IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 18, 2021, accepted November 30, 2021, date of publication December 3, 2021, date of current version December 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3132353

A New Unsupervised Online Early Fault Detection Framework of Rolling Bearings Based on Granular Feature Forecasting

KEYING LIU[1], WENTAO MAO[1,2], (Member, IEEE), HUADONG SHI[1], AND XIHUI LIANG[3], (Member, IEEE)

[1]School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
[2]Engineering Laboratory of Intelligence Business and Internet of Things of Henan Province, Xinxiang 453007, China
[3]Department of Mechanical Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada

Corresponding author: Wentao Mao (maowt@htu.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant U1704158, in part by the Henan Province Technologies Research and Development Project of China under Grant 212102210103, in part by the NSFC Development Funding of Henan Normal University under Grant 2020PL09, in part by the 2021 Scientific Research Project for Postgraduates of Henan Normal University of China under Grant YL202106, and in part by the Natural Science and Engineering Research Council of Canada under Grant RGPIN-2019-05361.

ABSTRACT In online scenarios, the monitoring signals are collected in the form of streaming data and would raise some requirements for early fault detection (EFD) of rolling bearings: 1) enhancing the detection accuracy of online data; 2) lowering the computational cost of real-time detection; 3) reducing false alarm rate; 4) deploying easily and working adaptively without manual initialization. To solve this problem, a new unsupervised online EFD framework of rolling bearings is proposed based on granular feature forecasting. First, the proposed framework considers two different online scenarios in extracting granular feature representations of online data. If the offline monitoring data are available, a deep stacked denoising autoencoder (SDAE) network with domain adaptation is introduced to extract common feature representation via decreasing the data distribution differences between offline and online working conditions. If only initial online data are available, a SDAE model is directly used to extract deep features. Second, for the obtained features, a forecasting model with tensor Tucker decomposition and ARIMA is run to predict the degradation trend of all feature sequences quickly and simultaneously. Finally, the deviation degree between the predicted sequence and sequentially-arrived data is calculated for setting alarm threshold. The proposed framework adopts an unsupervised learning mode and has three advantages: 1) flexible applicability to two different online scenarios; 2) automatic detection and easy deployment without manual intervention; 3) high reliability and extremely low false alarm rate. Experimental results on the IEEE PHM Challenge 2012 dataset and XJTU-SY dataset verify the advantages of this proposed framework.

INDEX TERMS Early fault detection, anomaly detection, streaming data, denoising autoencoder, alarm threshold.

I. INTRODUCTION

As one of the critical step of bearings Prognostics and Health Management (PHM), early fault detection (EFD) focuses on detecting the change of running status in the starting stage of fault occurrence [1]. Traditional EFD methods usually employ signal analysis techniques. In the past decade, data-driven EFD technology has been a research hotspot. Traditional machine learning algorithms such as Support Vector Machines (SVM) [2], naïve Bayesian [3], Adaboost [4] and so on have been successfully applied to EFD of rolling bearings. Although these methods can achieve fault detection well, they generally rely on manual feature extraction. Thanks to the excellent capabilities of end-to-end modeling and adaptive feature extraction, deep neural networks have also been proven effective in solving the problem of EFD. Building a suitable fault detection model for particular applications is of significant importance. Please refer to Section 2 for more specific literature analysis.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaojun Steven Li.

In recent years, with the rapid development of measurement technology, online EFD without system halt began to receive wide attention in industry and academia. Online EFD focuses on identifying the change of running status in a short time (usually between the arrivals of two data blocks). Therefore, online EFD is quite different from the traditional EFD methods with the following challenges: 1) the detection accuracy of online data, especially when the data amount is insufficient, should be enhanced; 2) the detection speed, as well as the computational efficiency, should be improved in real time; 3) the false alarm rate, i.e., avoiding recognizing normal state sample as faulty sample, should be reduced as much as possible; 4) deploying easily and working adaptively with no need of manual initialization. The challenges listed above pose new requirements for the construction of EFD model.

For the challenge 1), it is necessary to extract as many discriminative features as possible from online data, even the data merely have a small amount. Besides online data, we need to consider two cases: offline data available and unavailable. Generally speaking, some historical monitoring data of the bearings with the same manufacturing specifications can be accumulated in the laboratory or under other working conditions before going online. However, these offline data usually have different distribution characteristics with online data due to varying working conditions. In this case, the classifier trained with such offline data is prone to be biased. Therefore, the feature extraction should consider the problem of distribution shift of training data. In the other side, if the training data are merely collected online, we have to extract as granular features as possible from limited online data.

For the challenge 2), it is necessary to complete fault detection between the arrivals of two consecutive data blocks, since the monitoring signals reach sequentially. Thus, the detection model should be adjusted to fit the form of streaming data, identifying fault occurrence adaptively and quickly. Specially, streaming data is a common form, where the data blocks are sequentially collected without prior knowledge. Please note that most of traditional methods need to repeatedly train the classification model with the newly-collected data block, which will cause high computational cost and is not suitable for the online EFD problem.

For the challenge 3), it is necessary to set a reliable alarm threshold and avoid false alarms caused by the unexpected data fluctuations in the initial stage. Such data fluctuations are usually generated by machine running-in, noise interference and so on. For online EFD in non-stop scenarios, false alarms will cause unnecessary shutdowns and huge losses. The setting of alarm threshold should consider comprehensively the characteristics of streaming data to improve the reliability. Meanwhile, the threshold should have good adaptability and does not be adjusted repeatedly and manually.

For the challenge 4), the EFD model should work in an unsupervised learning mode, that is, the online data for model training are unlabeled while the detection rules are adaptively learned from such data. This mode is very appropriate for practical applications. Moreover, the detection model can be deployed directly, without assuming that the online data in the initial stage is in normal state.

Due to the unsupervised learning mode, the solutions of the challenges 1) - 3) need to be further exploited. Specifically, for the challenge 1), we adopt unsupervised deep features instead of hand-crafted statistical features or supervised deep features like CNN. For the challenge 2), we identify the abnormal status by calculating the deviation between the predicted degradation trends and the newly-arrived data blocks. Obviously, this operation is suitable to tackle streaming data. More importantly, the obtained deviation degree is helpful to build an online alarm threshold with high confidence for the challenge 3).

Following the ideas stated above, this paper proposes a new unsupervised online EFD framework of rolling bearings based on granular feature forecasting. This framework can handle two online scenarios which have, besides online data, offline data and no offline data respectively. The proposed framework consists of three inter-related generic blocks: granular feature decomposition, feature trend forecasting and anomaly recognition based on prediction deviation. First, if the offline data available, we introduce a deep transfer learning strategy to tackle the distribution drift problem. A Stacked Denoising Autoencoder (SDAE) with domain adaptation is employed to obtain multi-dimensional common feature representation of the offline and online data. If only online data available (the offline data unavailable), a SDAE model is directly used to obtain the multi-dimensional deep features. Then the granular decomposition of the online data is achieved. Second, a forecasting model with tensor Tucker decomposition and ARIMA is run to forecast the degradation trend of all the feature sequences. Finally, the deviation degree between the predicted sequence and sequentially-arrived data is calculated, and an alarm threshold is built to realize fault detection. Experimental results on the IEEE PHM Challenge 2012 dataset and XJTU-SY dataset verify the effectiveness of the proposed framework.

Rather than theoretical innovation of fault detection technology, this paper utilizes some existing techniques to build a practical and efficient online EFD framework with high deployability. This framework can provide a new integrated solution for bearing online health management under non-stop scenarios. In particular, the main contribution of this paper can be highlighted as follows:

1) An unsupervised online EFD framework is proposed. This framework can cover two online scenarios which have offline data and no offline data respectively, and realize self-adaptive detection without any human interventions. Thus it has good stability and deployability. Moreover, different from most existing EFD methods that employ black-box models, the proposed framework can provide an insight into how an anomaly is determined. Consequently, this framework can obtain detection results with high confidence.

2) A temporal anomaly detection method is proposed for streaming data. By adopting a strategy of granular decomposition, this method can extract multi-dimensional feature representations for a single sequence. Then the degradation trends of such feature sequences are jointly predicted to achieve stable and reliable EFD in real time. This method has low dependence on the amount of training data, even a very limited amount of online data can get satisfactory EFD results. According to our best knowledge, no similar work about granular anomaly detection with streaming data is found.

The remaining part of this paper is structured as follows. Section 2 provides a detailed literature analysis and presents the limitations of current bearing EFD methods in the online scenarios. Section 3 elaborates three generic blocks of the proposed framework. Section 4 verifies the effectiveness of the proposed framework on the IEEE PHM Challenge 2012 dataset and XJTU-SY dataset, followed by a conclusion in Section 5.

II. PRELIMINARY WORKS

At present, vibration signals are widely used in fault detection. Current EFD methods for rolling bearings can be divided into signal analysis-based methods and machine learning-based methods. Traditional signal analysis-based methods mainly use time-frequency analysis of weak vibration signals to identify the occurrence of early fault [5-7]. As a state-of-the-art method, Li *et al.* [5] used the bandwidth empirical mode decomposition (BEMD) to reconstruct the original vibration signal, and then exploits adaptive multi-scale morphological analysis (AMMA) to demodulate the reconstructed signal for judging fault occurrence. These methods have some drawbacks: 1) less of robustness and adaptive feature extraction, 2) insensitive to early faults and easy to delay alarm due to the low signal-to-noise ratio.

In the past decade, machine learning techniques have been applied to the EFD problem. The traditional methods usually include two steps [2]–[4]: 1) feature extraction, and 2) construction of classification model. Various algorithms such as SVM [2], naïve Bayesian [3] and Adaboost [4] were introduced to build bi-classification model based on some statistical features. Another way is utilizing one-class classification, e.g., Support Vector Data Description (SVDD) [8], to build anomaly detection model only using the data of the normal state. Although having got satisfactory results, these methods can not automatically adapt to the irregular fluctuations in the normal state, which is easy to generate false alarms. Moreover, the features are hand-crafted and fail to automatically work. There are also a few methods that can adaptively extract bearing fault characteristics under a certain working condition. For example, Tao *et al.* [9] used Generalized Gaussian Density (GGD) estimation to adaptively capture the wavelet subband information, improving the EFD accuracy significantly. However, such methods can not effectively work under varying working conditions.

