Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks

Sungho Suh [a,1], Paul Lukowicz [a], Yong Oh Lee [b,*]

[a] *German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany*
[b] *Smart Convergence Group, Korea Institute of Science and Technology Europe Forschungsgesellschaft mbH, 66123 Saarbrücken, Germany*

A R T I C L E   I N F O

A B S T R A C T

Bearing is a key component in industrial machinery and its failure may lead to unwanted downtime and economic loss. Hence, it is necessary to predict the remaining useful life (RUL) of bearings. Conventional data-driven approaches of RUL prediction require expert domain knowledge for manual feature extraction and may suffer from data distribution discrepancy between training and test data. In this study, we propose a novel generalized multiscale feature extraction method with generative adversarial networks. The adversarial training learns the distribution of training data from different bearings and is introduced for health stage division and RUL prediction. To capture the sequence feature from a one-dimensional vibration signal, we adapt a U-Net architecture that reconstructs features to process them with multiscale layers in the generator of the adversarial network. To validate the proposed method, comprehensive experiments on two rotating machinery datasets have been conducted to predict the RUL. The experimental results show that the proposed feature extraction method can effectively predict the RUL and outperforms the conventional RUL prediction approaches based on deep neural networks.

1. Introduction

Prognostics and health management (PHM) technology collects status information from industrial systems, such as manufacturing machines, facilities, and power plants, to detect failures of the system and enables maintenance schedule in advance by predicting the point of failure through analysis and predictive verification [1]. One of the PHM technologies predicts the remaining useful life (RUL) of the rolling element bearings to prevent unexpected failures and improve reliability [2,3]. A large number of studies have been conducted for RUL prediction, which can be categorized into model-based [4] and data-driven methods [5]. Model-based methods estimate the RUL through analytical models based on physical laws and mathematical functions. These methods include physics-based methods [4], empirical-based methods [1], Kalman filter [6], and particle filter [7]. However, they require expert domain knowledge to build a precise model in an increasingly complex industrial systems [8]. Recently, data-driven methods have garnered significant interest along with substantial development in machine learning and minimal

requirement of expert knowledge. The data-driven methods capture the direct relationship between collected machine data and degradation status with machine learning techniques.

Recently, deep-learning-based RUL prediction methods have been proposed and have achieved better prediction performance than conventional data-driven methods [9–16]. Although these deep-learning-based RUL prediction methods have been successfully developed, less importance has been given to three challenging issues: (1) These methods require domain knowledge to extract features or we have to manually specify the feature types. (2) The aforementioned data-driven methods assume that training and test data are collected by the same sensors under the same operating conditions or are from the same distribution. However, these assumptions can be impractical in industries, since machinery working conditions usually change with respect to tasks and the training and test data can be collected from different entities. (3) For accurate RUL prediction, it is essential to properly determine the health stage (HS) of the machinery; because the machine in a healthy state would not have any remarkable difference in the run-to-failure training dataset. However, extant conventional methods predict the RUL without determining the first predicting time (FPT) [17–19], which is the start time of the unhealthy stage.

In this study, we propose a generalized multiscale feature extraction method for the RUL prediction. Generative adversarial

---

network(GAN) is used to learn the distributions of multiple training data from different bearings and extract domain-invariant generalized prognostic features. For the determination of FPT and RUL prediction, the proposed feature extraction method consists of two steps; the first of which trains multiscale adversarial neural networks to reconstruct the vibration input signals to the generalized prognostic features. Here, three different levels of a one-dimensional U-Net [20] architecture are trained to minimize the loss functions of the GAN-based proposed feature extraction method; the second step transforms the generalized features into a nested-scatter plot (NSP) [21] image to determine the FPT and to predict the RUL. NSP is an imaging method for multi-variable correlation analysis. NSP is heuristic, but the NSP images transformed from raw vibration signals reduce the efforts for feature engineering based on domain knowledge. Also, NSP images are useful for the feature extraction in fault diagnosis of rotating machinery, when combined with convolutional neural networks (CNN) [21]. A CNN-based binary regression model determines the FPT, and a CNN–long short-term memory (CNN–LSTM) model predicts the RUL.

The main contributions of this work are summarized as follows.

