IET Image Processing

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Multi-Channel Deep Pulse-Coupled Net: A Novel Bearing Fault Diagnosis Framework

Yanxi Wu[1] | Yalin Yang[2] | Zhuoran Yang[1] | Zhizhuo Yu[1] | Jing Lian[3] | Bin Li[1] | Jizhao Liu[1,4] | Kaiyuan Yang[5]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou, China | [2]Henan Costar Group Co. Ltd., Nanyang City, China | [3]School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China | [4]National-local joint Engineering Laboratory of Building Health Monitoring and Disaster Prevention Technology, Hefei City, China | [5]School of Electronics & Electrical Engineering, University of Sheffield, Sheffield, UK

Correspondence: Jizhao Liu (liujz@lzu.edu.cn) | Kaiyuan Yang (kyang27@sheffield.ac.uk)

Received: 20 September 2024 | **Revised:** 5 February 2025 | **Accepted:** 22 February 2025

Funding: This study is supported by the open fund project of National-local joint Engineering Laboratory of building health monitoring and disaster prevention technology under Grant GG23KF004 and funded in part by the Natural Science Foundation of Gansu Province under Grant 22JR5RA543 and 22JR5RA683.

ABSTRACT

Bearings are a critical part of various industrial equipment. Existing bearing fault detection methods face challenges such as complicated data preprocessing, difficulty in analysing time series data, and inability to learn multi-dimensional features, resulting in insufficient accuracy. To address these issues, this study proposes a novel bearing fault diagnosis model called multi-channel deep pulse-coupled net (MC-DPCN) inspired by the mechanisms of image processing in the primary visual cortex of the brain. Initially, the data are transformed into greyscale spectrograms, allowing the model to handle time series data effectively. The method introduces a convolutional coupling mechanism between multiple channels, enabling the framework can learn the features on all channels well. This study conducted experiments using the bearing fault dataset from Case Western Reserve University. On this dataset, a 6-channel (adjustable to specific tasks) MC-DPCN was utilized to analyse one normal class and three fault classes. Compared to state-of-the-art bearing fault diagnosis methods, our model demonstrates one of the highest diagnostic accuracies. This method achieved an accuracy of 99.96% in normal vs. fault discrimination and 99.89% in fault type diagnosis (average result of ten-fold cross-validation).

1 | Introduction

In the field of industrial production, bearings play a central role in the operational integrity of machinery, making bearing failure detection critical [1]. As a key component of many mechanical systems, the health of bearings has a direct impact on the efficiency, safety, and longevity of the overall equipment [2]. Timely and accurate identification of bearing failures is essential to prevent potential machine breakdowns, which can lead to costly downtime, reduced productivity, and safety incidents [3]. Traditionally, bearing fault detection has been performed using a variety of mechanical and signal processing techniques, ranging from vibration analysis, acoustic analysis, temperature detection, etc. [4] to more complicated methods involving machine learning algorithms. To analyse these features, methods such as support vector machine (SVM) [5] and multilayer perceptron (MLP) [6] have been used to solve this problem, providing a basic understanding and moderate success in identifying fault characteristics. However, as the need for higher accuracy and detection continues to grow, these traditional techniques often fail to meet the demand due to their reliance on manual feature extraction and their inability to handle nonlinear fault data well.

With the advent of deep learning techniques, more sophisticated models such as convolutional neural network (CNN) [7] and recurrent neural network (RNN) [8] have been introduced to the

https://doi.org/10.1049/ipr2.70033

field of fault detection. These methods have achieved significant improvements by automatically extracting features and providing greater predictive capabilities. Despite these advantages, there are still shortcomings in the application of these advanced models in bearing fault diagnosis, especially in terms of model generalization and handling of different fault types in different mechanical setups [9].

To address this challenge, our study proposes a new framework that combines cognitive mechanisms of human visual processing with advanced neural network architectures. Inspired by the way the human brain processes visual information [10–14], we proposed MC-DPCN to simulate these mechanisms. Specifically, this is an auto-wave phenomenon. The key advantage of the auto-wave phenomenon over conventional methods lies in its adaptive threshold adjustment and regional consistency. Conventional methods typically rely on fixed thresholds, which are difficult to adapt to the complexity of images. In contrast, the auto-wave phenomenon dynamically adjusts the threshold based on local image features, enabling more precise adaptation to the needs of different regions and maintaining consistency within the internal areas of the image. Moreover, the auto-wave phenomenon effectively suppresses noise through pulse propagation and accumulation effects, improving the robustness of classification, particularly demonstrating stronger adaptability and accuracy in complex regions, thereby overcoming the limitations of conventional methods [10, 15].

First, for processing time series data, we use the time-frequency conversion method to convert the time series data into spectrograms. The spectrogram can intuitively display the characteristics of the signal in the frequency domain, making complex time series data easier to analyse and process [16]. For multi-dimensional data, we introduce a multi-channel convolution coupling mechanism to couple the features of multiple channels into one channel (shown in Figure 3). Through these improvements, MC-DPCN is able to handle multi-dimensional time series data. Specifically, our method converts the raw acceleration data of each fault type of the bearing into a greyscale spectrogram of multiple channels, which is then processed by our MC-DPCN. This method not only enhances the model's ability to discern subtle patterns in the data, but also improves the model's adaptability to various fault conditions, and spectrograms can be generated based on existing tools without extensive preprocessing.

The main contributions of this work are summarized as follows:

- Our work brings the brain-inspired computing to the field of machine fault diagnosis, enhancing the ability to detect and classify faults with higher accuracy and efficiency by mimicking the neural processes of the human brain.
- Our work proposes a novel framework for the bearing fault diagnosis. This framework is able to process time series data and multi-dimensional features, then classify them into different fault types through the mechanism of visual cortex of the human brain.
- The proposed bearing fault diagnosis framework was tested in CWRU bearing dataset, which is public and widely used in the area of bearing fault diagnosis. The accuracy of this framework reached 99.96% in normal vs. fault detection and 99.89% (average of ten-fold cross-validation and it ranges from 99.64% to 100.00%, which is shown in Table 4) in fault type diagnosis, which can effectively finish task of bearing fault diagnosis.

2 | Related Works

2.1 | Bearing Fault Detection Based on Machine Learning

Bearing fault diagnosis constitutes a critical challenge in industrial applications. Existing studies primarily utilize two signal acquisition approaches: vibration signal analysis and motor current monitoring. Kankar et al. [5] investigated the use of machine learning methods for defect detection and diagnosis. They used both artificial neural networks (ANN) and SVM to identify specific types of defects such as outer ring cracks, inner ring surface roughness and rolling element corrosion pitting by analysing the bearing vibration signals. The experimental results show that this method achieves an accuracy of 71.2329% in ANN and 73.9726% in SVM. The results of the study show that SVM performs slightly better than ANN in classifying bearing faults, although the accuracy of both methods decreases when dealing with combined faults, mainly due to the smaller training dataset. This study emphasizes the critical role of feature extraction and selection in the fault diagnosis process and demonstrates the potential of machine learning algorithms in developing an early fault detection system, thus contributing to condition-based maintenance, preventing catastrophic failures and reducing operational costs. Toma et al. [17] developed a hybrid methodology using motor current signals with genetic algorithm (GA) optimization, achieving over 97% accuracy across three distinct classifiers: K-nearest neighbours (KNN), decision tree, and random forest. This work established motor current monitoring as a cost-effective alternative to vibration analysis, particularly advantageous for multi-motor remote monitoring systems. Wu et al. [18] further advanced the field by proposing a multiscale reduction clustering (MRC) method for unsupervised fault diagnosis, which demonstrated enhanced clustering accuracy and noise robustness through multiscale convolutional representation compression.

These efforts reveal two persistent challenges: (1) suboptimal classification accuracy in complex fault conditions; (2) limited model generalization capabilities. Current solutions focus on multi-modal signal integration and advanced algorithms capable of processing complex datasets [19].

2.2 | Bearing Fault Detection Based on Deep Learning

Deep learning has great potential [20] in bearing fault diagnosis [21, 22]. Gan et al. [23] proposed a deep learning based hierarchical diagnostic network (HDN) utilizing a deep belief networks (DBN) for bearing fault pattern recognition with an accuracy of 99.03%. Sun et al. [24] proposed a bearing fault diagnosis method combining compressive sampling and a deep neural network based on stacked sparse self-encoders, which achieves highly accurate automatic fault identification with an accuracy of

97.47%. Chen et al. [25] proposed a bearing fault diagnosis method combining cyclic spectral cCoherence (CSCoh) and CNN, which improves the diagnostic accuracy and generalization ability by group normalization technique with an accuracy of 99.02%. Eren et al. [26] presented a comprehensive study on the use of compact adaptive 1D convolutional neural networks (1D-CNNs) for real-time bearing fault diagnosis from raw time-series sensor data. This study showcases the potential of 1D-CNNs to perform feature extraction and classification tasks efficiently without the need for pre-determined transformations or manual feature engineering. The research emphasized the classifier's ability to operate with limited training data and fewer backpropagation iterations, making it suitable for real-time applications. Verstraete et al. [7] introduced a method using time-frequency image analysis of rolling bearings, applying deep learning to convert vibration signals into images processed by a CNN. This simplifies feature extraction and improves diagnostic accuracy under noisy conditions. The study highlights the effectiveness of combining various time-frequency analysis methods, like the short-time Fourier transform, wavelet transform, and Hilbert-Yellow transform, to enhance the CNN's fault diagnosis capabilities. Miao et al. [27] proposed an improved method based on interactive channel attention (ICA), which helps CNN focus on channel correlations, especially under noisy conditions, making it highly effective for rotating component fault detection in scenarios with low signal-to-noise ratios.