In recent years, deep learning technology has been applied to bearing fault detection. Due to adaptive feature extraction directly from original data, deep neural networks can outperform traditional machine learning algorithms in the end-to-end modeling. Mao *et al.* [10] extracted the SDAE features of early fault data and then utilized a sliding window to determine the area of fault occurrence via a feature matching strategy. Eren [11] used one-dimensional Convolutional Neural Networks (CNN) to extract deep features of bearing vibration signals and performed an end-to-end EFD classification. Kumar *et al.* [12] employed CNN with an adaptive gradient optimizer to automatically extract deep features for the bearing EFD in squirrel cage induction motor. Although deep learning techniques have been proven effective in solving the EFD problem, they still need to update the modeling according to the requirements of specific applications, for example, online fault detection.

According to our literature survey, current bearing EFD methods mainly use offline data for model training and run detection in online scenario. But the research on online EFD, mainly using online data to conduct detection, is still in its infancy. As a pioneer work, Lu *et al.* [13] used the Long Short-Term Memory (LSTM) network to detect early fault of rolling bearings in online mode by calculating the deviation degree between the generative feature sequence and real online data. The alarm threshold in this work was built by means of the fluctuation characteristics of online data. Mao *et al.* [14] introduced a semi-supervised SVM with SDAE deep feature to incrementally update anomaly detection model in online scenarios. But these methods also require certain offline data to initialize a discriminative model or build an auxiliary CNN model for feature extraction.

Another key issue is that offline data and online data usually have distribution drift, e.g., caused by varying working conditions. Direct modeling using offline data may cause model bias and decrease the detection effect as well. As a potential solution to solve this problem, transfer learning aims to transfer the learned domain knowledge from one task (called source domain) to a different but related task (called target domain). In the last 2-3 years, transfer learning has been introduced to solve the problem of online EFD [15], [16]. The main idea is running deep domain adaptation to seek the common feature representation of normal state data between different working conditions. The enhancement of online features in the normal state can facilitate the construction of model detection. However, two limitations of transfer learning-based methods should be considered: 1) the commonly-used transfer strategy such as fine-tuning and layer frozen depends on data label, and 2) negative transfer would appear if the data distribution difference of the two domains is large. How to build a reliable anomaly detection model mainly using online data has become a key challenge of realizing online EFD with the applicability.

In addition, as the data in online scenario are collected sequentially, the model of anomaly detection should be updated to tackle streaming data. It is worth noting that anomaly detection with streaming data should work in an unsupervised learning mode, without assuming the initial part of online data is in normal state. There are a few works engaged in anomaly detection with streaming data. For instance, Wen *et al.* [17] regarded anomaly detection with streaming data as an image segmentation problem, and employed a U-net network to extract the temporal change pattern in the data. But this method has to set the normal state label, which belongs to supervised learning. Ahmad *et al.* [18] proposed an online sequential memory algorithm with a hierarchical structure. This algorithm can detect the change of latent data distribution over time and learn the fluctuation trend directly from the streaming data. Although this method is unsupervised, the model training relies on a large amount of data. If the online samples are insufficient, the temporal information will be not accurately extracted.

In summary, although the existing EFD methods have achieved good results, they still have the following challenges if applied to the online environment: 1) most of EFD methods require offline data for modeling training, but very few works directly use online data to build an EFD model. According to the literature survey, there is no integrated EFD solution that is applicable to the both two online scenarios: with or without offline data in training process; 2) current EFD methods usually assume the training data to be in normal state, which fails to recognize anomalies adaptively and reduces the deployability in practical applications. Unsupervised anomaly detection is essential to streaming data, especially the data are with insufficient amount.

III. PROPOSED FRAMEWORK OF EARLY FAULT ONLINE DETECTION

In this section, a new unsupervised online EFD framework of rolling bearings is proposed based on granular feature forecasting. This framework is suitable for two different types of online EFD problems: with and without offline data. The main idea is to decompose the streaming data into granular features, then calculate the deviation degree between the predicted trends of the feature sequences and sequentially-arrived data. The framework is divided into three generic and inter-related blocks: granular feature decomposition, feature trend forecasting and anomaly recognition based on prediction deviation. The overall flowchart is shown in Fig. 1. Each generic block will be elaborated in the following parts.

A. BLOCK 1: GRANULAR FEATURE DECOMPOSITION

To obtain granular features of the online data, we utilize SDAE to run unsupervised feature extraction. Two scenarios need to be treated differently. If only online data available, i.e., without any offline data accumulated before, we directly run SDAE on the online data. If some offline data available, we introduce the strategy of deep transfer learning to tackle the potential problem of distribution drift caused by varying working conditions. The SDAE with deep domain adaptation is then employed to obtain the domain-invariant feature representation of the online and offline data. The implementation methods for these two online scenarios are as follows.

1) ONLY ONLINE DATA AVAILABLE

Since online EFD is essentially an unsupervised learning problem, we run SDAE to achieve granular feature decomposition for online data. To improve the readability, we provide a brief introduction of SDAE, as follows. Please refer to the reference [19] for a detailed analysis.

SDAE is stacked by multiple denoising autoencoders (DAEs). DAE is an unsupervised learning neural network for dimensionality reduction and feature extraction. DAE's structure includes encoders and decoders, and is mainly composed of an input layer, a hidden layer and an output layer. Different from the general autoencoders (AEs), DAEs add a small level of noise in the training data. To eliminates the influence of noise on the input training data and extract more robust features at the hidden layer, the input data containing noise is encoded by the encoder from the input layer to the hidden layer, and the hidden layer output is reconstructed by the decoder from the hidden layer to the output layer. The network structure of the SDAE is shown in Fig. 2.

Suppose the training sample set is $x = [x_1, x_2, \ldots, x_n]^T \in \Re^n$, where n represents represents the number of sample. For the original training data, the DAE obtains a new data set $x' = \left[{x_1}', {x_2}', \ldots, {x_n}'\right]^T \in \Re^n$ by adding some noise, and then feeds it into the encoder to get a hidden feature representation $h = [h_1, h_2, \ldots, h_m]^T \in \Re^m$.

$$h = f\left(Wx' + b\right) \tag{1}$$

where W and b are the weight matrix and bias term in the encoder respectively, f represents the activation function of the encoder.

Then the reconstructed data set $\bar{x} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n]^T \in \Re^n$ can be obtained after feeding h into the decoder.

$$\bar{x} = g\left(W'h + b'\right) \tag{2}$$

where W' and b' are the weight matrix and bias term in the decoder respectively, g represents the activation function of the decoder. In this paper, the activation functions of the encoder and decoder are both set with the Sigmoid function: $sigmoid\left(x\right) = 1/\left(1 + e^{-x}\right)$.

Finally, the reconstruction error between input data and reconstruction data can be minimized by employing cross entropy, as shown in Equation (3). After running a gradient descent algorithm, the optimal DAE model weight and bias term can be obtained [20]. In this way, the $m-dimensional$ deep feature of the original online sequence is obtained, that is, the granular feature decomposition of the online data.

$$L\left(x', \bar{x}\right) = -\sum_{i=1}^{n}\left[{x_i}' \log\left(\bar{x}_i\right) + \left(1 - {x_i}'\right)\log\left(1 - \bar{x}_i\right)\right] \tag{3}$$

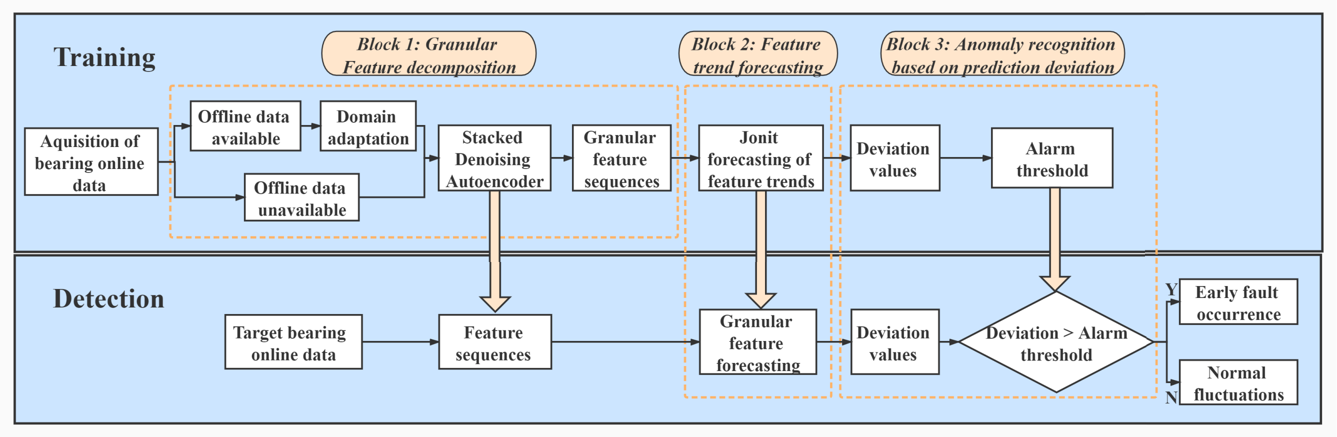

Training
Block 1: Granular Feature decomposition
Block 2: Feature trend forecasting
Block 3: Anomaly recognition based on prediction deviation
Aquisition of bearing online data
Offline data available
Domain adaptation
Offline data unavailable
Stacked Denoising Autoencoder
Granular feature sequences
Jonit forecasting of feature trends
Deviation values
Alarm threshold
Detection
Target bearing online data
Feature sequences
Granular feature forecasting
Deviation values
Deviation > Alarm threshold
Y
N
Early fault occurrence
Normal fluctuations

FIGURE 1. Flowchart of the proposed framework.

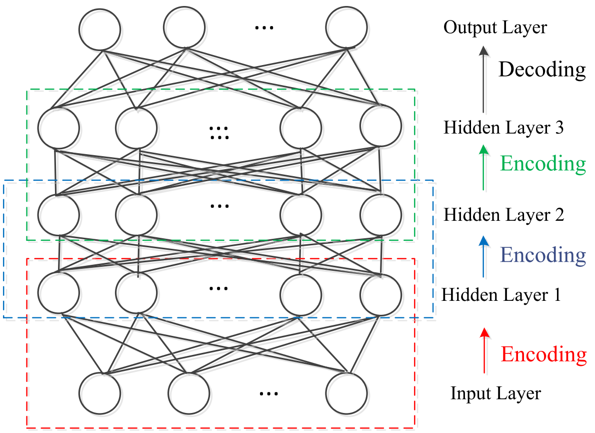


FIGURE 2. Network structure of SDAE.

