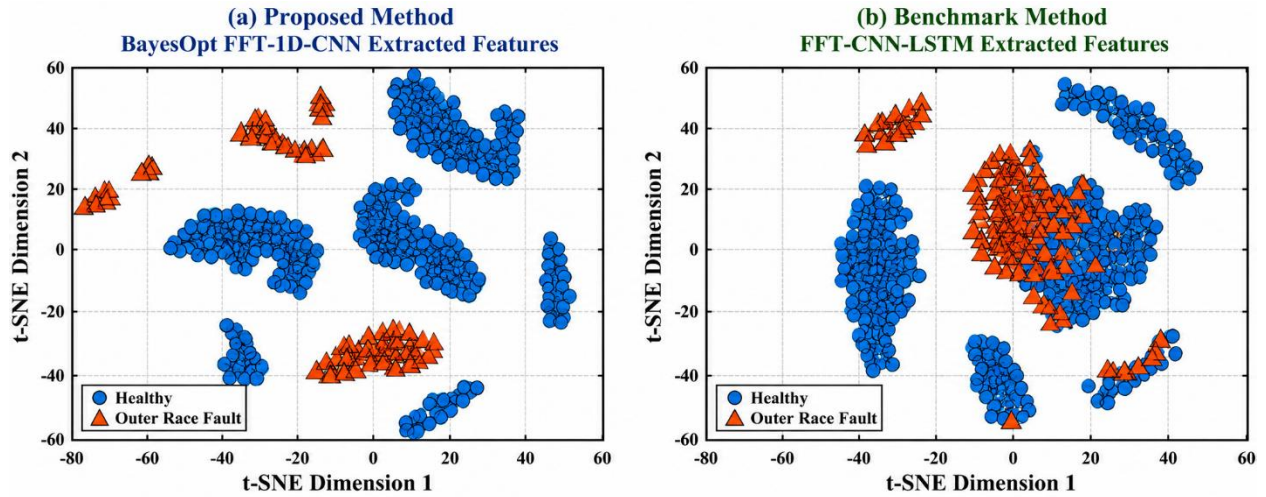


Figure 10. t-SNE visualization of the extracted feature distributions for IMS Dataset #2: (a) proposed Bayesian-optimized FFT-1D-CNN method and (b) benchmark FFT-CNN-LSTM method. The proposed method exhibits clearer separation between healthy and outer-race-fault conditions, whereas the benchmark representation shows greater class overlap in the latent feature space.

5 Conclusion and future outlook

This study developed a frequency-domain-based Bayesian-optimized 1-D CNN feature-learning framework for the integrated diagnosis and RUL prediction of rolling-element bearings under run-to-failure conditions. The framework transforms vibration signals into frequency-domain representations and employs Bayesian optimization to systematically determine influential CNN hyperparameters, including the learning rate, mini-batch size, L-2 regularization, feature dimension, and convolutional filters. The resulting deep features are subsequently utilized for Gaussian-kernel SVM-based fault diagnosis and GPR-based degradation modeling and RUL prediction, providing a unified diagnostic and prognostic framework. The framework was validated using three IMS run-to-failure datasets representing inner-race, rolling-element, and outer-race failure modes under different recording durations and sensor-channel configurations. The proposed method demonstrated consistently strong diagnostic performance, achieving overall classification accuracies of 99.85%, 100.00%, and 99.98% for IMS Datasets #1, #2, and #3, respectively, compared with 99.69%, 96.69%, and 99.86% obtained using the benchmark CNN-LSTM method. The most substantial improvement was observed for Dataset #2, where the proposed method achieved perfect classification and substantially improved sensitivity to the outer-race fault condition. The t-SNE analysis further supported these quantitative results, showing clearer and more compact separation between healthy and outer-race-fault features for the proposed method than for the benchmark approach. For prognostic assessment, the learned bearing-wise features were used to construct a normalized health indicator, followed by Bayesian-optimized GPR to predict future degradation and estimate RUL based on an adaptive failure threshold. Across the failed bearings in the three datasets, the proposed framework generated a more progressive and degradation-sensitive HI trajectory, preserving the evolution from relatively healthy operation through accelerated degradation toward failure. In contrast, the benchmark

