A Physics-Informed Feature Weighting Method for Bearing Fault Diagnostics

Hao Lu[1,2], Venkat Pavan Nemani[1], Vahid Barzegar[3], Cade Allen[1], Chao Hu[4,*], Simon Laflamme[2,3], Soumik Sarkar[1], and Andrew T. Zimmerman[5,6]

[1]Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA

[2]Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA

[3]Department of Civil, Environmental and Construction Engineering, Iowa State University, Ames, IA 50011 USA

[4]Department of Mechanical Engineering, University of Connecticut, Storrs, CT 06269 USA

[5]Percēv LLC, Davenport, IA 52807, USA

[6]Grace Technologies, Davenport, IA 52807, USA

* Indicates corresponding author

Author email addresses: hlu1@iastate.edu, vnemani@iastate.edu, barzegar@iastate.edu, allen1@iastate.edu, chao.hu@uconn.edu, laflamme@iastate.edu, andyz@gracetechnologies.com

Abstract

Intelligent bearing diagnostics has gained popularity over the last few years. However, most of the diagnostic methods are developed under the assumption that training and test data sets are collected under the same working conditions. This assumption is rare in practical scenarios because rotating machinery usually works under wide ranges of rotational speeds and loads. As bearings work under complex and time-varying operating conditions, the test data might come from a data distribution outside the training distribution. Purely data-driven diagnostic models often cannot provide reliable classifications for out-of-distribution test data. To tackle this challenge, this paper proposes a physics-informed feature weighting method for bearing diagnostics. First, a signal processing step is proposed that leverages physical knowledge of bearing faults to extract discriminative features that are robust to bearing speed variation. Then, a novel physics-informed feature weighting layer is developed to assign higher weights for features located closer to bearing fault characteristic frequencies. The feature weighting layer enhances the model's sensitivity towards the fault-related features among the speed invariant features. Through experiments on three bearing datasets, the effectiveness of the proposed method is validated and shown to have promise for bearing fault diagnostics under different operating conditions. This study also details the deployment of a physics-informed convolutional neural network model on an Industrial Internet of Things (IIoT) device, where edge computing gives users a real-time evaluation of bearing health.

1. Introduction

Rolling element bearings are essential in rotating machineries, such as wind turbine drive trains [1], hydraulic motors on agricultural machines [2], and fans, pumps, and blows on industrial equipment [3]. Since bearings often work under heavy loads or in harsh environments, they may suffer from unexpected failure, often more likely than other machine components. Thus, the long-term reliability and real-time health of a bearing significantly affect the machine's performance and working safety. Taking large induction motors as an example, bearing failures account for around 45-50% of motor failures [4].

Unexpected bearing failure might damage adjacent machine components, lead to unexpected downtime, and cause severe financial loss. Effective bearing fault diagnostics methods that can detect early-stage bearing faults and possibly identify the fault types are critical to avoiding high maintenance costs and accidents.

Over the past few decades, various sensing technologies, such as acoustic emission monitoring [5, 6], motor current analysis [7, 8], and vibration-based diagnostics [9, 10], have been applied for bearing fault diagnostics. Acoustic emission sensors are known for detecting cracks inside bearing components, motor current-based fault diagnostics is performed by detecting fault-related frequency components in the current frequency spectrum, and vibration-based diagnostics is carried out by detecting fault-related features in vibration signals. Although these sensing techniques have their unique advantages and benefits, there is still no consensus on which technique is the best choice for all applications. The remainder of this paper is confined to vibration-based fault diagnostics.

Traditional vibration-based fault diagnostics rely on signal processing-based feature extraction to analyze and process vibration signals. Some techniques have been widely used in industrial applications, including fast Fourier transforms (FFT), wavelet transforms, and Hilbert transforms [11]. In recent years, many new signal processing techniques have been developed to capture incipient fault patterns. Two examples are the adaptive period matching enhanced sparse representation algorithm developed by Yao et al. [12] and the fault information-guided variational model decomposition method developed by Ni et al. [13]. Both methods aim to identify and extract fault-induced weak repetitive transients from raw vibration signals. These methods have shown effectiveness in revealing fault-related features. For most traditional signal processing-based diagnostic approaches, after revealing the fault-related features, the final decision is typically made by a vibration expert [14] or a simple rule-based algorithm (e.g., a threshold-based algorithm [15]).

It is worth noting that some recent studies have used novel data-driven signal processing methods to extract features. For example, Mao et al. [16] incorporated discriminant information into the loss function to develop a novel deep auto-encoder for bearing fault diagnostics. Similarly, Mohammad et al. [17] adopted a convolutional auto-encoder to extract features for bearing failure prognostics. These data-driven features are also predictive of bearing health but are less physically meaningful than features extracted by traditional signal processing techniques.

In the age of big data, fully data-driven condition monitoring and fault diagnostic techniques have recently gained popularity. These data-driven methods take raw sensor signals or features extracted from raw signals as input and automatically classify or estimate bearing health. Various machine learning techniques, such as support vector machines [18], k-nearest neighbors [19], and random forests [20], have been applied. Zhang et al. [21] fed time- and frequency-domain statistical features extracted from vibration data and developed an ensemble learning-based incremental support vector machine for fault diagnostics. Jing et al. [22] developed a health index using principal component analysis and k-nearest neighbors; the health index allowed for detecting bearing faults and monitoring the bearing degradation process. Xu et al. [20] proposed a diagnostic method that first converts vibration signals into 2D gray-scale images by continuous wavelet transform, then trains a random forest ensemble model. One limitation of traditional machine learning techniques such as these is that their performance highly depends on the predictive power of the features (e.g., how sensitive these features are to machine health) that can be manually extracted and selected. Unfortunately, highly predictive features for large-volume training datasets are typically engineered by domain experts, and that process can be both time-consuming and very costly. As a result, traditional machine learning techniques may not apply to big data scenarios [23].

As a new branch of machine learning, deep learning is gaining popularity for its ability to automate the learning of complicated input-output relationships. In contrast to traditional machine learning, deep learning typically does not require extensive human intervention or domain knowledge. It can automate feature engineering by algorithmically identifying the best features from a training dataset [24, 25]. This unique property makes deep learning applicable to large-volume datasets. Deep learning models are developed based on neural networks, including deep neural networks, convolutional neural networks, and recurrent neural networks such as long short-term memory networks [26] and gated recurrent units [27]. A deep neural network comprises several layers (typically >3) for feature extraction, and each layer can be treated as a feature extractor [28, 29]. The deep neural network automatically learns discriminative, fault-related features from training data and has been widely applied to bearing fault diagnostics

Generally, the effectiveness of traditional deep learning models is based on the assumption that the training data and test data come from the same or similar distributions. However, a well-known drawback of purely data-driven models is that they may have a low level of compliance with physics and may provide results that do not conform to physical knowledge of bearing faults [2]. A further effect of this lack of physical compliance is low generalizability, which means that purely data-driven deep learning models may have difficulties extrapolating to test data falling outside the training data distribution. In practical industrial settings, bearings work under complex and time-varying operating conditions (e.g., rotational speed and radial and axial loads), and the operating conditions of one bearing may differ vastly from those of another. Additionally, one bearing may operate under a noisier environment than a different bearing, or readings of one sensor may contain a higher level of noise than readings by a different sensor, leading to differences in the signal-to-noise ratio. As a result, data from test bearings whose health class is unknown and needs to be classified might come from a data distribution outside of a training distribution which data from training bearings tend to follow. Due to the lack of generalizability, purely data-driven diagnostic models often cannot provide reliable health classifications for those out-of-distribution test data.

As mentioned above, the causes of out-of-distribution data generally can be categorized into (1) training-test differences in operating conditions and (2) training-test differences in signal-to-noise ratio. In many diagnostic studies, training and test data are collected from bearings under the same or similar rotational speeds or loads. Data-driven models capable of learning the input-output relationship from training data can, therefore, yield decent accuracy on test data. However, in practical implementation, machinery typically works under various operating conditions that may deviate substantially from the conditions under which a training dataset has been generated. These complex and varied working conditions can lead to significant changes in vibration signals, making it difficult for pre-trained data-driven models to provide reliable diagnostic results. Because it is time-consuming and sometimes impossible to gather data under all possible operating conditions, developing a fault diagnostics model with robust performance under different test operating conditions has been a hot topic.

The second challenge is that the difference in signal-to-noise ratio between training and test data tends to be high due to increased environmental noise in field deployments. Training data are typically collected from a lab test stand; the signal is clean without noise. In practical implementation, background noise and interference almost always exist due to vibrations generated by other machinery. The background noise may interfere with data collection, and more noisy test data may lead to worse classification results [30, 31].

Two approaches have been attempted to address the challenge of distribution differences: (1) transfer learning and (2) physics-informed deep learning. Transfer learning focuses on learning common knowledge from one or more related but different scenarios to help the deep learning model perform better in the target scenario. Domain adaptation, as one of the transfer learning techniques, has been applied to bearing diagnostic applications to guide data-driven models to extract domain-invariant features that are robust to changes in operating conditions [32]. For instance, Li et al. [33] adopted maximum mean discrepancy as a distance metric to evaluate the feature difference between training and unlabeled test data, facilitating knowledge generalization across data collected under different operating conditions. Another example is that Zhu et al. [34] developed a multi-adversarial learning strategy for bearing fault diagnostics. In their strategy, a feature extractor is optimized to extract domain-invariant features, which are then fed into a condition predictor to estimate bearing health. A third example is a gearbox fault diagnostics study presented by Wei et al. [35]. The authors proposed a multisource domain adaptation framework, where each source domain is assigned a unique weight according to its distributional similarity to the target working condition.

The other approach to dealing with distribution differences is developing physics-informed deep learning models incorporating physical knowledge. Note that physics-informed deep learning has gained popularity across different engineering fields. Notable applications of physics-informed deep learning have been attempted by the scientific computing community, focusing on solving partial differential equations [36-39], fractional equations [40, 41], integral-differential equations [42, 43], etc. A notable example is the physics-informed neural network proposed by Raissi et al. [39]. In this example, physical knowledge is described by a nonlinear partial differential equation. A custom loss function is designed to guide the model to fit the training data and yield predictions that approximately satisfy the physical constraints. This research focuses on physics-informed deep learning for bearing diagnostics.

Numerous studies have shown that early bearing faults can be detected by analyzing the vibration amplitudes at the bearing fault characteristic frequencies [2, 44]. Incorporating this knowledge into data-driven deep learning models yields physics-informed models that generalize better to unseen data and are less likely to produce predictions that violate physics. Physics-informed deep learning models for bearing degradation modeling and diagnostics can be built by (1) designing model architecture by developing custom layers [44-47] or including signal processing algorithms to enhance the feature learning of fault information or (2) modifying the loss function often by including an additional, physics-informed loss term [2, 48].

The *first* approach to building physics-informed models modifies the model architecture to emphasize the hidden fault information in the vibration signal. Physics-informed models can be developed by imitating the signal processing steps done by vibration experts. Mohammad et al. [44] proposed a physics-based (or more appropriately, physics-informed) convolutional neural network (CNN) that consisted of (1) a spectral kurtosis (SK) analysis layer, (2) an envelope analysis layer, (3) a physics-informed convolutional layer, and (4) an FFT layer, followed by (5) a standard one-dimensional CNN (1D CNN). The first four layers were designed to obtain a processed frequency-domain signal with enhanced fault-related features, aiming to maintain the diagnostic performance on data collected under different operating speeds. Li et al. [45] proposed a specially designed CNN called WaveletKernelNet for bearing fault diagnostics, where a continuous wavelet convolutional layer is added as the first layer of the CNN to extract features capturing repetitive vibration impulses excited by bearing faults. Another way is to guide a deep learning model to focus on informative features and pay less attention to features that contribute less to the final output. The attention mechanism, which assigns importance to features according to their relevance to the final output, has been adopted to allow data-driven models to learn hidden physical knowledge from training data. Ding et al. [46] proposed a time-frequency transformer, which learns useful information from time-frequency representation using an attention mechanism. Similarly, Wang et al. [47] adopted the attention mechanism in a 1D CNN; a channel attention module and an excitation attention module are designed to help the deep learning model learn discriminant features of 1-D signals.

The *second* approach is achieved by adding a loss term to penalize results not compliant with physical knowledge. Sheng et al. [2] created a physics-informed deep learning approach for bearing diagnostics that adds a penalty to the training loss of a CNN when the CNN and a simple physics-informed threshold model disagree in predicting the healthy and heavy damage classes. The threshold model classifies the bearing health class by comparing the amplitudes of envelope spectrum sub bands to predefined thresholds. This penalty helps guide the CNN model to learn the physical knowledge in the threshold model and also helps reduce false positive classifications. Tongtong et al. [48] proposed an architecturally explainable network to model machine degradation; a knowledge-guided loss function was designed to constrain the health index value remains constant at the normal stage and follows a monotonic trend when the machine enters the degradation stage.

In addition to the challenges caused by distribution differences, some industry-relevant requirements for a data-driven diagnostic model are also worth noting. One requirement is that the diagnostic model output a minimal number of false alarms. False alarms cause unnecessary machine shutdown and negatively affect the model's reliability; furthermore, with repeated false alarms, users might develop alarm fatigue and start ignoring most alarms [49]. A second requirement is that the deployment of the diagnostic model should minimize deployment cost and response time. This is especially important in deployments that utilize battery-powered wireless sensors (which can be affordably and quickly deployed in large quantities). Typically, diagnostics are performed in an offline environment utilizing powerful computers where a local sensing node sends data to a computer, a pre-trained deep learning model is used to estimate bearing health conditions, and the computer sends the results back to the sensing node. This centralized approach can lead to unacceptable delays in safety-critical applications; transmitting raw data increases power consumption, affecting a sensing node's battery life and operating costs [50].

This paper proposes a physics-informed feature weighting method for bearing diagnostics. The proposed physics-informed CNN (PICNN) contains a novel feature weighting layer. The physics of bearing faults is incorporated in the signal processing and feature weighting layers. The main contributions are summarized as follows:

1) The proposed feature weighting layer incorporates the physics of bearing faults by adding constraints to the distribution of attention parameters, inspired by the adoption of the attention mechanism in [32]. The features that are located nearer bearing fault characteristic frequencies are assigned with higher weights so that the resulting diagnostic model focuses more on the features related to the bearing faults. A case study shows that the proposed method is more sensitive to fault-related features and provides more interpretable results when compared with a vanilla CNN.