Despite these advancements, critical limitations persist: (1) excessive computational demands from multi-sensor data fusion; (2) inadequate temporal feature extraction capabilities; (3) difficulty handling high-dimensional time-series data.

2.3 | Brain-Inspired Computing

The human brain is composed of billions of neurons, with the dendrite serving as its input terminal [28–30]. The cell body integrates incoming spikes from different dendritic branches, producing a spike upon reaching the threshold membrane potential. These spikes then propagate along axons to communicate with other neurons through synapses [31–35].

Spiking neural network (SNN) is a popular network model in recent years. Currently, the enthusiasm for research on SNNs is rising in the academic community [28, 36, 37]. For example, multi-threshold augmented SNN models [38], vform [39], deep SNN with knowledge distillation [36] and time-to-first-spike (TTFS) in SNN [40]. In recent years, brain-inspired computing has been used in various fields such as affective computing [41], mental health monitoring [42], and smart home [43].

Currently, there are a number of brain-inspired computational models, the more typical one being pulse-coupled neural network (PCNN). PCNN originated in the 1990s and was first proposed by Eckhorn et al. [44]. They proposed this model to simulate synchronized impulse activity between neurons based on their observation of the phenomenon of synchronized oscillations in cat visual cortex [44, 45]. Many studies have shown significant advantages of PCNN over other neuronal models for image classification tasks [45, 46].

Combining the advantages of PCNN and SNN algorithms, in recent years, Yi et al. [47] proposed the DPCNN model. Compared to PCNN, which is limited to a single-layer structure and lacks learning capability [48–51], DPCNN utilizes convolutional architecture and introduces the spatiotemporal backpropagation algorithm from SNN, endowing the model with learning capability.

However, there is no fault diagnosis method based on DPCNN. And the current DPCNN cannot handle multi-channel data and time series data effectively, which largely limits the application of DPCNN, especially in bearing fault diagnosis.

3 | Method

In this section, we elaborate on a framework for diagnosing bearing faults, leveraging the image processing mechanism of the human brain's visual cortex.

As is shown in Figure 1, it shows the flowchart of the whole framework. The acceleration data are first processed into spectrograms, and then multiple spectrograms of a sample are fed into each of the six channels of the MC-DPCN in a consistent order.

3.1 | Time-Frequency Transformation

Due to the limitations of the model in handling time-series data effectively, we employed spectrogram processing techniques to preprocess these time-series data.

3.1.1 | Spectrogram

In signal processing, a spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time [52]. For the purpose of this study, we focus on using the spectrogram to analyse acceleration data from machinery, aiming to classify different operational states or detect faults.

3.1.1.1 | Fourier Transform. The Fourier transform is a mathematical tool used to transform a time-domain signal into its constituent frequencies. For a continuous signal $f(t)$, the Fourier transform $\mathcal{F}(f(t))$ is defined as:

$$\mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}\,\mathrm{d}t \tag{1}$$

For discrete signals, we use the discrete Fourier transform (DFT), which is defined for a sequence of N samples $x(n)$ as:

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi mn/N}, \quad m, n = 0, 1, \ldots, N-1 \tag{2}$$

where N is the number of samples, $x(n)$ is the signal in the time domain, and $X(m)$ is the corresponding frequency component.

3.1.1.2 | Fast Fourier Transform. The fast Fourier transform (FFT) is an efficient algorithm to compute the DFT. The computational complexity of the DFT is $O(N^2)$, whereas the

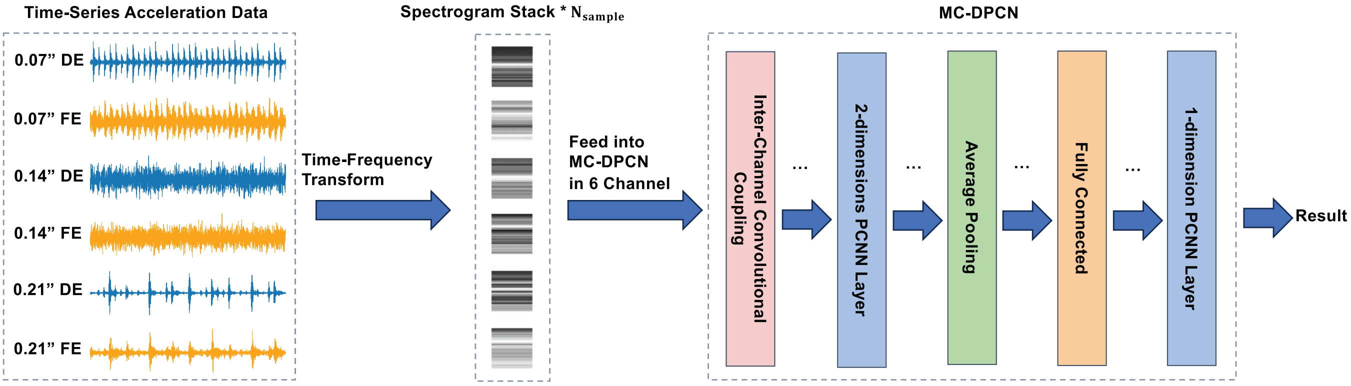

FIGURE 1 | Flowchart of the bearing fault detection framework.

FFT reduces this to $O(N \log_2 N)$. This efficiency is particularly beneficial for processing large datasets. The FFT algorithm transforms the time-domain signal into its frequency-domain representation. Currently, the most commonly used FFT algorithm is the Cooley–Tukey algorithm, which uses the "divide and conquer" strategy to decompose a DFT of arbitrary size into several smaller DFTs that are recursively computed step by step. The specific implementation method can be found in the work of Cooley et al. [53].

3.1.1.3 | Computing the Spectrogram. To compute the spectrogram, we first apply the FFT to the time-domain signal. The magnitude of the FFT results represents the amplitude of different frequency components. For a time-domain signal $x(t)$ sampled at a rate f_s, the frequency bins are given by:

$$f_m = \frac{kf_s}{N}, \quad m = 0, 1, \dots, N-1 \tag{3}$$

where N is the number of samples in the FFT.

3.1.2 | Application to Acceleration Data

In our approach, we apply a windowing technique to the acceleration data before computing the spectrogram. As is shown in Figure 2, each window has a size of 256 samples, and we use a step size of 64 samples between consecutive windows.

For each window of 256 samples, we compute the FFT to obtain the frequency spectrum. The resulting spectrum is then converted into a greyscale image, where the intensity of each pixel represents the amplitude of the corresponding frequency component. This greyscale spectrogram provides a visual representation of the frequency content of each windowed segment of the acceleration data.

By analysing these greyscale spectrograms, we can identify patterns and spectral characteristics associated with different operational states or faults in machinery. Peaks and patterns in the spectrograms can reveal dominant frequencies and their variations over time, aiding in the diagnosis and classification of machinery conditions.

3.2 | Multi-Channel Deep Pulse-Coupled Net

The classic PCNN and its improved models are limited to single-layer structures and lack learning capabilities, making them unable to complete complex tasks. To address this limitation, the MC-DPCN incorporates the spatiotemporal backpropagation algorithm from SNN into its architecture.

The MC-DPCN is structured into two main components: the convolutional block and the fully connected layers. The 6-dimensional input initially enters the convolutional unit, where it sequentially passes through multiple convolutional layers. Each convolutional layer comprises three elements: convolution, batch normalization, and a PCNN layer. Following the convolutional layers, the data proceeds to the pooling layer and subsequently to multiple fully connected layers. Each fully connected layer includes three components: a fully connected layer, batch normalization, and a non-linking PCNN layer. For each time step, the PCNN layer from the previous time step influences the PCNN layer of the subsequent time step. The output at time step N is taken as the final result.

The overall network framework is shown in Figure 3.

3.2.1 | Convolutional Layer

Our model incorporates a novel inter-channel coupling mechanism involving six channels within the convolutional layer. This coupling strategy, illustrated in the convolutional unfolding section, merges inputs from all six channels into channel 2, fostering comprehensive interactions among channels. Within this framework, neurons expand their receptive fields to encompass outputs not only from neighbouring neurons within their own channel's feature map but also from neurons at corresponding spatial positions across feature maps of other channels, including their adjacent neurons. This inter-channel coupling paradigm facilitates a synergistic integration of global features and exploits feature correlations across channels. As a result, our model achieves heightened feature representation capabilities, essential for nuanced feature extraction and effective learning in complex tasks.