- A novel multiscale feature extraction method designed for the HS division and the RUL prediction. We formulate one-dimensional feature extraction as a principal signal separation task and introduce the use of U-Net to reconstruct the prognostic features for the RUL prediction. A novel domain-invariant generalized solution based on the GAN scheme is introduced to learn the invariant representation and predict the RUL.
- We transform the multiscale prognostic features into an NSP image without any domain knowledge and manual setting to combine the generalized multiscale features and reduce the computational cost.
- A CNN-based binary regression model for determination of the HS without any threshold and a CNN–LSTM model for the RUL prediction are proposed, which can predict the RUL with less errors and higher prognosis accuracy than other existing methods.
- To validate the proposed method, we conducted experiments with two rotating machinery datasets: the Fanche-Comte Electronics Mechanics Thermal Science and Optics—Sciences and Technologies Institute (FEMTO) dataset [22] and the Xi'an Jiaotong University and the Changxing Sumyoung Technology Company (XJTU-SY) dataset [23]. By the experiments on multiple datasets, we can verify the effectiveness of the proposed method on different patterns of bearing wear.

The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 provides the details of the proposed method. Section 4 presents quantitative and qualitative experimental results on the two datasets. Finally, Section 5 concludes the paper.

2. Related works

2.1. Deep learning and adversarial domain adaptation for RUL

Ren et al. [9] proposed an integrated deep neural network (DNN) that extracts both the time and frequency-domain features of vibration signals and predicts the bearing RUL. Guo et al. [10] employed a recurrent neural network (RNN) based on health indicator to predict the RUL of bearings, where six related-similarity features with eight time–frequency features were fed into the recurrent neural networks(RNN). Ren et al. [11] proposed

a deep learning-based RUL prediction method using a deep auto-encoder (DAE) and DNN. Hinch et al. [12] proposed an end-to-end RUL prediction method based on convolutional and LSTM neural networks by taking raw data as input. Zhu et al. [13] proposed a multiscale CNN-based RUL prediction method with the time–frequency representation using the wavelet transformation. Li et al. [14] proposed a deep learning-based RUL estimation method with multiscale feature extraction and FPT determination from the kurtosis of vibration signals. Ren et al. [15] proposed a deep learning method using the multiscale dense gated RNN for the RUL prediction. Qin et al. [16] proposed a gated dual attention unit for the RUL prediction.

These works have achieved satisfactory performance in RUL prediction, nonetheless the three challenges still remain a concern. Firstly, these works require domain knowledge to extract features or we have to manually specify the feature types. The various time-domain statistical features, such as signal root mean square (RMS), crest factor, kurtosis, and entropy, and the frequency-domain features, such as fast Fourier transform (FFT), Hilbert–Huang transform (HHT), and wavelet transformation, were extracted and used to predict the RUL [9–11,15]. In other words, only one time-domain feature or time–frequency feature by wavelet transformation [13], short-time Fourier transform (STFT) [14], or RMS [16] was used to predict the RUL. The RUL prediction can remove the feature extraction by simply taking raw data as input to the deep neural networks (DNNs) [12], but it results in high computation cost. Even worse, such a method cannot correct the training error, if the extracted feature is used to predict RUL with the long short-term memory (LSTM) neural networks. Secondly, though the machinery working conditions, such as motor power and rotating speed, and degradation generation conditions, such as radial force, are identical in an experimental platform to validate bearings fault detection, the bearing faults usually vary. Generally, the degradation trends are not the same for different bearings under the same conditions, resulting in significant data distribution discrepancies between entities. To overcome these problems, adversarial domain adaptation approaches and CNN-based health stage prediction methods have been proposed for RUL prediction [17–19,24]. The adversarial domain adaptation approaches have been proposed to generalize the feature extraction of bearing wear. Da Costa et al. [17] proposed a deep domain adaptation method for the RUL prediction by integrating the LSTM model with domain classification loss to reduce the deviation between source and target domains. Ragab et al. [18,19] proposed a contrastive adversarial domain adaptation approach that enables automatic feature extraction and the RUL prediction from a source domain to a target domain. However, these approaches predict the RUL without determining the HS and FPT and only focus on the single-source single-target adaptation setting which is impractical because the network of domain adaptation should be trained for data from different domains with different distributions. However, those approaches only focus on the single-source single-target adaptation setting. In real-world applications, the training data may come from multiple domains with different distributions so that it is expensive in both time and cost to train the network of domain adaptation for every condition and bearing.