2) The proposed diagnostic model has a simple architecture designed for quick deployment within a battery-powered wireless vibration sensor. The proposed model is embedded within the computational core of a commercial off-the-shelf wireless sensing platform, and the model's performance is verified through online diagnostics tests. To the best of our knowledge, this is the first time a physics-informed deep learning model has been reported to be implemented on an Industrial Internet of Things (IIoT) device for online bearing fault diagnostics.

3) The proposed signal processing step extracts features by converting vibration data from the time to order domain. A comparative study of order- vs. frequency-domain features shows that models using order-domain features as input are more robust to rotational speed changes than models using frequency-domain features.

The remainder of the paper is organized as follows. Section 2 introduces the proposed physics-informed feature weighting method. Section 3 presents three case studies used to evaluate the proposed method. Section 4 discusses a demonstration of truly online fault diagnostics where a purely data-driven CNN model and a PICNN model are implemented in an embedded system for on-the-edge bearing health classification. Section 5 summarizes conclusions.

2. Methodology

Figure 1 shows the standard CNN-based pipeline for fault diagnostics and the proposed PICNN-based pipeline. An accelerometer is mounted close to the target bearing, acquiring vibration data while the bearing operates. After data acquisition, signal processing is applied to generate an envelope order spectrum based on the acquired time-domain data. Then, the proposed physics-informed CNN (PICNN) model, composed of a feature weighting layer and a CNN, takes the envelope spectrum as input and estimates the health condition of the target bearing. The CNN-based pipeline comprises two main steps: signal processing and CNN; the PICNN-based pipeline adds a feature weighting step between signal processing and CNN. Detailed discussions of signal processing and the PICNN architecture are presented in sections 2.1 and 2.2, respectively. In section 2.3, we introduce the optimization algorithm for the PICNN model.

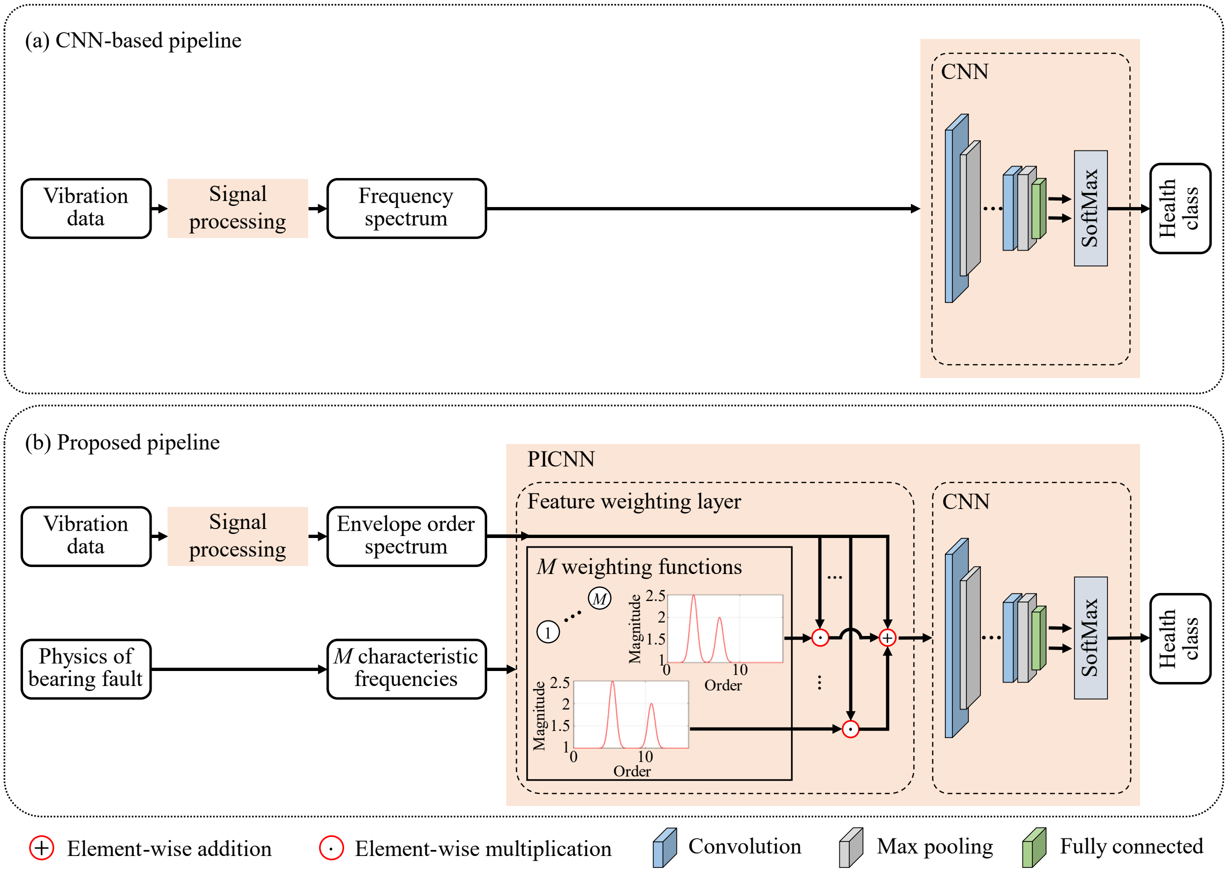


Figure 1. Standard CNN-based pipeline and proposed PICNN pipeline

2.1 Signal processing

The physics of bearing failure is incorporated in the signal processing step to extract fault-related features. A rolling element bearing has four main components: inner ring, outer ring, rollers, and cage. This research focuses on fault detection of two common fault types: the inner race fault (located on the inner raceway of the outer ring) and the outer race fault (located on the outer raceway of the inner ring). In the early stage of bearing degradation, mostly local defects are present, manifesting as dents due to plastic deformation of the rolling surface. While a bearing is rotating, contact between the rolling element and the dented area generates vibration impulses that excite the high-frequency resonance of the bearing.

The frequency of the bearing fault impulse is called the fault characteristic frequency, which is determined by the fault type, the rotational speed, and the geometric parameters of the bearing. The formulas of bearing fault characteristic frequency are as follows:

$$f_{\mathrm{IRD}} = f_r \times \frac{Z}{2}\left(1 + \frac{d}{D}\cos\alpha\right) \tag{1}$$

$$f_{\mathrm{ORD}} = f_r \times \frac{Z}{2}\left(1 - \frac{d}{D}\cos\alpha\right) \tag{2}$$

where f_{IRD} and f_{ORD} denote the fault characteristic frequencies of the inner race and outer race, respectively, d is the roller or ball diameter, D is the pitch diameter, Z denotes the number of rollers, f_{r} denotes the shaft rotational speed, and α denotes the contact angle.

Theoretically, when analyzing the bearing vibration signal, the bearing diagnostics can be performed by looking at the amplitude at fault characteristic frequencies; a higher amplitude indicates a higher chance of bearing component failure. However, the bearing fault impulse could excite the structural resonance of the machine, which leads to the amplitude modulation phenomenon. Due to amplitude modulation, the energy in the low-frequency band (where bearing fault characteristic frequencies and their harmonics are located) is significantly weak. A high-energy resonance frequency band can be observed around the system resonance frequency. Therefore, amplitude modulation makes it challenging to directly infer bearing health from the frequency spectrum. A well-established solution to this challenge is demodulation.

Envelope spectrum analysis is one of the well-known demodulation techniques for bearing vibration analysis [51]. To this end, our research uses the Hilbert transform to construct the demodulated signal from the sample vibration signal. If $a(t)$ denotes the time-domain signal, then its Hilbert transform $H\big(a(t)\big)$ is calculated by:

$$H\big(a(t)\big) = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{a(\tau)}{t-\tau}\, d\tau \tag{3}$$

The analytical signal is defined as $A(t) = a(t) + jH\big(a(t)\big)$, where j denotes the unit imaginary number. By performing the Hilbert transform and envelope analysis, a clear representation of the fault characteristic frequencies can be extracted from the vibration signal.

A sample vibration signal in the presence of an outer race fault is shown in Figure 2. It is hard to extract the diagnostic information by directly observing the raw signal (the blue waveform in Figure 2a). Even after applying Fourier transform to convert the data into the frequency domain (the blue spectrum in Figure 2b), the feature amplitudes in the low-frequency band are relatively small. The red dashed waveform in Figure 2a shows the enveloped signal, demodulated using the Hilbert transform. From the red spectrum in Figure 2c, it can be seen that the spectrum of the signal envelope reveals useful information for fault diagnostics, such as the fault characteristic frequency (f_{ORD}), its first harmonic ($2 \times f_{\mathrm{ORD}}$), and other harmonics.

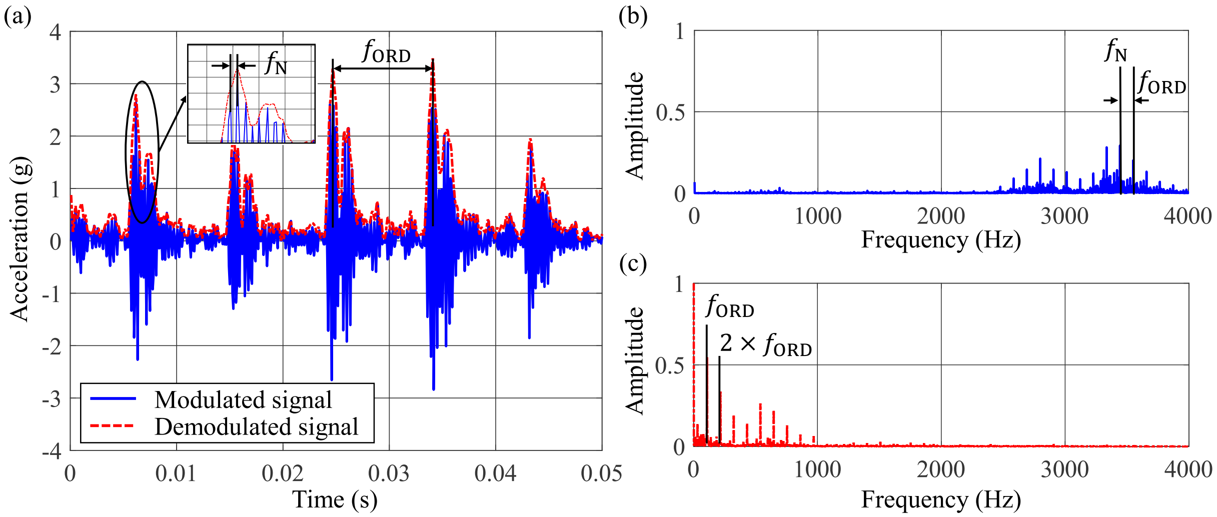


Figure 2. Bearing vibration signals in the presence of an outer race fault: (a) time-domain signals, (b) modulated signal in frequency domain, and (c) demodulated signal in frequency domain.

Equations (1-2) show that a bearing fault characteristic frequency can be computed as the product between the shaft speed and a constant value. If the bearing rotational speed changes, the fault characteristic frequencies will shift accordingly. To remove the influence of speed variation, we apply computed order tracking (COT). This resampling technique samples a vibration signal at constant increments of the shaft angle rather than constant increments of time. After performing computed order tracking and envelope analysis, the vibration signal is converted from a time series signal to an enveloped order spectrum. The order is defined as the frequency normalized by the reference speed:

$$o = \frac{f}{f_{\mathrm{r}}} \tag{4}$$

where o denotes the order, f denotes the frequency of the observed vibration, and f_{r} denotes the reference speed, which is the shaft's rotational speed.

2.2 PICNN model

2.2.1 Physics-informed feature weighting layer

Inspired by these earlier studies on [44, 45, 47], we design a physics-informed feature weighting layer and use data to optimize the parameters of this layer through backpropagation. After being converted into the frequency domain, the features close to fault characteristic frequencies and harmonics are pivotal for fault classification. Therefore, more attention needs to be assigned to these fault-related features, while the features far from those locations are less important to the final prediction. And for a deep learning model, prompting the first layer to reasonably extract fault-related information is of significant help for subsequent feature extraction and classification [9, 52]. Here, the physics-informed weighting layer is designed as the first layer of the PICNN model. The weighting layer consists of several weighting functions.

The physics-informed weighting layer assigns higher attention weights to the features more related to that fault. Given input **X,** each weighting function targets a particular bearing fault and returns weighted features by:

$$\mathbf{Y}_m = \mathbf{W}_m \odot \mathbf{X} \tag{5}$$

where m indicates the index of a fault type and $m = 1, \ldots, M$, with M being the total number of fault types, $\mathbf{W}_m$ denotes the weighting vector, $\odot$ is the Hadamard product that denotes the element-wise multiplication of two matrices or vectors of the same size.