In our network, we use 3×3 convolutional kernels with a stride of 1. Channels 1, 3, 4, 5, and 6 each have a coupling connector, which

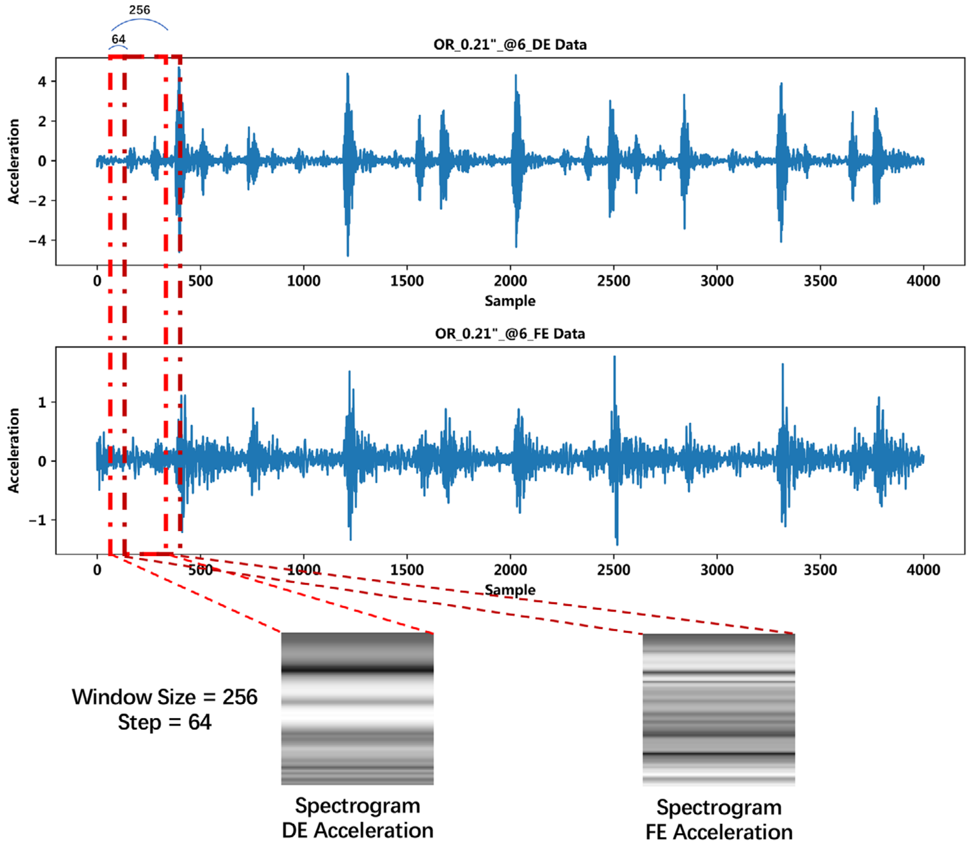

FIGURE 2 | Generate greyscale spectrogram from acceleration data.

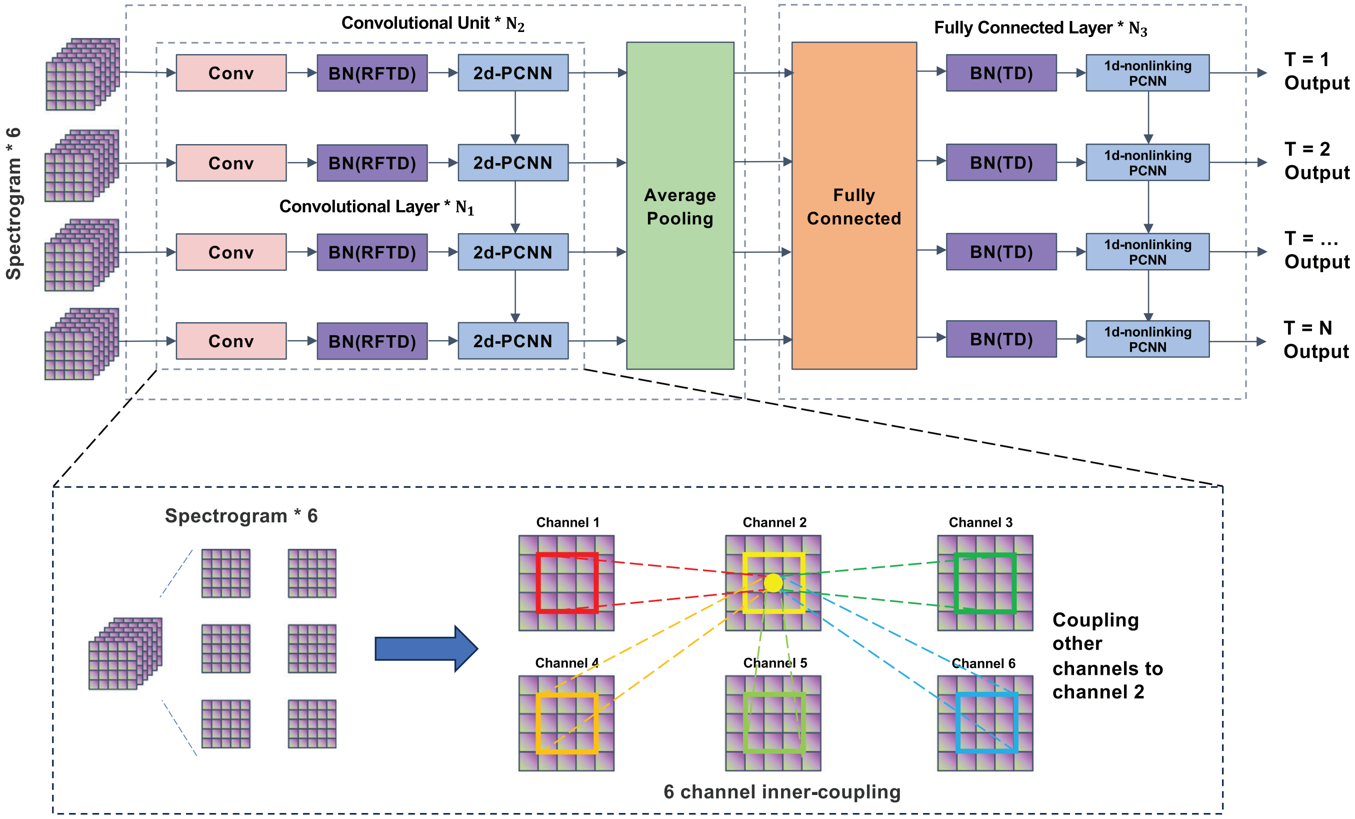

FIGURE 3 | MC-DPCN. "1D/2D" stands for *N*-dimension.

TABLE 1 | Notation list of PCNN neuron.

Symbol	Description
L	Couple linking
F	Feeding input
U	Modulation product
E	Dynamic activity
Y	Continuous output
W, M	Trainable weight matrix
$\alpha_F, \alpha_L, \alpha_E$	Leak factors of state variables F, L, and E
V_E	Amplitude gain
t	Time steps
n	Current PCNN layer
$*$	Convolution operation
$\odot$	Element-wise multiplication
H	Step function

couples them into channel 2. Compared to traditional three-channel configurations, the multi-channel approach allows for processing more data and better learning of data characteristics. Moreover, this method can be extended to include more channels, although for our work, we only need to utilize six channels.

This process is illustrated in the expanded section of the network framework, as depicted in Figure 3.

3.2.2 | PCNN Neuron

The MC-DPCN introduces convolutional connections into the PCNN layer. The PCNN neuron with convolutional feedforward connections can be formulated by the following equation:

$$\begin{cases} \boldsymbol{F}^{t,n} = \alpha_F \boldsymbol{F}^{t-1,n} + \boldsymbol{W}^n * \boldsymbol{Y}^{t,n-1} \\ \boldsymbol{L}^{t,n} = \alpha_L \boldsymbol{L}^{t-1,n} + \boldsymbol{M}^n * \boldsymbol{Y}^{t-1,n} \\ \boldsymbol{E}^{t,n} = \alpha_E \boldsymbol{E}^{t-1,n} + V_E \boldsymbol{Y}^{t-1,n} \\ \boldsymbol{U}^{t,n} = \boldsymbol{F}^{t,n} \odot (1 + \boldsymbol{L}^{t,n}) \\ \boldsymbol{Y}^{t,n} = H(\boldsymbol{U}^{t,n} - \boldsymbol{E}^{t,n}) \end{cases} \tag{4}$$

where the symbols are interpreted in Table 1.

The Heaviside step function $H(x)$ can be defined as:

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

If we disregard the connected inputs of PCNN, that is, ignore the coupled connections between PCNN neurons, PCNN is simplified into a non-linking PCNN (PCNN without linking channels):

$$\begin{cases} \boldsymbol{U}^{t,n} = \boldsymbol{F}^{t,n} = \alpha_F \boldsymbol{F}^{t-1,n} + \boldsymbol{W}^n * \boldsymbol{Y}^{t,n-1} \\ \boldsymbol{E}^{t,n} = \alpha_E \boldsymbol{E}^{t-1,n} + V_E \boldsymbol{Y}^{t-1,n} \\ \boldsymbol{Y}^{t,n} = H(\boldsymbol{U}^{t,n} - \boldsymbol{E}^{t,n}) \end{cases} \tag{5}$$

When iterating the PCNN calculations, it is necessary to specify the initial values of the state variables. Following the method outlined in reference [47], the following initialization approach is adopted:

$$\begin{cases} \boldsymbol{F}^0 = 0 \\ \boldsymbol{L}^0 = 0 \\ \boldsymbol{Y}^0 = 0 \end{cases} \tag{6}$$

and

$$\boldsymbol{E}^0 = \frac{V_E}{\alpha_E} \tag{7}$$

3.2.3 | Rapid Feature Transformation Dynamics Batch Normalized Layer

RFTD-BN (rapid feature transformation dynamics batch normalization) is a layer based on batch normalization (BN) techniques.

Batch normalization (BN) is a crucial technique for training deep networks. It accelerates convergence and improves performance by normalizing input data. The BN formula is given by:

$$\mathrm{BN}(\boldsymbol{X}) = \boldsymbol{\gamma} \odot \frac{\boldsymbol{X} - \boldsymbol{\mu}_{\mathcal{B}}}{\boldsymbol{\sigma}_{\mathcal{B}}} + \boldsymbol{\beta} \tag{8}$$

where $\mathcal{B}$ denotes a minibatch, $\boldsymbol{X}$ is an input sample, and $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learnable scale and shift parameters, respectively. $\boldsymbol{\mu}_{\mathcal{B}}$ and $\boldsymbol{\sigma}_{\mathcal{B}}^2$ are the mean and variance of the minibatch, which can be calculated as follows:

$$\boldsymbol{\mu}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{X} \in \mathcal{B}} \boldsymbol{X} \tag{9}$$

$$\boldsymbol{\sigma}_{\mathcal{B}}^2 = \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{X} \in \mathcal{B}} (\boldsymbol{X} - \boldsymbol{\mu}_{\mathcal{B}})^2 + \epsilon \tag{10}$$

where ϵ is a small positive constant to ensure numerical stability.