2) OFFLINE DATA AVAILABLE

Transfer learning aims to improve the generalization ability of the discriminative model in target domain by getting help from the source domain data with different distribution characteristics. Domain adaptation [21] is a representative method of transfer learning. The main idea is to map the data from different domains to a consistent feature space, and make them as close as possible in this space. Therefore, this section introduces the idea of domain adaptation to conduct granular feature decomposition of online data when offline data is available. Specifically, considering that online data is in streaming form while offline data can be labeled, a previous work of domain adaptation, proposed by the authors in [15], is introduced. We built SDAE with domain adaptation model to reduce the distribution difference between the online data and the normal state data of offline bearings. The loss function of this method is as follows:

$$Loss = Loss_{SDAE} + \lambda Loss_{MMD} + \frac{\mu}{2} Loss_{weight} \tag{4}$$

where:

$Loss_{SDAE} = \frac{1}{2n} \left\| \bar{x} - x' \right\|_F^2$ is the reconstruction loss function of SDAE. Here $\|\cdot\|_F$ indicates the Frobenius norm of matrix.

$$Loss_{MMD} = \sum_{j=1}^{J} \left\| \frac{1}{n_j} \sum_{p=1}^{n_j} \phi\left(x_{j,p}\right) - \frac{1}{n_j{}'} \sum_{q=1}^{n_j{}'} \phi\left({x'}_{j,q}\right) \right\|_H^2$$

is the maximum mean discrepancy (MMD) regularizer [22]. Here the symbol J represents the combination of the online target bearing and multiple offline bearings, x_j and $x_j{}'$ represent the offline bearing samples and online bearing samples respectively, n_j and $n_j{}'$ denote the number of $j-th$ offline bearing samples and online bearing samples respectively.

$$Loss_{weight} = \sum_{m=1}^{M} \exp\left(-\left\|W_m\right\|_F^2 \big/ \sigma\right)$$

is the weight regularizer which aims to reduce model complexity. Here σ is the width parameter, M is the number of SDAE hidden layers, W_m is the weight matrix of the $m-th$ layer.

To obtain $m-dimensional$ deep feature, the final loss function can be minimized by using the stochastic gradient descent algorithm. Please refer to the reference [15] for more details.

B. BLOCK 2: FEATURE TREND FORECASTING

This section predicts the trend of the $m-dimensional$ feature sequences extracted in Section 3.1 to achieve fault detection by calculating the deviation values between the predicted values and the sequentially-arrived data. Considering that the amount of online data is relatively limited, most of the existing deep learning models such as LSTM may have low prediction accuracy and stability in time series forecasting. Moreover, the deep learning methods are generally time-consuming in model training, which is not suitable for online EFD. Since the feature sequences are from the same original signals and then intrinsically correlated, this section adopts a multivariate time series forecasting method proposed in the literature [23]. Running with block Hankel tensor and ARIMA model, this method, called BHT-ARIMA, can capture well the intrinsic relationships among multiple feature sequences to improve the forecasting accuracy. More importantly, the computational cost is very low. The algorithmic details are described as follows.

1) Employ a multi-way delay embedding transform (MDT) [24] to convert the feature sequences $X_l = [x_1, x_2, \ldots, x_l]^T \in \Re^{l\times m}$ to a third-order tensor $\chi_{\hat{l}} = [\hat{\chi}_1, \hat{\chi}_2, \ldots, \hat{\chi}_{\hat{l}}]^T \in \Re^{m\times\tau\times(l-\tau+1)}$:

$$\chi_{\hat{l}} = \hbar_\tau\left(X_l\right) = Fold_{(l,\tau)}\left(X_l \times {}_1S_1 \times \cdots \times {}_lS_l\right) \tag{5}$$

where τ and l are the time window and sample length respectively, $\hat{l} = l - \tau + 1$ is the reconstructed sample length, S is the mapping matrix.

2) Use Tucker decomposition technology [25] to obtain the core tensors $\Delta^d G_{\hat{l}} = [\Delta^d g_1, \Delta^d g_2, \ldots, \Delta^d g_{\hat{l}}]^T$ (new features) by Equation (6):

$$\begin{aligned} &\Delta^d G_{\hat{l}} = \Delta^d \chi_{\hat{l}} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times \cdots \times_c U^{(c)^T} \\ &s.t. U^{(c)^T} U^{(c)} = I, \quad c = 1, \ldots, C \end{aligned} \tag{6}$$

where $\left\{U^{(C)}\right\}$ is a set of joint orthogonal factor matrices.

3) Utilize the obtained core tensors $\Delta^d G_{\hat{l}}$ to train a tensor ARIMA model [26]. Then predict a new core tensor $\Delta^d g_{\hat{l}+1}$ at the next time point by Equation (7) and calculate the predicted value $\Delta^d \chi_{\hat{l}+1}$ by Equation (8):

$$\Delta^d g_{\hat{l}+1} = \sum_{i=1}^{p} \alpha_i \Delta^d G_{\hat{l}-i} - \sum_{i=1}^{q} \beta_i \varepsilon_{\hat{l}-i} \tag{7}$$

$$\Delta^d \chi_{\hat{l}+1} = \Delta^d g_{\hat{l}+1} \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_c U^{(c)} \tag{8}$$

Then we conduct inverse Tucker decomposition for $\Delta^d \chi_{\hat{l}+1}$ and get $\chi_{\hat{l}+1}$ in the embedded space. Finally, a predicted sequence $X'_{l+1} = [x_1, x_2, \cdots, x_l, x'_{l+1}]^T \in \Re^{(l+1)\times m}$ can be obtained via an operation of inverse MDT in Equation (9), where x'_{l+1} indicates the predicted value:

$$\begin{aligned} X'_{l+1} &= \hbar_\tau^{-1}\left(\chi_{\hat{l}+1}\right) \\ &= Unfold_{(l+1,\tau)}\left(\chi_{\hat{l}+1}\right) \times {}_1S_1^\dagger \times \cdots \times {}_{l+1}S_{l+1}^\dagger \end{aligned} \tag{9}$$

where the superscript † means the Moore-Penrose pseudo-inverse.

The steps listed above can be illustrated in Fig. 3. After running one-step ahead forecasting with BHT-ARIMA, the trend of all m-dimensional feature sequences obtained in Section 3.1 can be predicted, as $X'_i = [x'_{l+1}, \ldots, x'_{l+n}]^T \in \Re^{n\times m}$. Since the intrinsic interaction information between different feature sequences is utilized, BHT-ARIMA can effectively improve the prediction accuracy and stability. Thus, it provides an essential confidence for calculating deviation degree between the predicted values and sequentially-arrived data. Moreover, since BHT-ARIMA runs at a fast speed, it can also meet the requirements of online scenarios.

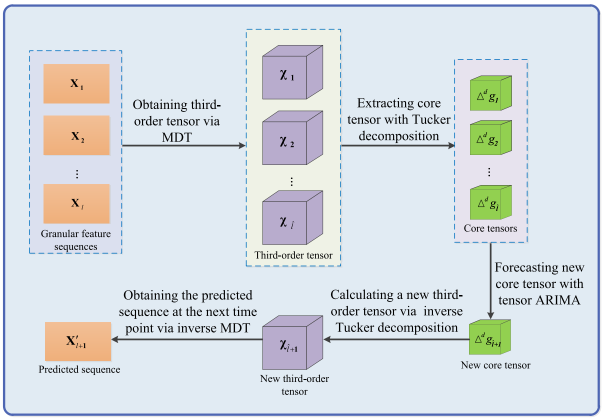

X_1
X_2
X_l
Granular feature sequences
Obtaining third-order tensor via MDT
χ_1
χ_2
$\chi_{\hat{l}}$
Third-order tensor
Extracting core tensor with Tucker decomposition
$\Delta^d g_1$
$\Delta^d g_2$
$\Delta^d g_{\hat{l}}$
Core tensors
Forecasting new core tensor with tensor ARIMA
$\Delta^d g_{\hat{l}+1}$
New core tensor
Calculating a new third-order tensor via inverse Tucker decomposition
$\chi_{\hat{l}+1}$
New third-order tensor
Obtaining the predicted sequence at the next time point via inverse MDT
X'_{l+1}
Predicted sequence

FIGURE 3. Building a granular feature forecasting model using BHT-ARIMA.

C. BLOCK 3: ANOMALY RECOGNITION BASED ON PREDICTION DEVIATION

To detect fault occurrence with streaming data in real-time, it is necessary to construct an effective alarm threshold. Considering the streaming data is unlabeled and arrives sequentially, it is very challenging to directly construct a two-class or one-class classifier for identifying the occurrence of anomalies. Because the initial state of online samples is unknown, the alarm threshold should be set based on the anomaly detection model. No matter the initial state of online data is normal or faulty, the detection rules should both be learned adaptively from the unlabeled data, without being adjusted repeatedly and manually. Based on the predicted values of the granular feature sequences (see Equation 9), the deviation degree between the predicted values and sequentially-arrived data (called the real values) are expressed as:

$$Dev(i) = \frac{1}{m}\left\|X'_i - X_i\right\|_2^2 \tag{10}$$

where $X_i{}'$ and X_i are the predicted values and real values at the $i-th$ time point respectively. A larger $Dev(i)$ indicates that the $i-th$ sample is more likely to be an abnormal point.

Moreover, an alarm threshold is set as the 95% confidence interval of the maximum deviation of the initial online data. Generally, the 95% confidence interval of the alarm threshold in the project is commonly used. For the newly-arrived online sample, a same operation including granular feature decomposition and BHT-ARIMA forecasting is conducted. Then the deviation degree can be calculated. If the deviation degree is larger than the threshold on three consecutive samples, an anomaly is determined.

From this process, the proposed framework does not adopt a supervised classifier to recognize state change, but employs a deviation indicator between the predicted feature trend and real online data. This framework does not set the initial online data as normal in advance, thus it meets the characteristics of streaming data. Specifically, if the available online data is in normal state, the deviation between the predicted feature trend and real fault data will arise, which can determine the location of faulty occurrence. In contrast, if the available online data cover a certain fault data (usually in the later part), the feature trend will correspondingly rise up. In this scenario, the alarm threshold can still work after counting the maximum deviation of the available online data. And the proposed framework can also identify fault occurrence adaptively in the newly-arrived data. Therefore, the proposed framework is suitable for the online EFD problem.

IV. EXPERIMENTS

To verify the effectiveness of the proposed framework, some comparative experiments are conducted on two bearing data sets: IEEE PHM Challenge 2012 and XJTU-SY. The programming environment is Matlab2014 and Python 3.6, with CPU i5-8400HQ and 16 GB memory. All data are linearly normalized to $[-1, +1]$ before training.