method exhibited comparatively abrupt HI changes and, in several cases, premature saturation near the upper normalization limit. These characteristics can limit the representation of progressive degradation and reduce the interpretability of long-horizon RUL extrapolation. Overall, the results demonstrate that integrating frequency-domain signal representation, Bayesian-optimized deep feature learning, fault classification, degradation-sensitive health-indicator construction, and probabilistic RUL prediction provides an effective approach for intelligent bearing condition monitoring. The proposed framework not only achieves highly accurate fault diagnosis but also provides an interpretable representation of progressive bearing degradation, supporting the transition from conventional fault detection toward predictive maintenance and condition-based maintenance decision-making. Future work will extend the framework toward agentic AI-based autonomous decision-making, with particular emphasis on investigating how progressive bearing faults influence overall structural vibration behavior and using this information for system-level health assessment and proactive maintenance decisions.

Author's contributions: Methodology conceptualization: Hicham Chaoui and Abdullah Al Mamun; literature review and investigation: Charles Lowe and Sujatha Alla; data collection: Charles Lowe; manuscript drafting, MATLAB code implementation, and results analysis: Md Hedayetul Islam Chy and Abdullah Al Mamun; manuscript review and supervision: Hicham Chaoui, Abdullah Al Mamun, and Sujatha Alla.

Statements and declarations

Funding: The authors received no financial support, grants, or other funding for the research, preparation, or publication of this manuscript.

Ethics approval: Not applicable.

Competing interests: The authors declare that they have no financial or non-financial interests that could be considered relevant to the work presented in this manuscript.

Consent for publication: All authors have reviewed and approved the final version of the manuscript, agreed to the listed authorship, and provided their consent for submission and publication. The authors further confirm that this work has not been previously published elsewhere.

Data availability: The datasets generated and/or analyzed during this study are available from <https://data.nasa.gov/docs/legacy/IMS.zip>.