Many functions can be used as weighting functions. Appendix A discusses several options for weighting functions and their performance. For example, the radial basis function with a Gaussian decay form takes the following form:

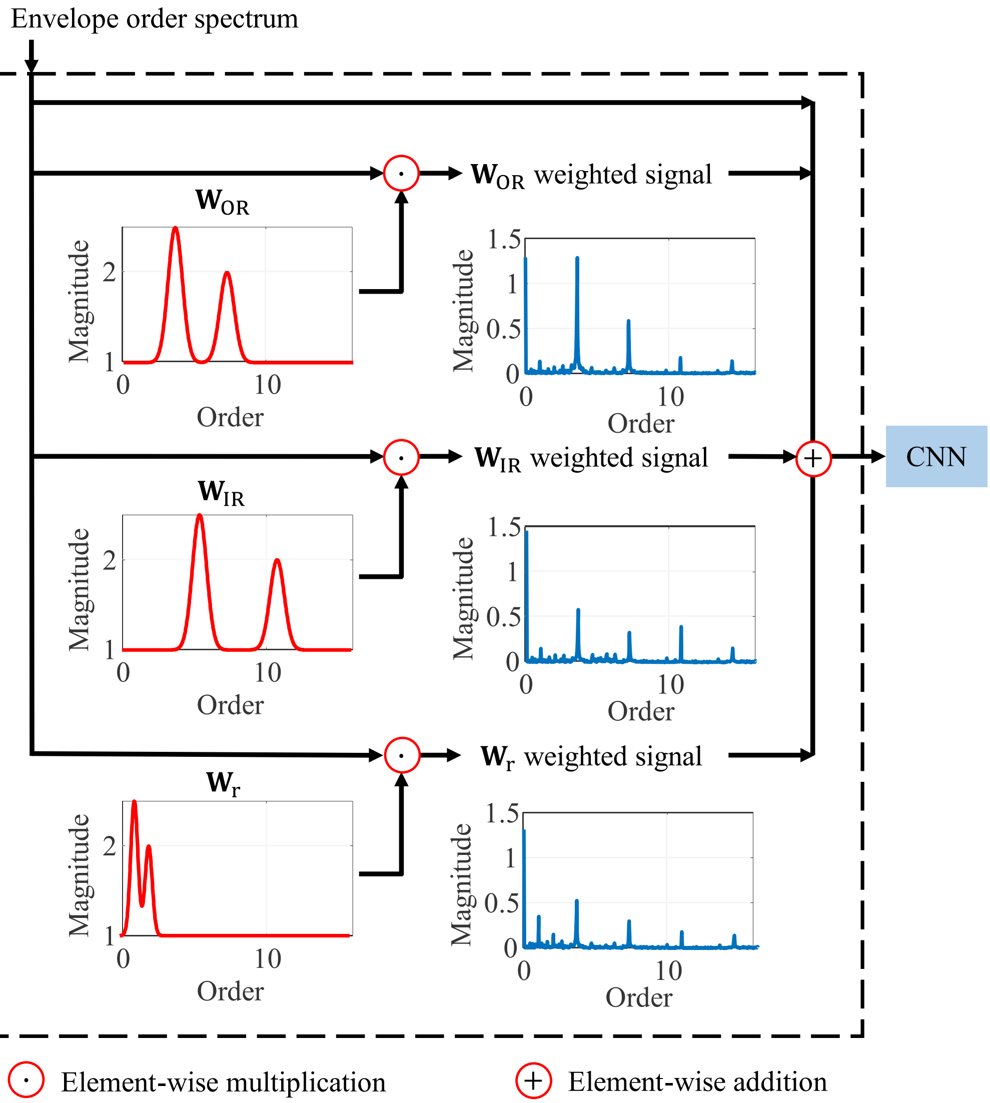
$$F^m(l, l^m_{\mathrm{fault}}) = 1 + \sum_{n=1}^{N+1} a^m_n \exp\left(-\frac{\left(l - n \cdot l^m_{\mathrm{fault}}\right)^2}{2(\sigma^m_n)^2}\right) \tag{6}$$

where l denotes the frequency position where the feature to be weighted is located, l^m_{fault} denotes the characteristic frequency related to bearing fault m, N indicates the number of harmonics that the weighting function considers, and a^m_n and σ^m_n are the shape parameters of the weighting function, quantifying the weighting strength and the decay rate, respectively. The input of the feature weighting layer is envelope order spectrum, and the frequency position of each input element is predefined after signal processing. With the help of the weighting function, the weighting vector $\mathbf{W}_m = [F^m(l_1, l^m_{\mathrm{fault}}), F^m(l_2, l^m_{\mathrm{fault}}), \ldots, F^m(l_{\max}, l^m_{\mathrm{fault}})]$ can be calculated.

An example of a feature weighting layer is visualized in Figure 3. In this research, three weighting vectors are created, marked as $\mathbf{W}_{\mathrm{OR}}$,$\mathbf{W}_{\mathrm{IR}}$, and $\mathbf{W}_{\mathrm{r}}$, and each of these weighting vectors allows assigning higher weights to outer race fault features, inner race fault features, and features related to bearing rotational speed, respectively. For each weighting function, the features located far from any fault-related frequency are given a weight of one to prevent any undesired loss of information. Each weighting function weighs the raw features by performing element-wise multiplication. After feature weighting, all three weighted signals (vectors of weighted features) and the raw signal (vector of raw features) are merged by element-wise addition. The final output of the feature weighting layer is:

$$\mathbf{Y}_{\text{Weighted}} = \mathbf{W}_{\text{OR}} \odot \mathbf{X} + \mathbf{W}_{\text{IR}} \odot \mathbf{X} + \mathbf{W}_{\text{r}} \odot \mathbf{X} + \mathbf{X} \tag{7}$$

This final output of the weighting layer is then fed into the CNN for further feature extraction and classification.



Envelope order spectrum
$\mathbf{W}_{\text{OR}}$
$\mathbf{W}_{\text{OR}}$ weighted signal
$\mathbf{W}_{\text{IR}}$
$\mathbf{W}_{\text{IR}}$ weighted signal
$\mathbf{W}_{\text{r}}$
$\mathbf{W}_{\text{r}}$ weighted signal
Magnitude
Order
0
10
1
2
0.5
1.5
CNN
⊙ Element-wise multiplication
⊕ Element-wise addition

Figure 3. A general pipeline of a feature weighting layer

2.2.2 PICNN model architecture

The PICNN model is composed of a customized feature weighting layer (see section 2.2.1) followed by several convolutional layers, pooling layers, and dense layers. The detailed architecture of a PICNN model is listed in Appendix B. Given that the input consists of one-dimensional features (output by the physics-informed feature weighting layer), the 1D convolutional layer is adopted in this study. Each 1D convolutional layer uses a set of learnable kernels to perform convolution; each kernel operates on local segments of the input data and generates a feature vector.

Mark $\mathbf{X}^l$ as the lth layer input and $\boldsymbol{h}_k^l$ as the kth feature vector of the lth layer output. The input data $\mathbf{X}^l$ is split into K segments $\mathbf{X}^l = [\mathbf{x}_1^l, \mathbf{x}_2^l, \mathbf{x}_3^l, \ldots, \mathbf{x}_K^l]$, where the length of $\mathbf{x}_k^l$ is equal to the length of $\mathbf{w}^l$. The output of each segment can be expressed as:

$$\boldsymbol{h}_k^l = f\left(\mathbf{w}^{l^{\mathrm{T}}} \cdot \mathbf{x}_k^l + b^l\right) \tag{8}$$

where $\mathbf{w}^l$, b^l, and f denote the weights of the convolution kernel, the bias of the convolutional kernel, and the activation function of the convolutional layer, respectively [53]. One important property of a convolutional layer is weight sharing. The same convolution kernels traverse the input once in a fixed stride, which leads to fewer network parameters and a lower risk of over-fitting in the training process.

After the convolutional layer, the average pooling layer is used to down-sample the extracted features. The average pooling helps reduce the feature size and minimize the possibility of overfitting. At the flatten layer, the features extracted from the previous average pooling layer are flattened to form a fixed-dimensional vector, which is then fed into two dense layers. Finally, the SoftMax activation function computes the estimated probabilities of all the health classes. The final output is the class with the highest probability.

2.3 Training strategy

At the model initialization stage, the parameters of each convolutional layer and fully connected layer are initialized according to the Gaussian distribution with a mean of 0 and standard deviation of 0.01, and the bias values of all of the convolutional and dense layers are initialized to 0.

At the model training stage, the categorical cross-entropy function is adopted as the loss function. Given one training sample, the model performs forward propagation to generate classification output $\mathbf{y}^P = [y_1^p, y_2^p, \ldots y_C^p]$. The true label (after converting the categorical value into a binary vector by using one-hot encoding) is $\mathbf{y}^T = [y_1^T, y_2^T, \ldots y_C^T]$, then the classification loss is denoted as:

$$loss(\mathbf{y}^P, \mathbf{y}^T) = -\sum_{c=1}^{C} y_c^T \log(y_c^P) \tag{9}$$

where C is the total number of classes, $y_c^T \in \{0,1\}$ denotes whether the cth label is the true label, and y_c^P is the prediction probability towards label c. The batch loss, noted as $LOSS$, represents the prediction loss ($loss$ calculated by Eq. 9) averaged over batched samples. The Adam optimization algorithm is employed for model training. Like many other optimizers for deep learning models, the Adam optimizer utilizes the backpropagation of a classification loss to calculate its gradient at a specific combination of the trainable parameters (i.e., a vector of partial derivations of the loss with respect to the trainable parameters) [54]. For the feature weighting layer, in each training epoch, the update process of a_n^m and σ_n^m is shown as:

$$\begin{cases} a_n^m = a_n^m - \eta \dfrac{\partial LOSS}{\partial a_n^m} = a_n^m - \eta \dfrac{\partial LOSS}{\partial F_m} \dfrac{\partial F_m}{\partial a_n^m} \\ \sigma_n^m = \sigma_n^m - \eta \dfrac{\partial LOSS}{\partial \sigma_n^m} = \sigma_n^m - \eta \dfrac{\partial LOSS}{\partial F_m} \dfrac{\partial F_m}{\partial \sigma_n^m} \end{cases} \tag{10}$$

where η is the learning rate, ∂ is the partial derivative operator. According to Eq. 7, The partial derivatives of the weighting function F_m with respect to parameters a_n^m and σ_n^m are derived as:

$$\frac{\partial F_m}{\partial a_n^m} = \exp\left(-\frac{\left(l - n \cdot l_{\text{fault}}^m\right)^2}{2(\sigma_n^m)^2}\right) \tag{11}$$

$$\frac{\partial F_m}{\partial \sigma_n^m} = a_n^m \exp\left(-\frac{\left(l - n \cdot l_{\text{fault}}^m\right)^2}{2(\sigma_n^m)^2}\right) \frac{\left(l - n \cdot l_{\text{fault}}^m\right)^2}{(\sigma_n^m)^3} \tag{12}$$

The learning rate is set as 0.0005, and the number of training epochs is 100. The training batch size is set as 32 samples. In addition, 20% of the training dataset is randomly selected as validation data during model training to avoid overfitting. The "ModelCheckpoint" function in Keras is adopted, which evaluates the validation loss of an intermediately trained model at the end of each training epoch, saves the model when the validation loss is lower than the current minimum, and finally returns the model with the lowest validation loss.

2.4 Related work on attention and physics-informed feature extraction

In this section, we compare the proposed PICNN with related work. The motivation for designing a customized layer for the deep learning model is to enhance the model's generalizability. The attention mechanism has been applied in bearing diagnostics to improve the model's performance in obtaining discriminative fault-related features. Wang et al. [47] developed a *Multiattention 1D CNN*, where attention modules are designed to enhance discriminative features adaptively and suppress irrelevant features. The input features of the attention module are marked as an *N*-dimensional vector, $\mathbf{X} = [x_1, x_2, \dots, x_N]^{\mathrm{T}}$, where x_n, $n = 1, 2, \dots, N$, corresponds to the signal measurement at the nth temporal location. The temporal attention vector **W** is generated by:

$$\mathbf{W} = [w_1, w_2, \dots, w_N]^{\mathrm{T}} = \sigma\big(C(\mathbf{X})\big) \tag{13}$$

where $C(\cdot)$ is a 1×1 convolutional layer with one channel, and $\sigma(\cdot)$ is a sigmoid function.

Then, the temporal attention vector is used to weigh the input features. Also, the residual connection is introduced to prevent the reduction of feature response value in the attention module. The final output of this attention module is:

$$\mathbf{Y}_{\mathrm{Attention}} = \mathbf{W} \odot \mathbf{X} + \mathbf{X} \tag{14}$$

The attention vector **W** helps the model focus on the meaningful features. A common practice in determining these attention weights is to initialize them randomly and optimize them using backpropagation. These data-driven attention weights may help improve the diagnostic accuracy over no use of attention, but they still lack physical meaning.

Along the same line, but unlike using a data-driven attention module, Mohammad et al. [44] proposed incorporating physical knowledge of bearing faults by developing a physics-based (or, more appropriately, physics-informed) convolutional layer. A reference signal is generated using a model that simulates bearing fault physics; the reference signal is adopted as a physics-based convolution kernel to help reveal the fault-related information carried by the time-domain input signal. Li et al. [45] developed a customized convolutional layer, called the continuous wavelet convolutional layer, as the first layer of a modified CNN model. The waveform of kernels in the customized convolutional layer is constrained by the wavelet function, which guides the model to extract fault-related impact components from the raw vibration signal. A common characteristic of the methods mentioned above is that they design a customized layer for feature weighting or signal processing, optimized either purely based on data or a combination of data and physics.

Figure 4 summarizes the methods of constructing a physics-informed model by designing a customized layer. For methods (1) and (2), the models learn the physics of bearing fault through pure data-driven optimization. For methods (3) and (4), the physics of bearing fault is incorporated while initializing the model. Method (3) generates convolutional kernels with frequencies equal to bearing fault characteristic frequencies. Method (4) designs a weighting layer that assigns higher weights for features close to bearing fault characteristic frequencies. Compared to method (3), the proposed method performs feature engineering in the frequency domain rather than the time domain, with lower computational complexity, and can provide similar performance in highlighting fault-related features (see Appendix C for a proof of the equivalence between time-domain convolution and frequency-domain multiplication). Additionally, the design of the reference signal in the physics-based convolutional layer needs to manually set the amplitude, length of the signal, and damping coefficient. Here in the proposed method, the shape parameters in the feature weighting layer are optimized during the training of the PICNN model.