A. DATASET INTRODUCTION

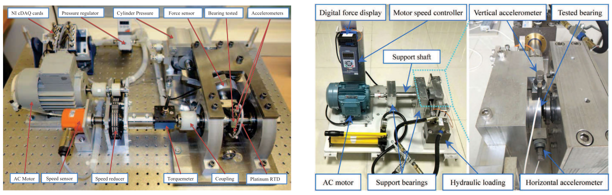
The dataset of IEEE PHM Challenge 2012 [27] is collected from PRONOSTIA test platform shown in Fig. 4(a). This platform provided the run-to-failure degradation data rolling bearings. The dataset includes three working conditions. Under the first condition, the motor speed is 1800rpm and the load is 4000N. The second condition is the motor speed of 1650rpm and the load of 4200N. The third condition is the motor speed of 1500rpm and the load of 5000N. Among them, the first condition and the second condition contain 7 bearings, and the third condition contains 3 bearings.

The dataset of XJTU-SY [28] is collected from the test platform, as shown in Fig. 4(b). The platform can provide accelerated degradation experiments of various rolling bearings and sliding bearings, and obtain whole-life monitoring data. This dataset includes three working conditions. Under the first condition, the motor speed is 2100rpm and the load is 12N. The second condition is the motor speed of 2250rpm and the load of 11N. The third condition is the motor speed of 2400rpm and the load of 10N. Each working condition contains 5 bearings.

It is noted that the two datasets have different data samples. One sample in the IEEE PHM challenge 2012 dataset and the XJTU-SY dataset has 2560 points and 32000 points respectively. For deep neural networks, the more sample points, the better the effect of feature extraction.

B. EXPERIMENTAL SETUP

According to including offline data in the training process or not, this section sets up three comparative experiments, as shown in Table 1. For the IEEE PHM Challenge 2012 dataset, the bearings 1, 3, 4 and 7 under the first working condition are selected in turn as the online target bearing (i.e., the target domain). Experiment 1 only has initial online data. Experiment 2 chooses six bearings under the first working condition as the offline data (i.e., the source domain). Experiment 3 also chooses six bearings under the second working condition as the offline data. For the XJTU-SY dataset, the bearings 1, 2, 3 and 5 under the first working condition are as the online target bearings in turn. The experimental setup is the same as the IEEE PHM Challenge 2012 dataset.

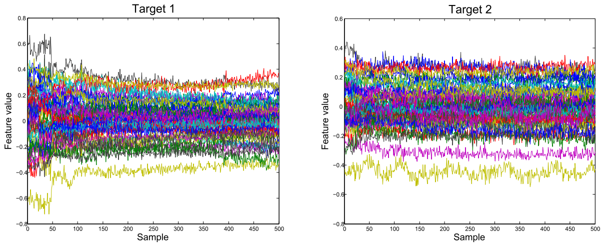


(a) (b)

FIGURE 4. Test environments used in our experiments, with (a) PRONOSTIA test platform [27] and (b) XJTU-SY test platform [28].

TABLE 1. Experimental settings on the two datasets. The word "Source" and "Target" indicates the source domain and target domain respectively. The first number in the bearing name indicates the number of working condition, and the second number indicates the number of the bearing. The data in source domain is for training, and the data in target domain is for test in the unsupervised learning mode.

		IEEE PHM Challenge 2012		XJTU-SY	
	Type	Bearing	Name	Bearing	Name
Experiment 1	Only initial online data	Bearing1_1	Target 1	Bearing1_1	Target 1
		Bearing1_3	Target 2	Bearing1_2	Target 2
		Bearing1_4	Target 3	Bearing1_3	Target 3
		Bearing1_7	Target 4	Bearing1_5	Target 4
Experiment 2	Offline data under the same working condition available	Bearing1_2	Source 1	Bearing1_4	Source 1
		Bearing1_5	Source 2	Bearing1_1	Target 1
		Bearing1_6	Source 3	Bearing1_2	Target 2
		Bearing1_1	Target 1	Bearing1_3	Target 3
		Bearing1_3	Target 2	Bearing1_5	Target 4
		Bearing1_4	Target 3		
		Bearing1_7	Target 4		
Experiment 3	Offline data under different working conditions available	Bearing2_1	Source 1	Bearing2_2	Source 1
		Bearing2_2	Source 2	Bearing2_3	Source 2
		Bearing2_3	Source 3	Bearing2_4	Source 3
		Bearing2_4	Source 4	Bearing2_5	Source 4
		Bearing2_6	Source 5	Bearing1_1	Target 1
		Bearing2_7	Source 6	Bearing1_2	Target 2
		Bearing1_1	Target 1	Bearing1_3	Target 3
		Bearing1_3	Target 2	Bearing1_5	Target 4
		Bearing1_4	Target 3		
		Bearing1_7	Target 4		



(a) (b)

FIGURE 5. The extracted 50-dimensional features of online bearing in Experiment 1 with (a) Target 1 and (b) Target 2.

C. RESULTS ON THE IEEE PHM CHALLENGE 2012 DATASET

1) RESULTS OF GRANULAR FEATURE DECOMPOSITION

Experiment 1 (Only Initial Online Data Available): To conduct granular feature decomposition, the first 500 samples

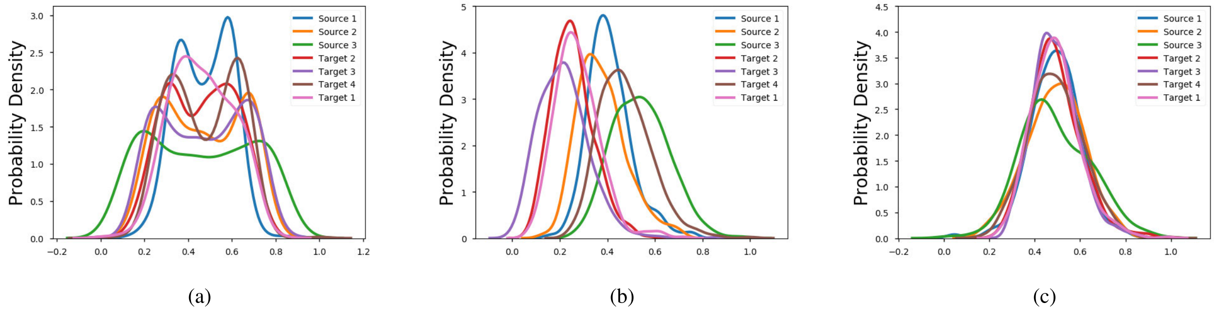


FIGURE 6. PDFs of the online bearing Target 1 and the offline data from the same working condition in Experiment 2, with (a) raw signal, (b) Hilbert-Huang Transform (HHT) [29] marginal spectrum, and (c) the common features extracted by SDAE with domain adaptation. To calculate the PDF of the common features, the features are reduced to one dimension by using principal component analysis (PCA).

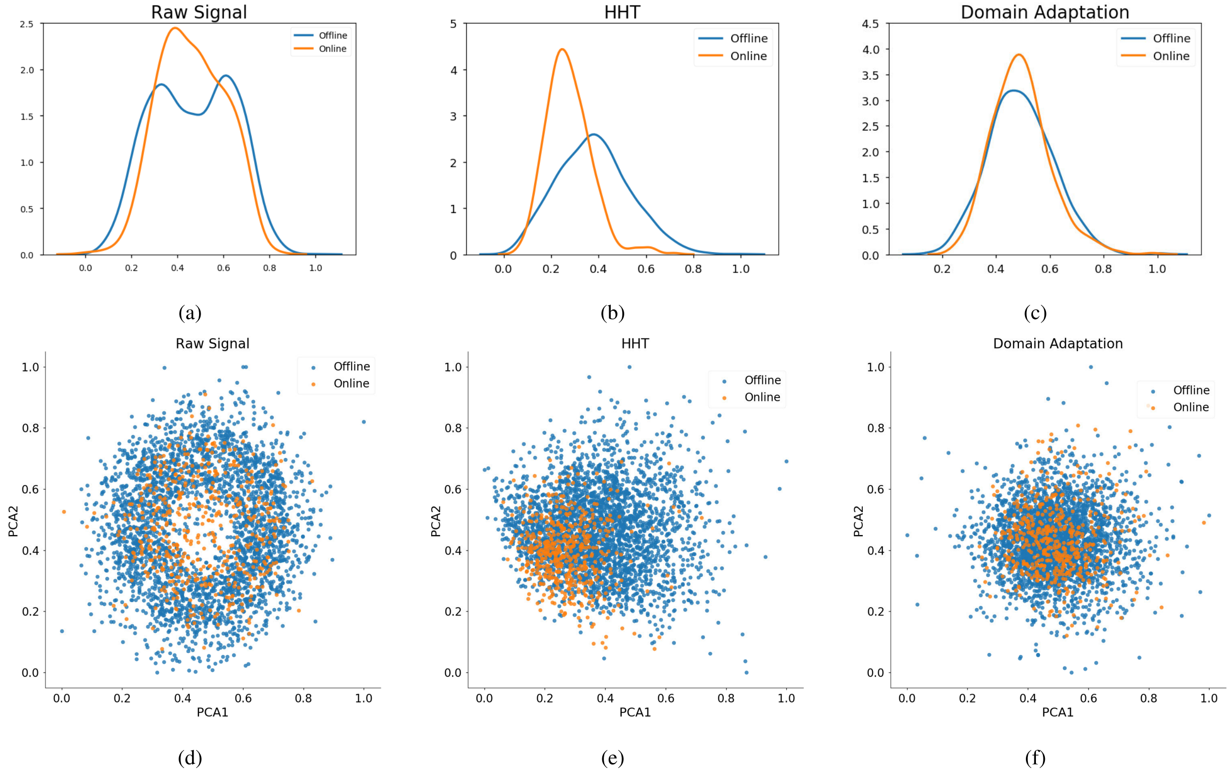
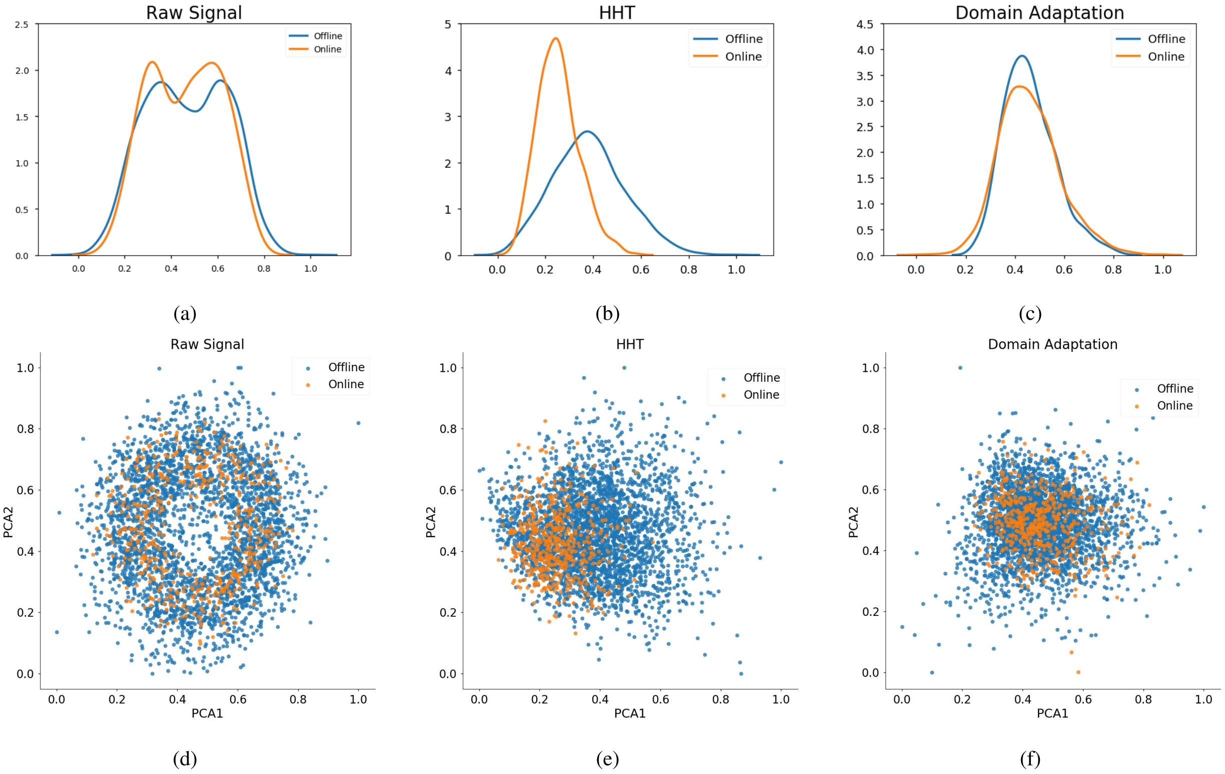


