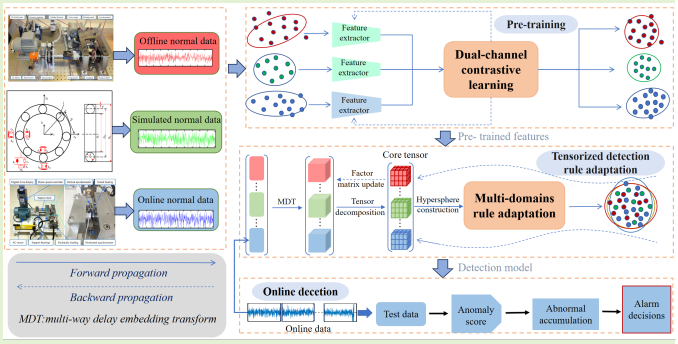
Dynamic Model-assisted Online Early Fault Detection Method of Rolling Bearings with Tensorized Multi-Domain Adaptation

Le Yang, Yuan Li, Shuang Jin, Yanyan Zhang, Wentao Mao *Member, IEEE*, Shubin Du

***Abstract*— In recent years, deep anomaly detection transfer learning techniques have successfully been applied to online early fault detection (EFD) under non-stopping scenarios. When facing irregular noise interference in online data, current methods generally suffer from extracting discriminative features for incipient fault, resulting in delayed alarm and a high false alarm rate. This paper proposes a dynamic model-assisted anomaly detection transfer learning method to extract discriminative features that are fault-sensitive and noise-robust. First, a new pre-training model with dual-channel contrastive learning is designed. Guided by the distribution of normal-state simulation data, this model reduces the distribution divergence between online and offline data. Then the pre-trained feature representation with better discriminative capability to fault occurrence is obtained. Second, an online EFD task is designed in which tensor Tucker decomposition is utilized to extract the core tensor for describing the anomaly detection rule. A multi-domain rule adaptation mechanism is proposed for collaborative adaptation among offline, online, and simulation data. Through an alternating optimization between rule adaptation and Tucker decomposition, the optimal domain-invariant feature representation that is robust to noise interference is extracted. Experiments on two public bearing datasets and an actual dataset of width-setting machine bearing from a large steel enterprise of China show that the proposed method not only significantly advances the alarm location compared to the existing methods, but also has a lower false alarm rate. The proposed method provides a reliable solution for online EFD of rolling bearings in practical industrial scenarios.**



***Index Terms*— Contrastive learning, Domain adaptation, Dynamic model, Early fault detection, Tensor decomposition**

I. Introduction

Early fault of rolling bearings usually refers to the initial early-stage defect caused by various factors such as wear, fatigue, improper installation, or contamination by foreign particles. Incipient faults will gradually progress and worsen over time. Early fault detection (EFD) is of great significance as it allows for preventive maintenance to be carried out, reducing unexpected downtime and extending the service life of the bearing and the associated equipment [1]. From a theoretical perspective, EFD is essentially a one-class classification problem of anomaly detection, where a detection rule is established using normal data to identify anomalies. Given the growth of artificial intelligence, a variety of machine learning algorithms, including support vector machines (SVMs) [2], Gaussian process [3], deep domain-adversarial contrastive network(DDCN) [4], and interpretable subdomain enhanced adaptive network (ISEANet) [5], have been successfully employed in resolving the EFD problem. For more details regarding the literature survey, please refer to Section 2.

This paper focuses on EFD in online scenarios (also called online EFD) that enable real-time assessment of status change under non-stopping scenarios and trigger an alarm as early as possible. Different from current EFD techniques, online EFD is dedicated to fast deployment without accumulating historical data in advance. That means identifying early fault occurrence mainly with sequentially collected monitoring data of target bearing. Thanks to this merit, online EFD can avoid delays caused by shutdown

This work was supported in part by National Natural Science Foundation of China under [Grant 62472146], in part by the Henan Province Science and Technology Research Project of China under [Grant 242102211014], in part by the Key Technologies Research Development Joint Foundation of Henan Province of China under [Grant 225101610001], and in part by Innovation and Entrepreneurship Training Program for College Students in Henan Province under [Grant 202410476051].

Corresponding author: Yuan Li (liyuan@htu.edu.cn)

Le Yang, Shuang Jin, and Yanyan Zhang are with the School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China.

Yuan Li and Wentao Mao are with the School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China, and also with the Engineering Lab of Intelligence Business and Internet of Things, Xinxiang 453007, Henan, China.

Shubin Du is with the Department of Equipment and Mechanical Power, HBIS Group Tangsteel Company, Tangshan 063000, China.

inspections and has more deployment potential for practical industrial applications. However, training EFD models directly with collected online data might result in model bias due to unexpected data fluctuation and irregular noise interference. For instance, training a support vector data description (SVDD) is fragile if the normal state data of the target bearing contains some outliers. In online settings, both outliers and the lack of sufficient pre-obtained normal-state information can significantly skew detection rules, leading to delayed alarms. To address this concern, transfer learning strategies are introduced to enhance detection performance on online data of target bearing (called target domain) by borrowing information of detection rule from offline data from auxiliary machines (called source domain). Due to abundant normal-state information in source-domain data, the negative impact of outliers can be effectively mitigated. The benefit of applying transfer learning to EFD, especially with deep neural networks (called deep transfer learning), has been proven in various fields, such as machine tools [6] and wind turbines [7] via adversarial training and domain adaptation [8]. This provides critical technical support for solving online EFD with insufficient training samples.

In practical industrial scenarios, due to the different mechanical configurations and severe interference of irregular noise, we found that existing transfer learning methods are susceptible to the following factors in addressing online EFD problems: (1) Large distribution divergence between online/offline data (e.g., due to run-in, lubrication, assembly) degrades domain-invariant feature extraction from normal data, reducing model fault sensitivity. For instance, irregular vibrations induced by the bearing run-in process will probably lead to a deviated classifier boundary, thereby making a certain tolerance for the real fault samples, and in turn causing alarm delay. (2) Noise interferences caused by transient shocks and time-varying lubricant viscosity in online scenarios, which are prone to be misclassified as faults, i.e., false alarms. Taking the width-setting machine, a core steel rolling equipment, as an example, the impact load during the slab entry process will raise transient noise with abrupt amplitude variations in vibration signals. The energy spectrum of such impulsive noise exhibits severe aliasing with incipient crack features in bearings, thus leading to exceptionally high false alarm rates at this location. How to obtain features sensitive to real faults and robust to noise is the key challenge for deep transfer learning in online EFD.

To address the aforementioned challenges, this paper adopts the following ideas: (1) Introducing a dynamic model to generate simulation data that reflects the operation mechanism of the target bearing during the normal stage. Such simulation data are expected to provide prior information for anomaly detection as a new domain (called the simulation domain). We believe that incorporating the simulation data (as illustrated in Fig. 1) will enhance the feature representation capability under noise interference while improving feature sensitivity to fault occurrence. (2) Employing tensor decomposition theory to design a core tensor-based rule adaptation mechanism among source domain, simulation domain, and target domain. Core tensors are capable of capturing essential information from monitoring data by suppressing interference from irregular noise. Aligning feature distribution and constructing detection rules based on core tensors will reduce the false alarm rate. This finding has been verified through our previous work [9].

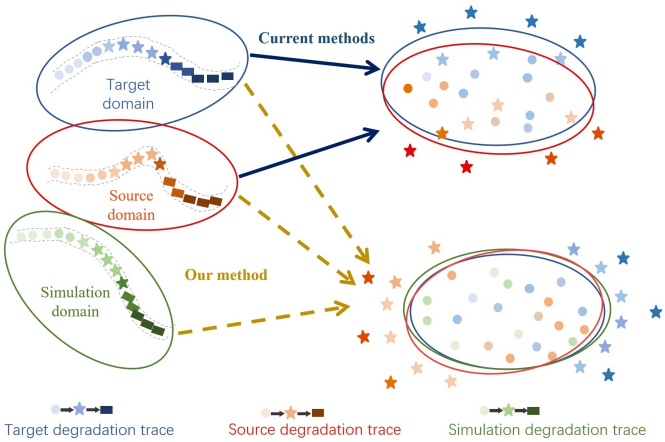


Fig. 1: Schematic diagram for introducing simulation data in transfer learning. Current domain adaptation techniques usually suffer from large distribution divergence between online and offline data in real-world industrial scenarios. The dynamic model is employed in this paper to serve as a guide for better domain adaptation.

Specifically, a novel transfer learning method for anomaly detection enhanced by a dynamic model and tensorized multi-domain adaptation is proposed. This method adopts a two-stage architecture of "pre-training & online" to improve detection performance as well as efficiency in online scenarios. In the pre-training stage, a dual-channel contrastive learning network is constructed to reduce distribution divergence among source, simulation, and target domains to obtain discriminative pre-trained features. In the online stage, a tensorized multi-domain rule adaptation mechanism is designed to improve the robust capability of features against noise interference. Please be aware that simulation data is generated using a 5-DOF differential equation dynamic model [10], though finite element, lumped parameter models or lower-freedom equations are also applicable. Experiments were carried out on two public bearing datasets (IEEE PHM 2012 and XJTU-SY) and a real-world dataset from a Chinese steel enterprise's width-setting machine bearing.

The main contributions of this study are as follows:

1) This paper finds a new way to incorporate simulation data generated by dynamic models into online EFD of rolling bearings. Unlike existing EFD methods using source and target domains for anomaly detection transfer learning, this method introduces an extra simulation domain and proposes a multi-domain transfer learning. The proposed method significantly improves the model's sensitivity to weak faults while remaining robust to irregular noise interference in a normal state. Literature review shows that this is the first attempt at applying a dynamic model to online EFD, while our work demonstrates great potential in practical industrial scenarios.

2) This paper proposes a new tensorized multi-domain

rule adaptation mechanism to realize the effective transfer of the detection rule. One-class anomaly detection rules are constructed by means of tensor decomposition, while the transfer of such rules is achieved by developing a simulation-domain-referenced rule adaptation strategy. Consequently, the sensitivity of the extracted features to fault occurrence, as well as the model's robustness to irregular noise interference, is highly preserved.

The organization of the rest of this paper is as follows. In Section 2, previous relevant works are reviewed and discussed. Section 3 provides a detailed introduction to the proposed methodology. Section 4 presents and analyzes the experimental results, and Section 5 concludes with the closing remarks.

II. Related works

Existing EFD methods include the methods based on signal analysis, the methods based on machine learning, and the methods based on deep learning. Most signal analysis-based methods focus on the statistical characteristics of signals in time, frequency, or time-frequency domains [11], [12]. However, weak fault signals are easily masked by noise in real-world industrial scenarios. Despite high accuracy in identifying fault characteristic frequency, the alarm is usually triggered only if the fault evolves to a certain extent. This limit can be found in the section Experiment. Machine learning-based methods also depend on the statistical characteristics of signals. Classic algorithms such as SVM [2] and Gaussian process [3] are utilized to construct classification models. However, since the selection of features relies on expert experience, inappropriate features will negatively affect the performance of the model. Deep learning-based methods introduce deep neural network architectures, including convolutional neural networks (CNN) [1], stacked long short-term memory networks (LSTM) [13], stacked deep auto-encoders (SDAE) [14], and ISEANet [5], into the field to carry out end-to-end fault detection by means of self-adaptive feature extraction. Nevertheless, these methods require large-scale training data. The high cost of industrial data acquisition often makes it difficult to meet the training needs of large-scale data. In a few-sample setting, Liu et al. [15] proposes a multiscale residual antinoise network (MRANet) via using multi-branch dilated convolution and improved residual blocks to extract the shallow mechanism and deep discriminable features. Due to the scarcity of fault data in real-world industrial scenarios, fault samples can be regarded as anomalous data relative to normal state data. Therefore, EFD can be treated as a one-class anomaly detection problem, with commonly employed methods such as one-class SVM [16], SVDD [17], and dictionary learning [18]. Ruff et al. [19] developed a classical framework where they combined SVDD with CNN and put forward the Deep SVDD method. This method constructs a hypersphere decision boundary on the basis of self-adaptive feature extraction. However, the detection rules constructed by these methods are highly susceptible to irregular noise interference, resulting in model bias and failing to meet the detection requirements of EFD in online scenarios.