However, the batch normalization method mentioned above cannot be used due to the coupled connections in PCNN. RFTD-BN was proposed by Yi et al. [47]. The RFTD-BN performs batch normalization operations on the feedforward input current I_F and the connected input current I_L at different moments, respectively. In RFTD-BN, different BN parameters are used for feedforward inputs and link inputs. And for input channels, different BN parameters are used at different time steps. The mathematical expression is as follows:

$$\hat{\boldsymbol{I}}_F^t = \mathrm{BN}_F(\boldsymbol{I}_F^t) = \boldsymbol{\gamma}_F \odot \frac{\boldsymbol{I}_F^t - \boldsymbol{\mu}_{\mathcal{B},F}^t}{\boldsymbol{\sigma}_{\mathcal{B},F}^t} + \boldsymbol{\beta}_F \tag{11}$$

$$\hat{\boldsymbol{I}}_L^t = \mathrm{BN}_L(\boldsymbol{I}_L^t) = \boldsymbol{\gamma}_L \odot \frac{\boldsymbol{I}_L^t - \boldsymbol{\mu}_{B,L}^t}{\boldsymbol{\sigma}_{B,L}^t} + \boldsymbol{\beta}_L \tag{12}$$

where F denotes the feedforward channel, L denotes the connected channel, and t denotes the time step.

In addition, since non-linking PCNN does not have a feedforward input F with a connected input L, a simplification of RFTD-BN is required. The simplified RFTD-BN is called TD-BN (time dependent batch normalization). The TD-BN utilize the same BN parameters at time step t to normalize both $\boldsymbol{I}_F^t$ and $\boldsymbol{I}_F^t$, as given by:

$$\hat{\boldsymbol{I}}_F^t = \mathrm{BN}^t(\boldsymbol{I}_F^t) = \boldsymbol{\gamma}^t \odot \frac{\boldsymbol{I}_F^t - \boldsymbol{\mu}_{B,F}^t}{\boldsymbol{\sigma}_{B,F}^t} + \boldsymbol{\beta}^t \tag{13}$$

$$\hat{\boldsymbol{I}}_L^t = \mathrm{BN}^t(\boldsymbol{I}_L^t) = \boldsymbol{\gamma}^t \odot \frac{\boldsymbol{I}_L^t - \boldsymbol{\mu}_{B,L}^t}{\boldsymbol{\sigma}_{B,L}^t} + \boldsymbol{\beta}^t \tag{14}$$

In addition, there is no batch normalization operation between the non-linking PCNN used for output and the fully connected layer.

4 | Experiments

4.1 | Datasets

We utilized a bearing dataset collected from the Case Western Reserve University (CWRU) data center.

The CWRU bearing dataset is a standard dataset used for mechanical fault diagnosis and prediction. It comprises vibration signal data collected under various fault conditions, including inner race fault (IR), outer race fault (OR), ball fault (BF), and normal operating conditions. These data are extensively used to assess the performance of various fault diagnosis and prediction algorithms. Figure 4 illustrates the experimental platform.

This study examines three fault states along with one normal state. The fault states include IR, BF, and OR occurring at the 6 o'clock position. Three accelerometers were used to collect acceleration data: the drive end accelerometer (DE), the fan end accelerometer (FE), and the base accelerometer (BA). Due to missing data from the BA for all fault classes, we ensure data consistency by relying on acceleration data solely from the DE and FE accelerometers.

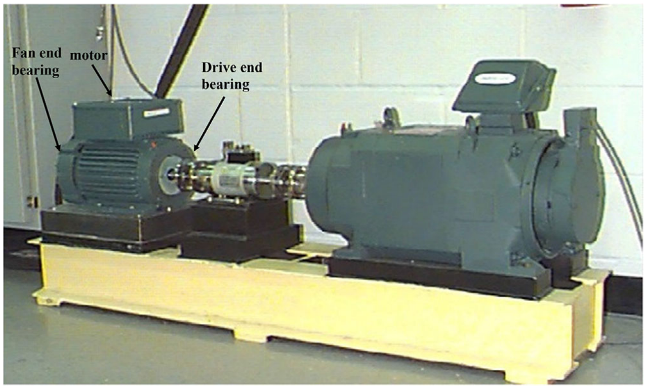

FIGURE 4 | CWRU bearing dataset experimental platform.

4.2 | Network Structure and Hyperparameters

As is shown in Table 2, we put the MC-DPCN under VGG9 Net. The VGG network follows a classic structure, consisting of alternating convolutional and pooling layers, culminating in fully connected layers [54]. However, instead of merely relying on standard convolution operations, each convolutional block is followed by a 2-dimensions PCNN module (PCNN2d), which introduces temporal dynamics through iterative updates across multiple time steps (T). This adjustment allows for enhanced temporal feature extraction, making the model more suitable for dynamic data analysis.

For the MC-DPCN leak factors, use the following values:

$$\alpha_F = \alpha_L = 0.5, \alpha_E = 0.7, V_E = 1.0 \tag{15}$$

and set the time step T to 6.

4.3 | Normal and Fault Diagnosis

We placed FE and DE into channel 1 and channel 2 and conducted a binary classification task for normal and abnormal detection. In this task, we used data from 2 channels only and trained the model with DE (feature 1) and FE (feature 2) of normal class data and three abnormal class data.

After training for 10 epochs, our model achieved 100.00% accuracy on the training dataset with a loss of 0.000. On the test dataset, the model achieved 100% accuracy with a test time of 1 s, and the maximum test accuracy reached 99.96%.

The confusion matrix on test dataset is shown in Figure 5.

It can be seen that our model performed excellently on this task, demonstrating high classification accuracy and low misclassification rates.

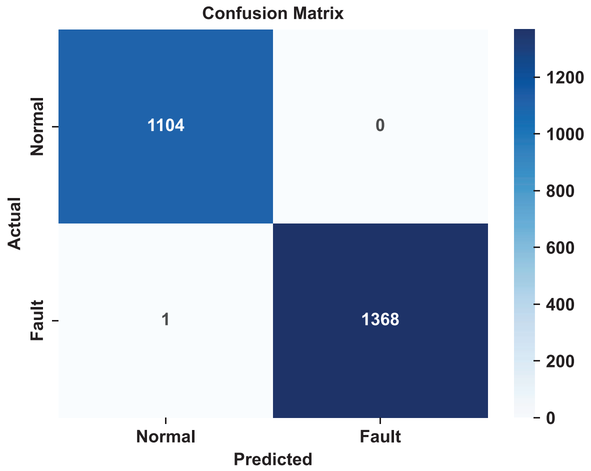

FIGURE 5 | Fault diagnosis confusion matrix on test dataset.

TABLE 2 | MC-DPCN under the structure of VGG9.

Layer name	Type	Input shape	Output shape	Parameters
Input	[N, T, 6, 32, 32]	[N, T, 6, 32, 32]	—	
Conv2d-1	Conv2d (6–>64)	[N, T, 6, 32, 32]	[N, T, 64, 32, 32]	Kernel size: 3x3, Padding: 1, Bias: False
PCNN2d-1	PCNN2d (linking=True)	[N, T, 64, 32, 32]	[N, T, 64, 32, 32]	Linking and coupling layers enabled
Conv2d-2	Conv2d (64–>64)	[N, T, 64, 32, 32]	[N, T, 64, 32, 32]	Kernel size: 3x3, Padding: 1, Bias: False
PCNN2d-2	PCNN2d (linking=True)	[N, T, 64, 32, 32]	[N, T, 64, 32, 32]	Linking and coupling layers enabled
Pool-1	AvgPool2d (2×2)	[N, T, 64, 32, 32]	[N, T, 64, 16, 16]	—
Conv2d-3	Conv2d (64–>128)	[N, T, 64, 16, 16]	[N, T, 128, 16, 16]	Kernel size: 3x3, Padding: 1, Bias: False
PCNN2d-3	PCNN2d (linking=True)	[N, T, 128, 16, 16]	[N, T, 128, 16, 16]	Linking and coupling layers enabled
Conv2d-4	Conv2d (128–>128)	[N, T, 128, 16, 16]	[N, T, 128, 16, 16]	Kernel size: 3x3, Padding: 1, Bias: False
PCNN2d-4	PCNN2d (linking=True)	[N, T, 128, 16, 16]	[N, T, 128, 16, 16]	Linking and coupling layers enabled
Pool-2	AvgPool2d (2×2)	[N, T, 128, 16, 16]	[N, T, 128, 8, 8]	—
Conv2d-5	Conv2d (128–>256)	[N, T, 128, 8, 8]	[N, T, 256, 8, 8]	Kernel size: 3x3, Padding: 1, Bias: False
PCNN2d-5	PCNN2d (linking=True)	[N, T, 256, 8, 8]	[N, T, 256, 8, 8]	Linking and coupling layers enabled
Conv2d-6	Conv2d (256–>256)	[N, T, 256, 8, 8]	[N, T, 256, 8, 8]	Kernel size: 3x3, Padding: 1, Bias: False
PCNN2d-6	PCNN2d (linking=True)	[N, T, 256, 8, 8]	[N, T, 256, 8, 8]	Linking and coupling layers enabled
Pool-3	AvgPool2d (2×2)	[N, T, 256, 8, 8]	[N, T, 256, 4, 4]	—
Flatten	Flatten	[N, T, 256, 4, 4]	[N, T, 4096]	—
Linear-1	Linear (4096–>1024)	[N, T, 4096]	[N, T, 1024]	Bias: False
PCNN1d	PCNN1d (linking=False)	[N, T, 1024]	[N, T, 1024]	Linking disabled
Linear-2	Linear (1024–>Classes)	[N, T, 1024]	[N, T, Classes]	—

Note: N denotes batch size and *T* denotes time step.