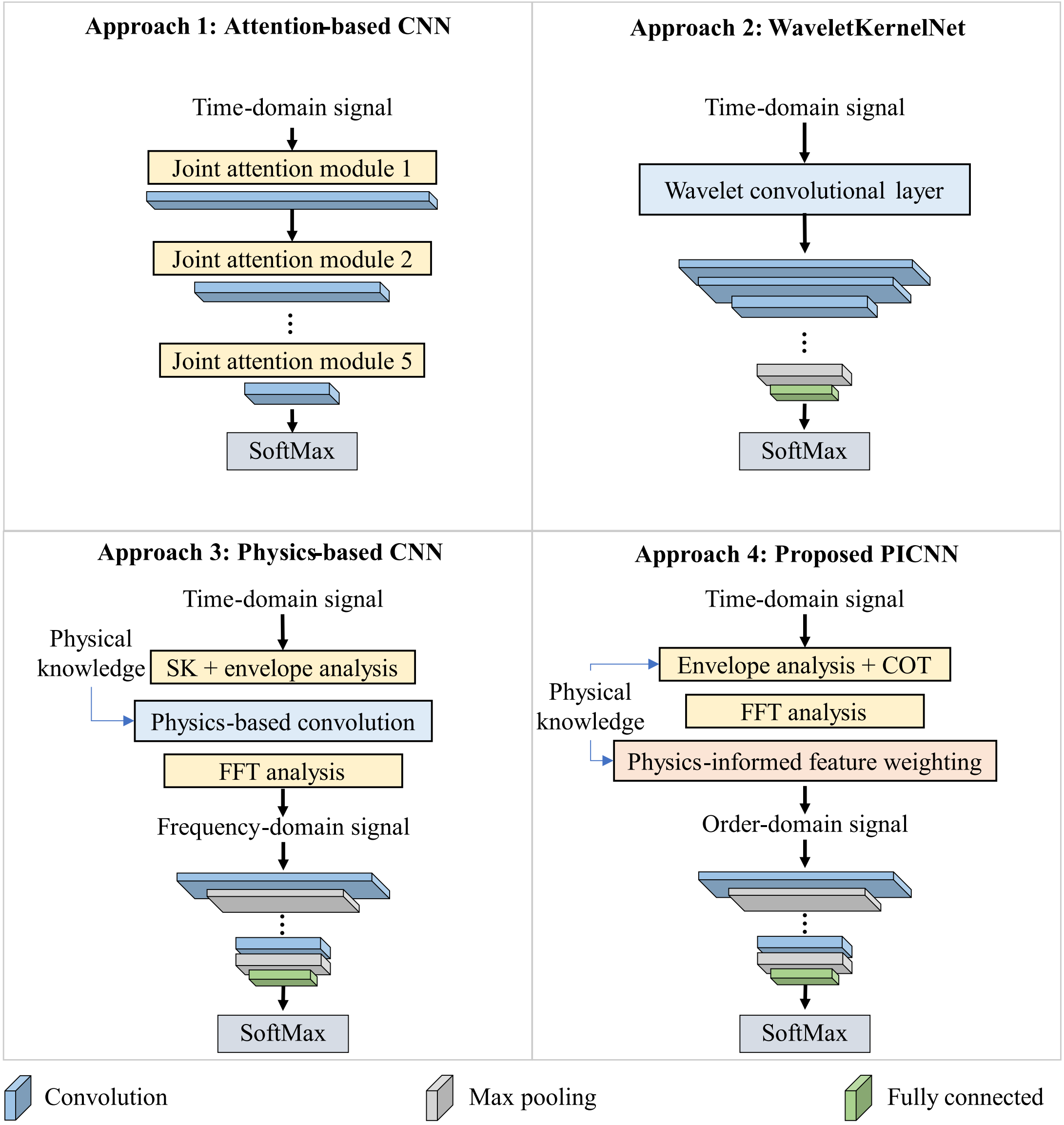


Figure 4. Four methods of constructing a physics-informed ML model by developing a customized layer: (1) Designing model architecture by adopting attention mechanism [47]; (2) Using wavelet convolutional kernel to extract interpretable output [45]; (3) Using SK and envelope analysis to highlight fault-related information, then design a physics-based convolutional layer [44]; and (4) The proposed physics-informed feature weighting model.

3. Case studies

We conduct a total of three case studies to demonstrate the effectiveness of the proposed method. Case study 1 uses an experimental dataset collected from a machinery fault simulator under different shaft speeds in our lab at Iowa State University. This dataset is named Iowa State University Machinery Fault Simulator (ISU-MFS) dataset. The ISU-MFS dataset and codes are provided at https://github.com/SalieriLu/BearingFaultDiagnostics. In the first case study, the training dataset is collected under rotational speeds that are different from the test dataset; we also examine the model's performance when there is significant noise interference. In the second case study, we evaluate the model's capability using data collected from an agricultural machine. During data collection, the rotational speeds are varied within the range of 21 to 54 Hz with a stable and an unstable stage. Case study 2 represents an effort to transition from a lab-based study to a field study with time-varying speeds. Finally, in case study 3, the open-source Case Western Reserve University (CWRU) bearing dataset is employed to compare the proposed method with other attention-based methods.

Two commonly used methods are introduced as benchmark methods for comparison. The methods are a) CNN and b) random forest.

a) CNN

We use a vanilla CNN as a benchmark method. The only difference between the vanilla CNN model and PICNN is that CNN does not have the feature weighting layer. Similar to the PICNN model, the categorical cross-entropy loss function is adopted to optimize model parameters.

b) Random forest

We also included a random forest model. Following the settings in [2], the random forest model uses 22 trees in the forest and has no limit to the maximum depth of the tree; this setting guarantees that each tree node will keep expanding until all leaves are pure (containing no more than 1 sample). The Gini impurity loss function is adopted to train the random forest model.

In addition to the benchmark methods listed above, PICNN is also compared with models introduced in section 2.4. A brief summary of the three case studies is given in Table 1.

Table 1. Summary of case studies

No.	Dataset name	Experimental platform	Total number of samples	Model output	Methods for comparison
1	ISU-MSF dataset	Machinery fault simulator	4,560	Bearing fault type	CNN, random forest, and PICNN
2	Agricutural machine dataset	Agricultural machine	34,168	Bearing fault severity	CNN, random forest, physics-based CNN [2], and PICNN
3	CWRU bearing dataset	Custom-built bearing test stand	2,000	Bearing fault type	CNN, random forest, attention-based CNN [52], and PICNN

3.1 Case study 1: ISU-MSF dataset

For case study 1, we evaluate the accuracy and robustness of the proposed method in real-world scenarios such as variation in rotational speed or interference from external noise sources. An experiment is carried out on a machinery fault simulator. As shown in Figure 5 (a), two bearings are mounted on the shaft of the simulator and driven by an electric motor. An accelerometer is mounted to the bearing housing to acquire vibration signals.

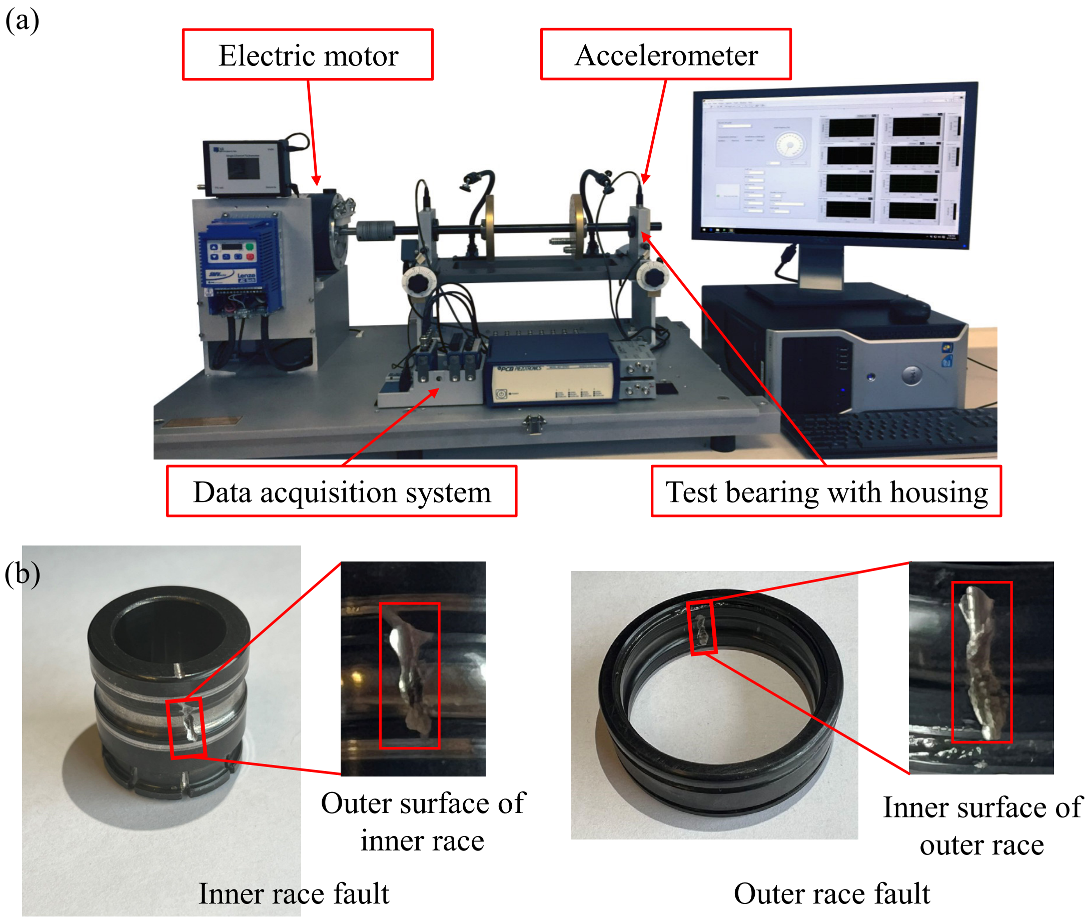


Figure 5. (a) Machinery fault simulator and (b) manually injected faults of an experimental bearing with combination faults

The bearings used in this experiment are rolling element bearings, each consisting of 13 balls. The inner ring, outer ring, and ball diameters of the bearing are 22.1 mm, 29.1 mm, and 3.5mm, respectively. In this case study, we collect data from several healthy/undamaged bearings as well as bearings with three types of defects (1) an inner race fault, (2) an outer race fault, and (3) a combination of faults. In Figure 5 (b), a bearing with combination faults is dissembled to show the manually introduced inner and outer race faults. These defects are pre-seeded into the bearings by electrical discharge machining, where the size of each fault is approximately 1.5 mm × 1.0 mm × 0.1 mm, which could represent the early stage of bearing degradation [44]. Based on Eqns. (1) and (2) mentioned in section 2, the characteristic frequency of inner race fault and outer race fault is $f_{\mathrm{IRD}} = 3.048 \times f_r$ and $f_{\mathrm{ORD}} = 4.950 \times f_r$, respectively.

The electric motor is operated such that the shaft rotational speeds vary from 15.5 Hz to 30 Hz in increments of 0.5 Hz. Four bearings, each with a different health class, are used to collect data. The 608-A11 accelerometer collects 20 s of vibration data from the bearings at a sampling frequency of 12.8 kHz. All collected data is then divided into train/test datasets based on the shaft rotational speed (see Table 2). Model training is performed on the dataset with integer shaft rotation speed, and a trained model is tested on fractional shaft rotational speed. The hypothesis in such a train/test split is to ensure that the test dataset is within the distribution of the training set and mimic more realistic testing scenarios where the test dataset does not have the exact shaft rotational speed as that of the training dataset. A total of 12,800 × 20 data points are collected for each bearing for each shaft speed. Then the vibration data are split into thirty-eight samples using a sliding window of stride length of 6,400. These samples are analyzed using the proposed signal processing approach of envelope order spectrum with the order range clipped between 0 to 16 (length = 1,600), which serves as the input for the deep learning model. In total, there are 38 × 4 × 15 = 2,280 samples for both training and test datasets.

Table 2. Data summary

Parameter	Value(s)
Shaft speed for training dataset (Hz)	16, 17, 18, …, 30
Shaft speed for test dataset (Hz)	15.5, 16.5, 17.5, …, 29.5
Bearing condition (health class)	Healthy, inner race fault, outer race fault, the combination of faults
Sampling rate (kHz)	12.8
Sampling time (s)	20
SNR settings (dB)	-12, -10, -8, -6, -4, -2, 0, 2 ,4, 6, 8, 10, 12, 16, 20

The objective of Case Study 1 is to demonstrate the positive effect of employing physics-informed signal processing as well as the physics-informed feature weighting layer. The fault classification accuracy of the proposed method is compared with that of a random forest and a vanilla CNN model. To establish the importance of using features in the order domain rather than the frequency domain, we also compare the PICNN model with another CNN model, named CNN-FFT, which uses envelope spectrum in the frequency domain as the input (ranging from 0 to 500 Hz, length = 1,600). To capture the effect of run-to-run variation, each model is independently trained five times with a randomized training-validation split ratio of 4:1.

3.1.1 Evaluation of physics-informed signal processing

In this section, we will first demonstrate the benefit of using order-domain features as input relative to using frequency-domain features. The proposed method (PICNN) and three benchmark methods (CNN-FFT, random forest, and CNN) are evaluated using the training and test datasets collected from the machinery fault simulator. We show the test dataset classification accuracy of all four methods in Table 3. First, when comparing CNN-based models, it can be observed that CNN-FFT produces the lowest diagnostic accuracy. The CNN and PICNN take the envelope order spectrum as input, while the input of CNN-FFT is trained on the frequency-domain data. This poor performance of the CNN-FFT model can be attributed to the shift in fault characteristic frequencies due to the change in rotational speed. These results suggest that the model learning is more robust when considering the speed invariant envelope order spectrum as the input.

Table 3. Diagnostic accuracy

Model	Mean accuracy (%)	Best accuracy (%)
CNN-FFT	79.23 ± 0.19	82.11
Random forest	99.40 ± 0.21	99.69
CNN	99.60 ± 0.18	99.78
PICNN	99.55 ± 0.16	**99.87**

To demonstrate the shift in fault frequencies, we plot in Figure 6 the order- and frequency-domain spectra of two samples collected from a bearing with an inner race defect. The two samples vary in rotational speed, with sample 1 collected at a rotational speed of 25.5 Hz and sample 2 collected at a rotational speed of 16.5 Hz. The shift of frequency peaks can be observed in Figure 6(a). This shift in the peaks can sometimes cause overlap with other bearing defects confusing the CNN-FFT model learning and resulting in low diagnostic accuracy. On the other hand, in Figure 6(b), the high amplitude features appear in the same order with no shift.

Note that, in Table 3, the test dataset is free of noise interference; the only challenge is to extract and identify fault-related features for samples collected at different speeds. Using order spectra eliminates shifts due to speed changes, making it easier for data-driven models to classify health correctly. As a result, all the methods that take order-domain features as input yield high diagnostic accuracy. However, test data are often gathered under strong noise interference in industrial applications. In the next section, we will evaluate the robustness of each method to varying levels of noise interference present in the test dataset.