For the online EFD problem, since the monitoring data is collected sequentially, it is hard to accumulate sufficient samples for training. To alleviate the dependence on online data, various deep transfer learning methods [20], [21] have been applied to EFD. The essence of these methods is the transfer and re-utilization of domain knowledge. Most current researchers worked on the Deep SVDD framework to construct a one-class anomaly detection model by utilizing transfer strategies such as adversarial training and domain adaptation [8]. Yang et al. [8] proposed an invariant representation anomaly detection (IRAD) methodology. By means of adversarial learning mechanisms, it enables the extraction of invariant feature representations from normal-state data. Han et al. [22] proposed the Log-TAD method that utilizes an adversarial training strategy to achieve cross-domain common feature extraction. Subsequently, an anomaly detection model for the target domain is developed. Michau et al. [23] employed a domain adversarial training strategy to design an unsupervised one-class transfer learning framework, which is capable of learning all positive-class features across different scenarios in source domains and identifying the location of anomalies from unlabeled data. Wu et al. [4] constructs a deep domain-adversarial contrastive network (DDCN) to realize selective information transfer according to frequency band's significance. Xie et al. [24] proposed a transfer learning framework that integrates SVM with one-class dictionary learning to derive a discriminative rule. In recent years, pre-training techniques have been introduced into deep transfer learning methods to enhance the stability of adversarial training. For instance, Ding et al. [25] employed contrastive learning to derive discriminative representations from extensive unlabeled datasets, followed by parameter optimization for the EFD task. Mao et al. [26] designs a new tensorized rule adaptation mechanism, which is applied to pre-training to learn the task-invariant detection rule. Liu et al. [27] enhanced the alignment of domain features by combining pre-trained features extracted from time and frequency domain information. Chen et al. [28] and Zhao et al. [29] showed that the pre-training model can improve the numerical stability of adversarial training and speed up the training of network models in the anomaly detection phase. In summary, the key idea of these methods is to find domain-invariant feature representation for normal state data through domain adversarial training. However, the transfer effect relies on a large amount of source-domain data to extract and represent domain knowledge. Once the source-domain data is of poor quality, model bias is inevitable.

To address the concern about poor-quality training data in EFD, e.g., involving irregular noise in monitoring data, the simulation model grounded in physical principles offers a novel perspective for enhancing feature representation in deep learning. For rolling bearings, simulation data generated from a dynamic model replicates the ideal operational states under various working conditions, i.e., free from the interference of irregular noise. Incorporating simulation data into training data is beneficial to assist the model in decoupling noise from feature representation at the physical level. At present, there are many successful applications of simulation data in bearing fault diagnosis. For instance, Yu et al. [30] applied the diagnostic knowledge derived from the simulation data of

a rotor-bearing system to achieve intelligent fault diagnosis. Mei et al. [31] developed a fault signal simulation model capable of accurately replicating different damage levels on inner and outer rings and introduced a novel bearing fault diagnosis method driven by simulation data. The accuracy of fault diagnosis is improved by generating extensive simulation data across various fault types. Simulation data that models the degradation process has also been utilized to handle the challenge of remaining useful life prediction. For example, Chen et al. [32] proposed a reliability assessment method that combines numerical simulation with zero-fault data for the life prediction of aero-engine compressor disks. From these studies, the accuracy of feature representation can be improved by using simulation data. Exploring the application of simulation data in the EFD field to mitigate the potential model bias is a worthwhile approach. To the best of our knowledge, the study about dynamic model or simulation data-assisted EFD has not yet been found.

Irregular noise in online data not only degrades the accuracy of discriminative information but also reduces feature robustness, even raising false alarms. Tensor decomposition, as a powerful tool for processing high-dimensional data, can effectively extract core information while suppressing the impact of noise interference. For instance, Hu et al. [33] proposed a fault diagnosis method based on tensor alignment. By aligning the underlying tensor subspaces of the source and target domains, this method effectively retains discriminative features while filtering out irrelevant noise, thereby enhancing the effectiveness of cross-domain fault diagnosis. Liu et al. [34] reconstructed the feature tensor into matrix form through the joint optimization of factor matrices and tensor-invariant subspaces for fault diagnosis. In this process, the tensor decomposition technique effectively separates noise components from the signal and extracts key features. The support tensor machine, proposed by He et al. [35], mitigated the impact of noise and improved classification performance by leveraging tensor decomposition techniques when processing high-order monitoring data with imbalanced distributions. Mao et al. [9] integrated tensor decomposition into EFD and designed a novel tensor hypersphere adversarial network, demonstrating the advantages of core tensor information in characterizing detection rules. Considering the capability of optimizing feature representation, tensor decomposition is applied to the online EFD problem to improve the robustness of detection model.

Despite successful applications of deep transfer learning methods into EFD, they still have the following limitations: (1) Current methods are negatively affected by the joint effects of different mechanical configurations, heterogeneous testing environments, and divergent operational conditions, thus leading to a large distribution divergence of data between source domain and target domain. (2) Irregular noise interference in the online data leads to the lack of robustness of the extracted features, which easily triggers an increased false alarm rate.

III. Methodology

This section presents a multi-domain transfer learning framework using simulation data, as shown in Fig. 2. This framework includes two stages: (1) Pre-training stage with dual-channel contrastive learning, which aims to obtain good pre-trained features; (2) Online anomaly detection stage, which is devoted to fast and efficient transfer of detection rules based on tensor representation. The key to the method is to extract fault-sensitive and noise-robust features with the help of mechanism information and tensor decomposition. The specific implementations are as follows.

A. Problem description

The three domains involved in our method are defined as follows: 1) The operating data of the bearing in the normal state collected offline is set as the source domain(abbreviated as SD), such as the data from machines in laboratories or factories. 2) The data from another machine is taken as the target domain((abbreviated as TD). 3) The data simulated by a dynamic model corresponding to the TD is introduced as the simulation domain((abbreviated as SimD). Assume that:

1) There is one machine in SD, The normal-state data from SD is represented by $X_s = \left\{X_s^1, X_s^2, \ldots, X_s^{n_s}\right\}$, where the superscript n_s denotes the number of offline samples, the subscript s specifies SD. For the i-th sample $X_s^i = \left\{x_s^{ij}\right\}_{j=1}^{m}$, m denotes the sample feature dimension, i.e., the number of sampling points contained in each sample. The SD, denoted by $D_s = \{\chi_s,\ P(\chi_s)\}$, is composed of the sample space χ_s and the corresponding data distribution $P(\chi_s)$.

2) The normal-state data from TD is denoted by $X_t = \left\{X_t^1, X_t^2, \ldots, X_t^{n_t}\right\}$, where the subscript t specifies TD. Similarly, the TD is denoted by $D_t = \{\chi_t,\ P(\chi_t)\}$, where $P(\chi_t)$ represent the corresponding data distribution.

3) The simulation data is denoted by $X_{sim} = \left\{X_{sim}^1, X_{sim}^2, \ldots, X_{sim}^{n_{sim}}\right\}$, where the subscript sim denotes SimD. And the SimD D_{sim} is denoted by $D_{sim} = \{\chi_{sim},\ P(\chi_{sim})\}$, where $P(\chi_{sim})$ represents the corresponding data distribution.

4) For the TD, the goal of constructing detection rules is to find an optimal mapping function $f : \chi_t \rightarrow \gamma_t$, where γ_t represents the state space. Due to the discrepancy between production parameters and operational environments of offline machinery and idealized modeling for simulation, $P(\chi_s) \neq P(\chi_t) \neq P(\chi_{sim})$. But the detection rules of the three domains are inherently identical because they follow the same dynamic law of rolling bearings. The rules obtained from D_s and D_{sim} are then considered to be beneficial in constructing the rules of D_t.

B. Preparation of simulation data

The dynamic model [10] of rolling bearing in Fig. 3(a) is adopted to generate simulation data of vibration signals in the normal state. This model simplifies the rolling bearing to a 5-DOF system in which the horizontal displacement x_s and vertical displacement y_s in the inner ring, the horizontal displacement x_p and vertical displacement y_p in the outer ring, and the vertical displacement y_r at the resonator are involved.

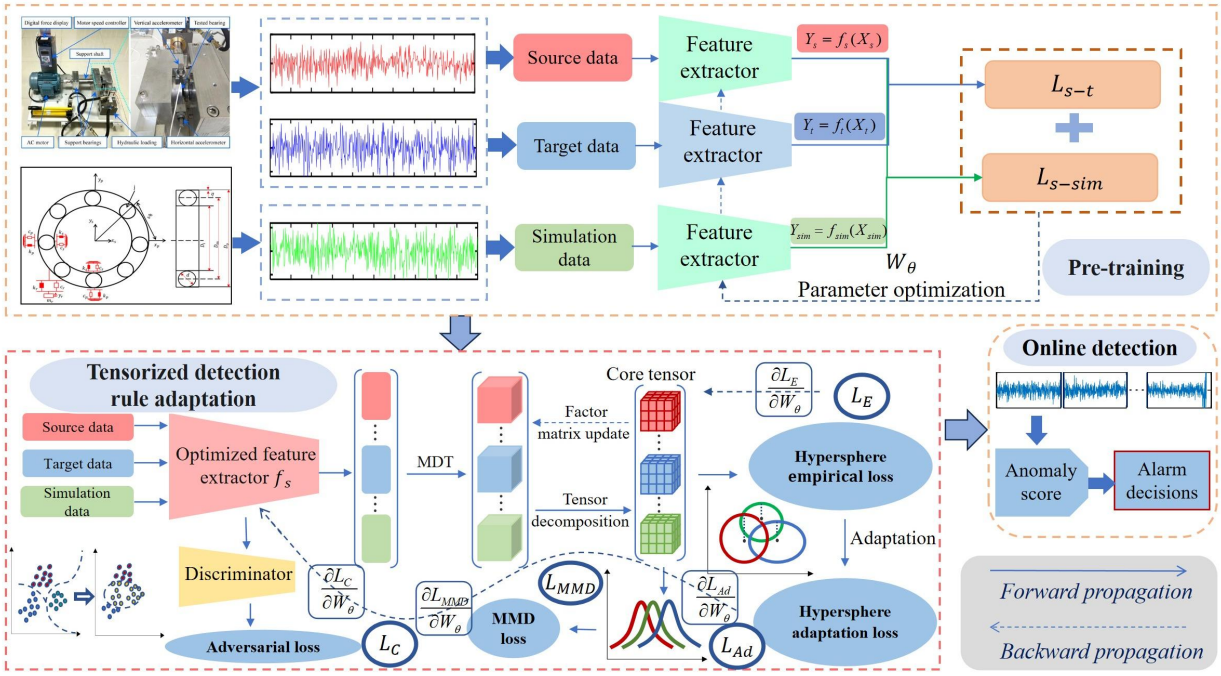


Fig. 2: Diagrammatic representation of the suggested approach

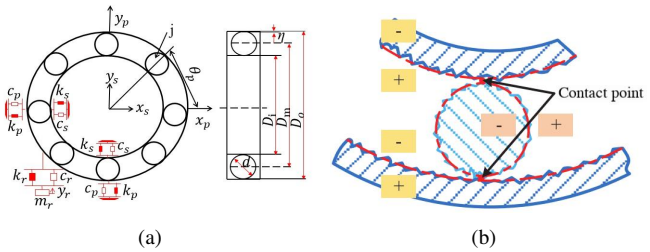


Fig. 3: Schematic diagram of a dynamic 5-DOF system of rolling bearing, in which (a) is the overall view and (b) is the partially enlarged view corresponding to contact surface [10]

$$\begin{cases} m_s\ddot{x}_s + c_s\dot{x}_s + k_s x_s + f_x = 0 \\ m_s\ddot{y}_s + c_s\dot{y}_s + k_s y_s + f_y = F_s + m_s g \\ m_p\ddot{x}_p + c_p\dot{x}_p + k_p x_p = f_x \\ m_p\ddot{y}_p + (c_p + c_r)\dot{y}_p + (k_p + k_r)x_p = \\ \quad k_r y_r + c_r\dot{y}_r + f_y + m_p g \\ m_p\ddot{y}_p + c_r\dot{y}_p + k_r y_r = k_r y_p + c_r\dot{y}_p \end{cases} \tag{1}$$

where m, k, and c represent the equivalent mass, equivalent stiffness, and equivalent damping of the system, respectively. The subscripts s, p, and r represent the inner ring, outer ring, and resonator, respectively. x and y represent horizontal and vertical vibration displacement, respectively. Fs is the external radial load acting on the inner ring, and f is the nonlinear contact force. Please refer to Ref. [10] for the details of calculating f. The numerical results of Eq. (1) can be obtained with the Runge-Kutta method. It is worth noting that the formula of f in Eq. (1) takes into account the roughness between contact surfaces (as shown in Fig. 3(b)), which can simulate the vibration response caused by manufacturing errors of bearings. This provides conditions for simulating the vibration of rolling bearings in the normal state.