FIGURE 7. Data distribution of the online bearing Target 1 and the offline data from the same working condition with (a)-(c) are the PDF curves, (d)-(f) are the feature distribution. PCA is used to visualize the feature distribution.

of the bearing whole lifetime are used to extract deep features by SDAE, and the remaining samples are sequentially collected and employed for test. For the SDAE network, the added noise level is set 0.01 and three hidden layers with the size [2048, 512, 50] are employed. After 100 iterations, the 50 dimensional deep features are obtained. Then the sequentially-arrived data is fed into the trained SDAE model to get online features. Taking Target 1 and Target 2 as examples, Fig. 5 shows the extracted features. From Fig. 5, due to the reason of run-in, the feature values of the first 50 samples are more different from the remaining ones. It is obvious that the overall trend of the features is smooth, indicating that the extracted features have good predictability. Thus it is suitable to predict the feature trends using BHT-AMRIMA and other time series forecasting methods.

Experiment 2 (Offline Data Under the Same Working Condition Available): Due to the degradation characteristics of rolling bearings, the actual degradation process still has a certain difference even under the same working condition. To reduce the difference of data distribution, this section uses the method in Section 3.1.2 to seek the common features of the offline bearings data. The first 500 samples of each bearing are taken to train SDAE with domain adaptation. The probability density functions (PDF) before and after domain



(a)

(b)

(c)

(d)

(e)

(f)

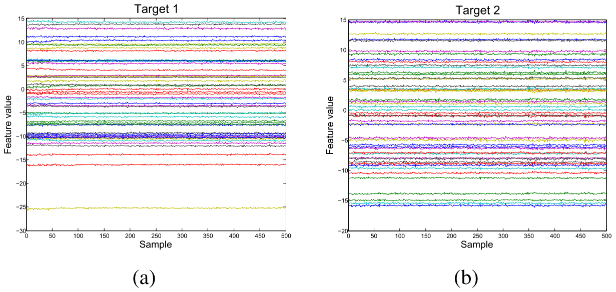
FIGURE 8. **Data distribution of the online bearing Target 2 and the offline data from the same working condition with (a)-(c) are the PDF curves, (d)-(f) are the feature distribution. PCA is used to visualize the feature distribution.**

adaptation are shown in Fig. 6. The PDF of the bearings under the same working condition have a bit differences. But after domain adaptation, the PDF of all bearings tends to be consistent.

Here Target 1 and Target 2 are selected for test. The data of these two bearings are fed into the SDAE model to obtain the online features, as shown in Fig. 7 and Fig. 8. Please note that the PDF of the offline bearings is calculated by splicing the features of multiple bearings and then reducing it to one dimension. It is obvious in Fig. 7(d) and Fig. 8(d) that the original data of different bearings are scattered, which indicates that the data distributions of the initial state are significantly different in time domain. The HHT features illustrated in Fig. 7(e) and Fig. 8(e) are certainly gathered towards the center. After domain adaptation, Fig. 7(f) and Fig. 8(f) show that the bearings data converge to a large extent. With a roughly identical distribution, the features are applicable for the further feature trend forecasting.

Fig. 9 provides the online feature sequences extracted from the common feature representations that are shown in Fig. 7(f) and Fig. 8(f). It is clear that the online features have a smooth trend after domain adaptation and can be regarded as granular features of the online data. Obviously, these feature sequences have good predictability.

Experiment 3: Offline data under different working conditions available.



(a)

(b)

FIGURE 9. **The extracted 50-dimensional features of online bearing in Experiment 2 with (a) Target 1 and (b) Target 2.**

In this experiment, the first 500 samples of each bearing under the first working condition and the first 100 samples of the six bearings under second working condition are chosen to train the SDAE model with domain adaptation. The PDFs before and after domain adaptation are shown in Fig. 10.

The effect of Fig. 10 is similar to Fig. 6, which means that the SDAE with domain adaptation is also suitable for the granular feature extraction under different working conditions. In addition, Target 1 and Target 2 are selected as the online target bearing respectively, and the PDF and feature distribution before and after domain adaptation are shown in Fig. 11 and Fig. 12.

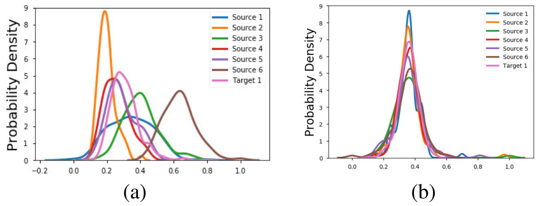


FIGURE 10. PDFs of the online bearing Target 1 and the offline data from the different working condition in Experiment 3, with (a) HHT marginal spectrum, and (b) the common features extracted by SDAE with domain adaptation. Same to Fig. 6, PCA is also used to calculate the PDF of the common features.

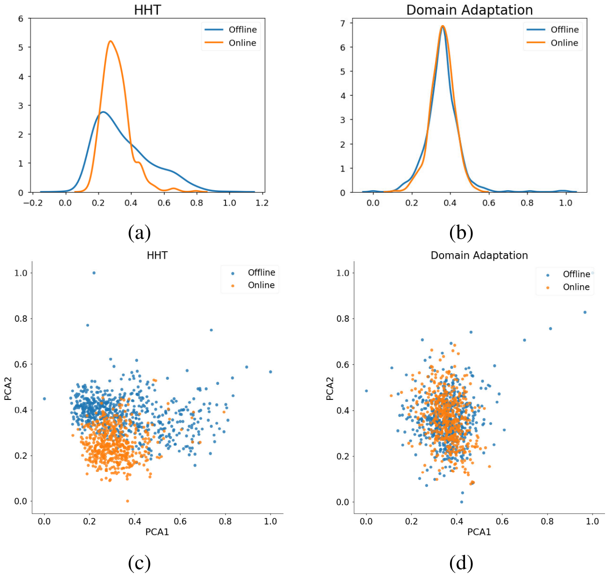


FIGURE 11. Data distribution of the online bearing Target 1 and the offline data from different working condition, with (a)-(b) are the PDF curves of HHT marginal spectrum and the extracted common features, (c)-(d) are the corresponding feature distribution. Same to Fig. 7, PCA is used to visualize the feature distribution.

Similar to Fig. 9, Fig. 13 provides the online feature sequences extracted from the common feature representations that are shown in Fig. 11(d) and Fig. 12(d). As in Fig. 13, the online features are smooth and of good predictability. These feature sequences prove again that the granular feature decomposition can provide a solid basis for the online data, especially with an insufficient amount, to make an accurate trend forecasting.

2) RESULTS OF FEATURE TREND FORECASTING

As described in Section 3.2, we need to forecast the extracted granular feature sequences to calculate the deviation between the predicted values and sequentially-arrived data. For the BHT-ARIMA algorithm, we take the first 300 samples of the extracted feature sequences as input, set the autoregressive coefficient p to be 3, the difference order d to be 2, and the moving average coefficient q to be 1. One-step ahead forecasting is used to obtain the trend of the multi-dimensional deep feature sequences. Taking the online bearings Target 1 and Target 2 as examples, Fig. 14, Fig. 15, and Fig. 16 show the 50-dimensional feature trends of the Experiment 1 to Experiment 3, respectively.

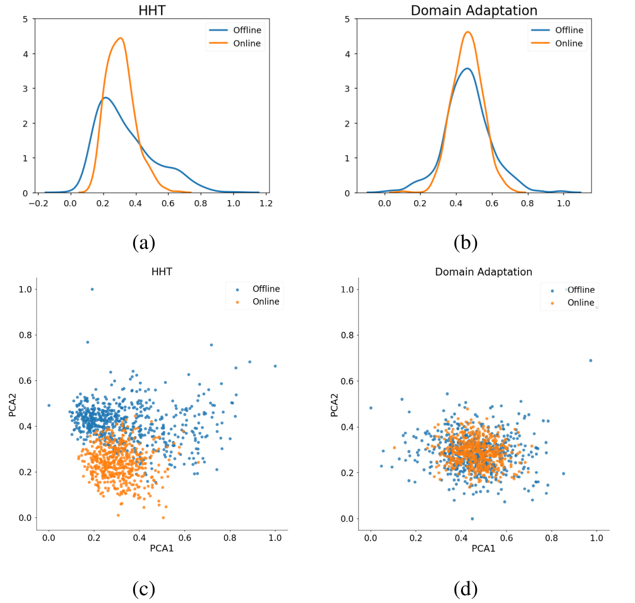


FIGURE 12. Data distribution of the online bearing Target 2 and the offline data from different working condition, with (a)-(b) are the PDF curves of HHT marginal spectrum and the extracted common features, (c)-(d) are the corresponding feature distribution.