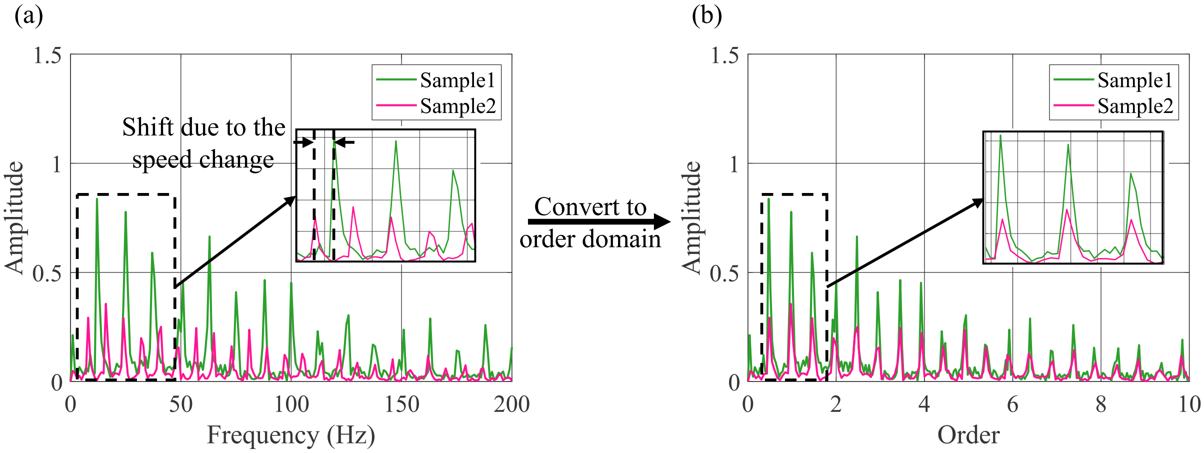


Figure 6. Comparison between (a) envelope frequency spectrum and (b) envelope order spectrum

3.1.2 Evaluation of physics-informed feature weighting

We will now evaluate the robustness of the proposed method in the presence of noise. To mimic real-world scenarios, we train random forest, CNN, and PICNN models on the original training dataset. After training, the model is then evaluated on the test dataset with the addition of noise. The degree of noise is parametrized by the signal-to-noise ratio (SNR), which is defined as [55]:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right) \tag{15}$$

where P_{signal} and P_{noise} are the average power of signal and noise, respectively. The range of SNR used in this study is listed in Table 2, and Figure 7 shows the variation of the diagnostic accuracy with SNR. Three observations can be made from Figure 7. First, when SNR is larger than 10 dB, the noise strength is relatively low, and the diagnostic accuracies of all the models are around 99% (as shown in Table 3). This is expected because the amount of noise is insignificant and doesn't alter the original test dataset. Second, with a decrease in SNR (i.e., an increase in the strength of noise), the accuracy of all three models decreases. The random forest model is most impacted by noise, with a sudden decrease in accuracy as the noise increases. PICNN and CNN deep learning models are relatively more robust to noise than random forest , but PICNN produces the highest diagnostic accuracy across the range of SNR values. Finally, PICNN outperforms CNN and random forest by showing the least run-to-run variation. The high standard deviation of the CNN model implies that the performance of CNN is unstable and hence not reliable. Note that the only difference between the PICNN model and the CNN model is that the CNN model does not contain the feature weighting layer. This comparison confirms that assigning a higher weight to fault-related features helps significantly improve the model's robustness in the presence of noise which is unavoidable in real-world applications.

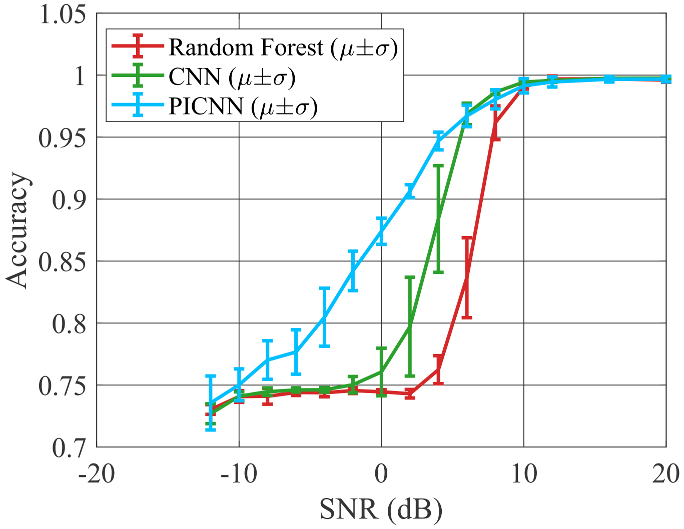


Figure 7. Classification accuracy under different noise levels

3.1.3 Model interpretability

PICNN achieves higher accuracy when the test data contains Gaussian noise. The only difference between PICNN and CNN is that PICNN contains the physics-informed feature weighting layer. In this section, we will explore how the inclusion of the physics-informed feature weighting layer helps improve the model performance. To do this, let us first consider the confusion matrix in Table 4 obtained from evaluating random forest, CNN, and the proposed PICNN models on the test dataset with the addition of Gaussian noise (SNR = 4dB).

Table 4. Confusion Matrix (when tested with SNR = 4 dB)

Item		Random forest				CNN				PICNN			
		Predicted health class											
		🟩	🟧	🟪	🟥	🟩	🟧	🟪	🟥	🟩	🟧	🟪	🟥
True health class	H 🟩	38	0	532	0	300	0	270	0	566.6	0	3.4	0
	IR 🟧	0	561	0	9	0	563.4	0.2	6.4	0	558.8	0	11.2
	OR 🟪	0.8	0	569.2	0	18.2	0	551.8	0	104	0	466	0
	Comb 🟥	0	0	0	570	0	0	3.2	566.8	0	0	0.2	569.8
Accuracy		76.24%				86.93%				94.79%			

From the confusion matrix in Table 4, it can be observed that both the CNN and the random forest model misclassify a high portion (> 50%) of the healthy samples as containing an outer race fault, while the PICNN provides > 99% accuracy for healthy test samples. However, the PICNN model misclassifies 18% of samples with outer race fault as healthy. These results indicate that the presence of noise makes the clear distinction between outer race fault and healthy bearings a challenge (although the PICNN generally outperforms the other two approaches). In other words, for this dataset, correct classification between healthy bearings and bearing with outer race fault determines which of the three models has the best overall accuracy. Upon the introduction of Gaussian noise, the fault-related features in the test dataset might be masked, effectively reducing the visibility of the corresponding fault signatures. In such scenarios, the CNN and random forest provide false-positive results. On the contrary, the presence of a feature weighting layer in PICNN helps the model to pay more attention to the range of frequencies pertaining to bearing faults. Unless a clear fault pattern is observed, PICNN classifies the bearing to be healthy. Although this can sometimes lead to false negatives, the false positives of the overly sensitive CNN and random forest models outweigh the false negatives of the PICNN model, making PICNN have the best diagnostic accuracy.

Let us now focus on CNN and PICNN models and understand the causes of misclassification. Fundamentally, each layer of the CNN and PICNN models acts as a feature extractor that extracts the most relevant information to be passed on to the next layer. The quality of the extracted features significantly affects the diagnostic performance. t-distributed stochastic neighbor embedding (t-SNE) is used to visualize the extracted features in the last convolutional layer of both models. The t-SNE method embeds the high-dimensional features into two-dimensional space, and the distance between samples in the t-SNE plot indicates the similarity between samples. Figure 8 shows the t-SNE plot of training and testing data without any external noise. The training samples of the four classes are shown as "+", and the test samples are shown as "O". The misclassified test samples are specifically highlighted with two black edge semicircles: the left semicircle showing the true class and the right semicircle showing the predicted health class for the sample. Physics-informed signal processing helps pool similar class samples irrespective of the shaft rotational speed. For both CNN and PICNN models, the t-SNE plots formed by the features extracted from the last convolutional layer have distinct boundaries for all the classes, which indicates that the deep learning models are successfully differentiating across different bearing health conditions. Moreover, the distributions of test and training data are similar, as observed by the proximity of training and test samples for each class. This successful differentiation between classes explains why both CNN and PICNN models achieved close to 99% classification accuracy.

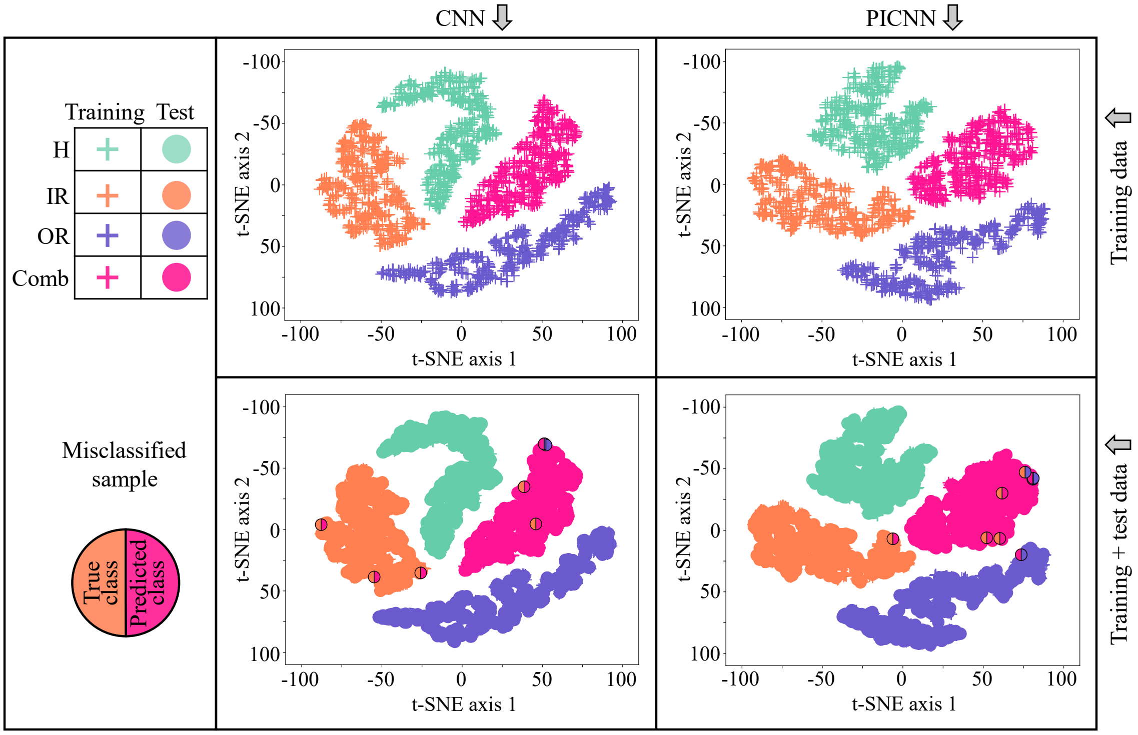


Figure 8. t-SNE visualization for extracted features from clean training and test data

Adding Gaussian noise to the test data (SNR = 4 dB) slightly affects the CNN model's ability to have clear boundaries between different classes, especially of the healthy bearing class, as shown in Figure 9. In the case of the CNN model, the healthy training samples are clustered into two regions, with one region close to the training sample distribution for the outer race fault. When evaluating the model using the noisy test set, the healthy test samples located between the two healthy training sample regions are misclassified as having outer race fault. This is due to the proximity of the outer race training sample distribution confusing the model's distinction across class boundaries. In the case of the PICNN model, all the healthy training samples are clustered into one region, and the healthy test samples are offset from the training samples due to the addition of noise. The majority of misclassified outer race test samples are located in the healthy test sample region, indicating that the misclassification is due to the lack of distinguishable features. Hence, PICNN classifies those samples as healthy samples.

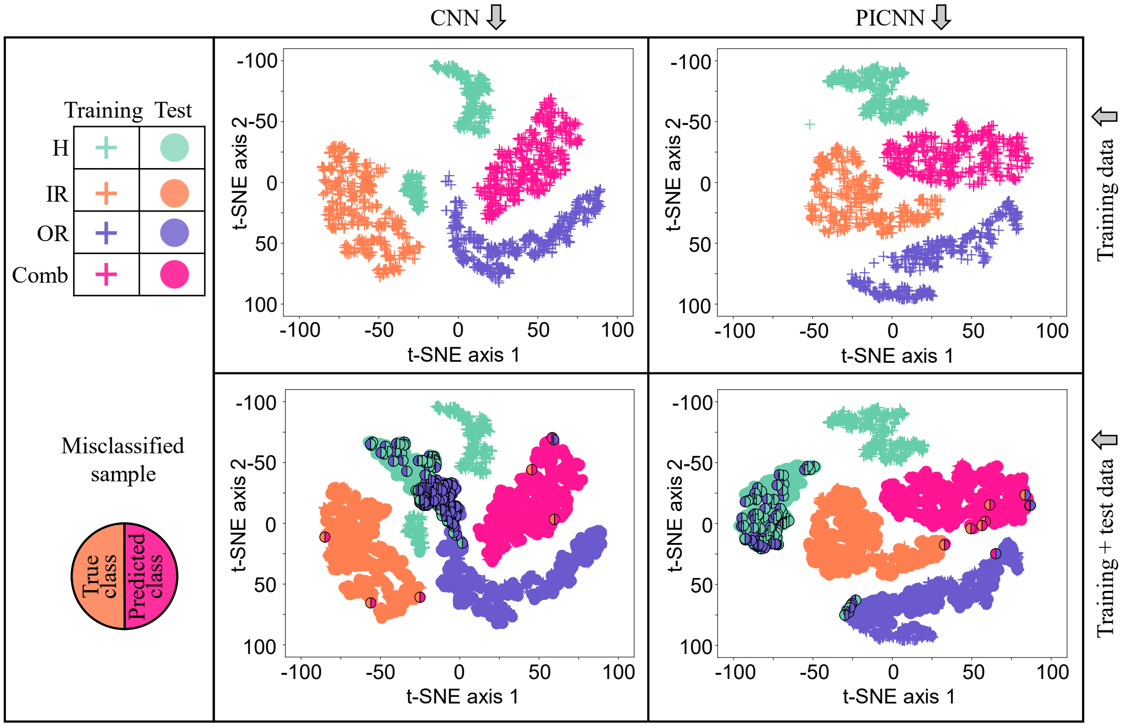


Figure 9. t-SNE visualization for extracted features from training and test data (SNR = 4 dB)

To further gain insight into which order-domain features are activated during prediction, the Gradient-weighted class activation mapping (Grad-CAM) is used to visualize the relative importance of all the input features when predicting a particular class. The Grad-CAM uses the gradient information that flows into the target convolutional layer to assign importance values to each extracted feature regarding target prediction results [56]. Combining the Grad-CAM localizations with the original input provides interpretable visual explanations for model predictions.