C. Pre-training with dual-channel contrastive learning

To facilitate the training of online detection task, this section designs a dual-channel contrastive learning model for pre-training, as shown in Fig. 4, to extract pre-trained features for the SD, TD, and SimD. The pre-trained features are used to provide an initial detection rule for the online task, with the training cost much reduced.

In a specific implementation, this paper adopts the classic contrastive learning model, Bootstrap Your Own Latent (BYOL) [36], as the basic framework. Different from the traditional contrastive learning with two domains (i.e., SD and TD), this paper introduces an additional SimD into this framework to construct a dual-channel contrastive learning network: the contrastive channel between the SD and the TD, and the contrastive channel between the SD and the SimD.

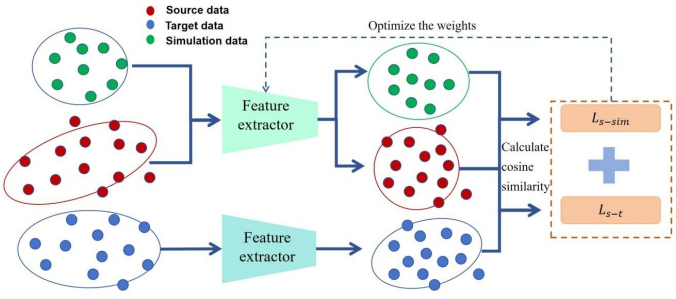


Fig. 4: Diagram of the dual-channel contrastive learning model for pre-training

Considering the characteristic of simulation data, we believe that the data distribution divergence between the SD and TD is larger than that between the TD and SimD. Specifically, set an individual feature extractor shared by the SimD and the SD, and another feature extractor for the TD. This design not only avoids the increase of model complexity in building an extra feature extractor for the SimD, but also speeds up the training process by sharing the feature space. Due to space limitation, we will not elaborate on the details of the whole pre-training algorithm. Here we only provide the specific implementation in the BYOL framework, as follows.

First, construct an SD feature extractor $f_s(W_s)$ using D_s and D_{sim} jointly, and a TD feature extractor $f_t(W_t)$ using D_t independently, where W_s and W_t are the parameters of f_s and f_t, respectively. The features of the SD, TD, and SimD are represented as $Y_s = f_s(X_s)$, $Y_t = f_t(X_t)$, and $Y_{sim} = f_s(X_{sim})$, respectively.

Second, calculate the cosine similarities between the features of SD and TD and between the features of SD and SimD in the following two loss functions, respectively:

$$\mathcal{L}_{s-t} = 1 - \text{cosine\_similarity}(Y_s,\ Y_t) \tag{2}$$

$$\mathcal{L}_{s-sim} = 1 - \text{cosine\_similarity}(Y_s,\ Y_{sim}) \tag{3}$$

Consequently, the total loss function is obtained as:

$$\mathcal{L}_{\text{contrastive}} = \mathcal{L}_{s-t} + \mathcal{L}_{s-sim} \tag{4}$$

By minimizing $\mathcal{L}_{\text{contrastive}}$, we optimize the parameters of $f_s(W_s)$ and $f_t(W_t)$ for reducing the distribution divergence of the three domains. $f_s(W_s^*)$ is finally selected as the initial feature extractor for the online detection task, where W_s^* is the optimized weights of the f_s.

D. Online detection task with tensorized multi-domain adaptation

In this section, an anomaly detection transfer learning model is designed based on the pre-trained features. The specific implementation is as follows: 1) tensor hypersphere construction, which is to construct a hypersphere-form detection rule on the basis of core tensors extracted from the features of the three domains; 2) multi-domain rule adaptation, which is to adapt the detection rules by overlapping the hyperspheres of the three domains as much as possible in an adversarial training scheme. During the domain adaptation process, the SimD is serving as a prototype to guide the feature alignment among the three domains, i.e., $collaborative\ \ adaptation$.

1) Tensor hypersphere construction: First, the feature extractor $f_s(W_s^*)$ obtained through pre-training is employed as the feature extractor of the anomaly detection model. The aim of $f_s(W_s^*)$ is to find a mapping $\phi(W_\theta) : X \to \Psi$, where W_θ represents the network weights. Input the data from $X = \{X_s,\ X_{sim},\ X_t\}$ into $f_s(W_s^*)$, and the output of features in a shared subspace Ψ are $F = \{F_s,\ F_{sim},\ F_t\}$, where $F_s,\ F_{sim},\ F_t$ are respectively the features of the SD data, the SimD data, and the TD data .

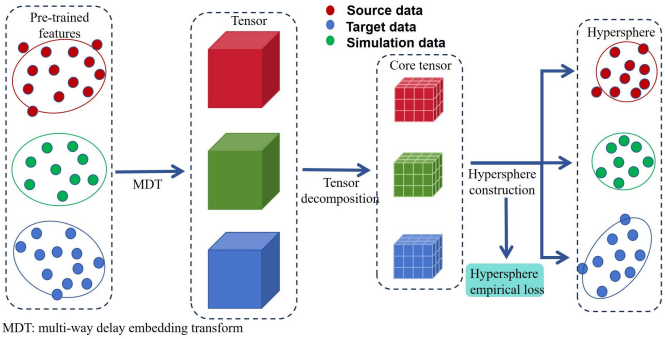


Fig. 5: Diagram of tensor Tucker decomposition for hypersphere construction

To further explore the high-order relationships between features and reduce the data dimensionality, F is transformed into a tensor form. Subsequently, tensor Tucker decomposition is conducted on the tensors to extract their core tensors that retain the essential information of raw data, providing a more concise and efficient feature representation for subsequent hypersphere construction, as shown in Fig. 5.

Taking the features from the SD as an example, we transform F_s into a high-order tensor $T_s = \{\chi_1 \cdots \chi_{n\_core}\}$ along the time direction using multi-way delay embedding transform (MDT) [37], in which:

$$\chi_1 = H(F_s) = Fold_{(n_s,\tau)}(F_s \times_1 S_1 \times \ldots \times_{n\_core} S_{n\_core}) \tag{5}$$

where τ is the embedding dimension, n_s is sample length, $n\_core = n - \tau + 1$ reprensents the sample number subsequent to the MDT, where S denotes the mapping matrix. Similarly, the tensors T_t and T_{sim} are obtained for the TD and SimD. Then, use Tucker decomposition [38] technique to get the core tensor for each tensor, as (take SD as an example):

$$g^s = T_s \times_1 U_1 \times_2 U_2 \times \cdots \times_N U_N \tag{6}$$

where g^s is the core tensor of SD, $\{U_i\}_{i=1}^N$ is the factor matrix of mode i, and $\times_i$ represents the tensor-matrix multiplication in mode i. Similar to Eq. (6), the core tensors g^t and g^{sim} can be obtained from T_t and T_{sim}, respectively.

With reference to the concept of the hypersphere in the SVDD [17], tensor-based hyperspheres are constructed respectively utilizing g^s, g^t, and g^{sim}, which contain most normal-state samples. Through minimizing the hyperspheres' volume, the decision boundary for anomaly detection, namely the detection rule, can then be obtained. Specifically, the minimization objective for the hypersphere of the SD is:

$$L_E^s(R^s,\ W_\theta) = (R^s)^2 + \frac{1}{n\_core^s} \sum_{i=1}^{n\_core^s} \max\left\{0, \|g_i^s - C^s\|^2 - (R^s)^2\right\} \tag{7}$$

For the hypersphere of the TD:

$$L_E^t(R^t,\ W_\theta) = \left(R^t\right)^2 + \frac{1}{n\_core^t}\sum_{j=1}^{n\_core^t}\max\left\{0, \|g_j^t - C^t\|^2 - \left(R^t\right)^2\right\} \tag{8}$$

And for the hypersphere of the SimD:

$$L_E^{sim}\ \left(R^{sim}, W_\theta\right) = \left(R^{sim}\right)^2 + \frac{1}{n\_core^{sim}}\sum_{i=1}^{n\_core^{sim}}\max\left\{0, \|g_i^{sim} - C^{sim}\|^2 - \left(R^{sim}\right)^2\right\} \tag{9}$$

where n_core^s, n_core^t and n_core^{sim} represent the number of samples after MDT from the SD, TD, and SimD data, respectively. C^s, C^t, and C^{sim} are the centers of the three hyperspheres, and R^s, R^t, and R^{sim} are the corresponding radii.

The total empirical loss of the hyperspheres is expressed as:

$$L_E(R,\ W_\theta) = L_E^s(R^s,\ W_\theta) + L_E^t(R^t,\ W_\theta) + L_E^{sim}(R^{sim},\ W_\theta) \tag{10}$$

The purpose of minimizing Eq. (10) is to minimize the volume of the hyperspheres as much as possible, and impose penalties on the samples outside the boundaries of the hyperspheres as well. This approach preserves the generalization ability of the detection model.

2) Simulation-domain-referenced multi-domain rule adaptation: Based on the obtained tensor hypersphere, a multi-domain rule adaptation strategy is designed to achieve the effective transfer of detection rules. The core of this strategy is introducing the simulation data as a rule prototype under ideal operating conditions. Then the simulation data is able to guide the extraction of the domain-invariant feature representation in a more discriminative direction. Meanwhile, domain adversarial training and the maximum mean difference (MMD) constraint are utilized to jointly complete the feature alignment of the three domains, as shown in Fig. 6.

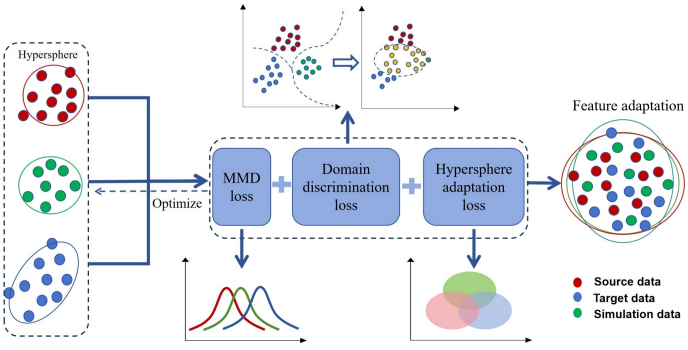


Fig. 6: Diagram of the proposed multi-domain rule adaptation mechanism

From a geometric perspective, the optimization objective of rule adaption is equivalent to minimizing the hypersphere parameter difference across the domains. Thus, the hypersphere adaptation loss is formulated as:

$$\begin{cases} L_1\left(R,\ W_\theta\right) = \|C^s - C^t\|^2 + |(R^s)^2 - (R^t)^2| \\ L_2\left(R,\ W_\theta\right) = \left\|C^s - C^{sim}\right\|^2 + |(R^s)^2 - \left(R^{sim}\right)^2| \\ L_3\left(R,\ W_\theta\right) = \left\|C^t - C^{sim}\right\|^2 + |(R^t)^2 - \left(R^{sim}\right)^2| \end{cases} \tag{11}$$

$$L_{Ad}(R,\ W_\theta) = L_1 + \gamma_1 L_2 + \gamma_2 L_3 \tag{12}$$

where L_1 , L_2, and L_3 represent the hypersphere divergences between the SD and TD, between the SD and SimD and between the SimD and TD respectively. The hypersphere divergence is evaluated by the hypersphere parameters (C and R). γ_1 and γ_2 ($\gamma_1, \gamma_2 > 1$) are regularization parameters. As shown in Fig. 7, when $\gamma_1, \gamma_2 > 1$, the model narrows the gap between the hyperspheres of TD and SD with that of the SimD, ultimately achieving that SD and TD are guided to align with SimD. Consequently, minimizing Eq. (12) is able to achieve an enhanced cross-domain adaptation.