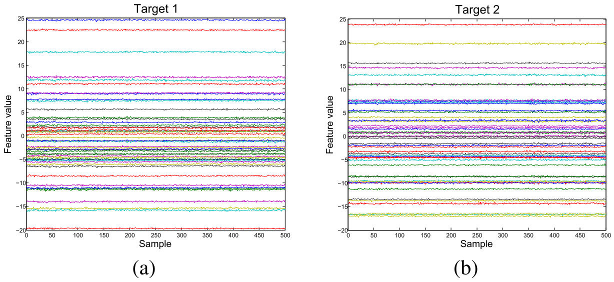


FIGURE 13. The extracted 50-dimensional features of online bearing in Experiment 3 with (a) Target 1 and (b) Target 2.

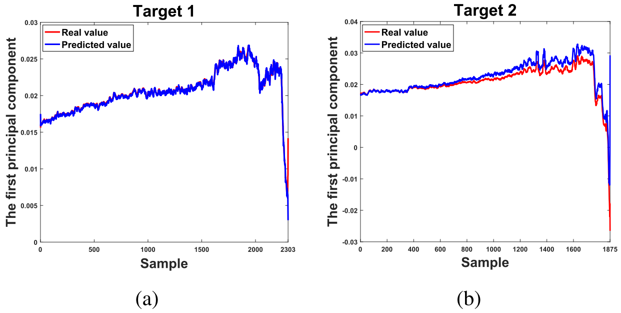
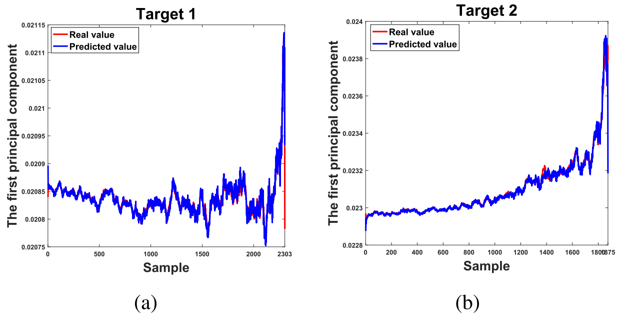
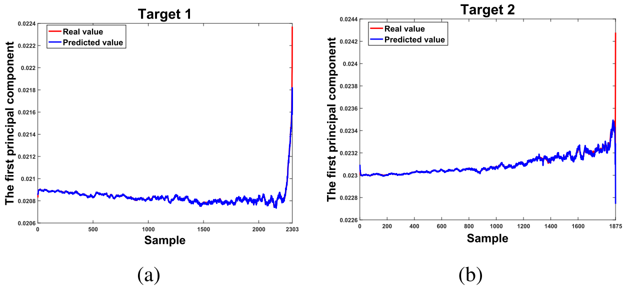


FIGURE 14. Feature trend forecasting in Experiment 1 with (a) Target 1 and (b) Target 2. Here PCA is utilized to visualize the forecasting performance by reducing the 50-dimensional features into the first principal component.



(a) (b)

FIGURE 15. **Feature trend forecasting in Experiment 2 with (a) Target 1 and (b) Target 2.**



(a) (b)

FIGURE 16. **Feature trend forecasting in Experiment 3 with (a) Target 1 and (b) Target 2.**

It is clear from Figs. 14-16 that the predicted trend of the two bearings keeps line with the real online data in the three experiments. It indicates that the BHT-ARIMA model has good capability of time series forecasting with limited data. Moreover, the forecasting performance demonstrates that the proposed framework can obtain well the public information of multiple granular feature sequences from the same original data. More importantly, the computational cost is extremely low. Therefore, the forecasting results listed in Figs. 14-16 are suitable to calculate the deviation from the real data and recognize fault occurrence as well.

3) FAULT OCCURRENCE DETECTION

In this section, the alarm threshold is built according to the 95% confidence interval of the maximum deviation in the initial online data. The mean square error of each feature sequence is calculated between the forecasting results in Section 4.3.2 and the sequentially-arrived online data. Then all 50-dimensional error sequences are reduced to one-dimensional error sequence. After smoothing operation by using moving average method, the sample in the error sequence whose deviation exceeds the alarm threshold will be recognized as an anomaly. The results of Experiment 1 to Experiment 3 are shown in Figs. 17-19 respectively.

From Figs. 17-19, the deviation value line crosses the alarm threshold multiple times, which is the scenario of false alarm. Obviously, we cannot determine the fault occurrence at the first cross point that may be caused by irregular signal fluctuation. Only consecutive anomalies can indicate a real fault alarm. In this experiment, a fault is supposed to occur once the deviations on three successive samples are all larger than the threshold. Otherwise the sample is set as false alarm. Apparently, the number of false alarm is extremely low in Figs. 17-19. More interestingly, the locations of fault detection in Fig. 18 and Fig. 19 are mostly earlier than the location in Fig. 17. The reason is introducing offline data for model training can improve the effect of granular feature decomposition and obtain common features with better representative capability. Such features are certainly beneficial to forecast future trends. The specific numerical results can be referred to Table II (in Section 4.5). The results also demonstrate the stability of the proposed framework, both applying to two different online scenarios: with offline data and without offline data.

D. RESULTS ON THE XJTU-SY DATASET

In this section, the experimental steps are the same as those of the IEEE PHM Challenge 2012 dataset. Due to space limitation, we only provide detection results rather than detailed description of experimental process. Here the first 300 samples of each bearing whole lifetime are selected for granular feature decomposition and domain adaptation. In such samples, the first 200 samples are used to train the BHT-ARIMA model, and the remaining 100 samples are used to build an alarm threshold via the 95% confidence interval of the maximum deviation. The fault detection results of Experiment 1 to Experiment 3 are shown in Figs. 20-22 respectively.

From Figs. 20-22, the number of false alarm is also low, which is similar with the results of IEEE PHM Challenge 2012 dataset. It indicates that the proposed framework can handle two types of online detection problems on the XJTU-SY dataset: with and without offline data. In addition, the locations of fault occurrence in Fig. 21 and Fig. 22 are relatively later than the location in Fig. 20, which is exactly the opposite of the results on the IEEE PHM Challenge 2012 dataset. The reason is offline data has large fluctuations trend, affecting the effect of domain adaptation. As a result, the fault detection results are correspondingly delayed. However, the overall results is earlier than other EFD models (please refer to Table III), which proves again the stability of the proposed framework.

E. COMPARATIVE RESULTS

To verify the effectiveness of the proposed framework, we introduce 10 widely-used EFD methods for comparison, as follows.

1) BEMD+AMMA[5]
2) Kurtosis + SVDD
3) SDAE+ SVDD
4) Kurtosis + LOF
5) SDAE+LOF
6) Kurtosis + iFOREST
7) SDAE+ iForest
8) FDDA[13]
9) SDFM[10]
10) S4VM+SODRMB[14]

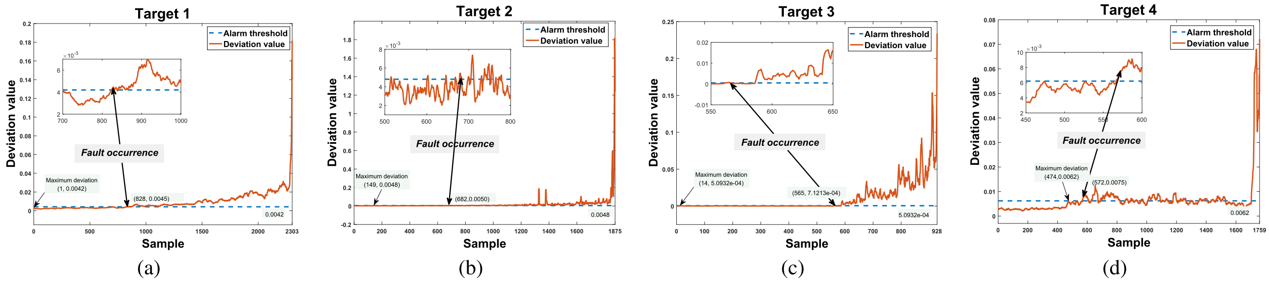


FIGURE 17. Fault detection results in Experiment 1 with (a) Target 1, (b) Target 2, (c) Target 3 and (d) Target 4. The small window is an enlarged view of the location of fault occurrence, and the blue dotted line is the obtained alarm threshold. The alarm threshold is built according to the 95% confidence interval of the maximum deviation in the initial online data.

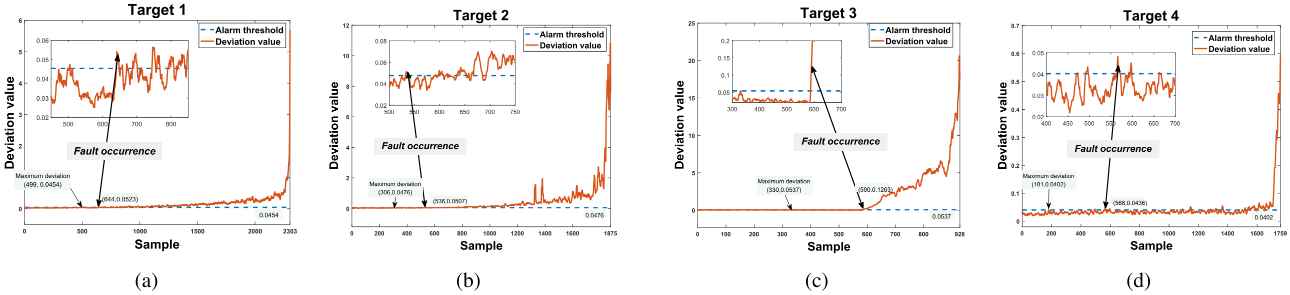


FIGURE 18. Fault detection results in Experiment 2 with (a) Target 1, (b) Target 2, (c) Target 3 and (d) Target 4.