References

1. Yadav E, Chawla VK, Angra S, Yadav S (2025) The Fault Diagnosis of Different Rotating Machine Elements by Using Infrared Thermography Images and Extended Adaptive Neuro-Fuzzy Inference System: An Experimental Evaluation. *MAPAN* 40:731–749. <https://doi.org/10.1007/s12647-025-00838-6>
2. Kang T-J, Yang C, Park Y, et al (2018) Electrical Monitoring of Mechanical Defects in Induction Motor-Driven V-Belt–Pulley Speed Reduction Couplings. *IEEE Trans Ind Appl* 54:2255–2264. <https://doi.org/10.1109/TIA.2018.2805840>
3. Wong VW, Tung SC (2016) Overview of automotive engine friction and reduction trends—Effects of surface, material, and lubricant-additive technologies. *Friction* 4:1–28. <https://doi.org/10.1007/s40544-016-0107-9>
4. Nikolakopoulos PG, Papadopoulos CI, Kaiktsis L (2011) Elastohydrodynamic analysis and Pareto optimization of intact, worn and misaligned journal bearings. *Meccanica* 46:577–588. <https://doi.org/10.1007/s11012-010-9319-7>
5. Xu Z, Chen Y (2026) Dynamic Modeling and Analysis of Local Defect System of Rolling Bearing Outer Ring. *J Vib Eng Technol* 14:24. <https://doi.org/10.1007/s42417-025-02186-w>
6. Li C, Cao Z, Li S, Dai J (2024) Frequency Estimation of Vibration Signals: A Subspace Approach for Bearing Fault Diagnosis. *IEEE Sens J* 24:449–459. <https://doi.org/10.1109/JSEN.2023.3331355>
7. McInerny SA, Dai Y (2003) Basic vibration signal processing for bearing fault detection. *IEEE Trans Educ* 46:149–156. <https://doi.org/10.1109/TE.2002.808234>
8. Mongia C, Goyal D, Sehgal S (2022) Vibration response-based condition monitoring and fault diagnosis of rotary machinery. *Mater Today Proc* 50:679–683. <https://doi.org/10.1016/j.matpr.2021.04.395>
9. Hassan IU, Panduru K, Walsh J (2024) An In-Depth Study of Vibration Sensors for Condition Monitoring. *Sensors* 24:740. <https://doi.org/10.3390/s24030740>
10. Prawin J (2025) Deep learning neural networks with input processing for vibration-based bearing fault diagnosis under imbalanced data conditions. *Struct Heal Monit* 24:883–908. <https://doi.org/10.1177/14759217241246508>
11. Bharatheedasan K, Maity T, Kumaraswamidhas LA, Durairaj M (2025) Enhanced fault diagnosis and remaining useful life prediction of rolling bearings using a hybrid multilayer perceptron and LSTM network model. *Alexandria Eng J* 115:355–369. <https://doi.org/10.1016/j.aej.2024.12.007>
12. Zhou S, Xiao M, Bartos P, et al (2020) Remaining Useful Life Prediction and Fault Diagnosis of Rolling Bearings Based on Short-Time Fourier Transform and Convolutional Neural Network. *Shock Vib* 2020:1–14. <https://doi.org/10.1155/2020/8857307>
13. Frazier PI (2018) Bayesian Optimization. In: *Recent Advances in Optimization and Modeling of Contemporary Problems*. INFORMS, pp 255–278
14. Mamun A, Guerra-Zubiaga D, Peng Y (2025) Smart systems for real-time bearing faults diagnosis by using vibro-acoustics sensor fusion with Bayesian optimised 1-D CNNs. *Nondestruct Test Eval* 40:2113–2137. <https://doi.org/10.1080/10589759.2024.2375567>
15. Li Z, Wang Y, Ma J (2021) Fault Diagnosis of Motor Bearings Based on a Convolutional Long Short-Term Memory Network of Bayesian Optimization. *IEEE Access* 9:97546–97556. <https://doi.org/10.1109/ACCESS.2021.3093363>
16. Lim Y-F, Ng CK, Vaitesswar US, Hippalgaonkar K (2021) Extrapolative Bayesian Optimization with Gaussian Process and Neural Network Ensemble Surrogate Models. *Adv Intell Syst* 3:. <https://doi.org/10.1002/aisy.202100101>
17. Lin S-L (2021) Application of Machine Learning to a Medium Gaussian Support Vector Machine in the

Diagnosis of Motor Bearing Faults. *Electronics* 10:2266. <https://doi.org/10.3390/electronics10182266>