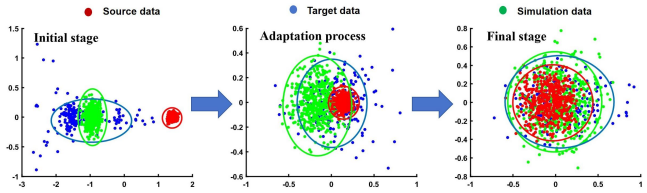


Fig. 7: Multi-domain adaptation process guided by simulation domain.

In domain adversarial training, the role of domain classifier is to discriminate whether the training data originates from the SD, the TD, or the SimD. The corresponding loss function is as follows:

$$L_{\mathrm{C}}(W_{DC}) = \sum_{x\sim X_s}\left[\log DC(F_s)\right] + \sum_{x\sim X_t}\left[\log DC(F_t)\right] + \sum_{x\sim X_{\mathrm{sim}}}\left[\log DC(F_{\mathrm{sim}})\right] \tag{13}$$

where W_{DC} denotes the model parameters of the domain classifier, and $DC(\cdot)$ represents the domain classifier. By maximizing Eq. (13), the model is prevented from distinguishing the originating domains (SD, TD, or SimD) of the data. This mechanism achieves alignment of feature distributions across the three domains, thereby enhancing the model's generalization capability in cross-domain scenarios.

Usually speaking, it is difficult for the adversarial training process to converge when there is a large divergence in data distribution. MMD constraint is imposed on the core tensor of the three domains to accelerate domain adaptation. The total MMD loss is formulated as:

$$L_{MMD}(W_\theta) = L_{MMD1} + L_{MMD2} + L_{MMD3} \tag{14}$$

where:

$$\begin{aligned} L_{MMD1}(W_\theta) &= \left\| \frac{1}{n\_core^s} \sum_{i=1}^{n\_core^s} g_i^s - \frac{1}{n\_core^t} \sum_{j=1}^{n\_core^t} g_j^t \right\| \\ L_{MMD2}(W_\theta) &= \left\| \frac{1}{n\_core^t} \sum_{i=1}^{n\_core^t} g_i^t - \frac{1}{n\_core^{sim}} \sum_{j=1}^{n\_core^{sim}} g_j^{sim} \right\| \\ L_{MMD3}(W_\theta) &= \left\| \frac{1}{n\_core^s} \sum_{i=1}^{n\_core^s} g_i^s - \frac{1}{n\_core^{sim}} \sum_{j=1}^{n\_core^{sim}} g_j^{sim} \right\| \end{aligned} \tag{15}$$

Minimizing Eq. (14) can effectively reduce the distribution divergence among the three domains data in the high-dimensional common space.

In summary, the optimization objective for the anomaly detection transfer learning model is formulated as:

$$\min_{W_\theta, R^S, R^T, R^{Sim}} \max_{W_{DC}} = L_E + \lambda_1 L_{Ad} + \lambda_2 L_C + \lambda_3 L_{MMD} \tag{16}$$

where λ_i $(i = 1,\ 2,\ 3)$ denotes the weighting hyperparameter for balancing the contributions of different loss components.

E. Model training of online EFD

Eq. (16) is a multi-objective optimization problem that involves both iterative updating of the network model parameter weight $W = \{W_\theta,\ W_{DC}\}$ and radius $R = \{R^s,\ R^t,\ R^{sim}\}$, as well as the optimization of the tensor representation. Since the model parameters and the tensor representation are optimized respectively by stochastic gradient descent (SGD) and alternating least squares method (ALSM) [39], we adopt an alternating optimization strategy: the tensor factor matrices $\{U_i\}_{i=1}^N$ are first fixed, and the W and R are optimized by SGD; Then, W and R are fixed, and $\{U_i\}_{i=1}^N$ is updated by using the ALSM. The two steps alternate and continue until convergence is achieved. When the difference between the factor matrices obtained from two consecutive tensor optimizations is less than a pre-fixed threshold, the update of $\{U_i\}_{i=1}^N$ stops, and both W and R are updated until the model reaches convergence. The systematic approach to optimization is described below:

1) Fix $\{U_i\}_{i=1}^N$, and update W and R

Considering the $\{W_\theta,\ W_{DC}\}$ and $\{R^s,\ R^t,\ R^{sim}\}$ have different dimensions and are coupled, the alternating optimization strategy is further utilized here: First, given the initial $\{W_\theta,\ W_{DC}\}$ and $\{R^s,\ R^t,\ R^{sim}\}$, the weights $\{W_\theta,\ W_{DC}\}$ are updated. Second, $\{R^s,\ R^t,\ R^{sim}\}$ are updated with the updated weights. Iteratively update the above process until convergence.

The optimization of $W_\theta,\ W_{DC}$ is performed using the SGD method as:

$$\begin{aligned} W_\theta &\leftarrow W_\theta - \eta\Big(\frac{\partial L_{\mathrm{E}}}{\partial W_\theta} + \lambda_1 \frac{\partial L_{\mathrm{Ad}}}{\partial W_\theta} + \lambda_3 \frac{\partial L_{\mathrm{MMD}}}{\partial W_\theta} - \lambda_2 \frac{\partial L_{\mathrm{C}}}{\partial W_\theta}\Big) \\ W_{DC} &\leftarrow W_{DC} - \eta \frac{\partial L_{\mathrm{C}}}{\partial W_{DC}} \end{aligned} \tag{17}$$

where η indicates the learning rate. Then, $\{R^s,\ R^t,\ R^{sim}\}$ is updated as follows:

$$\begin{aligned} R^s &\leftarrow R^s - \eta(\tfrac{\partial L_E}{\partial R^s} + \lambda_1 \tfrac{\partial L_{Ad}}{\partial R^s}) \\ R^t &\leftarrow R^t - \eta(\tfrac{\partial L_E}{\partial R^t} + \lambda_1 \tfrac{\partial L_{Ad}}{\partial R^t}) \\ R^{sim} &\leftarrow R^{sim} - \eta(\tfrac{\partial L_E}{\partial R^{sim}} + \lambda_1 \tfrac{\partial L_{Ad}}{\partial R^{sim}}) \end{aligned} \tag{18}$$

where:

$$\begin{aligned} &\tfrac{\partial L_E}{\partial R^s} = 2\left(1 - \tfrac{n_{out}^s}{n^s}\right) R^s, \quad \tfrac{\partial L_{Ad}}{\partial R^s} = (1+\gamma_1)\, R^s \\ &\tfrac{\partial L_E}{\partial R^t} = 2\left(1 - \tfrac{n_{out}^t}{n^t}\right) R^t, \quad \tfrac{\partial L_{Ad}}{\partial R^t} = (1+\gamma_2)\, R^t \\ &\tfrac{\partial L_E}{\partial R^{sim}} = 2\left(1 - \tfrac{n_{out}^{sim}}{n^{sim}}\right) R^{sim}, \quad \tfrac{\partial L_{Ad}}{\partial R^{sim}} = (\gamma_1+\gamma_2)\, R^{sim} \end{aligned} \tag{19}$$

in which, n_{out}^s, n_{out}^t, and n_{out}^{sim} respectively represent the number of samples outside the hypersphere in the SD, TD and SimD.

2) Fix W and R and update the $\{U_i\}_{i=1}^N$

After $\{U_i\}_{i=1}^N$ randomly initializes, we can optimize the core tensor by minimizing the tensor reconstruction error (Re_error) as follows:

$$\min Re\_error = \left\| T - \hat{T} \right\|_F^2 \tag{20}$$

where T is the original tensor, $\hat{T} = g \times_1 U_1 \times_2 U_2 \times \cdots \times_N U_N$ be the reconstructed tensor from core tensor g of T via Eq. (6), and $\|\cdot\|_F$ is the Frobenius norm. Since $\{U_i\}_{i=1}^N$ is an orthogonal factor matrix, Eq. (20) can be written as:

$$\begin{aligned} \left\| T - \hat{T} \right\|_F^2 &= \|\mathrm{vec}(T) - (U_1 \otimes U_2 \otimes \cdots \otimes U_N) \cdot \mathrm{vec}(g)\|_F^2 \\ &= \|\mathrm{vec}(T) - (U_1 \otimes \cdots \otimes U_N) \\ &\quad \cdot (U_1 \otimes \cdots \otimes U_N)^T \cdot \mathrm{vec}(T)\|_F^2 \\ &= \|\mathrm{vec}(T)\|_F^2 - \left\|(U_1 \otimes \cdots \otimes U_N)^T \cdot \mathrm{vec}(T)\right\|_F^2 \end{aligned} \tag{21}$$

To minimize Eq. (21), we can maximize $\left\|(U_1 \otimes U_2 \otimes \cdots \otimes U_N)^T \cdot vec(T)\right\|_F^2$ by using the ALSM.

This optimization process is summarized in **Algorithm 1**.

For online task, the online data $X_{test} = \left\{X_{test}^i\right\}_{i=1}^{n_{test}}$ are sequentially input into $\phi(x, W_\theta^*)$ to determine whether an early fault has occurred by calculating the anomaly score as follows:

$$Score(x_{test}) = \left\| \phi(x_{test}, W_\theta^*) - \tilde{C}^t \right\|^2 - \tilde{R}^t \tag{22}$$

where $\tilde{C}^t$ and $\tilde{R}^t$ represent the center and radius of the target hypersphere reconstructed using X_t and W_θ^*. $Score(x_{test}) \leq 0$ indicates that X_{test} locates inside the hypersphere and is classified as being in normal state, otherwise, as an anomaly.

In the proposed method, the discriminative features are extracted through the following ways: 1) Introduce simulation domain from a dynamic model (Eq. (1)) to the domain adaptation process in transfer learning. Through dual-channel contrastive learning (see Fig. 4 and Eq. (4) and simulation-domain-referenced multi-domain rule adaptation (see Fig. 6 and Eq. (12)), the model is guided to extract fault-sensitive features. 2) Employ tensor Tucker decomposition to extract the core tensors of features (see Fig. 5 and Eq. (6)), serving to remove redundant information and yield noise-robust features.

By alternately optimizing the feature representation and the core tensor (see Eqs. (17)-(21)), the optimal domain-invariant feature representation that is fault-sensitive and noise-robust is ultimately obtained.

Algorithm 1 Optimization algorithm of the proposed tensorized multi-domain rule adaptation training mechanism

Input: The training data $X = \{X_s, X_t, X_{sim}\}$, the pre-trained feature representation W_s^*, the threshold for optimizing tensor reconstruction error $Thre\_re$, the threshold for optimizing the objective function $Thre\_ob$, the learning rate η, and the regulation parameters $\gamma_i (i = 1, 2)$ and $\lambda_j (j = 1, 2, 3)$, and hyperparameter of iteration n_1.
Step 1: Initialize the feature extractor parameters W_θ using W_s^*. Initialize randomly the weights W_{DC} for domain classifiers and $\{U_i\}_{i=1}^N$. Initialize $k = 0$.
Step 2: Input X into the feature extractor $f(W_\theta)$ to generate features $F = \{F_s, F_t, F_{sim}\}$ for the three domains data.
Step 3: Transform features into a higher-order tensor T, and calculate the core tensor g using Eq. (6). Calculate the hypersphere centers $C = \{C^s, C^t, C^{sim}\}$ and the radius $R = \{R^s, R^t, R^{sim}\}$ for the SD, TD, and SimD.
Step 4: Fix C, R, and W_θ, W_{DC}. Update $\{U_i\}_{i=1}^N$:
  while $(Re\_error > Thre\_re)$ **do**
    Calculate the core tensor g using Eq. (6);
    Based on the obtained g, update the factor matrix $\{U_i\}_{i=1}^N$ according to Eq. (21);
    Calculate the reconstruction error Re_error via Eq. (20);
  end while
  $k = k + 1$;
Step 5: Fix $\{U_i\}_{i=1}^N$, update C, R, and W_θ, W_{DC}:
  while (The change in the value of Eq. (16) for two consecutive iterations $> Thre\_ob$) **do**
    Calculate the objective function value in Eq. (16);
    Fix W_θ and W_{DC}, and update R of the three domain hyperspheres by solving Eq. (18);
    Fix R, and update W_θ and W_{DC} via Eq. (17);
    Re-calculate Eq. (16);
  end while
Step 6: If $(k < n_1)$, go to Step 4. Otherwise, return to Step 5 and train until convergence.
Output: The optimal weight W_θ^*.