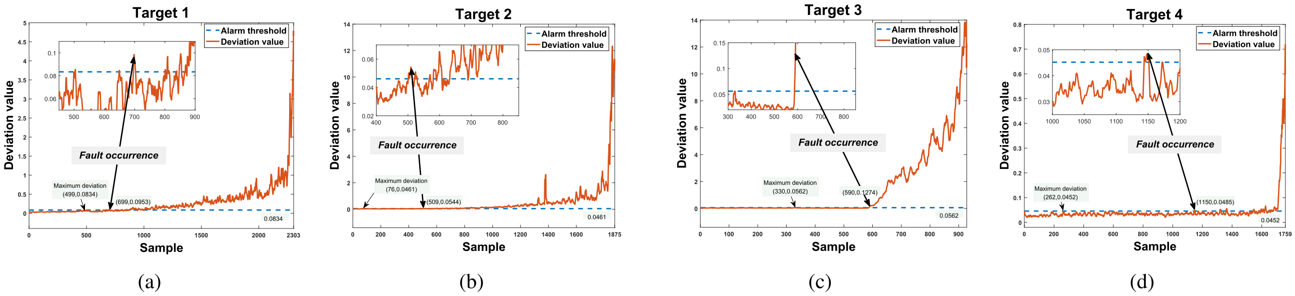


FIGURE 19. Fault detection results in Experiment 3 with (a) Target 1, (b) Target 2, (c) Target 3 and (d) Target 4.

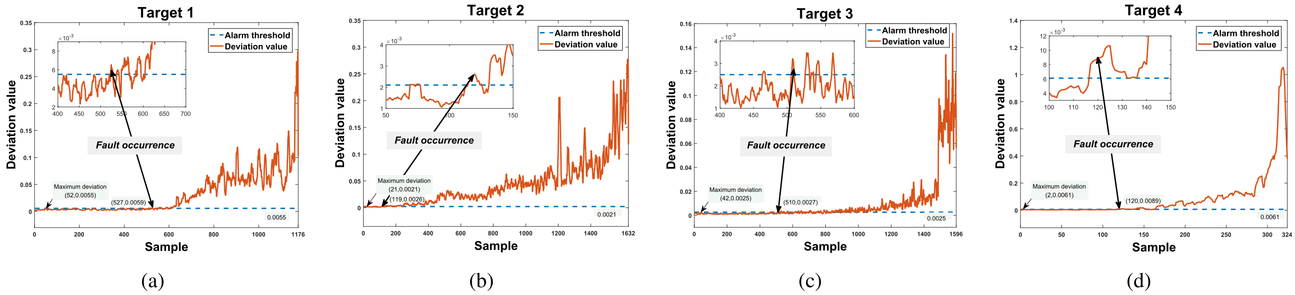
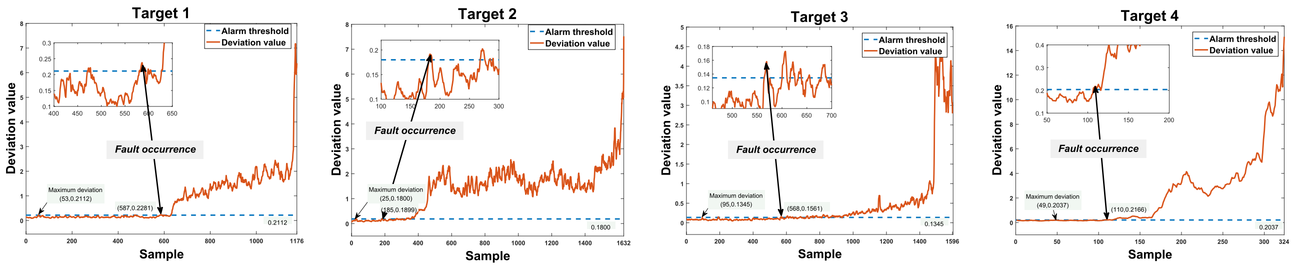


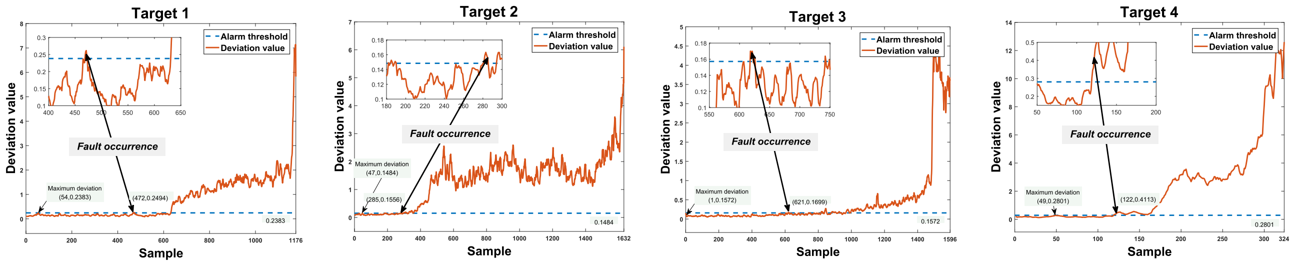
FIGURE 20. Fault detection results in Experiment 1 with (a) Target 1, (b) Target 2, (c) Target 3 and (d) Target 4. The small window is an enlarged view of the location of fault occurrence, and the blue dotted line is the obtained alarm threshold. The alarm threshold is built according to the 95 confidence interval of the maximum deviation in the initial online data.

Specifically, BEMD+AMMA [5] is a state-of-the-art EFD method based on weak signal analysis. This method utilizes Bandwidth EMD to decompose the signal in time-frequency domain and compares it with the prefixed fault characteristic frequency. LOF [30] is a typical anomaly detection algorithm based on distance metric. SVDD [8] is a one-class



(a) (b) (c) (d)

FIGURE 21. Fault detection results in Experiment 2 with (a) Target 1, (b) Target 2, (c) Target 3 and (d) Target 4.



(a) (b) (c) (d)

FIGURE 22. Fault detection results in Experiment 3 with (a) Target 1, (b) Target 2, (c) Target 3 and (d) Target 4.

classification algorithm for anomaly detection. iForest [31] is an integrated algorithm based on data cutting. Besides the SDAE features, we also adopt a typical early fault feature Kurtosis. By combining the two features and the three methods, we have 6 methods (Methods 2-7) for comparison.

Methods 8-10 are three state-of-the-art online EFD methods. Among them, FDDA [13] utilizes LSTM to predict the trend of the normal state data of online bearing and determine fault occurrence by evaluating the deviation degree between the predicted distribution and real distribution. SDFM [10] realizes online EFD based on abnormal sequence matching with deep learning. This method mainly detects the early fault occurrence by matching the fault feature and the known faults in a sliding window. S4VM+SODRMB [14] utilizes offline data to construct an initial semi-supervised SVM and incrementally updates the SVM model with the unlabeled data batch to achieve anomaly detection in online scenario. These three methods all adopt the strategy of offline-modeling and online detection, and can be viewed as the state-of-the-art online EFD works.

For the IEEE PHM Challenge 2012 dataset, we choose the first 500 samples of the online target bearing in all methods as the normal samples and then construct the EFD model. For the XJTU-SY dataset, we choose the first 300 samples of the online target bearing in all methods complete the same operation. Moreover, we employ cross-validation to optimize the hyper-parameters of SVDD and set the k-value of LOF to be 10. For iForest, the number of trees is set to 100, and each tree trains 256 samples.

TABLE 2. Comparative results of online EFD on IEEE PHM Challenge 2012 dataset.

Method	IEEE PHM Challenge 2012							
	Target 1		Target 2		Target 3		Target 4	
	Alarm location	False alarm number	Alarm location	False alarm number	Alarm location	False alarm number	Alarm location	False alarm number
1.BEMD+AMMA[5]	1900	-	1600	-	1098	-	1350	-
2.Kurtosis +SVDD	1642	58	2042	104	1193	45	1180	75
3.SDAE+ SVDD	1525	116	1285	118	1788	150	1063	109
4.Kurtosis +LOF	2381	65	2037	73	1092	56	1191	89
5.SDAE+LOF	2050	4	1236	1	1094	3	1523	7
6.Kurtosis +iForest	2057	159	1783	146	1100	175	2207	149
7.SDAE+ iForest	1556	82	1341	37	1290	49	2198	56
8.FDDA[13]	1392	45	1837	38	1263	28	1123	37
9.SDFM[10]	1374	42	1193	57	1145	38	970	47
10.S4VM+SODRMB[14]	1480	3	1290	1	1258	8	1098	7
11.Experiment 1	1328	2	1182	2	1063	0	1071	1
12.Experiment 2	1144	3	1036	0	1090	0	1068	0
13.Experiment 3	1199	1	1009	3	1090	0	1650	0

To make a comprehensive comparison, we choose two widely-used evaluation metrics [10], [14]: *alarm location* and *false alarm number*. Alarm location means the number of the signal snapshot where the fault alarm appears. False alarm number means the number of anomalies before the fault alarm appears. For all methods, we define the maximum deviation of the initial online data as the threshold. If the deviations on three successive samples are all larger than the threshold, the fault occurrence is recognized. The comparative results are listed in Table 2 and Table 3.

On the whole, the proposed framework performs better than all methods in the three experiments. For the IEEE PHM Challenge 2012 dataset, the proposed framework gets earlier alarm location in Experiment 1 (only online data available) than all methods except SDFM on Target 4. However, the false alarm number of the proposed framework is 1, far

TABLE 3. **Comparative results of online EFD on XJTU-SY dataset.**

Method	XJTU-SY							
	Target 1		Target 2		Target 3		Target 4	
	Alarm location	False alarm number	Alarm location	False alarm number	Alarm location	False alarm number	Alarm location	False alarm number
1.BEMD+AMMA[5]	1130	-	550	-	824	-	450	-
2.Kurtosis +SVDD	1163	134	541	114	923	154	432	123
3.SDAE+ SVDD	879	93	538	75	863	64	433	88
4.Kurtosis +LOF	1372	33	541	15	825	22	421	25
5.SDAE+LOF	828	5	541	7	814	4	431	8
6.Kurtosis +iForest	1257	93	758	88	1484	101	426	79
7.SDAE+ iForest	908	25	557	14	1489	19	482	33
8.FDDA[13]	1004	28	787	35	1199	24	522	15
9.SDFM[10]	930	27	561	34	910	29	425	19
10.S4VM+SODRMB[14]	925	5	554	4	921	8	545	7
11.Experiment 1	827	0	419	0	810	2	420	0
12.Experiment 2	887	3	485	0	868	3	410	0
13.Experiment 3	772	0	585	1	921	0	422	0

less than the number 47 of SDFM. In Experiment 2 (offline data under the same working condition available), the alarm location of the proposed framework is earlier than the location of Experiment 1 on all target bearings except Target 3. But the false alarm number is lower, indicating that the degradation data under the same working conditions can effectively improve the online feature representation and then develop the detection results. But for Experiment 3, the comparative results are sort of complex. On Target 2 and Target 3, the results of Experiment 3 are slightly better than the one of Experiment 2. But on Target 4, the result of experiment 3 is far inferior to the result of Experiment 2. The reason is that the bearing data under different working conditions may have a quite different distribution characteristic. In this case, the SDAE with domain adaptation is not effective enough to extract common features. Therefore, it is necessary to introduce a stronger unsupervised domain adaptation method.