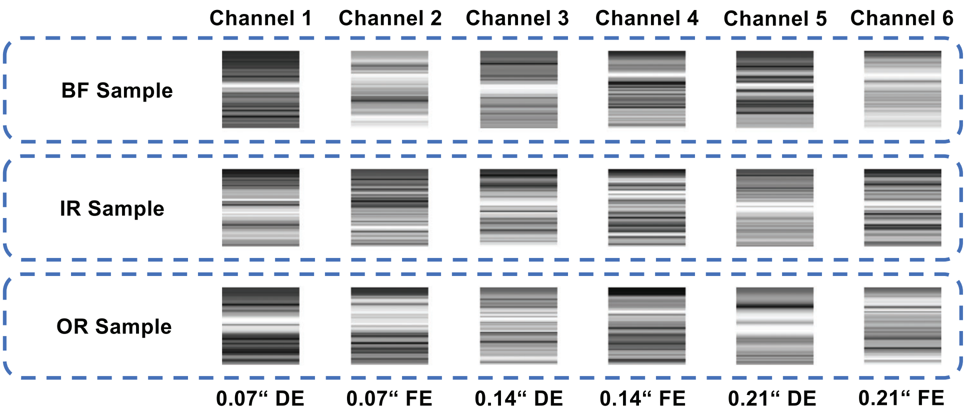

FIGURE 6 | Data sample.

4.4 | Classification of Bearing Fault Types

To ensure consistency, we used bearing data with diameters of 0.07, 0.14, and 0.21 inches at a sampling frequency of 12 kHz. We generated a series of spectrograms with a window size of 256. A sample from a fault class comprises six spectrograms, representing the spectrograms of fan blade acceleration and ball bearing acceleration at diameters of 0.07, 0.14, and 0.21 inches, respectively. The composition of sample data is shown in Figure 6.

The number of samples is shown in Table 3. Among them, there are 1850 samples for each of the BF, IR, and OR fault type. We use a randomized division strategy to divide the training dataset and test dataset in the ratio of 80% to 20%. After the division, in the training dataset, there are 1462 samples for the BF class, 1495 samples for the IR class, and 1483 samples for the OR class, totalling 4440 samples. In the test dataset, there are 388 samples in the BF class, 355 samples in the IR class, and 367 samples in the OR class, totalling 1110 samples.

4.4.1 | Performance Measure

Our analysis primarily centres on scrutinizing the model's accuracy and loss assessment. Furthermore, to enhance our

TABLE 3 | Sample's amount.

Fault type			Amount	Total
BF	0.07″	DE	1850	1850
		FE	1850	
	0.14″	DE	1850	
		FE	1850	
	0.21″	DE	1850	
		FE	1850	
IR	0.07″	DE	1850	1850
		FE	1850	
	0.14″	DE	1850	
		FE	1850	
	0.21″	DE	1850	
		FE	1850	
OR	0.07″	DE	1850	1850
		FE	1850	
	0.14″	DE	1850	
		FE	1850	
	0.21″	DE	1850	
		FE	1850	

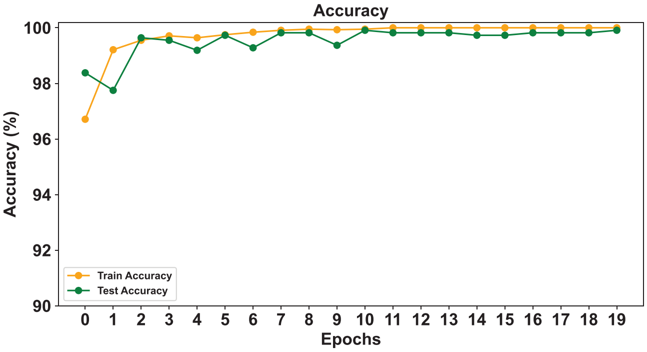

FIGURE 7 | Fault type detection accuracy.

evaluation of the model's performance and ensure its generalization ability, we conduct a ten-fold cross-validation procedure.

As shown in Figures 7 and 8, the loss and accuracy curves during training show a typical convergence trend.

It can be seen intuitively that the model converges very quickly, thanks to the fact that MC-DPCN uses the perception of the visual cortex of the human brain. Specifically, as shown in Figure 7, in the initial stage (epoch = 0), the test accuracy is slightly higher than the training accuracy, which may indicate a phenomenon: the model's learning on the train dataset has not yet been fully demonstrated, and some features or patterns in the test dataset may be different from the train dataset, making it easier for the model to accurately categorize the test data at that stage. This may be due to sample distributions, noise, or the presence of specific patterns in the test data that may not have been fully demonstrated in the training data.

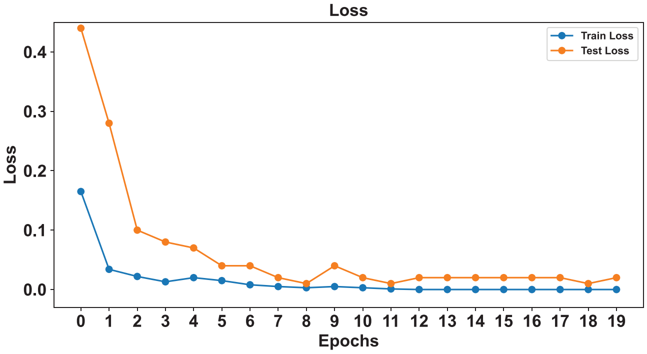

FIGURE 8 | Fault type detection loss.

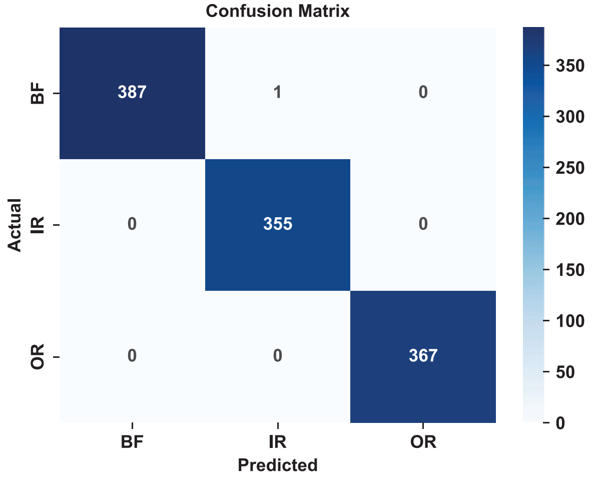

FIGURE 9 | Fault type diagnosis confusion matrix on test dataset.

As training continues, we see that the accuracy of the model gradually stabilizes and converges on both the train dataset and test dataset. This indicates that the model is gradually mastering the features and patterns of the data during the learning process. This stabilization and convergence phenomenon of accuracy indicates that our model avoids overfitting to some extent.

In Figure 8, we observe that the loss curve decreases faster in the early stage of training, while the decrease gradually slows down until it levels off as training progresses. This reflects that the model gradually learns more complex and abstract features during training.

Additionally, there are significant fluctuations within the first three training epochs. This is attributed to the adaptive learning rate of the Adam optimizer. Research has shown that the variance of the adaptive learning rate in the early stages of the Adam algorithm is relatively high [55]. This can lead to instability in the model parameter updates, resulting in substantial fluctuations in both loss and accuracy.

Figure 9 shows the confusion matrix of the model on the test dataset after 20 epochs of training. It can be seen that the model predicts the three types of faults almost entirely accurately.

TABLE 4 | Ten-fold cross-validation results.

Fold	Train accuracy	Train loss	Test accuracy	Test loss
1	99.98	0.0019	100.00	0.0003
2	99.98	0.0009	100.00	0.0005
3	100.00	0.0006	100.00	0.0024
4	100.00	0.0008	100.00	0.0003
5	100.00	0.0008	99.82	0.0024
6	100.00	0.0007	99.82	0.0010
7	100.00	0.0008	99.82	0.0024
8	100.00	0.0008	99.82	0.0003
9	100.00	0.0008	99.82	0.0010
10	100.00	0.0008	99.64	0.0009
Average	99.99	0.0009	99.89	0.0012

Taken together, these phenomena demonstrate the effectiveness and feasibility of our proposed model for the problem of bearing fault type classification. Through in-depth analysis and reasonable adjustment of the training process, we were able to obtain a model with good performance and provide strong support and guidance for solving practical problems.

4.4.2 | Ten-Fold Cross-Validation

We also performed a tenfold cross-validation of the model on the greyscale spectrogram dataset we constructed. As shown in Table 4, we trained 10 epochs for each fold, where each fold contains training accuracy, training loss, testing accuracy, and testing loss.

It can be seen that the average training accuracy of the ten-fold cross-validation reaches 99.99%, while the average testing accuracy is as high as 99.89%, which indicates that the model achieves high classification accuracy on different datasets. This consistent performance proves the robustness and generalization ability of the model. This is due to the high robustness of PCNN in processing images using human brain mechanisms.

Meanwhile, the average training loss is 0.0009 and the average testing loss is 0.0012. This indicates that the model is able to fit the data well in both training and testing without overfitting.

4.4.3 | Performance Comparison with Existing Models

As shown in Table 5, we retrieved studies that performed fault diagnosis on the CWRU bearing dataset from 2011 to 2024. The research methods of these studies include machine learning, deep learning, transfer learning, and other methods. The accuracy of these studies for fault diagnosis on the CWRU bearing dataset ranges from 67% to 100%. The result of our diagnostic framework is 99.89%, which shows that our diagnostic framework is one of the most advanced bearing fault diagnosis methods available.