F. Analysis of Computational Complexity

The running cost of Algorithm 1 mainly divided into two parts: the feature extractor and the tensor Tucker decomposition. The deep autoencoder (DAE) adopted as the feature extractor is essentially a linear model. Therefore, the time complexity is $O\big(E \cdot n \cdot \big(m_{D_{\text{input}}} \cdot m_{D_1} + \sum_{l=2}^{L} m_{D_{l-1}} \cdot m_{D_l} + m_{D_L} \cdot m_{D_{\text{output}}}\big)\big)$, where E denotes the training epoch, n denotes the number of training samples, $m_{D_{input}}$ represents the size of the input data, m_{D_l}, $(l = 1, 2, 3, \ldots, L)$ denotes the neuron number of the l-th hidden layer, and L denotes the number of hidden layers. In the DAE, $m_{D_{input}}$ is equal to $m_{D_{output}}$ and greater than m_{D_l}, so the time complexity can be represented as $O\left(E \cdot n \cdot m_{D_{input}} \cdot m_{D_1}\right)$. Moreover, since the core tensor is obtained by performing ALSM for each factor matrix, the main time complexity in ALSM lies in the singular value decomposition (SVD) [39]. Its time complexity equals to that of SVD, i.e. $O\big(E \cdot \big(\sum_{c=1}^{C-1} n^2 \cdot m_c + m_c^3\big)\big)$, where m_c represents dimension in the c-th direction, C denotes all directions of the higher-order tensor. Thus, the total time complexity can be represented as $O\big(E \cdot n \cdot m_{D_{\text{input}}} \cdot m_{D_1}\big) + O\big(E \cdot \big(\sum_{c=1}^{C-1} n^2 \cdot m_c + m_c^3\big)\big)$.

In this work, the number of model parameters is used to measure the space complexity of Algorithm 1. The space complexities of feature extractor and tensor Tucker decomposition are as follows: $O\big(m_{D_{\text{input}}} \cdot m_{D_1} + m_{D_1} + \sum_{l=2}^{L}\big(m_{D_{l-1}} \cdot m_{D_l} + m_{D_l}\big) + m_{D_L} \cdot m_{D_{\text{output}}}\big)$ and $O\left(\sum_{c=1}^{C-1} n \cdot m_c\right)$ respectively. Therefore, the total space complexity is represented as: $O\left(m_{D_{\text{input}}} \cdot m_{D_1} + m_{D_1} + \sum_{l=2}^{L}\big(m_{D_{l-1}} \cdot m_{D_l} + m_{D_l}\big) + m_{D_L} \cdot m_{D_{\text{output}}}\right) + O\left(\sum_{c=1}^{C-1} n \cdot m_c\right)$.

IV. Experiments

To validate the effectiveness of our method, a set of comparative experiments has been conducted on three bearing datasets: the IEEE PHM Challenge 2012 (PHM for short) dataset, the XJTU-SY dataset, and a real-world dataset of width-setting machine bearing (WSM for short) from a large steel factory in China. The algorithmic framework is executed within a Python 3.10 environment, featuring an AMD Radeon(TM) graphics card, an AMD Ryzen 7 5800H processor, and 16GB of RAM.

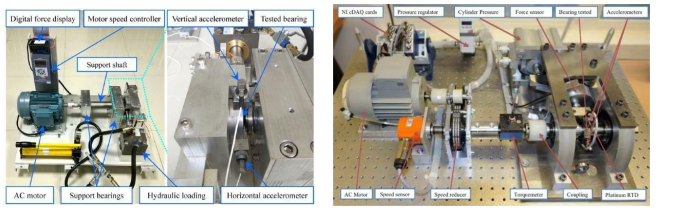
A. Dataset introduction

The test platform of the XJTU-SY dataset [40] is shown in Fig. 8(a). The dataset contains the full-lifecycle vibration signals of 15 bearings under three working conditions. The horizontal and vertical vibration signals of the test bearing were collected using two acceleration sensors, respectively. The data set is characterized by a sampling rate of 25.6kHz. Each sampling lasts 1.28s, and the sampling interval is 1min. Fig. 8(b) illustrates the PRONOSTIA experimental platform of the PHM dataset [41]. The dataset contains the full-lifecycle data of 17 bearings under three working conditions. The sampling frequency is 25.6kHz, while a sample is collected every 10s.

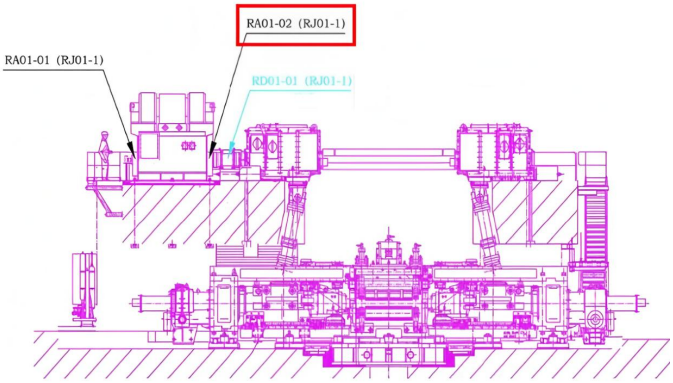
The WSM dataset was collected from the main motor bearings 23056-B-MB of an industrial width-setting machine (Fig. 8(c)) using the factory's official health monitoring system. This dataset contains full-lifecycle vibration signals from June 20th, 2022 to July 7th, 2024. The sample frequency is 80.9 kHz. Each sampling lasts 1-3s with varying sampling intervals.

For the three datasets, the acceleration signals in the vertical direction are chosen for the experiment. The vertical direction is affected by gravity. Certain faults, such as structural looseness or changes in clearances, will be superimposed with the gravitational force during vibration, resulting in larger changes in the dynamic range. Consequently, it will be

easier to detect weak fault.



(a) (b)



(c)

Fig. 8: Platforms used in this experiment, with (a) for the XJTU-SY dataset, (b) PRONOSTIA test platform for the PHM dataset and (c) the equipment diagram for the main motor bearing of the width-setting machine.

B. Experimental design

The degradation processes exhibited by bearings across the three datasets display marked divergence, primarily attributed to differences in physical sizes and operational environments, consequently leading to distinct data distributions. To simulate the real-world engineering environment, we set three cross-machine experiments in which the data from one dataset are chosen as source domain and the data from another dataset are chosen as target domain. The experimental setting is shown in Table I.

TABLE I: Experimental design of this paper.

Domain	Experiment 1		Experiment 2		Experiment 3	
	Bearing	Number	Bearing	Number	Bearing	Number
Source domain	XJTU-SY		IEEE PHM Challenge 2012		XJTU-SY	
	Bearing1_4	500	Bearing2_5	500	Bearing1_5	500
Target domain	IEEE PHM Challenge 2012		XJTU-SY		WSM	
	Bearing1_5	200	Bearing1_1	200	Motor_bearing	200
Simulation domain	Simulation Model		Simulation Model		Simulation Model	
	Signal1_1	500	Signal1_2	500	Signal1_3	500

In these experiments, the simulation domain data obtained by numerically solving the dynamic differential equations given in Section 3.1 are employed. The parameters in Eq. (1) are set according to the working conditions and structural parameters of target bearings. Regarding the source domain, the initial normal-state samples of the bearings are selected for training, which have already been verified as being in a normal state by our prior work [21]. For the target domain, a small number of initial normal-state samples from the target bearing are selected for online training. These samples are assumed to have been collected already. Fig. 9 to Fig. 11 show the original vibration signals and root mean square (RMS) curves of the data in the three experiments. It is obvious that the target bearing in the three experiments all contains irregular noise interference. Especially in Experiment 1, the initial stage of Bearing 1_5 in the PHM dataset contains heavy irregular noise. Motor_bearing from the WSM dataset also contains irregular noises, while the degradation trend in the original signals has been masked by the noises. This raises extra difficulty in extracting the features of early fault and will cause alarm delay as well.

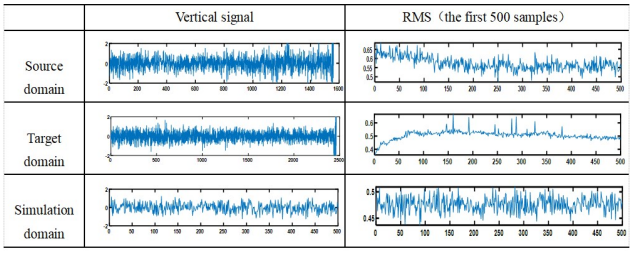


Fig. 9: Original signals and the RMS curves of the training data in Experiment 1.

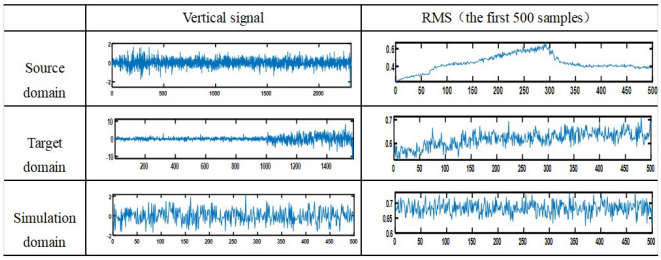


Fig. 10: Original signals and the RMS curves of the training data in Experiment 2.

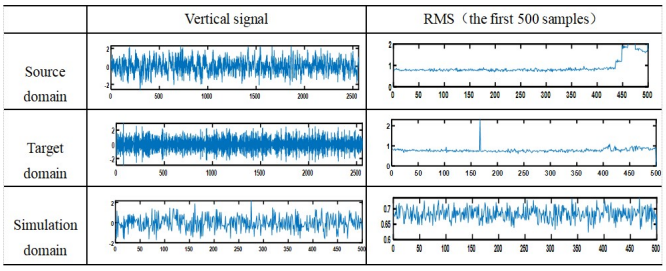


Fig. 11: Original signals and the RMS curves of the training data in Experiment 3.

In the subsequent experiments, the following parameters are adopted in this paper: the optimizer is Adam. η is 0.05, and it follows the quadratic decay characteristic (polynomial

decay) from 0.05 to 0.00005. γ_1 and γ_2 are set as 10. λ_1, λ_2 and λ_3 are 1, 0.1 and 100 respectively. The threshold for optimizing the tensor reconstruction error $There\_re$ is 0.001. The threshold for optimizing the objective function $There\_ob$ is 0.002. And n_1 is 3. All the comparison methods strictly adopt the parameter settings listed in the original papers.

C. Experimental results

1) Experiment 1 (XJTU → PHM): Fig. 12 illustrates the comparison of feature distributions across the three domains before and after dual-channel contrastive learning pre-training. It is obvious that there is a large distribution divergence between the data in source domain and target domain before pre-training, as shown in Fig. 12(a), because the data of the two domains come from different machines and operating conditions. The data distributions between the simulation domain and the target domain are highly similar due to the consistent structural parameters and operating conditions. After pre-training, the feature distribution of the three domains all shows a certain similarity, as shown in Fig. 12(b). This similar distribution characteristic makes it easier to further adapt features with each other, even though the data in the three domains are still separated.

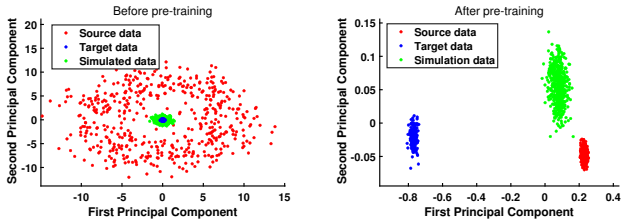


Fig. 12: Feature distribution of the three domains, where (a) is before pre-training and (b) is after pre-training. PCA is used here for visualization. The feature distribution after pre-training shows a certain similarity among the three domains, thus indicating a high-quality initial representation for online detection.

The effect of pre-trained features in the online tasks is further validated. In our method, the multi-domain rule adaptation is employed. Therefore, the change of hypersphere adaptation loss L_{Ad} (see Eq. (12)) with different iterations is able to intuitively evaluate the pre-training effect, as shown in Fig. 13. The results show that the pre-trained features significantly accelerate the convergence of adversarial training while enhancing the training stability. Notably, our method achieves effective domain adaptation in the 12th training round, while the method keeping the same structure but without pre-training always fails to reach a stable adaptation. The feature distribution at the lowest point of the two L_{Ad} values has also been checked. It is obvious that the method with pre-training gets satisfactory feature adaptation in the three domains, while the features without pre-training still remain quite divergent. This suggests that the pre-trained feature greatly facilitates domain adaptation in the online task.