18. Kumar PS, Kumaraswamidhas LA, Laha SK (2021) Selection of efficient degradation features for rolling element bearing prognosis using Gaussian Process Regression method. *ISA Trans* 112:386–401. <https://doi.org/10.1016/j.isatra.2020.12.020>
19. Aye SA, Heyns PS (2017) An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission. *Mech Syst Signal Process* 84:485–498. <https://doi.org/10.1016/j.ymssp.2016.07.039>
20. Li X, Ma Z, Yuan Z, et al (2024) A review on convolutional neural network in rolling bearing fault diagnosis. *Meas Sci Technol* 35:072002. <https://doi.org/10.1088/1361-6501/ad356e>
21. Alqunun K, Bechiri MB, Naoui M, et al (2025) An efficient bearing fault detection strategy based on a hybrid machine learning technique. *Sci Rep* 15:18739. <https://doi.org/10.1038/s41598-025-02439-4>
22. Beiranvand V, Hare W, Lucet Y (2017) Best practices for comparing optimization algorithms. *Optim Eng* 18:815–848. <https://doi.org/10.1007/s11081-017-9366-1>
23. Cong S, Zhou Y (2023) A review of convolutional neural network architectures and their optimizations. *Artif Intell Rev* 56:1905–1969. <https://doi.org/10.1007/s10462-022-10213-5>
24. Ibrahim MQ, Hussein NK, Guinovart D, Qaraad M (2025) Optimizing Convolutional Neural Networks: A Comprehensive Review of Hyperparameter Tuning Through Metaheuristic Algorithms. *Arch Comput Methods Eng* 32:5123–5160. <https://doi.org/10.1007/s11831-025-10292-x>
25. Wojciuk M, Swiderska-Chadaj Z, Siwek K, Gertych A (2024) Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization. *Heliyon* 10:e26586. <https://doi.org/10.1016/j.heliyon.2024.e26586>
26. Rom ARM, Ibrahim S, Fadzil AFA, et al (2025) A Review of Hyperparameter Tuning Methods in Machine Learning. In: 2025 6th International Conference on Artificial Intelligence and Data Sciences (AiDAS). IEEE, pp 1–6
27. Luo G (2016) A review of automatic selection methods for machine learning algorithms and hyperparameter values. *Netw Model Anal Heal Informatics Bioinforma* 5:18. <https://doi.org/10.1007/s13721-016-0125-6>
28. Komer B, Bergstra J, Eliasmith C (2014) Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn. In: Proceedings of the 13th Python in Science Conference. pp 32–37
29. Bie F, Zhang Y, Zhang Y, et al (2024) Research on compound fault pattern recognition of rotor system based on grid search VMD parameters combined with RCMDE-Relief-F-GRNN. *J Mech Sci Technol* 38:5899–5912. <https://doi.org/10.1007/s12206-024-1011-2>
30. Li X, Zheng A, Zhang X, et al (2013) Rolling element bearing fault detection using support vector machine with improved ant colony optimization. *Measurement* 46:2726–2734. <https://doi.org/10.1016/j.measurement.2013.04.081>
31. Arias Velásquez RM (2024) Bearings faults and limits in wind turbine generators. *Results Eng* 21:101891. <https://doi.org/10.1016/j.rineng.2024.101891>
32. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J Mach Learn Res* 13:281–305
33. Mantovani RG, Rossi ALD, Vanschoren J, et al (2015) Effectiveness of Random Search in SVM hyperparameter tuning. In: 2015 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 1–8
34. Elansari T, Ouanan M, Bourray H (2023) Mixed Radial Basis Function Neural Network Training Using Genetic Algorithm. *Neural Process Lett* 55:10569–10587. <https://doi.org/10.1007/s11063-023-11339-5>