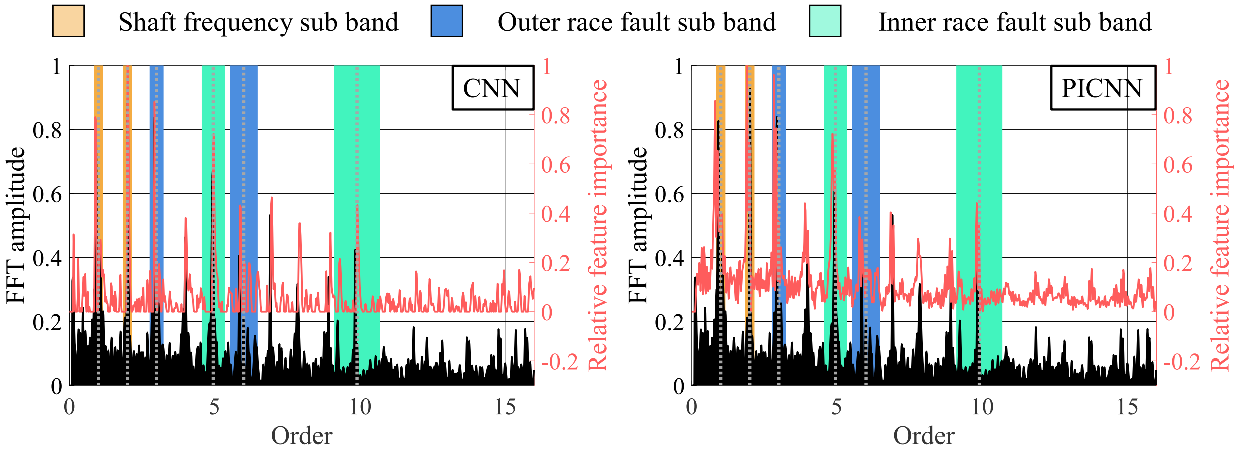


Figure 10. Relative importance value for CNN and PICNN Gaussian when the target label is set as a combination fault

A test sample with a true class of combination of faults is shown in Figure 10 along with the Grad-CAM derived feature importance when the target label is set as a combination of faults. As the bearing has a combination of faults, large FFT amplitudes are seen both at the inner race fault frequency and outer race fault frequency (along with their harmonics). Additionally, due to the inner race defect, the bearing suffers from shaft unbalance, leading to large FFT amplitudes in the synchronous fault regions of orders 1, 2, and 3. It can be observed that both CNN and PICNN models assign higher feature importance to these fault-related features; the high amplitude fault features are captured by the two models, eventually leading to correct classification.

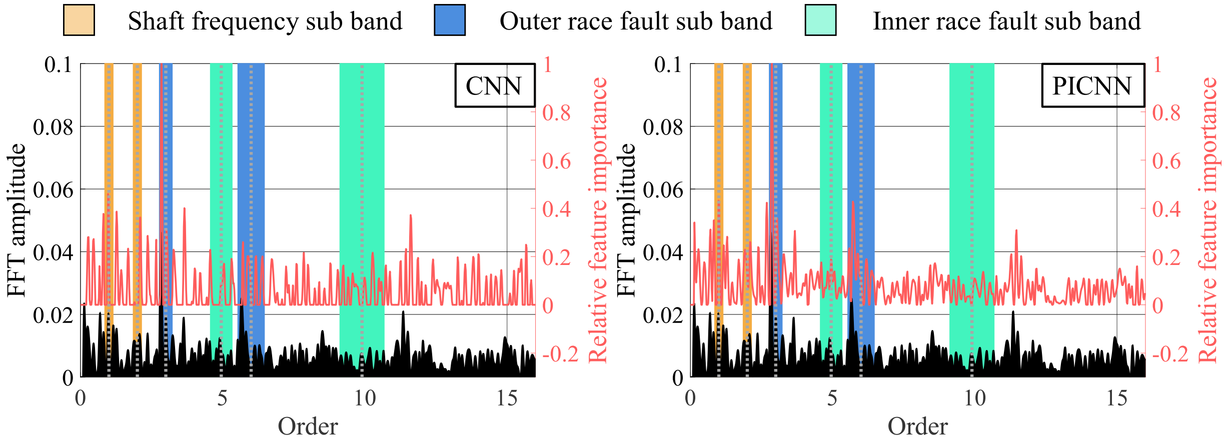


Figure 11. Relative importance value for CNN and PICNN Gaussian when the target label is set as outer race fault (clean test data)

Next, a test sample with a true class of outer race fault is shown in Figure 11, along with the Grad-CAM derived feature importance when the target label is set as outer race fault. The selected sample has outer race fault; therefore, a large FFT amplitude is seen at the outer race fault frequency and the first harmonic. CNN and PICNN assign higher feature importance to the outer race fault frequency. Note that PICNN also assigns higher importance to the feature located at the first harmonic (at order = 6.1).

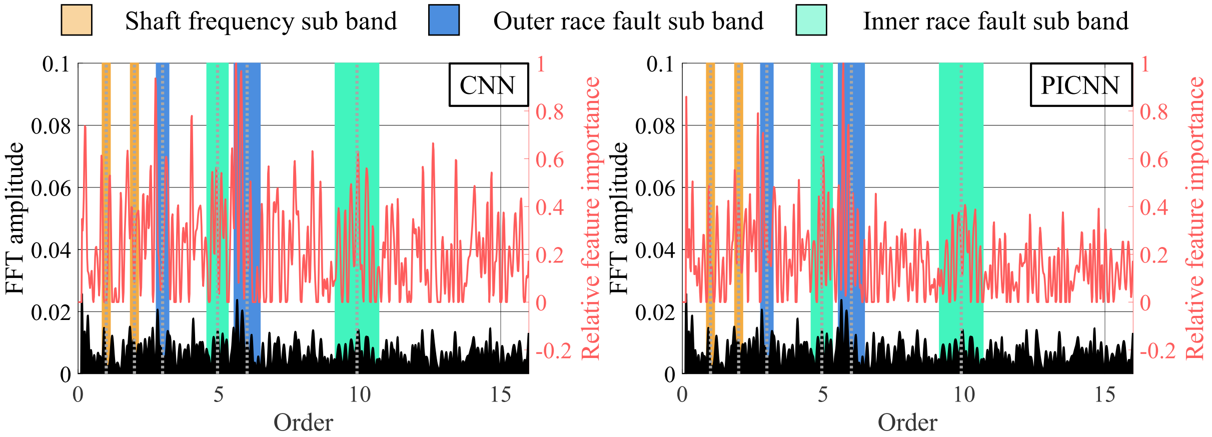


Figure 12. The relative importance value for CNN and PICNN Gaussian when the target label is set as outer race fault (noisy test data)

Finally, the noisy test sample with a true class of outer race fault is selected, and the Grad-CAM results when the target label is set as outer race fault is visualized in Figure 12. It is difficult to find the fault features on the envelope spectrum. Although CNN gives the correct classification results, the importance values of outer race fault-related features are close to other features. The CNN model provided outer race fault results by considering the overall amplitude of the features. This way, samples with high overall amplitude are classified as outer race fault. This inference is consistent with the fact that the majority of healthy samples in Table 4 are classified as outer race fault. For the PICNN model, the importance values are higher around the outer race fault frequency. When classifying the sample as outer race fault, PICNN is more interested in the feature values around the outer race fault frequency and its harmonics. As the feature amplitude located at the outer race fault sub-band does not significantly differ from the overall amplitude, the PICNN rejects classifying the sample as outer race fault. With the help of the physics-informed feature weighting layer, the PICNN is more sensitive to the amplitude of fault-related features than the overall amplitude. This helps the model avoid false alarm prediction results.

3.2 Case study 2: Model evaluation using agricultural machine dataset

In addition to evaluating the models' ability to classify fault types using laboratory data, we also evaluate PICNN's ability to identify bearing fault severity using data collected from experiments on an agricultural machine [2]. The main motivation for case study 2 is to mimic the bearing fault diagnostics in a real-world scenario. An accelerometer with a sampling frequency of 25,600 Hz is mounted close to the hydraulic motor, which contains a bearing inside. We collected the training and test data when operating the agricultural machine under various speed settings.

In this case study, the faults are pre-seeded into the bearings by introducing shallow peak milling slot cuts into the surface of bearing components. Three bearing fault types are considered: inner race fault, outer race fault, and roller fault. Figure 13 shows a manually introduced outer race fault. Two damage severity levels are designed for each fault type, leading to 12 bearings pre-seeded with faults. Six healthy bearings with no faults are also included in this study. These 18 bearings are assembled onto hydraulic motors, and the shaft rotational speeds are varied from 21 Hz to 54 Hz. Further details on this experiment/dataset can be found in [2].

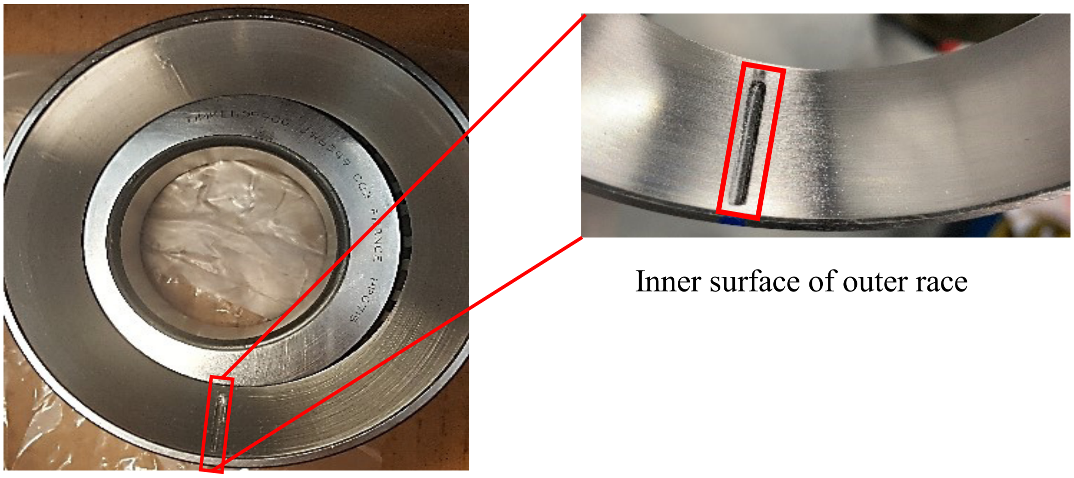


Figure 13. The outer race fault (single-point peck milling slot) of an experimental bearing in case study 2 [2].

One advantage of the proposed diagnostic method is that it can easily be implemented into a CNN-based deep learning model without significantly modifying its architecture. Case study 2 is a bearing fault detection problem, and the model output is the bearing damage severity (healthy, light damage, or severe damage). For this case study, the PICNN model is developed by adding the feature weighting layer to the CNN model developed in [2]. After five independent runs, the prediction results are averaged and are shown in Table 5.

Table 5. Classification accuracy results for agricultural machine dataset

Model	Input size	Layer	Mean accuracy (%)				
			Test 1	Test 2	Test 3	Test 4	Test 5
SVM	[1, 873]	–	93.29	92.28	90.34	98.77	99.63
Random forest	[1, 873]	–	89.32	92.85	82.47	98.78	98.76
CNN	[1, 2000]	5 Conv + 1 DC	97.75	**97.93**	90.40	98.88	99.72
Physics-informed deep learning [2]	[1, 873]	5 Conv + 1 DC	**99.19**	96.23	92.23	99.38	99.95
PICNN	[1, 2000]	Feature weighting + 5 Conv + 1 DC	99.13	97.58	**92.27**	**99.46**	**99.97**

The PICNN model in Table 5 is designed by adding a physics-informed feature weighting layer to the CNN model in [2]. The PICNN model yields more than 90% accuracy for all five tests, which indicates the proposed physics-informed feature weighting method applies to time-varying operating conditions. Compared to CNN, PICNN yields higher classification accuracy for tests 1, 3, 4, and 5, and the results for test 2 show comparable accuracy. This performance improvement is achieved by adding a physics-informed feature weighting layer in front of the first layer of the CNN model. Adding this feature weighting layer can be treated as an easy-to-implement solution for incorporating physical knowledge into deep learning models. The physics-informed deep learning approach incorporates physical knowledge by taking features located at fault-related sub-bands as input; features outside the predefined sub-bands are not considered. In contrast, PICNN takes all the features from order 0 to 16, and the features located farther from the fault-related sub-bands are assigned with lower attention weights, due to which the PICNN achieves slightly higher accuracy than the physics-informed deep learning approach. Also, the physics-informed deep learning approach incorporates physical knowledge by using a physics-informed loss function [2], the design of which requires statistical analysis of training data. The proposed PICNN method incorporates physical knowledge by assigning higher weights to more fault-related input features. With the help of backpropagation optimization, the proposed method optimizes feature weights automatically. The results of this field study show that the proposed method has the potential for deployment in industrial settings.

3.3 Case study 3: Model evaluation using CWRU bearing dataset

As a standard reference in the bearing diagnostic field, the CWRU dataset has been widely used to evaluate diagnostic models. The CWRU dataset contains vibration signals collected from healthy bearings, inner-race-fault bearings, roller-fault bearings, and outer-race-fault bearings. Each bearing fault type has three different fault severities.