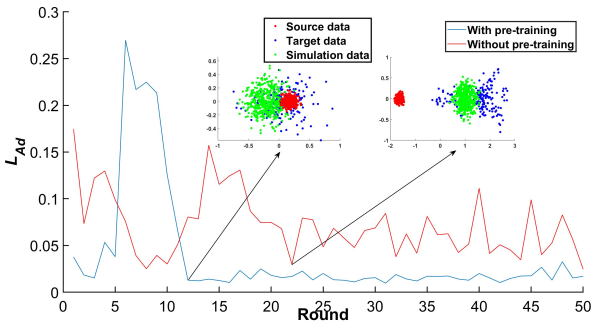


Fig. 13: Hypersphere adaptation loss of the model with and without pre-training. For better understanding, the corresponding adaptation effects are also provided. The adaptation loss with the pre-trained features exhibits less fluctuation and earlier convergence.

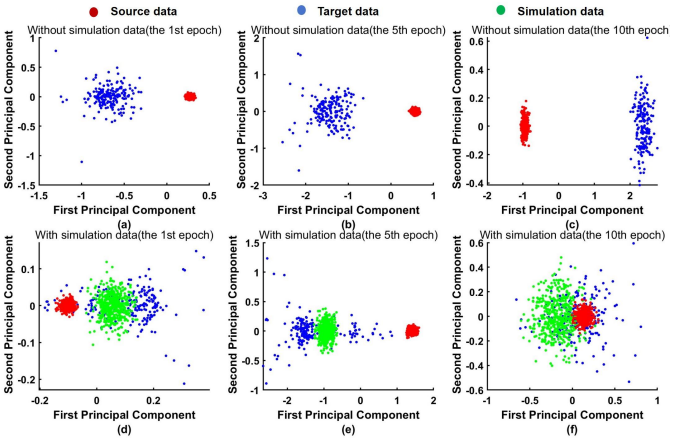


Fig. 14: Feature distribution effects in the adversarial training process, where (a)-(c) are without simulation data, and (d)-(f) are with simulation data.The introduction of simulation data significantly improves the adaptation performance, directly proving the guide effect of the dynamic model in the transfer learning.

To reveal the reason for fast adaptation with pre-trained features in Fig. 13, the role of simulation data in the hypersphere adaptation is checked, as shown in Fig. 14. Although the pre-trained features have been employed in Fig. 14(a)-(c), the samples still locate much separately. That means, without using simulation data, the adaptation is still hard to achieve when facing significant distribution divergence. In contrast, Fig. 14(d)-(f) shows a much better adaptation effect after introducing simulation data for adversarial training. It indicates the role of simulation data: acting as an intermediate domain to mitigate the distribution shift between source and target domains through adversarial training.

Fig. 15 exhibits the whole-life feature sequence of the target

bearing Bearing1_5 in the PHM dataset. It is obvious that the initial normal-state period exhibits obvious fluctuations under the noise interference caused by running-in. Once the simulation data are introduced, the feature sequence (blue curve) involves smaller bias than that without using simulation data. The stability of feature representation is then much improved. Moreover, the model with simulation data is able to capture early fault features before the fault evolves rapidly, as marked by the black circle in Fig. 15. It indicates that the proposed method is not only robust against noise interference but also sensitive to weak status change. The experimental results show that the simulation data plays a middle role in the domain adaptation process through adversarial training, preserving the discriminative features as well as suppressing noise-induced disturbances during the critical early fault phase.

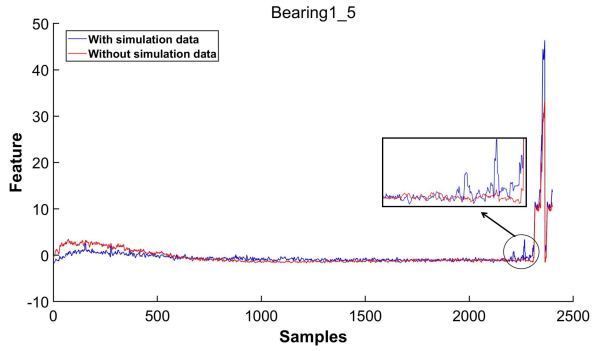


Fig. 15: Feature sequence of the target bearing Bearing1_5 in the PHM dataset with and without using simulation data.The feature sequence with simulation data is significantly more sensitive to real fault occurrence than that without simulation data.

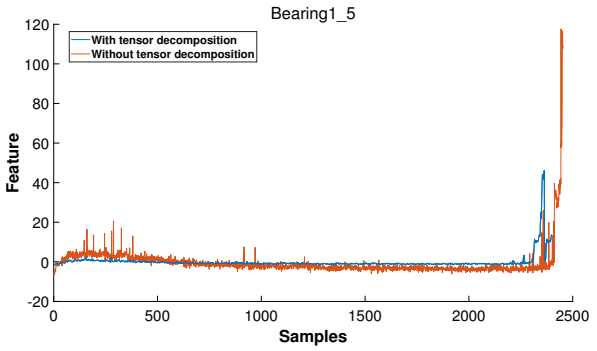


Fig. 16: Feature sequence of the target bearing Bearing1_5 in the PHM dataset with and without tensor decomposition. The feature sequence with tensor decomposition is significantly more robust to noise interference than that without tensor decomposition.

The role of tensor decomposition is further analyzed, as shown in Fig. 16. Without tensor decomposition, the feature sequence (red curve) exhibits severe fluctuations in the initial part of normal state. This is because the bearing start-up phase requires a running-in period due to assembly error and other factors. By introducing tensor decomposition, the feature sequence (blue curve) becomes notably smoother with suppressed noise. It verifies that tensor decomposition enhances the model's robustness to noise interference.

2) Experiment 2 (PHM → XJTU): The experimental design in this section is similar to that in Experiment 1. A series of ablation experiments is first carried out to evaluate the effectiveness of each component in the proposed method. Fig. 17 illustrates the influence of removing different components on the feature sequence of the target bearing Bearing1_1 in the XJTU dataset. The results show that our method can issue a fault warning at the 876th sample, which is significantly earlier than the actual fault degradation period. The feature sequence of our method presents a noticeable upward trend starting from the 876th sample, much earlier than the other models. These results indicate that our method has good discriminative capability for early fault occurrence.

We also observe that the other models do not work well. The model without pre-training fails in the transfer of detection rules due to the poor effect of hypersphere adaptation, demonstrating lower fault detection capability. Meanwhile, the model without simulation data shows less fault sensitivity due to the insufficiency of discriminative information. The absence of tensor decomposition negatively influences the feature robustness against noise interference, leading to an obvious delay in fault alarm. These results demonstrate again the significance of each component in the designed methodology.

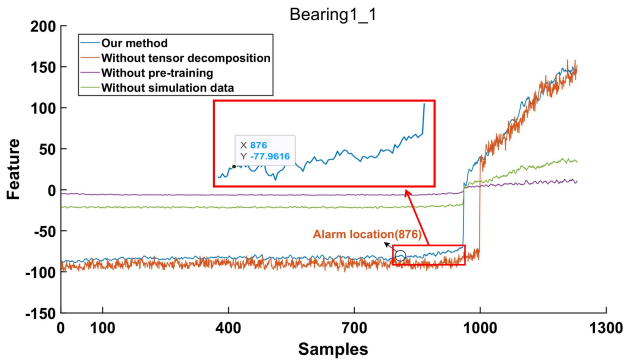
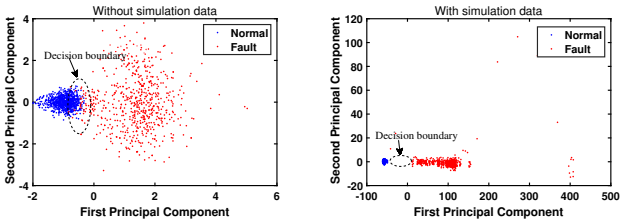


Fig. 17: Ablation results for the target bearing Bearing1_1 in the XJTU dataset. The results verify that pre-training facilitates domain adaptation, while simulation data improves model discriminative capability and tensor decomposition improves model robustness. Then noise-robust and fault-sensitive features are finally obtained.

Fig. 18 shows the feature distribution of the target bearing with and without using simulation data. As observed in Fig. 18(a), the model without simulation data has difficulty in extracting discriminative features, resulting in an unclear decision boundary between the normal state and the faulty state. Thanks to the simulation data, the margin between samples of the two states is significantly expanded as shown

in Fig. 18(b). This is because the simulation data enhances the separability of data distribution and facilitates capturing the detection boundary. In summary, the mechanism information contained in simulation data is beneficial for enhancing the discriminability of features.



(a) (b)

Fig. 18: Feature distribution for the target bearing Bearing1_1 in the XJTU dataset, where (a) is without simulation data and (b) with simulation data. The model using simulation data shows a clearer decision boundary between normal-state and faulty-state samples.

3) Experiment 3 (XJTU → WSM (real-world industrial dataset)): Compared with the laboratory data employed in the previous sections, the WSM dataset, a real-world industrial dataset from a large steel factory in China, involves stronger noise interference due to the complex working environment. The degradation tendency of the original signals in the WSM dataset has been masked by noise, as shown in Fig. 11. This raises extra difficulty in extracting the discriminative features for early fault. Similar to Fig. 17, Fig. 19 illustrates the influence of removing different components in the designed methodology. It is obvious that our method (blue curve) achieves the most desirable feature representation, that is, improves the sensitivity to early fault while maintaining robustness against noise interference.

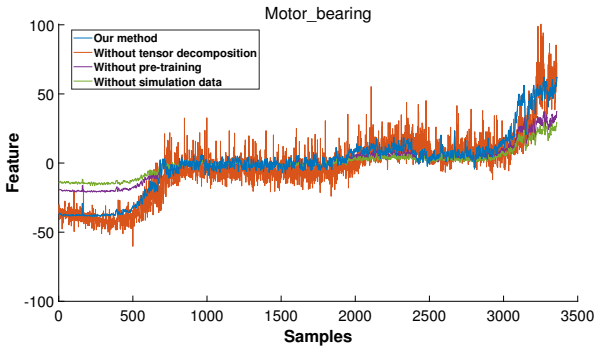
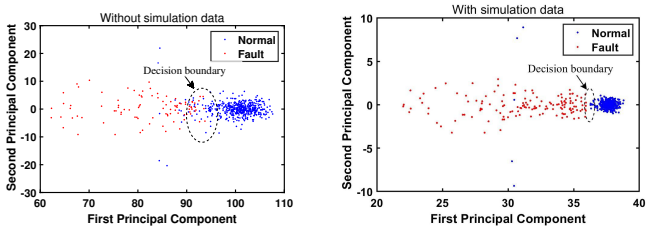


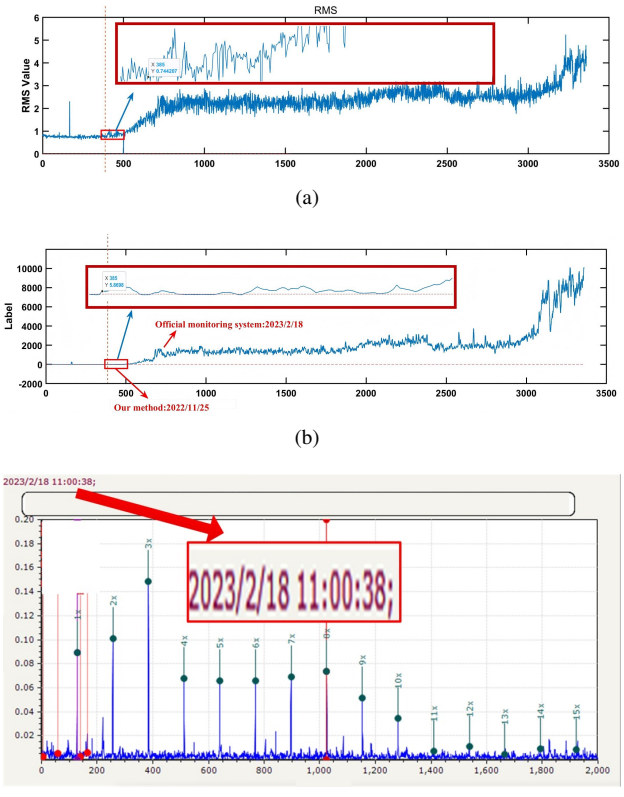
Fig. 19: Ablation results for the target bearing Motor_bearing in the WSM dataset. Similar to Fig. 17, All modules in the proposed method play critical roles.