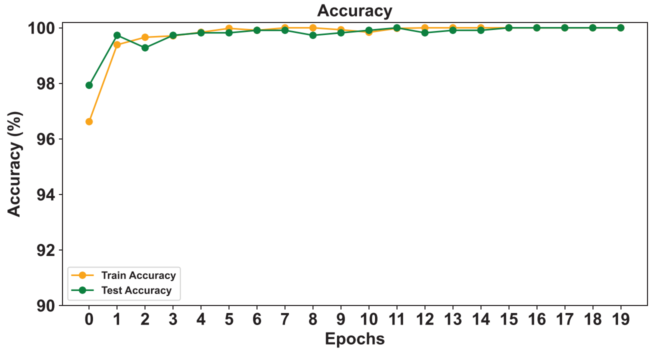

FIGURE 10 | Accuracy after added Gaussian noise.

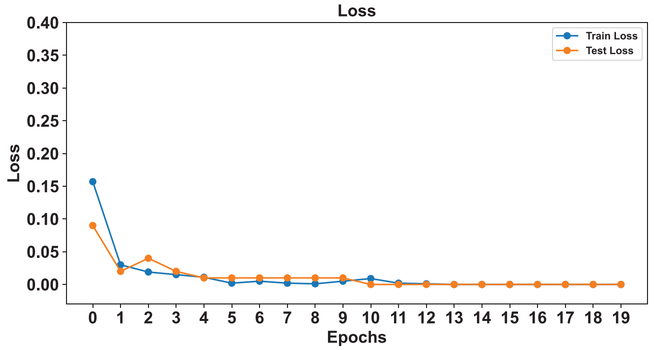

FIGURE 11 | Loss after added Gaussian noise.

4.4.4 | Robustness Analysis

We added Gaussian noise with mean 0 and standard deviation 0.2 to the constructed dataset. A training of 20 epochs was performed and the accuracy and loss on the training and test datasets were counted.

In Figure 10, both the training accuracy and testing accuracy exhibit an increasing trend. The training accuracy reaches 100% after 7 epochs, while the testing accuracy attains 100% at epoch 11 and remains stable thereafter. This indicates that the model performs exceptionally well on both the training and testing datasets, with no obvious signs of overfitting.

In Figure 11, the training loss and testing loss initially exhibit relatively high values. However, as the number of epochs increases, the loss decreases rapidly and approaches zero after the 5th epoch, suggesting that the model has effectively converged.

In summary, MC-DPCN maintains high accuracy even after the addition of Gaussian noise. The performance differences compared to the results without noise can be considered as random errors. This demonstrates that MC-DPCN possesses strong robustness.

4.4.5 | Model's Generalization Ability

We believe that conducting experiments solely at CWRU is insufficient to demonstrate the generalization ability of MC-DPCN. To address this, we trained MC-DPCN using the bearing

TABLE 5 | Accuracy comparison on CWRU bearing dataset.

Year	Venue	Model/Method	Accuracy
2011	Expert Syst. with Appl.	ANN [5]	71.2329%
2011	Expert Syst. with Appl.	SVM [5]	73.9726%
2014	Signal Process.	KNN [56]	91.2%
2015	Meas. Sci. Technol.	Opti-DBN [57]	87.45%
2016	Mech. Syst. Signal Process.	DBNs [23]	99.03%
2017	Shock Vib.	CNN [7]	84.1–99.4%
2017	IEEE Trans. on Instrum. and Meas.	CS-DNN [24]	97.47%
2017	IEEE Trans. Ind. Electron.	LeNet-5 CNN [58]	99.79%
2019	J. Signal Process. Sys.	1D-CNN [26]	93.2%
2019	IEEE Trans. Syst. Man Cybern. Syst.	DADA [59]	99.83%
2020	Adv. Eng. Inform.	DWT-GDA-WNN [60]	98.02%
2020	Adv. Eng. Inform.	WPT-GDA-WNN [60]	98.54%
2020	Adv. Eng. Inform.	VMD-GDA-WNN [60]	98.48%
2020	Mech. Syst. Signal Process	CSCoh-CNN [25]	99.02%
2021	Neurocomputing	CMD [61]	92.16%
2021	Measurement	CNN-gcForest [62]	99.79%
2021	Appl. Intell.	CRNN [22]	99.77%
2022	Knowl.-Based Syst.	DWQDAN [63]	84.95%
2022	Machines	GAF-CA-CNN [64]	96.9–99.7%
2022	Measurement	ADC-CNN+LATL [65]	99.80–100.00%
2023	Sensors	Lite-CNN [66]	99.75–99.97%
2024	Reliab. Eng. & Syst. Safe.	PCA-GMM [67]	32–54%
2024	Reliab. Eng. & Syst. Safe.	NCA-GMM [67]	50.5–52%
2024	Reliab. Eng. & Syst. Safe.	WPCA-GMM [67]	67–82.50%
—	—	**MC-DPCN(Ours)**	**99.89%**

Note: We take the average of ten-fold cross-validation as the result.

fault dataset from Xi'an Jiaotong University (XJTU)[68]. The XJTU dataset includes three operating conditions. Operating condition 1 has a speed of 2100 rpm and a load of 12 kN; operating condition 2 has a speed of 2250 rpm and a load of 11 kN; and operating condition 3 has a speed of 2400 rpm and a load of 10 kN. For each operating condition, five bearings with different fault types were tested. Acceleration data in both the vertical and horizontal directions were collected for each bearing. We converted the vertical and horizontal acceleration data collected under operating condition 1 into spectrograms using a window size of 256 and a step size. At this point, MC-DPCN only needs to open two channels (channel 1 and 2) to input the spectrograms of the vertical and horizontal accelerations. We performed 20 training runs and obtained the training and testing accuracy curves shown in Figures 12 and 13.

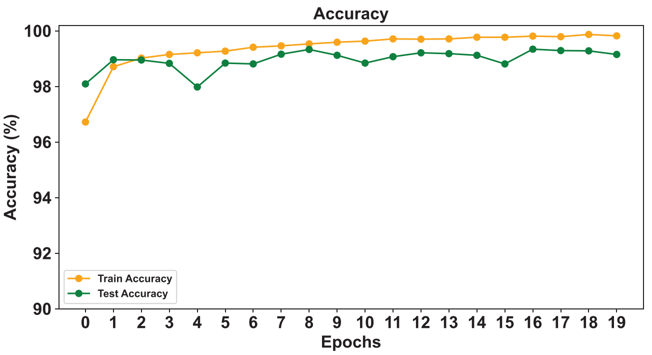

FIGURE 12 | Accuracy on XJTU bearing dataset of condition 1.

From Figures 12 and 13, the experimental results indicate that the training loss gradually decreased as the number of epochs increased, from an initial value of 0.111 to 0.005, demonstrating continuous improvement in the model's performance on the training data. In the early epochs, the training loss decreased sharply, while in the later epochs, the rate of decline slowed down, suggesting that the model was stabilizing as it approached the optimal solution. The test loss also exhibited a decreasing trend, from 0.08 to 0.04, although there were slight increases in certain epochs (e.g., epochs 4 and 5), the overall trend remained downward. These fluctuations may be related to variations in the model's generalization ability during specific phases, but they did not have a significant impact.

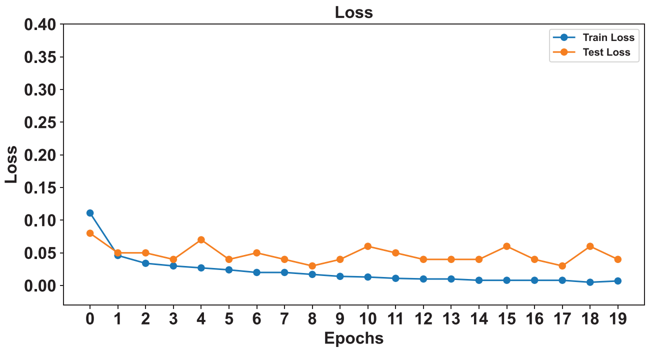

FIGURE 13 | Loss on XJTU bearing fault dataset of condition 1.

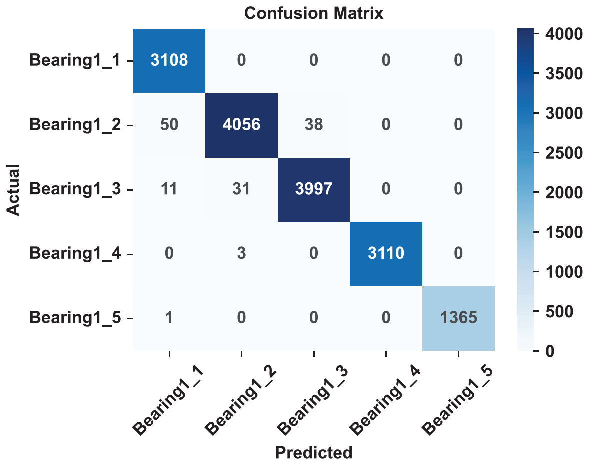

FIGURE 14 | Confusion matrix on test dataset of XJTU bearing fault dataset of condition 1.

The training accuracy increased steadily from 96.72% to 99.82%, indicating that the model's performance on the training dataset continued to improve, ultimately achieving near-perfect fit. The test accuracy also showed a similar upward trend, rising from 98.09% to 99.15%. Although there were slight fluctuations in test accuracy during some epochs (e.g., epochs 5 and 16), the overall trend demonstrated consistent improvement, with test accuracy stabilizing at around 99% in the final epochs. This suggests that the model exhibited strong generalization ability, performing well on unseen data.

Overall, the experimental results reflect the convergence and stability of the training process. Despite occasional fluctuations in test accuracy during individual epochs, the model's performance on both the training and test datasets consistently improved, ultimately achieving high accuracy, indicating that the model was able to adequately fit the training data and generalize well to new, unseen samples.