We compare PICNN with the Deep neural network for Domain Adaptation in Fault Diagnostic (DAFD) model proposed in Ref. [57] and the attention-based algorithm presented in Ref. [52]. The experimental setting follows the description in Refs. [45] and [57], where the labeled data under 0 hp load are used for training, then the data collected under 3 hp load are used to test the model. Both training and test datasets contain 1,000 samples with a sample length of 1,200. Two different diagnostic tasks are designed depending on whether the bearing failure severity is considered. When only considering the bearing fault type, the number of fault classes is four, and if the model also considers the bearing fault severity, the number of health classes is ten. The results are summarized in Table 6.

Table 6. Comparison of fault diagnostic accuracies using CWRU dataset

No. of health classes	Method	Accuracy
4	SVM	82.28%
	CNN	90.60%
	DAFD* [57]	**94.73%**
	PICNN	93.62%
10	SVM	80.50%
	CNN	84.10%
	Attention-based LSTM+CNN [52]	**91.54%**
	PICNN	91.27%

* Test data (unlabled) available during model training

For the experiment with only four different health classes, the proposed PICNN model performs better than CNN and achieves comparable accuracy compared to DAFD. However, note that for DAFD, unlabeled test data is required during model training. With the help of signal processing and physics-informed feature weighting, PICNN generates comparable results without using test data for model training.

For the more complex experiment with ten different health classes, the performance of the PICNN model is similar to the attention-based LSTM + CNN model. Note that PICNN is composed of feature weighting layer, convolutional layers, and dense layers, while the attention-based LSTM + CNN model contains convolutional layers, LSTM layer, attention layer, and dense layers. The model architecture of PICNN is less complicated than the attention-based model, and PICNN converges within 100 epochs while the attention-based LSTM + CNN takes 5,000 epochs.

4. IIoT implementation for online diagnostics

In Section 3, the proposed PICNN model is compared with other methods via offline tests. In Section 4, we bridge the gap between offline evaluation and the online implementation of diagnostic models, and we focus on the practical implementation of this model inside an IIoT edge device.

Although there have been significant advances recently in the field of bearing diagnostics, very little work focused on practical implementation. As a result, most diagnostics research evaluates models using pre-collected test data evaluated offline using high-performance computers. However, outside of certain high-value assets, it is generally not cost-effective for sensors monitoring each piece of rotating equipment to collect and transmit long time history records for analysis using a high-performance computer.

In the era of the IIoT, one alternative approach to this type of centralized model diagnostics is to use battery-powered data acquisition devices to collect data and then wirelessly transmit that data to the cloud. A deep learning model, deployed in the cloud, can then make predictions and send results both to system users and back to the local device (for model updating purposes) [58]. However, in the case of bearing diagnostics, the required transmission of raw vibration data to the cloud can drain available wireless bandwidth and increase overall power consumption, which in turn can negatively affect scalability, battery life, and operating costs [50]. As a potential solution to these limitations, edge computing has become an important technique for IIoT services, where an embedded system performs diagnostic tasks locally at the data source. The advantages of edge computing are not limited to reducing data transmission costs, as it can also provide real-time evaluation results, preserve a user's data privacy, and increase battery life.

This section presents an IIoT deployment of the proposed method for online bearing diagnostics. Here, the diagnostic model is implemented on a commercial wireless sensing node consisting of an accelerometer, a wireless radio, and two microprocessors (Figure 14). Both the signal processing algorithm and the model are written in C and executed within the ATSAMG55 processor.

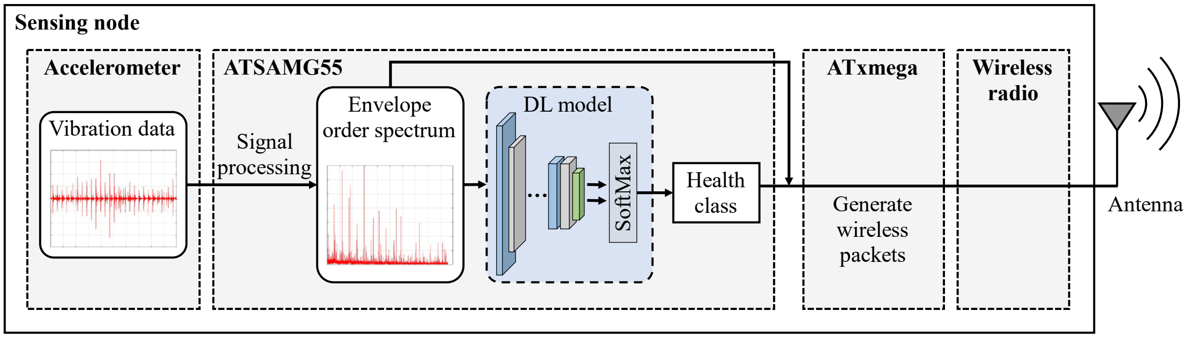


Figure 14. The architecture of the embedded diagnostic system

Compared to a high-performance computer, most embedded systems have limited memory. For example, the ATSAMG55 chip used here has 512 KB of program memory (used for storing program code) and 176 KB of data memory (used for storing program data). The proposed PICNN model contains 17,388 trainable parameters, and saving those model parameters consumes ≈ 70 KB of data memory. Additionally, the outputs of each layer consume ≈ 69 KB of data memory. Since embedded systems contain significantly more functionality than just PICNN model functions, these numbers help illustrate the extent to which memory optimization is needed to effectively deploy our model on the edge.

Traditionally, the output of each layer of a model would be computed and saved to an output buffer in memory. Then, if that output is required as an input to the next layer, the result would be accessed from memory. Figure 15(a) illustrates a naive method for memory allocation over two dimensions, time and memory size. In this naïve memory allocation strategy, memory usage increases over time as the outputs of each layer are saved. However, this type of consumption can be optimized by using the bin packing technique shown in Figure 15 (b), where layer outputs are erased from memory after these outputs are used for the next operation that requires them, and then they are reallocated for the output of the next layer [59]. We deploy this bin-packing strategy in this Section so that proposed method can be deployed into the sensing node.

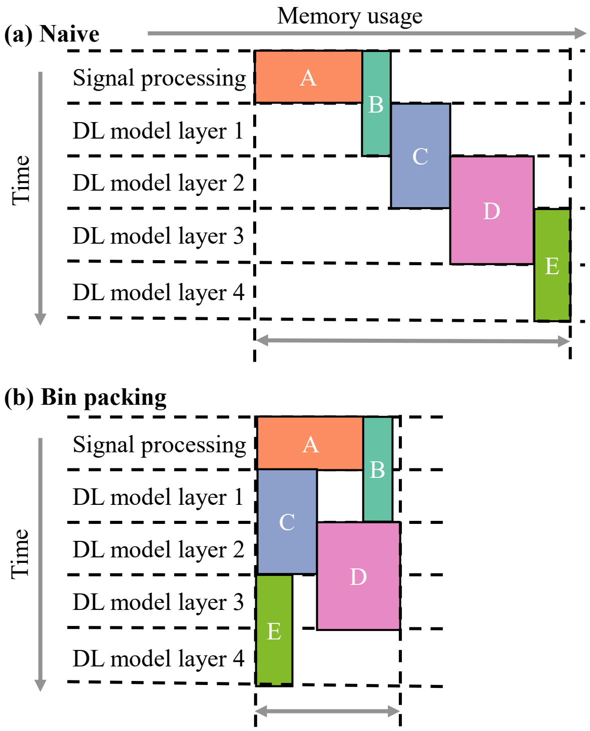
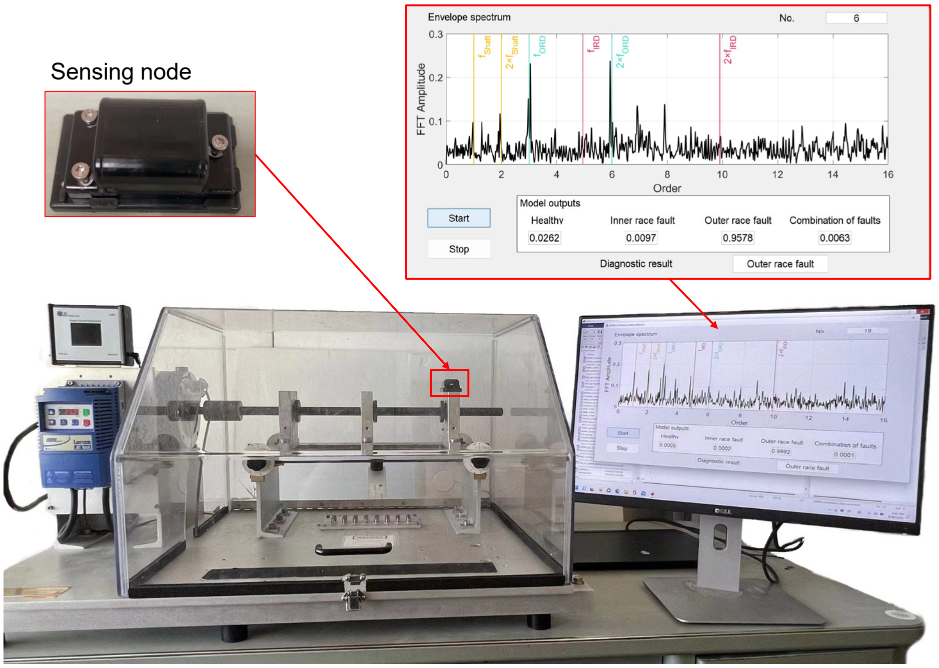


Figure 15. Memory allocation strategies

A preliminary version of this work was presented at the 2022 International Symposium on Flexible Automation Conference [60], where we demonstrated our online diagnostics approach using a commercially available wireless sensing node deployed on a vibration shaker. In this paper, our embedded diagnostic algorithm is evaluated on a machinery fault simulator (shown in Figure 5, section 3.1). As shown in Figure 16, a wireless sensing node is mounted to the bearing housing, gathering a vibration signal and identifying the bearing health condition in an online manner using both CNN and PICNN approaches, then sending diagnostic results and the order spectrum to the server.



Envelope spectrum

No. 6

Model outputs

Healthy	Inner race fault	Outer race fault	Combination of faults
0.0262	0.0097	0.9578	0.0063

Diagnostic result: Outer race fault

Figure 16. Experimental setup for online bearing diagnostics

Healthy bearings and bearings with inner race faults and outer race faults are used to evaluate each model's performance. Three speed settings, 15 Hz (i.e., 900 RPM), 20 Hz (1,200 RPM), and 25 Hz (1,500 RPM), are considered for this online diagnostic test. We record the diagnostic results provided by the embedded model ten times for each bearing test. Furthermore, after the server receives extracted features and diagnostic results from the embedded analysis, offline Keras model results are generated for the same collected data. Results are shown in Table 7.

Table 7. Diagnostic results by embedded and Keras implementations of CNN and PICNN

		CNN						PICNN					
Item		Embedded model			Keras model			Embedded model			Keras model		
		Predicted health class											
		🟩	🟧	🟪	🟩	🟧	🟪	🟩	🟧	🟪	🟩	🟧	🟪
True health class	H 🟩	28	0	2	28	0	2	30	0	0	30	0	0
	IR 🟧	0	30	0	0	30	0	0	30	0	0	30	0
	OR 🟪	0	0	30	0	0	30	1	0	29	1	0	29
Accuracy		97.78%			97.78%			**98.89%**			**98.89%**		

The results of the embedded system are identical to those of the Keras model, indicating that the diagnostic models have been successfully deployed in the embedded system. It is important to note that, compared to the offline evaluation performed in section 3.1, the online diagnostic implementation uses a different, lower-cost accelerometer. Thus, the SNR of online test data is lower. As a result, the classification accuracies decreased, the CNN model misclassified two healthy bearings as outer race fault bearings, and the PICNN misclassified one bearing with an outer race fault as healthy. Still, both CNN and PICNN provide classification accuracy higher than 95%. Relative to CNN, the PICNN model shows better accuracy regarding healthy bearings and is more conservative in identifying bearing faults, consistent with the discussion in section 3.1.3.

This work shows that for industrial implementations, IIoT devices can successfully use embedded models to perform diagnostics locally and can optimize wireless bandwidth and battery life by only sending classification results (1 byte data) to the server. This is a huge saving relative to centralized approaches that rely on Keras models to perform diagnostics on raw vibration signals (4×1600 bytes data).

5. Conclusion

This study presents a physics-informed feature weighting method to solve the model deterioration caused by the distribution difference between training and test data. This method first processes a vibration signal to obtain an envelope order spectrum. Then, the PICNN model, formed by a feature weighting layer and CNN, is used to predict the bearing health class. The proposed method has two desirable characteristics: (1) the extracted order envelope spectrum is robust to the speed variation, and (2) similar to the attention mechanism, the feature weighting layer assigns higher weights to discriminative fault features. The physical knowledge is incorporated by adding constraints to the distribution of feature weights.

The effectiveness of the PICNN is verified using data collected from a machinery fault simulator in a lab, an agricultural machine operating in the field, and a bearing test stand at the CWRU Bearing Data Center. The proposed model has the following advantages: (1) robust to the change of rotational speeds and SNR - compared to a vanilla CNN, the PICNN is more sensitive to changes in fault-related features and has less chance of a false alarm; and (2) easy to implement, the PICNN can easily be implemented in other models by adding the weighting layer as the model's first layer.

An online diagnostics implementation of the deep learning model is also included in this research. The signal processing and diagnostic algorithms are deployed on an IIoT device. The embedded model provides identical results to the Keras model that runs on the server.

The proposed method targets practical diagnostic problems in which test data is collected under different operating states and SNR settings. It is worth mentioning that in the industrial environment, the vibration of other mechanical components will generate vibration signals with a fixed frequency. These signals may be mixed in the vibration data collected from the bearing. A future research direction is investigating how to perform efficient and accurate diagnostics in these realistic industrial settings.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Grants IIP-1919265 and IIP-2036044. Any opinions, findings, or conclusions in this paper are those of the authors and do not necessarily reflect the sponsor's views.

References

[1] Z. Xu, C. Li, Y. Yang, Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks, Applied Soft Computing, 95 (2020) 106515.

[2] S. Shen, H. Lu, M. Sadoughi, C. Hu, V. Nemani, A. Thelen, K. Webster, M. Darr, J. Sidon, S. Kenny, A physics-informed deep learning approach for bearing fault detection, Engineering Applications of Artificial Intelligence, 103 (2021) 104295.