Similar to Fig. 18, Fig. 20 shows the feature distribution of the target bearing Motor_bearing with and without using simulation data. Obviously, the model with simulation data raises broader margin between the normal state and the faulty state, as shown in Fig. 20(b). With such discriminative features, the construction of detection model will be accordingly simplified.



(a) (b)

Fig. 20: Feature distribution for the target bearing Motor_bearing in the WSM dataset, where (a) is without simulation data and (b) with simulation data. The model using simulation data can extract the feature representation with better discriminative capability.



(a)

(b)

(c)

Fig. 21: Detection results on the target bearing in the WSM dataset, with (a) is the RMS curve of the test data, (b) the results of our method and (c) the official alerting results by the factory's official health monitoring system. In (b), the scores greater than 0 are identified as early fault states.

Fig. 21(b) a shows the detection results of our method on the WSM dataset. Our method successfully alarms an early fault at the 385th sample (on November 25, 2022), much earlier than the first alarm recorded by the factory's

official health monitoring system (on February 18, 2023). As can be seen from the RMS variation curve in Fig. 21(a), the signal near the 385th sample begins to exhibit obvious power fluctuations. Subsequently, the RMS starts to rise continuously, indicating that this location is exactly the starting point of the early fault. Please be aware that, as shown in Fig. 21(c), the alarm strategy in the official monitoring system depends heavily on the fault characteristic frequency, thereby triggering alarm only after the fault has evolved to a certain extent. From the comparative results, our method enables pre-alerting, providing a longer emergency response window for equipment condition monitoring.

D. Comparative experiments

As presented in Table II, this paper compare the proposed method with 13 classical anomaly detection methods to validate its advantages. These 13 methods include anomaly detection algorithms without transfer learning, anomaly detection algorithms with transfer learning, and EFD algorithms with transfer learning as follows.

(1) Method 1 is the outlier anomaly method LOF, and Method 2 employs the deep one-class classification model (Deep SVDD [19]). Both of them are two classical methods without transfer learning.

(2) Methods 3 - 5 are three anomaly detection methods that use deep transfer learning approaches with pre-training technique. In Method 3, DCAE pre-training [42] makes use of variational autoencoders to extract pre-trained features. Method 4 employs KD pre-training [13], which relies on knowledge distillation for this purpose. Meanwhile, Method 5 uses SSL pre-training [43], leveraging self-supervised learning to obtain pre-trained features. Subsequently, all three methods perform fine-tuning on the Deep SVDD model for the task of anomaly detection.

(3) Method 6 is a typical anomaly detection method Log-TAD [22] that adopts structural domain adaptation training.

(4) Methods 7-13 belong to the EFD methods with deep transfer learning. Method 7 SRD [18] realizes fault detection across different operating conditions based on the sparse residual distance. Method 8 OD-DTL [21] achieves online EFD via deep transfer learning. It makes use of a fine-tuning strategy to achieve this goal. Method 9 DAAD [44] realizes the extraction of cross-domain common features under the constraint of rule adaptation. Method 10 Tensor-DAAD [9] achieves the efficient transfer of anomaly detection rules between two domains by adversarial training and tensor rule adaptation. Method 11, named deep domain-adversarial contrastive network (DDCN) [4], extracts fine-grained information from different frequency bands for reliable transfer of anomaly detection rules. Method 12, named Tensor-DART+DAAD, applies the effective pre-trained feature representation obtained from Tensor-DART [26] to DAAD for fault detection. Method 13, named WSN+DAAD, utilizes the optimal features obtained through wavelet scattering network (WSN) [45] and inputs them into DAAD for anomaly detection.

Here are two indicators used to evaluate the detection effect [14]: (1) Detection location, which refers to the sample location where the alarm is triggered; (2) False alarm number, which represents the number of anomalous samples detected before the final fault alarm. To localize anomaly occurrences, this paper implement a threshold-based alert mechanism requiring 6 consecutive abnormal instances for alarm activation. The performance of 11 methods is systematically compared in Table II. To enable intuitive performance evaluation, corresponding anomaly visualization outcomes are comparatively illustrated in Fig. 22.

TABLE II: Numerical results by the total 14 methods on the three experiments.

Method type	Method name	Bearing1_5 In the PHM dataset		Bearing1_1 In the XJTU-SY dataset		Motor_bearing In the WSM dataset	
		Detection location	False alarm number	Detection location	False alarm number	Detection location	False alarm number
Anomaly detection methods without transfer learning	1.Kurtosis+ LOF	2450	143	1566	49	3427	235
	2.Deep SVDD [19]	2414	38	1001	34	572	33
Anomaly detection methods with deep transfer learning	3.DCAE pre-train [42]	2391	49	997	37	605	30
	4.KD pre-train [13]	2412	38	961	120	568	19
	5.SSL pre-train [43]	2409	72	988	17	570	45
	6.Log-TAD [22]	2402	17	981	60	570	37
Early fault detection methods with deep transfer learning	7.SRD [18]	2463	19	1574	21	3490	56
	8.OD-DTL [21]	777	43	1540	77	3232	250
	9. DAAD [44]	2395	14	994	29	633	80
	10.Tensor-DAAD [9]	689	15	728	85	211	12
	11.DDCN [4]	2449	1	1000	5	711	12
	12.Tensor-DAPT [26] + DAAD	2412	14	997	40	404	28
	13.WSN [45]+DAAD	2415	18	999	13	551	20
	Our method	**2210**	**37**	**876**	**12**	**385**	**9**

From Table II and Fig. 22, our method accurately detects fault occurrence at earlier locations than the other methods, and keeps a rather low false alarm number on the three datasets. The alarm location of Method 1 is delayed and the false alarm number is high. The training of Method 1 only utilizes the initial normal data of the target bearing. The potential running-in will make the detection method less effective. Method 2 that constructs a one-class classification model, the noise interferences in normal state are easy to deviate the hypersphere boundary, resulting in a delayed alarm. Methods 3-6 demonstrate limited effectiveness in transfer learning algorithms, indicating that conventional transfer strategies fail to enhance the detection performance and robustness when the target bearing possesses only limited data with irregular noise interference. Method 7 almost exhibits the poorest performance, indicating that sparse dictionary coding strategy is susceptible to noise interference. Although the alarm location of Method 9 is relatively early, it is still later than our method. Noise interference negatively influences the detection performance of the used transfer strategy. Compared with these methods, our method overcomes the limitations on transfer performance caused by noise interference and data distribution divergence. By introducing simulation data and constructing a tensorized multi-domain rule adaptation mechanism, the feature sensitivity to early fault occurrence, as well as the robustness

to noise interference, has been significantly enhanced. Although Method 11 has a relatively small number of false alarms, its alarm location has a delay due to noise interference from the monitoring data. Method 12 improves the deployment efficiency by pre-trained features, however, the insufficient normal state information hinders the full exploration of feature patterns of normal samples, which reduces the ability to distinguish abnormal samples. Although Method 13 can reduce the effect of noise interference, the detection rules are still difficult to realize effective transfer across domains due to the significant data distribution divergence between different domains.

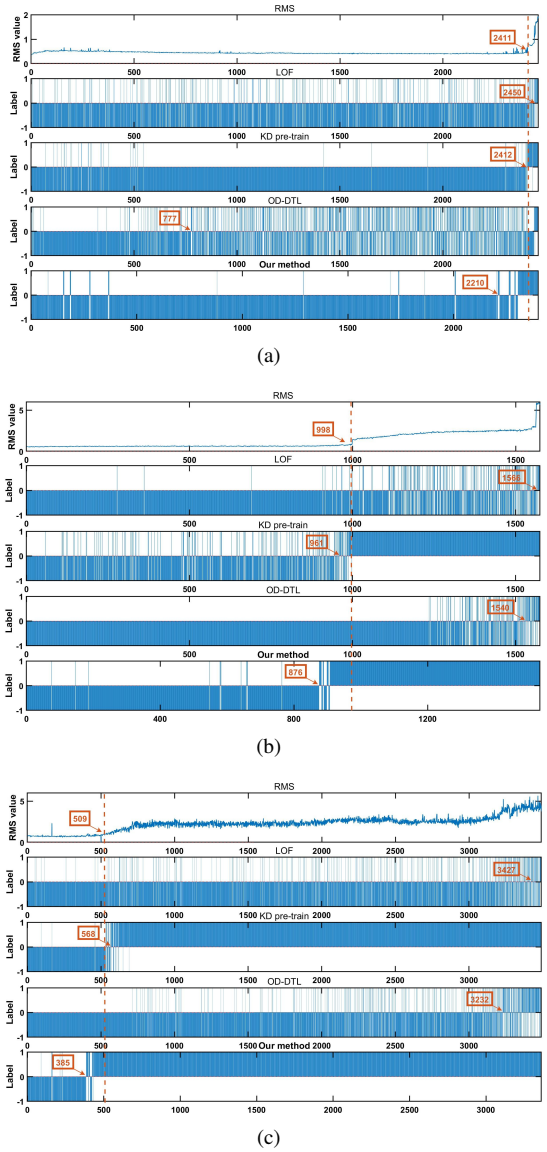


Fig. 22: Visual comparison between our method and some representative methods, where (a)-(c) are the detection results on Experiment 1 to Experiment 3, respectively. The *y*-axis label of -1 indicates that the sample point is identified to be normal, while the label of 1 indicates an anomaly. The red box numbers indicate the final alarm location.

We notice that Method 8 and Method 10 both get earlier locations and fewer false alarm numbers than our method in Experiment 1. For Method 8, the detection rule is easily affected by the fine-tuning transfer strategy when facing irregular noise interference. For Method 10 that also adopted the strategy of detection rule adaptation, the hypersphere adaptation is not achieved due to the large distribution divergence. The raw signals of each target bearing and the corresponding RMS curve are further checked, and it is found that the alarm samples identified by Method 8 and Method 10 are all located at the initial part of the normal state. The signals in these parts are all affected by the running-in (Bearing1_5) or impact shock (Motor_bearing). Then we believe that the obtained alarm locations are false and not trustworthy.

The time efficiency, parameter storage and floating point operations (FLOPs) [46] are also evaluated. The average of 10 experiments is taken as the final result. Here we take Experiment 1 as an example, the full network parameter storage is 30.8MB. for the pre-training phase, the network training time is 42.54s, and the FLOPs for one batch (100 samples) is 5.83GB. for the online EFD task phase, the model training time is 14.34min, and the FLOPs is 5.96GB. With the trained model, the online data block detection time is 0.002s, much shorter than the sampling interval. This easily meets the online detection task's time requirements.

E. Discussion

Based on the aforementioned experiments, we have the following notable observations:

1) Through the designed dual-channel contrastive learning algorithm, the pre-trained features of the three domains exhibit consistency in data distribution, as shown in Fig. 12. Thus, the divergence of data distribution can be reduced, even though the samples of the three domains are still separated.

2) Using the pre-trained features can accelerate the convergence speed of the hypersphere adaptation during the adversarial training, as shown in Fig. 13. The adaptation effect is also improved. Moreover, the pre-training strategy does not force the features of the three domains to be aligned. It only makes the feature distribution of the three domains to be consistent. This is beneficial for the deployment efficiency of the online tasks.

3) The integration of dynamic simulation data addresses the challenges posed by severe noise interference in real-world industrial data (such as in Experiment 3). Serving as a prototype, simulation data plays a crucial role in narrowing the gap between the distributions of the source domain and the target domain. Mechanism information provided by simulation data guides the domain adaptation to focus on the essential discriminative information and to eliminate the disturbance information as well, as shown in Fig. 18 and Fig. 20. Consequently, the sensitivity of features to real fault occurrence is enhanced.

4) Tensor decomposition improves feature robustness in strongly-noisy scenarios (e.g., the running-in phase in Experiment 1) by extracting pivot features and eliminating redundant information from original features. The false alarm rates are

then significantly reduced, as shown in Fig. 16. It is worth noting that excessive tensor decomposition would smooth out the weak information of early fault, thus diminishing the feature sensitivity. This issue is addressed by imposing constraints on the reconstruction error of factor matrices in Eq. (20).