Thirdly, the conventional methods predict RUL without determining the FPT. The FPT can be determined using conventional features, such as kurtosis and RMS [14,16]. Although unsupervised methods [25], such as auto-encoder, promise the extraction of efficient health indicators(HI); however, they possess a challenge assigning an appropriate threshold for determining FPT. Recently, the CNN-based HS division methods have been proposed, but they require an appropriate threshold to determine the FPT [24] or need to determine bandwidths, where there are

significant differences between normal and faulty conditions [21, 26]. Li et al. [24] proposed a GAN-based RUL prediction approach that consists of two stages. Initially, the GAN is used to learn the distributions of the data in the healthy state of the machine and determine the FPT in prognostics. The learned features are further adopted for the RUL prediction and adversarial training is also utilized for data alignments of different entities. It trains the DNN with the degradation data from multiple training data of different bearings and the healthy state data from different training and test bearings and can generalize prognostic features. However, it still requires an appropriate threshold to determine the FPT and needs the target domain data to train the DNN. Our previous works [21,26] proposed a supervised HS division method using CNN and an NSP, which transforms the time-series data into an image for feature extraction, rids the necessity of designating a threshold of health index. However, it still needs to determine bandwidths of bandpass filters manually in the data preprocessing procedure. Also, manual bandpass filter setting based on the sampled dataset can be limited, because it is hard to consider all machinery working conditions.

2.2. Nested Scatter Plot (NSP)

NSP is a data wrangling method that transforms correlated time series data into an image for multi-variate correlation analysis [21,26]. It is a matrix representation similar to the heat map of the quantized value of time series data. Each vibration signal is quantized and mapped into nested clusters. The cumulative number of signal values in the nested cluster is normalized to represent the density of the nested cluster. Although NSP removes the non-stationarity of the time sequence, it is an efficient imaging method for multi-variable correlation analysis [21].

By using NSP, we transform multi-channel vibration signals into a single fixed-size image. Continuous multi-channel vibration signals are split into given intervals. A three-step approach is used: feature extraction using bandpass filters, decomposition of the nested clusters in each bandwidth, and aggregation of the decomposed sections into a single RGB image. At least two different data channels as data sources are required and the first step is the incorporation extraction of signals in three different bandwidths. HHT and FFT are used to determine the bandwidth of bandpass filters [21]. In the second step, the extracted two-channel signals are compressed into nested clusters. Each channel signal is mapped on the x-axis and y-axis. Three different extracted signals are colored in red, green, and blue, respectively. In the final stage, three scatter plots are aggregated together to form a single RGB image.

In the previous works [21,26], the combination of NSP, which allowed us to integrate a frequency analysis that helped to extract faulty wear features, and a CNN-based binary regression model, rid the necessity of designating a threshold, one of the most crucial problems in early fault detection. The CNN based model effectively distinguished healthy and unhealthy states unambiguously by changing the anomaly detection problem into a binary regression problem, utilizing a simple trigger mechanism. To minimize the labeling process in supervised learning, a small portion of the dataset was utilized for training the model.

3. Proposed method

3.1. Generalized multiscale feature extraction

In this section, we introduce a generalized multiscale feature extraction method based on GAN scheme and U-Net architecture for the HS division and the RUL prediction. In the original formulation of GAN [27], a GAN model is trained through the min–max game between a generator network G and a discriminator D. The GAN aims to approximate the probability distribution function that certain data is assumed to be drawn from. The objective function of the min–max game between the generator and the discriminator is expressed as follows:

$$\min_G \max_D \mathbb{E}_{x \sim P_r(x)}[\log D(x)] + \mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z)))], \quad (1)$$

where x is real data sampled from the real data distribution $P_r(x)$, z is the noise vector sampled from a uniform distribution $P_z(z)$, and the generator G generates a synthetic data $G(z)$. The equation shows that treating the discriminator as a classifier minimizes the Jensen–Shannon (JS) divergence between the real data distribution and the one assumed by the generator.