Additionally, we have also plotted the confusion matrix after 20 epochs training, as shown in Figure 14.

In Figure 14, the rows correspond to the actual classes, and the columns represent the predicted classes. The majority of instances were correctly classified, with 3108 instances of Bearing1_1 correctly predicted, and no misclassifications for this class. For Bearing1_2, out of 4144 actual instances, 4056 were correctly predicted, while 50 instances were misclassified as Bearing1_1 and 38 as Bearing1_3. Similarly, for Bearing1_3, 3997 out of 4039 instances were correctly predicted, with 11 misclassified as Bearing1_1 and 31 as Bearing1_2. Bearing1_4 was correctly classified in all 3113 instances, with only 3 misclassifications as Bearing1_2. For Bearing1_5, 1365 instances were correctly predicted, with a single misclassification as Bearing1_1. The matrix shows that the model performs well overall, with a low number of misclassifications, primarily occurring between adjacent classes.

5 | Limitations and Future Work

Although the model constructed in this study exhibits high accuracy and robustness in fault detection, there are limitations. Our 6-channel MC-DPCN, while beneficial for feature extraction and fault diagnosis, increases the computational requirements to some extent, which may limit its use in real-time applications with limited computational resources, such as Internet of Things (IoT) devices. For further exploration in reducing computational overhead, decreasing the time steps is one potential approach. In our work, the MC-DPCN utilizes 6 channels, with the time step T set to 6. On our PC equipped with an NVIDIA GeForce RTX 3070 Laptop GPU, the training time is 661 s (averaging 33.05 s per epoch), and the validation time on the test dataset is 340 s (averaging 17 s per epoch). When the time step is reduced to $T = 3$, the training time decreases to 480 s (averaging 24 s per epoch), while the validation time on the test dataset is 320 s (averaging 16 s per epoch). This demonstrates a certain degree of reduction in computational overhead.

Our future work will focus on extending the applicability of the model by working on optimizing the model architecture to reduce the computational load without compromising the diagnostic performance. In addition, the model can be integrated with IoT devices to facilitate remote monitoring and predictive maintenance. The functionality of the model can also be extended to handle different types of mechanical failures beyond bearings, further exploring the adaptability of the MC-DPCN framework to different industrial environments.

Moreover, the optimal hyperparameters for MC-DPCN on the CWRU dataset require further grid search. We plan to conduct a more refined search with a step size of 0.01 for the optimal parameters in our future work, which will demand significant computational time. This effort aims to enhance model performance while maintaining efficiency, contributing to the broader applicability of MC-DPCN in industrial settings.

6 | Conclusion

The MC-DPCN proposed in this study has 99.96% accuracy in normal vs. fault discrimination and 99.89% accuracy in fault type diagnosis. Excellent fault detection accuracy is achieved by expanding to 6-channel inputs, introducing a 6-channel inter-channel coupling mechanism and converting data from time series to spectrogram input models. Experimental results confirm that our method outperforms traditional methods and classical deep learning methods in terms of accuracy. This research is capable of bearing fault diagnosis tasks and provides a novel methodological framework for the field of mechanical fault diagnosis.

Author Contributions

Yanxi Wu: conceptualization, methodology, software, investigation, formal analysis, writing – original draft. **Yalin Yang:** data Curation. **Zhuoran Yang:** visualization, investigation. **Zhizhuo Yu:** visualization, investigation. **Jing Lian:** data curation, visualization, investigation. **Bin Li:** visualization, investigation. **Jizhao Liu:** conceptualization, funding acquisition, resources, supervision, writing – review & editing. **Kaiyuan Yang:** conceptualization, funding acquisition, resources, supervision.

Acknowledgements

This study is supported by the open fund project of National-local joint Engineering Laboratory of building health monitoring and disaster prevention technology under Grant GG23KF004 and funded in part by the Natural Science Foundation of Gansu Province under Grant 22JR5RA543 and 22JR5RA683. Some experiments are supported by the Supercomputing Center of Lanzhou University.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The datasets used in this study are publicly available. The CWRU bearing fault dataset can be accessed via the Case Western Reserve University Bearing Data Center at https://engineering.case.edu/bearingdatacenter. Additionally, the XJTU-SY bearing fault dataset is available at https://biaowang.tech/xjtu-sy-bearing-datasets/.

References

1. M. Mao, K. Zeng, Z. Tan, et al., "Adaptive VMD–K-SVD-Based Rolling Bearing Fault Signal Enhancement Study," *Sensors* 23, no. 20 (2023): 8629.

2. L. Jiang, H. Sheng, T. Yang, H. Tang, X. Li, and L. Gao, "A New Strategy for Bearing Health Assessment With a Dynamic Interval Prediction Model," *Sensors* 23, no. 18 (2023): 7696.

3. Y. Yao, T. Han, J. Yu, and M. Xie, "Uncertainty-Aware Deep Learning for Reliable Health Monitoring in Safety-Critical Energy Systems," *Energy* 291 (2024): 130419.

4. L. Shi, S. Su, W. Wang, S. Gao, and C. Chu, "Bearing Fault Diagnosis Method Based on Deep Learning and Health State Division," *Applied Sciences* 13, no. 13 (2023): 7424.

5. P. K. Kankar, S. C. Sharma, and S. P. Harsha, "Fault Diagnosis of Ball Bearings Using Machine Learning Methods," *Expert Systems With Applications* 38, no. 3 (2011): 1876–1886.

6. B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi, "Bearing Fault Detection Using Artificial Neural Networks and Genetic Algorithm," *EURASIP Journal on Advances in Signal Processing* 2004 (2004): 785672.

7. D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, and M. Modarres, "Deep Learning Enabled Fault Diagnosis Using Time-Frequency Image Analysis of Rolling Element Bearings," *Shock and Vibration* 2017 (2017): 5067651.

8. H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault Diagnosis of Rolling Bearings With Recurrent Neural Network-Based Autoencoders," *ISA Transactions* 77 (2018): 167–178.

9. S. Qiu, X. Cui, Z. Ping, et al., "Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review," *Sensors* 23, no. 3 (2023): 1305.

10. J. Liu, J. Lian, J. C. Sprott, Q. Liu, and Y. Ma, "The Butterfly Effect in Primary Visual Cortex," *IEEE Transactions on Computers* 71, no. 11 (2022): 2803–2815.

11. Z. Yu, T. Bu, Y. Zhang, S. Jia, T. Huang, and J. K. Liu, "Robust Decoding of Rich Dynamical Visual Scenes With Retinal Spikes," *IEEE Transactions on Neural Networks and Learning Systems* 36, no. 2 (2024): 3396–3409.

12. X. Zhang, J. Lian, Z. Yu, H. Tang, D. Liang, J. Liu, and J. K. Liu, "Revealing the Mechanisms of Semantic Satiation With Deep Learning Models," *Communications Biology* 7, no. 1 (2024): 487.

13. X. Ji, Z. Dong, C. S. Lai, and D. Qi, "A Brain-Inspired In-Memory Computing System For Neuronal Communication Via Memristive Circuits," *IEEE Communications Magazine* 60, no. 1 (2022): 100–106.

14. X. Ji, Z. Dong, Y. Han, C. S. Lai, and D. Qi, "A Brain-Inspired Hierarchical Interactive In-Memory Computing System and Its Application in Video Sentiment Analysis," *IEEE Transactions on Circuits and Systems for Video Technology* 33, no. 12 (2023): 7928–7942.

15. U. A. Syed, F. Kunwar, and M. Iqbal, "Guided Autowave Pulse Coupled Neural Network (GAPCNN) Based Real Time Path Planning and an Obstacle Avoidance Scheme For Mobile Robots," *Robotics and Autonomous Systems* 62, no. 4 (2014): 474–486.

16. B. Boashash, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference* (Academic Press, 2015).

17. R. N. Toma, A. E. Prosvirin, and J.-M. Kim, "Bearing Fault Diagnosis of Induction Motors Using a Genetic Algorithm and Machine Learning Classifiers," *Sensors* 20, no. 7 (2020): 1884.

18. Y. Wu, C. Li, S. Yang, and Y. Bai, "Multiscale Reduction Clustering of Vibration Signals for Unsupervised Diagnosis of Machine Faults," *Applied Soft Computing* 142 (2023): 110358.

19. Y. Lei, J. Lin, Z. He, and M. J. Zuo, "A Review on Empirical Mode Decomposition in Fault Diagnosis of Rotating Machinery," *Mechanical Systems and Signal Processing* 35, no. 1–2 (2013): 108–126.

20. C. Li, W. Ma, L. Sun, et al., "Hierarchical Deep Network With Uncertainty-Aware Semi-Supervised Learning For Vessel Segmentation," *Neural Computing and Applications* 34 (2022): 3151–3164.

21. M. T. Pham, J.-M. Kim, and C. H. Kim, "Deep Learning-Based Bearing Fault Diagnosis Method for Embedded Systems," *Sensors* 20, no. 23 (2020): 6886.

22. A. Khorram, M. Khalooei, and M. Rezghi, "End-to-End CNN+LSTM Deep Learning Approach for Bearing Fault Diagnosis," *Applied Intelligence* 51, no. 2 (2021): 736–751.

23. M. Gan, C. Wang, and C. Zhu, "Construction of Hierarchical Diagnosis Network Based on Deep Learning and Its Application in the Fault Pattern Recognition of Rolling Element Bearings," *Mechanical Systems and Signal Processing* 72 (2016): 92–104.

24. J. Sun, C. Yan, and J. Wen, "Intelligent Bearing Fault Diagnosis Method Combining Compressed Data Acquisition and Deep Learning," *IEEE Transactions on Instrumentation and Measurement* 67, no. 1 (2017): 185–195.