The above experimental results show that the proposed method enables extracting fault-sensitive and noise-robust features, thus providing a reliable solution for online EFD in practical industrial scenarios. In particular, the results of Experiment 3 on real-world industrial data demonstrate the application value of the proposed method.

V. Conclusion

To tackle the challenges of online EFD of rolling bearings, this paper proposes a novel deep transfer learning method through integrating dynamic simulation data. For practical EFD methods, the sensitivity and robustness of features are equally important. Simulation data serves as a guide for domain adaptation, which helps to improve the sensitivity to early faults. Tensor decomposition is an effective means to suppress noise interference and play a role in enhancing robustness. The most interesting point of this paper is that the proposed method is applied to detect early faults in real industrial scenarios and it works well, demonstrating substantial potential in solving real industrial problems. From the experimental results, we have the following conclusions:

1) Features obtained by model with simulation data exhibit clearer decision boundary between normal-state and faulty samples. It indicates that the mechanism information in simulation data improves feature sensitivity to real faults and enhances detection accuracy in practical industrial scenarios.

2) Multi-domain adaptation guided by simulation domain achieves better feature distribution alignment. It reveals that the simulation domain, as a prototype, guides domain adaptation to focus on essential discriminative information, which helps narrow the gap between the target and source domains.

3) Feature sequence obtained via Tucker decomposition exhibits smoother and more pronounced trend in fault evolution. It demonstrates that core tensor representations improve noise robustness by eliminating redundant features, which plays an role in reducing the false alarm rate of online EFD.

4) Pre-training with dual-channel contrastive learning provides high-quality feature representations for online EFD by enhancing feature distributions' similarity across different domains. This improves the efficiency of online tasks.

In light of the potential distribution divergence and feature shift between simulation data and monitoring data, how to evaluate the effectiveness of simulation data and how to enable online updates to the dynamic model are a theoretical issue and worthy of in-depth exploration. In our future work, an updating mechanism of the dynamic model will be introduced for incremental detection under variable working conditions.

References

[1] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to rul prediction," *Mechanical systems and signal processing*, vol. 104, pp. 799–834, 2018.

[2] R. Liu, B. Yang, X. Zhang, S. Wang, and X. Chen, "Time-frequency atoms-driven support vector machine method for bearings incipient fault diagnosis," *Mechanical systems and signal processing*, vol. 75, no. 1, pp. 345–370, 2016.

[3] P. S. Kumar, L. Kumaraswamidhas, and S. Laha, "Selection of efficient degradation features for rolling element bearing prognosis using gaussian process regression method," *ISA transactions*, vol. 112, pp. 386–401, 2021.

[4] J. Wu, W. Mao, Y. Zhang, L. Fan, and Z. Zhong, "A novel few-shot deep transfer learning method for anomaly detection: Deep domain-adversarial contrastive network with time-frequency transferability analytics," *IEEE Internet of Things Journal*, 2024.

[5] B. Liu, C. Yan, Y. Liu, M. Lv, Y. Huang, and L. Wu, "Iseanet: an interpretable subdomain enhanced adaptive network for unsupervised cross-domain fault diagnosis of rolling bearing," *Advanced Engineering Informatics*, vol. 62, p. 102610, 2024.

[6] B. Luo, H. Wang, H. Liu, B. Li, and F. Peng, "Early fault detection of machine tools based on deep learning and dynamic identification," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 509–518, 2018.

[7] Q. Lu, W. Ye, and L. Yin, "Parallel multiple cnns with temporal predictions for wind turbine blade cracking early fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024.

[8] Z. Yang, I. Soltani, and E. Darve, "Anomaly detection with domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2958–2967.

[9] W. Mao, Z. Chen, Y. Zhang, and X. Liang, "Tensor-daad: When tensor meets online early fault detection with transfer learning," *Measurement*, vol. 208, p. 112478, 2023.

[10] L. Cui, X. Wang, H. Wang, and H. Jiang, "Remaining useful life prediction of rolling element bearings based on simulated performance degradation dictionary," *Mechanism and Machine Theory*, vol. 153, p. 103967, 2020.

[11] A. Soualhi, H. Razik, G. Clerc, and D. D. Doan, "Prognosis of bearing failures using hidden markov models and the adaptive neuro-fuzzy inference system," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2864–2874, 2013.

[12] M. D. Prieto, G. Cirrincione, A. G. Espinosa, J. A. Ortega, and H. Henao, "Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3398–3407, 2012.

[13] W. Lu, Y. Li, Y. Cheng, D. Meng, B. Liang, and P. Zhou, "Early fault detection approach with deep architectures," *IEEE Transactions on instrumentation and measurement*, vol. 67, no. 7, pp. 1679–1689, 2018.

[14] W. Mao, S. Tian, Z. Dou, D. Zhang, and L. Ding, "A new deep transfer learning-based online detection method of rolling bearing early fault," *Acta Automatica Sinica*, vol. 48, no. 1, pp. 302–314, 2022.

[15] B. Liu, C. Yan, Y. Liu, Z. Wang, Y. Huang, and L. Wu, "Multiscale residual antinoise network via interpretable dynamic recalibration mechanism for rolling bearing fault diagnosis with few samples," *IEEE Sensors Journal*, vol. 23, no. 24, pp. 31 425–31 439, 2023.

[16] Y. Xue and P. Beauseroy, "Transfer learning for one class svm adaptation to limited data distribution change," *Pattern Recognition Letters*, vol. 100, pp. 117–123, 2017.

[17] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, pp. 45–66, 2004.

[18] X. Guo, S. Liu, and Y. Li, "Fault detection of multi-mode processes employing sparse residual distance," *Acta Automatica Sinica*, vol. 45, no. 3, pp. 617–625, 2019.

[19] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[20] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, J. Woo *et al.*, "Deep unsupervised domain adaptation: A review of recent advances and perspectives," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, p. e25, 2022.

[21] W. Mao, L. Ding, S. Tian, and X. Liang, "Online detection for bearing incipient fault based on deep transfer learning," *Measurement*, vol. 152, p. 107278, 2020.

[22] X. Han and S. Yuan, "Unsupervised cross-system log anomaly detection via domain adaptation," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 3068–3072.

[23] G. Michau and O. Fink, "Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer," *Knowledge-Based Systems*, vol. 216, p. 106816, 2021.

[24] H. Xie, B. Liu, and Y. Xiao, "Transfer learning-based one-class dictionary learning for recommendation data stream," *Information Sciences*, vol. 547, pp. 526–538, 2021.

[25] Y. Ding, J. Zhuang, P. Ding, and M. Jia, "Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings," *Reliability Engineering & System Safety*, vol. 218, p. 108126, 2022.

[26] W. Mao, Z. Chen, Y. Zhang, and Z. Zhong, "Harmony better than uniformity: A new pre-training anomaly detection method with tensor domain adaptation for early fault evaluation," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107427, 2024.

[27] Y. Liu, W. Wen, Y. Bai, and Q. Meng, "Self-supervised feature extraction via time–frequency contrast for intelligent fault diagnosis of rotating machinery," *Measurement*, vol. 210, p. 112551, 2023.

[28] J. Chen, Z. Yan, C. Lin, B. Yao, and H. Ge, "Aero-engine high speed bearing fault diagnosis for data imbalance: A sample enhanced diagnostic method based on pre-training wgan-gp," *Measurement*, vol. 213, p. 112709, 2023.

[29] B. Zhao, C. Cheng, Z. Peng, Q. He, and G. Meng, "Hybrid pre-training strategy for deep denoising neural networks and its application in machine fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[30] K. Yu, Q. Fu, H. Ma, T. R. Lin, and X. Li, "Simulation data driven weakly supervised adversarial domain adaptation approach for intelligent cross-machine fault diagnosis," *Structural Health Monitoring*, vol. 20, no. 4, pp. 2182–2198, 2021.

[31] S. Mei, T. Xu, Q. Zhang, Y. Fang, and S. Zhang, "Intelligent fault diagnosis of rolling bearing under unbalanced samples based on simulation data fusion," *Measurement Science and Technology*, vol. 36, no. 1, p. 0161a6, 2024.

[32] R. Chen, G. Chen, X. Liu, X. Ai, and H. Zhu, "Reliability prediction method for low-cycle fatigue life of compressor disks based on the fusion of simulation and zero-failure data," *Applied Sciences*, vol. 12, no. 9, p. 4318, 2022.

[33] C. Hu, Y. Wang, and J. Gu, "Cross-domain intelligent fault classification of bearings based on tensor-aligned invariant subspace learning and two-dimensional convolutional neural networks," *Knowledge-Based Systems*, vol. 209, p. 106214, 2020.

[34] Z.-H. Liu, L. Chen, H.-L. Wei, F.-M. Wu, L. Chen, and Y.-N. Chen, "A tensor-based domain alignment method for intelligent fault diagnosis of rolling bearing in rotating machinery," *Reliability Engineering & System Safety*, vol. 230, p. 108968, 2023.

[35] Z. He, H. Shao, J. Cheng, X. Zhao, and Y. Yang, "Support tensor machine with dynamic penalty factors and its application to the fault diagnosis of rotating machinery with unbalanced data," *Mechanical systems and signal processing*, vol. 141, p. 106441, 2020.

[36] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[37] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. Hontani, "Missing slice recovery for tensors using a low-rank model in embedded space," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8251–8259.

[38] Y. Zhou and Y.-M. Cheung, "Bayesian low-tubal-rank robust tensor factorization with multi-rank determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 62–76, 2019.

[39] P. Comon, X. Luciani, and A. L. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 23, no. 7-8, pp. 393–405, 2009.

[40] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, 2018.

[41] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier, "Pronostia: An experimental platform for bearings accelerated degradation tests." in *IEEE International Conference on Prognostics and Health Management*, 2012, pp. 1–8.

[42] Y. Zhou, X. Liang, W. Zhang, L. Zhang, and X. Song, "Vae-based deep svdd for anomaly detection," *Neurocomputing*, vol. 453, pp. 131–140, 2021.

[43] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, "Learning and evaluating representations for deep one-class classification," *arXiv preprint arXiv:2011.02578*, 2020.

[44] W. Mao, G. Wang, L. Kou, and X. Liang, "Deep domain-adversarial anomaly detection with one-class transfer learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 2, pp. 524–546, 2023.

[45] J. Shi, Y. Zhao, W. Xiang, V. Monga, X. Liu, and R. Tao, "Deep scattering network with fractional wavelet transform," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4740–4757, 2021.

[46] X. Wang, Z. Zheng, Y. He, F. Yan, Z. Zeng, and Y. Yang, "Soft person reidentification network pruning via blockwise adjacent filter decaying," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13 293–13 307, 2021.

Le Yang is pursuing a Bachelor of Engineering degree at Henan Normal University in Xinxiang, China. His major is Artificial Intelligence. His research interest is anomaly detection with deep transfer learning.

Yuan Li received the Ph.D. degree in mechanical engineering from Hunan University, Changsha, China, in 2015. She is currently serving as an associate professor with the School of Computer and Information Engineering, Henan Normal University. Her main research interests include machine learning and computational mechanics.

Shuang Jin is pursuing a Bachelor of Engineering degree at Henan Normal University in Xinxiang, China. Her major is Artificial Intelligence. Her research interest is unsupervised anomaly detection.

Yanyan Zhang is pursuing a Bachelor of Engineering degree at Henan Normal University in Xinxiang, China. Her major is Artificial Intelligence. Her research interest is early fault detection.

Wentao Mao (Member, IEEE) received the Ph.D. degree in engineering mechanics from Xi'an Jiaotong University, Xi'an, China, in 2011. He is currently serving as a Full Professor with the School of Computer and Information Engineering, Henan Normal University, Xinxiang, China. His current research interests include machine learning, time series analysis and fault prognostics.

Shubin Du obtained a Bachelor's degree in Engineering from Yanshan University in Qinhuangdao, China in 2008. He is currently engaged in the diagnosis of mechanical and electrical equipment faults at Hebei Iron and Steel Tangshan Company, serving as a senior engineer and holding the qualification of International Vibration Analyst Level III.