35. Shen C, Xie J, Wang D, et al (2019) Improved Hierarchical Adaptive Deep Belief Network for Bearing Fault Diagnosis. *Appl Sci* 9:3374. <https://doi.org/10.3390/app9163374>
36. Wang J, Mo Z, Zhang H, Miao Q (2019) A Deep Learning Method for Bearing Fault Diagnosis Based on Time-Frequency Image. *IEEE Access* 7:42373–42383. <https://doi.org/10.1109/ACCESS.2019.2907131>
37. Wang S, Zhang Y, Zhang B, et al (2022) On the Sparse Gradient Denoising Optimization of Neural Network Models for Rolling Bearing Fault Diagnosis Illustrated by a Ship Propulsion System. *J Mar Sci Eng* 10:1376. <https://doi.org/10.3390/jmse10101376>
38. Jiaocheng M, Jinan S, Xin Z, Peng Z (2022) Bayes-DCGRU with bayesian optimization for rolling bearing fault diagnosis. *Appl Intell* 52:11172–11183. <https://doi.org/10.1007/s10489-021-02924-z>
39. Pelikan M (2005) Bayesian Optimization Algorithm. pp 31–48
40. Binois M, Wycoff N (2022) A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization. *ACM Trans Evol Learn Optim* 2:1–26. <https://doi.org/10.1145/3545611>
41. Garrido-Merchán EC, Hernández-Lobato D (2020) Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing* 380:20–35. <https://doi.org/10.1016/j.neucom.2019.11.004>
42. Wen L, Li X, Gao L (2021) A New Reinforcement Learning Based Learning Rate Scheduler for Convolutional Neural Network in Fault Classification. *IEEE Trans Ind Electron* 68:12890–12900. <https://doi.org/10.1109/TIE.2020.3044808>
43. Wen L, Gao L, Li X, Zeng B (2021) Convolutional Neural Network With Automatic Learning Rate Scheduler for Fault Classification. *IEEE Trans Instrum Meas* 70:1–12. <https://doi.org/10.1109/TIM.2020.3048792>
44. Wan A, Li P, AL-Bukhaiti K, et al (2025) Fault diagnosis of air conditioning compressor bearings using wavelet packet decomposition and improved 1D-CNN. *Next Energy* 9:100424. <https://doi.org/10.1016/j.nxener.2025.100424>
45. Jiang K, Cai Y, Han D, et al (2026) MDEA-Net: multisource disentangled expert adaptation network for cross-domain bearing fault diagnosis. *Struct Heal Monit*. <https://doi.org/10.1177/14759217261421679>
46. Li S, Xuan J, Wang Z, et al (2024) Noisy Open Set Adversarial Domain Adaption for Bearing Fault Diagnosis Based on Optimized Divergence. *IEEE Trans Instrum Meas* 73:1–17. <https://doi.org/10.1109/TIM.2024.3450060>
47. Ni Y, Li S, Guo P (2025) Discrete wavelet integrated convolutional residual network for bearing fault diagnosis under noise and variable operating conditions. *Sci Rep* 15:16185. <https://doi.org/10.1038/s41598-025-99346-5>
48. Zheng S, Wu Z, Xu Y, et al (2025) DPNet: A Lightweight Directional-Aware Pointwise Network for Dropper Defect Detection in High-Speed Railway Preventive Maintenance. *IEEE Trans Instrum Meas* 74:1–14. <https://doi.org/10.1109/TIM.2025.3544329>
49. Alam TE, Ahsan MM, Raman S (2025) Multimodal bearing fault classification under variable conditions: A 1D CNN with transfer learning. *Mach Learn with Appl* 21:100682. <https://doi.org/10.1016/j.mlwa.2025.100682>
50. Liu H, Yao D, Yang J, Li X (2019) Lightweight Convolutional Neural Network and Its Application in Rolling Bearing Fault Diagnosis under Variable Working Conditions. *Sensors* 19:4827. <https://doi.org/10.3390/s19224827>
51. Li H, Wu W, Zhang M (2025) Beyond reconstruction: A deep convolutional critic with spectral normalization for superior bearing fault anomaly detection. *Results Eng* 28:108053. <https://doi.org/10.1016/j.rineng.2025.108053>

52. Iqbal M, Lee CKM, Keung KL (2024) An intelligent fault diagnosis method based on L2 regularization and deep transfer LSTM. In: 2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, pp 0282–0286
53. Su X, Han J, Chen C, et al (2025) Intelligent Workshop Bearing Fault Diagnosis Method Based on Improved Convolutional Neural Network. *Lubricants* 13:521. <https://doi.org/10.3390/lubricants13120521>
54. Song X, Zhu D, Liang P, An L (2021) A new bearing fault diagnosis method using elastic net transfer learning and LSTM. *J Intell Fuzzy Syst* 40:12361–12369. <https://doi.org/10.3233/JIFS-210503>
55. Xu M, Yu Q, Chen S, Lin J (2024) Rolling Bearing Fault Diagnosis Based on CNN-LSTM with FFT and SVD. *Information* 15:399. <https://doi.org/10.3390/info15070399>