For the XJTU-SY dataset, the proposed framework obtains a similar comparative effect. In Experiment 1, the proposed framework gets earlier alarm location and lower false alarms than all the other methods. Meanwhile, due to the introduction of offline data, the results of Experiment 2 and Experiment 3 are not very stable. The results of Experiment 2 are rather worse than the results of Experiment 1. The reason is that the initial offline data has obvious noise in the normal state, which influences negatively the feature representation. Meanwhile, the detection effect depends on the quality of the offline data when using offline data under different working conditions. In Experiment 3, some results are better than the ones in Experiment 2, but some are significantly inferior (e.g., on Target 2).

Furthermore, we also evaluate the computational cost of the proposed method. Due to space limitation, we choose two representative methods for comparison, e.g. SDFM [10] and S4VM+SODRMB [14]. For one data block with 10 samples, the running time of our proposed method is approximately 2.5s, while the running times of SDFM and S4VM+SODRMB are 6s and 10s respectively. Obviously, the computational cost of the proposed method is very low, which is suitable for the online fault detection.

In summary, the proposed framework always gets the best EFD effect than all the methods for comparison when only online data are available. After introducing offline data for model training, the proposed framework can get relatively earlier alarm location than the other methods on most of bearings, also with lower false alarm number. But still on very few bearings, the proposed framework is slightly less effective. The comparative effect depends on the data distribution characteristics of the offline data as well as the used domain adaptation method.

V. CONCLUSION

Aiming at the problem of bearings EFD without system halt, this paper proposes an unsupervised online EFD framework based on granular feature forecasting. This framework is divided into three inter-related generic blocks: granular feature decomposition, feature trend forecasting and anomaly recognition based on prediction deviation. The main conclusions can be highlighted as follows:

1) The proposed framework is applicable to two types of online scenarios: with and without offline data. Especially, this framework can realize effective EFD only using a limited amount of unlabeled online data, which is the biggest difference from the existing methods.

2) Granular feature forecasting with BHT-ARIMA can accurately forecast the fluctuation trend of single degradation sequence with very low computational cost, which is very suitable to online EFD.

3) Introducing some offline data may obtain better EFD results. But this effect depends on the quality of the offline data (for example, whether there is colored noise) as well as the transfer learning performance of the used domain adaptation method.

In the next work, we will construct an unsupervised deep learning algorithm with stronger adaptability to improve the detection robustness. Besides, the current alarm strategy is manually set. For online detection, an adaptive threshold setting is certainly vital. It is necessary to optimize alarm strategy to realize automatic setting of alarm threshold by considering various fault patterns comprehensively.

REFERENCES

[1] W. Mao, W. Feng, Y. Liu, D. Zhang, and X. Liang, ''A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis,'' *Mech. Syst. Signal Process.*, vol. 150, Mar. 2021, Art. no. 107233.

[2] Z. Wang, L. Yao, G. Chen, and J. Ding, ''Modified multiscale weighted permutation entropy and optimized support vector machine method for rolling bearing fault diagnosis with complex signals,'' *ISA Trans.*, vol. 114, pp. 470–484, Aug. 2021.

[3] M. K. Saini and A. Aggarwal, ''Detection and diagnosis of induction motor bearing faults using multiwavelet transform and naive Bayes classifier,'' *Int. Trans. Electr. Energy Syst.*, vol. 28, no. 8, p. e2577, Aug. 2018.

[4] I. Martin-Diaz, D. Morinigo-Sotelo, O. Duque-Perez, and R. de J. Romero-Troncoso, ''Early fault detection in induction motors using AdaBoost with imbalanced small data and optimized sampling,'' *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 3066–3075, Oct. 2017.

[5] Y. Li, M. Xu, X. Liang, and W. Huang, ''Application of bandwidth EMD and adaptive multiscale morphology analysis for incipient fault diagnosis of rolling bearings,'' *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6505–6517, Jan. 2017.

[6] Q. He, ''Vibration signal classification by wavelet packet energy flow manifold learning,'' *J. Sound Vibrat.*, vol. 332, no. 7, pp. 1881–1894, Apr. 2013.

[7] J. Antoni and R. B. Randall, "The spectral kurtosis: Application to the vibratory surveillance and diagnostics of rotating machines," *Mech. Syst. Signal Process.*, vol. 20, no. 2, pp. 308–331, Feb. 2006.

[8] C. Liu and K. Gryllias, "A semi-supervised support vector data description-based fault detection method for rolling element bearings based on cyclic spectral analysis," *Mech. Syst. Signal Process.*, vol. 140, Jun. 2020, Art. no. 106682.

[9] X. Tao, C. Ren, Y. Wu, Q. Li, W. Guo, R. Liu, Q. He, and J. Zou, "Bearings fault detection using wavelet transform and generalized Gaussian density modeling," *Measurement*, vol. 155, Apr. 2020, Art. no. 107557.

[10] W. Mao, J. Chen, X. Liang, and X. Zhang, "A new online detection approach for rolling bearing incipient fault via self-adaptive deep feature matching," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 443–456, Feb. 2020.

[11] L. Eren, "Bearing fault detection by one-dimensional convolutional neural networks," *Math. Problems Eng.*, vol. 2017, pp. 1–9, 2017.

[12] P. Kumar and A. S. Hati, "Deep convolutional neural network based on adaptive gradient optimizer for fault detection in SCIM," *ISA Trans.*, vol. 111, pp. 350–359, May 2021.

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

[14] W. Mao, S. Tian, J. Fan, X. Liang, and A. Safian, "Online detection of bearing incipient fault with semi-supervised architecture and deep feature representation," *J. Manuf. Syst.*, vol. 55, pp. 179–198, Apr. 2020.

[15] W. Mao, L. Ding, Y. Liu, S. S. Afshari, and X. Liang, "A new deep domain adaptation method with joint adversarial training for online detection of bearing early fault," *ISA Trans.*, Apr. 2021, doi: 10.1016/j.isatra.2021.04.026.

[16] G. Michau and O. Fink, "Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106816.

[17] T. Wen and R. Keyes, "Time series anomaly detection using convolutional neural networks and transfer learning," 2019, *arXiv:1905.13628*.

[18] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017.

[19] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1017–1027, Apr. 2017.

[20] L. Ren, Y. Sun, J. Cui, and L. Zhang, "Bearing remaining useful life prediction based on deep autoencoder and deep neural networks," *J. Manuf. Syst.*, vol. 48, pp. 71–77, Jul. 2018.

[21] J. Zhu, N. Chen, and C. Shen, "A new deep transfer learning method for bearing fault diagnosis under different working conditions," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8394–8402, Aug. 2020.

[22] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.

[23] Q. Shi, J. Yin, J. Cai, A. Cichocki, T. Yokota, L. Chen, M. Yuan, and J. Zeng, "Block Hankel tensor ARIMA for multiple short time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, Apr. 2020, pp. 5758–5766.

[24] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8251–8259.

[25] F. E. Correa, M. D. B. Oliveira, J. Gama, P. L. P. Corrêa, and J. Rady, "Analyzing the behavior dynamics of grain price indexes using tucker tensor decomposition and spatio-temporal trajectories," *Comput. Electron. Agricult.*, vol. 120, pp. 72–78, Jan. 2016.

[26] A. Shadab, S. Ahmad, and S. Said, "Spatial forecasting of solar radiation using ARIMA model," *Remote Sens. Appl., Soc. Environ.*, vol. 20, Nov. 2020, Art. no. 100427.

[27] P. Nectoux, R. Gouriveau, and K. Medjaher, "PRONOSTIA: An experimental platform for bearings accelerated life test," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, Denver, MA, USA, Jun. 2012, pp. 1–8.

[28] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Trans. Rel.*, vol. 69, no. 1, pp. 401–412, Mar. 2020.

[29] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London. A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

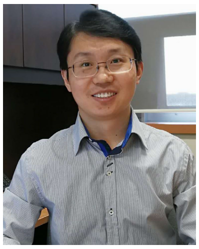
[30] H. Ding, K. Ding, J. Zhang, Y. Wang, L. Gao, Y. Li, F. Chen, Z. Shao, and W. Lai, "Local outlier factor-based fault detection and evaluation of photovoltaic system," *Sol. Energy*, vol. 164, pp. 139–148, Apr. 2018.

[31] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, Feb. 2018.

KEYING LIU received the bachelor's degree, in 2020. She is currently pursuing the M.S. degree with Henan Normal University, China. Her research interests include deep learning and anomaly detection.

WENTAO MAO (Member, IEEE) received the M.S. degree in computer science from the Chongqing University of Posts and Telecommunications, in 2006, and the Ph.D. degree in engineering mechanics from Xi'an Jiaotong University, China, in 2011. He is currently working as a Full Professor at Henan Normal University, China. He has conducted and is conducting about ten research projects, such as the National Natural Science Foundation of China as a Project Principal or a Main Researcher. His current research interests include machine learning, kernel methods, and evolutionary computation.

HUADONG SHI received the bachelor's degree, in 2020. He is currently pursuing the M.S. degree with Henan Normal University, China. His research interests include anomaly detection and transfer learning.



XIHUI LIANG (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering from Shandong University, China, in 2007 and 2009, respectively, and the Ph.D. degree in mechanical engineering from the University of Alberta, Canada, in 2016. After that, he worked as a Postdoctoral Research Fellow at the University of Alberta, for about two years. He was also an Assistant Professor with the Department of Mechanical Engineering, University of Manitoba, Canada. He has authored/coauthored more than 30 articles in prestigious journals, such as IEEE Transactions on Industrial Electronics, *Mechanical Systems and Signal Processing*, and *Reliability Engineering and System Safety*. His research interests include dynamic modeling of mechanical systems, condition monitoring, fault diagnostics and prognostics, reliability analysis, and intelligent maintenance.