25. Z. Chen, A. Mauricio, W. Li, and K. Gryllias, "A Deep Learning Method for Bearing Fault Diagnosis Based on Cyclic Spectral Coherence and Convolutional Neural Networks," *Mechanical Systems and Signal Processing* 140 (2020): 106683.

26. L. Eren, T. Ince, and S. Kiranyaz, "A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier," *Journal of Signal Processing Systems* 91, no. 2 (2019): 179–189.

27. J. Miao, C. Deng, H. Zhang, and Q. Miao, "Interactive Channel Attention for Rotating Component Fault Detection With Strong Noise and Limited Data," *Applied Soft Computing* 138 (2023): 110171.

28. Z. Yi, J. Lian, Q. Liu, H. Zhu, D. Liang, and J. Liu, "Learning Rules in Spiking Neural Networks: A Survey," *Neurocomputing* 531 (2023): 163–179.

29. Z. Yang, J. Lian, and J. Liu, "Infrared UAV Target Detection Based on Continuous-Coupled Neural Network," *Micromachines* 14, no. 11 (2023): 2113.

30. C. Li, W. Li, H. Liu, et al., “Flaws Can be Applause: Unleashing Potential of Segmenting Ambiguous Objects in SAM,” in *The Thirty-Eighth Annual Conference on Neural Information Processing Systems* (ACM, 2024), 1–22.

31. J. Di, L. Ren, J. Liu, et al., “FDNet: An End-to-End Fusion Decomposition Network for Infrared and Visible Images,” *Plos One* 18, no. 9 (2023): e0290231.

32. T. Ma, J. Mou, A. A. Al-Barakati, H. Jahanshahi, and M. Miao, “Hidden Dynamics of Memristor-Coupled Neurons With Multi-Stability and Multi-Transient Hyperchaotic Behavior,” *Physica Scripta* 98, no. 10 (2023): 105202.

33. J. Lian, J. Liu, Z. Yang, et al., “A Pulse-Number-Adjustable MSPCNN and Its Image Enhancement Application,” *IEEE Access* 9 (2021): 161069–161086.

34. Y. Zheng, Z. Yu, S. Wang, and T. Huang, “Spike-Based Motion Estimation for Object Tracking Through Bio-Inspired Unsupervised Learning,” *IEEE Transactions on Image Processing* 32 (2022): 335–349.

35. C. Li, X. Liu, W. Li, et al., “U-KAN Makes Strong Backbone for Medical Image Segmentation and Generation,” *arXiv:2406.02918* (2024).

36. Q. Xu, Y. Li, J. Shen, J. K. Liu, H. Tang, and G. Pan, “Constructing Deep Spiking Neural Networks From Artificial Neural Networks With Knowledge Distillation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2023), 7886–7895.

37. Y. Li, X. Fang, Y. Gao, et al., “Efficient Structure Slimming for Spiking Neural Networks,” *IEEE Transactions on Artificial Intelligence* 5, no. 8 (2024): 3823–3831.

38. X. Wang, Y. Zhang, and Y. Zhang, “MT-SNN: Enhance Spiking Neural Network With Multiple Thresholds,” *arXiv:2303.11127* (2023).

39. T. Huang, Y. Zheng, Z. Yu, et al., “1000× Faster Camera and Machine Vision With Ordinary Devices,” *Engineering* 25 (2023): 110–119.

40. K. Malcolm and J. Casco-Rodriguez, “A Comprehensive Review of Spiking Neural Networks: Interpretation, Optimization, Efficiency, and Best Practices,” *arXiv:2303.10780* (2023).

41. Z. Dong, X. Ji, C. S. Lai, and D. Qi, “Design and Implementation of A Flexible Neuromorphic Computing System for Affective Communication via Memristive Circuits,” *IEEE Communications Magazine* 61, no. 1 (2022): 74–80.

42. X. Ji, Z. Dong, Y. Han, C. S. Lai, G. Zhou, and D. Qi, “EMSN: An Energy-Efficient Memristive Sequencer Network for Human Emotion Classification in Mental Health Monitoring,” *IEEE Transactions on Consumer Electronics* 69, no. 4 (2023): 1005–1016.

43. Z. Dong, X. Ji, G. Zhou, M. Gao, and D. Qi, “Multimodal Neuromorphic Sensory-Processing System With Memristor Circuits For Smart Home Applications,” *IEEE Transactions on Industry Applications* 59, no. 1 (2022): 47–58.

44. R. Eckhorn, H. Reitbock, M. Arndt, and P. Dicke, “A Neural Network for Feature Linking Via Synchronous Activity: Results From Cat Visual Cortex and From Simulations,” in *Models of Brain Function* (Cambridge University Press, 1989), 1–2.

45. H. Liu, M. Liu, D. Li, W. Zheng, L. Yin, and R. Wang, “Recent Advances in Pulse-Coupled Neural Networks With Applications in Image Processing,” *Electronics* 11, no. 20 (2022): 3264.

46. J. Lian, Y. Ma, Y. Ma, et al., “Automatic Gallbladder and Gallstone Regions Segmentation in Ultrasound Image,” *International Journal of Computer Assisted Radiology and Surgery* 12 (2017): 553–568.

47. Z. Yi, J. Lian, Y. Qi, et al., “Deep Pulse-Coupled Neural Networks,” *arXiv:2401.08649* (2023).

48. K. Zhan, J. Shi, H. Wang, Y. Xie, and Q. Li, “Computational Mechanisms of Pulse-Coupled Neural Networks: A Comprehensive Review,” *Archives of Computational Methods in Engineering* 24 (2017): 573–588.

49. J. Lian, Z. Yang, J. Liu, et al., “An Overview of Image Segmentation Based On Pulse-Coupled Neural Network,” *Archives of Computational Methods in Engineering* 28 (2021): 387–403.

50. Z. Wang, X. Sun, Y. Zhang, Z. Ying, and Y. Ma, “Leaf Recognition Based on PCNN,” *Neural Computing and Applications* 27 (2016): 899–908.

51. L. Sun, C. Li, X. Ding, et al., “Few-Shot Medical Image Segmentation Using a Global Correlation Network With Discriminative Embedding,” *Computers in Biology and Medicine* 140 (2022): 105067.

52. F. Léonard, “Phase Spectrogram and Frequency Spectrogram as New Diagnostic Tools,” *Mechanical Systems and Signal Processing* 21, no. 1 (2007): 125–137.

53. J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation* 19, no. 90 (1965): 297–301.

54. K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint arXiv:1409.1556* (2014).

55. L. Liu, H. Jiang, P. He, et al., “On The Variance of the Adaptive Learning Rate And Beyond,” *arXiv:1908.03265* (2019).

56. R. Yan, R. X. Gao, and X. Chen, “Wavelets for Fault Diagnosis of Rotary Machines: A Review With Applications,” *Signal Processing* 96 (2014): 1–15.

57. H. Shao, H. Jiang, X. Zhang, and M. Niu, “Rolling Bearing Fault Diagnosis Using an Optimization Deep Belief Network,” *Measurement Science and Technology* 26, no. 11 (2015): 115002.

58. L. Wen, X. Li, L. Gao, and Y. Zhang, “A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method,” *IEEE Transactions on Industrial Electronics* 65, no. 7 (2017): 5990–5998.

59. Z.-H. Liu, B.-L. Lu, H.-L. Wei, L. Chen, X.-H. Li, and M. Rätsch, “Deep Adversarial Domain Adaptation Model for Bearing Fault Diagnosis,” *IEEE Transactions on Ssystems, Man, and Cybernetics: Systems* 51, no. 7 (2019): 4217–4226.

60. N. Gharesi, M. M. Arefi, R. Razavi-Far, J. Zarei, and S. Yin, “A Neuro-Wavelet Based Approach for Diagnosing Bearing Defects,” *Advanced Engineering Informatics* 46 (2020): 101172.

61. X. Li, Y. Hu, J. Zheng, M. Li, and W. Ma, “Central Moment Discrepancy Based Domain Adaptation for Intelligent Bearing Fault Diagnosis,” *Neurocomputing* 429 (2021): 12–24.

62. Y. Xu, Z. Li, S. Wang, W. Li, T. Sarkodie-Gyan, and S. Feng, “A Hybrid Deep-Learning Model For Fault Diagnosis Of Rolling Bearings,” *Measurement* 169 (2021): 108502.

63. Z. Fan, Q. Xu, C. Jiang, and S. X. Ding, “Weighted Quantile Discrepancy-Based Deep Domain Adaptation Network For Intelligent Fault Diagnosis,” *Knowledge-Based Systems* 240 (2022): 108149.

64. J. Cui, Q. Zhong, S. Zheng, L. Peng, and J. Wen, “A Lightweight Model for Bearing Fault Diagnosis Based on Gramian Angular Field and Coordinate Attention,” *Machines* 10, no. 4 (2022): 282.

65. C. Huo, Q. Jiang, Y. Shen, C. Qian, and Q. Zhang, “New Transfer Learning Fault Diagnosis Method of Rolling Bearing Based on ADC-CNN and LATL Under Variable Conditions,” *Measurement* 188 (2022): 110587.

66. Y. Yoo, H. Jo, and S.-W. Ban, “Lite and Efficient Deep Learning Model for Bearing Fault Diagnosis Using the CWRU Dataset,” *Sensors* 23, no. 6 (2023): 3157.

67. A. E. Chaleshtori and A. Aghaie, “A Novel Bearing Fault Diagnosis Approach Using the Gaussian Mixture Model and the Weighted Principal Component Analysis,” *Reliability Engineering & System Safety* 242 (2024): 109720.

68. B. Wang, Y. Lei, N. Li, et al., “XJTU-SY Bearing Datasets,” (2018).