[3] S. Sampath, Y. Li, S. Ogaji, R. Singh, Fault diagnosis of a two spool turbo-fan engine using transient data: A genetic algorithm approach, Turbo Expo: Power for Land, Sea, and Air, 2003, pp. 351-359.

[4] S. Nandi, H.A. Toliyat, X. Li, Condition monitoring and fault diagnosis of electrical motors—A review, IEEE transactions on energy conversion, 20 (2005) 719-729.

[5] F. Elasha, M. Greaves, D. Mba, Planetary bearing defect detection in a commercial helicopter main gearbox with vibration and acoustic emission, Structural Health Monitoring, 17 (2018) 1192-1212.

[6] M.S. Raghav, R.B. Sharma, A review on fault diagnosis and condition monitoring of gearboxes by using AE technique, Archives of Computational Methods in Engineering, 28 (2021) 2845-2859.

[7] S.J. Kim, K. Kim, T. Hwang, J. Park, H. Jeong, T. Kim, B.D. Youn, Motor-current-based electromagnetic interference de-noising method for rolling element bearing diagnosis using acoustic emission sensors, Measurement, 193 (2022) 110912.

[8] D.T. Hoang, H.J. Kang, A motor current signal-based bearing fault diagnosis using deep learning and information fusion, IEEE Transactions on Instrumentation and Measurement, 69 (2019) 3325-3333.

[9] H. Wang, Z. Liu, D. Peng, Z. Cheng, Attention-guided joint learning CNN with noise robustness for bearing fault diagnosis and vibration signal denoising, ISA transactions, (2021).

[10] K. Feng, J. Ji, Q. Ni, M. Beer, A review of vibration-based gear wear monitoring and prediction techniques, Mechanical Systems and Signal Processing, 182 (2023) 109605.

[11] I. Bediaga, X. Mendizabal, A. Arnaiz, J. Munoa, Ball bearing damage detection using traditional signal processing algorithms, IEEE Instrumentation & Measurement Magazine, 16 (2013) 20-25.

[12] R. Yao, H. Jiang, X. Li, J. Cao, Bearing incipient fault feature extraction using adaptive period matching enhanced sparse representation, Mechanical Systems and Signal Processing, 166 (2022) 108467.

[13] Q. Ni, J. Ji, K. Feng, B. Halkon, A fault information-guided variational mode decomposition (FIVMD) method for rolling element bearings diagnosis, Mechanical Systems and Signal Processing, 164 (2022) 108216.

[14] B. Li, M.-Y. Chow, Y. Tipsuwan, J.C. Hung, Neural-network-based motor rolling bearing fault diagnosis, IEEE transactions on industrial electronics, 47 (2000) 1060-1069.

[15] D. Dou, J. Yang, J. Liu, Y. Zhao, A rule-based intelligent method for fault diagnosis of rotating machinery, Knowledge-Based Systems, 36 (2012) 1-8.

[16] W. Mao, W. Feng, Y. Liu, D. Zhang, X. Liang, A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis, Mechanical Systems and Signal Processing, 150 (2021) 107233.

[17] M. Sadoughi, H. Lu, C. Hu, A Deep Learning Approach for Failure Prognostics of Rolling Element Bearings, 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2019, pp. 1-7.

[18] X. Yan, M. Jia, A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing, Neurocomputing, 313 (2018) 47-64.

[19] J. Lu, W. Qian, S. Li, R. Cui, Enhanced K-nearest neighbor for intelligent fault diagnosis of rotating machinery, Applied Sciences, 11 (2021) 919.

[20] G. Xu, M. Liu, Z. Jiang, D. Söffker, W. Shen, Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning, Sensors, 19 (2019) 1088.

[21] X. Zhang, B. Wang, X. Chen, Intelligent fault diagnosis of roller bearings with multivariable ensemble-based incremental support vector machine, Knowledge-Based Systems, 89 (2015) 56-85.

[22] J. Tian, C. Morillo, M.H. Azarian, M. Pecht, Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with K-nearest neighbor distance analysis, IEEE Transactions on Industrial

Electronics, 63 (2015) 1793-1803.
[23] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mechanical Systems and Signal Processing, 138 (2020) 106587.
[24] S. Khan, T. Yairi, A review on the application of deep learning in system health management, Mechanical Systems and Signal Processing, 107 (2018) 241-265.
[25] K. Feng, J. Ji, Y. Zhang, Q. Ni, Z. Liu, M. Beer, Digital twin-driven intelligent assessment of gear surface degradation, Mechanical Systems and Signal Processing, 186 (2023) 109896.
[26] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9 (1997) 1735-1780.
[27] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, (2014).
[28] D. Palaz, R. Collobert, M.M. Doss, Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks, arXiv preprint arXiv:1304.1018, (2013).
[29] N. Aloysius, M. Geetha, A review on deep convolutional neural networks, 2017 international conference on communication and signal processing (ICCSP), IEEE, 2017, pp. 0588-0592.
[30] M. Qiao, S. Yan, X. Tang, C. Xu, Deep convolutional and LSTM recurrent neural networks for rolling bearing fault diagnosis under strong noises and variable loads, Ieee Access, 8 (2020) 66257-66269.
[31] Q. Jiang, F. Chang, B. Sheng, Bearing fault classification based on convolutional neural network in noise environment, IEEE Access, 7 (2019) 69795-69807.
[32] K. Zhao, H. Jiang, K. Wang, Z. Pei, Joint distribution adaptation network with adversarial learning for rolling bearing fault diagnosis, Knowledge-Based Systems, 222 (2021) 106974.
[33] X. Li, W. Zhang, Q. Ding, J.-Q. Sun, Multi-layer domain adaptation method for rolling bearing fault diagnosis, Signal processing, 157 (2019) 180-197.
[34] J. Zhu, N. Chen, C. Shen, A new multiple source domain adaptation fault diagnosis method between different rotating machines, IEEE Transactions on Industrial Informatics, 17 (2020) 4788-4797.
[35] D. Wei, T. Han, F. Chu, M.J. Zuo, Weighted domain adaptation networks for machinery fault diagnosis, Mechanical Systems and Signal Processing, 158 (2021) 107744.
[36] V. Dwivedi, N. Parashar, B. Srinivasan, Distributed physics informed neural network for data-efficient solution to partial differential equations, arXiv preprint arXiv:1907.08967, (2019).
[37] A.D. Jagtap, G.E. Karniadakis, Extended Physics-informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition based Deep Learning Framework for Nonlinear Partial Differential Equations, AAAI Spring Symposium: MLPS, 2021.
[38] X.Y. Lee, J.R. Waite, C.-H. Yang, B.S.S. Pokuri, A. Joshi, A. Balu, C. Hegde, B. Ganapathysubramanian, S. Sarkar, Fast inverse design of microstructures via generative invariance networks, Nature Computational Science, 1 (2021) 229-238.
[39] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational physics, 378 (2019) 686-707.
[40] Y. Ye, H. Fan, Y. Li, X. Liu, H. Zhang, Deep neural network methods for solving forward and inverse problems of time fractional diffusion equations with conformable derivative, Neurocomputing, (2022).
[41] G. Pang, L. Lu, G.E. Karniadakis, fPINNs: Fractional physics-informed neural networks, SIAM Journal on Scientific Computing, 41 (2019) A2603-A2626.
[42] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next, arXiv preprint arXiv:2201.05624, (2022).
[43] W. Peng, J. Zhang, W. Zhou, X. Zhao, W. Yao, X. Chen, IDRLnet: A physics-informed neural network library, arXiv preprint arXiv:2107.04320, (2021).
[44] M. Sadoughi, C. Hu, Physics-based convolutional neural network for fault diagnosis of rolling element bearings, IEEE Sensors Journal, 19 (2019) 4181-4192.
[45] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, R.X. Gao, WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2021).

[46] Y. Ding, M. Jia, Q. Miao, Y. Cao, A novel time–frequency Transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings, Mechanical Systems and Signal Processing, 168 (2022) 108616.

[47] H. Wang, Z. Liu, D. Peng, Y. Qin, Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis, IEEE Transactions on Industrial Informatics, 16 (2019) 5735-5745.

[48] T. Yan, Y. Fu, M. Lu, Z. Li, C. Shen, D. Wang, Integration of a Novel Knowledge-guided Loss Function with an Architecturally Explainable Network for Machine Degradation Modeling, IEEE Transactions on Instrumentation and Measurement, (2022).

[49] R.G.C. Cunha, E.T. da Silva Jr, C.M. de Sá Medeiros, Machine learning and multiresolution decomposition for embedded applications to detect short-circuit in induction motors, Computers in Industry, 129 (2021) 103461.

[50] J. Azar, A. Makhoul, M. Barhamgi, R. Couturier, An energy efficient IoT data compression approach for edge machine learning, Future Generation Computer Systems, 96 (2019) 168-175.

[51] X. Ye, Y. Hu, J. Shen, R. Feng, G. Zhai, An improved empirical mode decomposition based on adaptive weighted rational quartic spline for rolling bearing fault diagnosis, IEEE Access, 8 (2020) 123813-123827.

[52] X. Li, W. Zhang, Q. Ding, Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism, Signal processing, 161 (2019) 136-154.

[53] J. Jiao, M. Zhao, J. Lin, K. Liang, A comprehensive review on convolutional neural network in machine fault diagnosis, Neurocomputing, 417 (2020) 36-63.

[54] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).

[55] X. Wang, D. Mao, X. Li, Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network, Measurement, 173 (2021) 108518.

[56] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, Proceedings of the IEEE international conference on computer vision, 2017, pp. 618-626.

[57] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, T. Zhang, Deep model based domain adaptation for fault diagnosis, IEEE Transactions on Industrial Electronics, 64 (2016) 2296-2305.

[58] S. Yang, B. Bagheri, H.-A. Kao, J. Lee, A unified framework and platform for designing of cloud-based machine health monitoring and manufacturing systems, Journal of Manufacturing Science and Engineering, 137 (2015).

[59] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang, TensorFlow lite micro: Embedded machine learning for tinyml systems, Proceedings of Machine Learning and Systems, 3 (2021) 800-811.

[60] H. Lu, C. Allen, V. Nemani, C. Hu, A. Zimmerman, IIoT Deployment of a Physics-Informed Deep Learning Model for Online Bearing Fault Diagnostics, Proceedings of the 2022 International Symposium on Flexible Automation, 2022.

Appendix A. Physics-informed feature weighting functions and their performance

The key to feature weighting is to assign larger weights to the features closer to the bearing fault frequencies. There are several weighting functions that can be selected for feature weighting. In this section, in addition to the Gaussian weighting function, the other three weighting functions are designed. Table B.1 summarizes the formula of each function.

Table A.1 Diagnostic results of PICNN when using different weighting functions

Model		Mean accuracy (%)	Best accuracy (%)
PICNN (Gaussian)	$1+\sum_{n=1}^{N+1} a_n^m \exp\left(-\frac{(l-n\cdot l_{\text{fault}}^m)^2}{2(\sigma_n^m)^2}\right)$	99.55 ± 0.16	99.87
PICNN (Linear)	$\sum_{n=1}^{N+1} \max\left(a_n^m - b_n^m \cdot \lvert l - n \cdot l_{\text{fault}} \rvert, 1\right)$	99.56 ± 0.27	99.80
PICNN (Quadratic)	$\sum_{n=1}^{N+1} \max\left(a_n^m - b_n^m \cdot (l - n \cdot l_{\text{fault}})^2, 1\right)$	99.53 ± 0.14	99.85
PICNN (Step)	$\begin{cases} 1 & if\ \lvert l - l_{\text{fault}} \rvert < c \cdot l_{\text{fault}} \\ 0 & \text{otherwise} \end{cases}$	99.50 ± 0.22	99.78

*N indicates the number of harmonics that the weighting function considers

Table B.1 indicates that changing the weighting function type does not significantly change diagnostic accuracy. The Gaussian function assigns attention weights relatively smoothly compared to the other three weighting functions. PICNNs with Gaussian weighting functions are used to generate the results for the proposed physics-informed feature weighting method in the three case studies presented in section 3.

Appendix B. PICNN model Architecture

Table B.8 PICNN model architecture

Layer name	Output shape	Number of parameters
Input layer	(None, 1600, 1)	12
Feature weighting	(None, 1600, 1)	12
Convolutional layer 1	(None, 793, 8)	128
Average pooling 1	(None, 198, 8)	0
Convolutional layer 2	(None, 96, 16)	912
Average pooling 2	(None, 32, 16)	0
Convolutional layer 3	(None, 14, 32)	2592
Average pooling 3	(None, 7, 32)	0
Flatten layer	(None, 224)	0
Dense layer 1	(None, 60)	13500
Dense layer 2	(None, 4)	244
SoftMax	(None, 4)	0

Appendix C. Equivalence between time-domain convolution and frequency-domain multiplication

Consider two time series signals, denoted as $x(t)$ and $y(t)$. The convolution between these two signals can be expressed as:

$$z(t) = \int_{-\infty}^{+\infty} x(s)y(t-s)ds$$

The Fourier transform of signal $x(t)$ is represented by $X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt$, and the Fourier transform of signal $y(t)$ is represented by $Y(f) = \int_{-\infty}^{+\infty} y(t)e^{-j2\pi ft}dt$.

The Fourier transform of the convolved signal can be expressed and further derived as follows:

$$
\begin{aligned}
Z(f) &= \int_{-\infty}^{+\infty} z(t)e^{-j2\pi ft}dt \\
&= \int_{-\infty}^{+\infty} \left\{\int_{-\infty}^{+\infty} x(s)y(t-s)ds\right\} e^{-j2\pi ft}dt \\
&= \int_{-\infty}^{+\infty} x(s)\left\{\int_{-\infty}^{+\infty} y(t-s)e^{-j2\pi ft}dt\right\} ds \\
&= \int_{-\infty}^{+\infty} x(s)\left\{\int_{-\infty}^{+\infty} y(r)e^{-j2\pi f(r+s)}dr\right\} ds \\
&= \int_{-\infty}^{+\infty} x(s)\left\{\int_{-\infty}^{+\infty} y(r)e^{-j2\pi fr}dr\right\}e^{-j2\pi fs}ds \\
&= Y(f)X(f)
\end{aligned}
$$

Therefore, a convolution in the time domain is equivalent to a multiplication in the frequency domain.