Although GAN can generate synthetic data close to the real samples, the GAN training procedure has instability with respect to loss function convergence. To solve this instability problem, the Wasserstein GAN with gradient penalty (WGAN-GP) [28], which uses the Wasserstein-K distance as the loss function, was proposed to guide the training process. The loss function of the WGAN-GP is defined as follows.

$$\begin{aligned} \mathbb{L} = &-\mathbb{E}_{x \sim P_r(x)}[D(x)] + \mathbb{E}_{z \sim P_z(z)}[D(G_\theta(z))] \\ &+ \alpha \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \end{aligned} \quad (2)$$

where α is the penalty coefficient and $P_{\hat{x}}$ is the uniform sampling along straight lines between pairs of points from the real data distribution P_r and the generated distribution. The motivation for this is that the constraint is enforced uniformly along the line as the optimal discriminator consists of straight lines connecting the two distributions. The WGAN-GP provides a training procedure that is faster and more stable than the original GAN.

Fig. 1 presents the overall framework with the detailed steps of the learning procedure of the proposed feature extraction model. The GAN structure is a kind of adaptation of our previous work, classification enhancement GAN (CEGAN) [29]. CEGAN is composed of three independent networks: a discriminator, a generator, and a classifier. The classifier in CEGAN is trained with both real and generated data for preventing generated minority data from overfitting to majority data in the imbalanced data, unlike conventional GAN methods. In other words, the generated data for minority classes can prevent the classifier from overfitting to the majority data and improve the performance of the classifier, while the classifier only trained with real data can be biased to the majority class data. Because the conventional GAN methods employ the auxiliary classifier and the classifier shares network structure and weight parameters with the discriminator, the performance of the auxiliary classifier cannot lead to generating high-quality images. To generate data to improve the performance of the classifier, CEGAN employs the structure of the classifier to the independent network in the GAN structure. As the structure of the classifier is employed in the GAN structure, the GAN can generate data to improve the performance of the classifier.

By adapting the structure of CEGAN, we define three types of independent networks in the proposed GAN scheme: (1) the multiscale feature extractor as a generator network, (2) a discriminator to separate the features generated by the generator from the real input data and to distinguish the data from different domains, (3) the CNN-based HS division and the LSTM-based RUL predictor as the classifier network.

3.1.1. Generator

The generator is composed of three multiscale generators as shown in Fig. 1. The proposed generator adopts basic concepts from U-Net [20] for one-dimensional(1D) time-series feature reconstruction by substituting the original 2D convolutions with
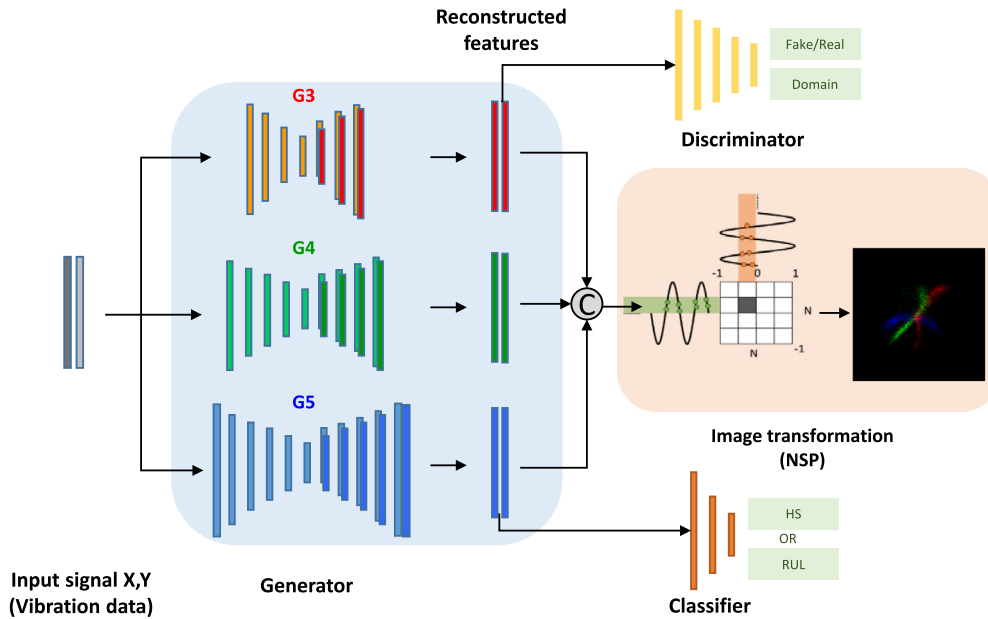
Fig. 1. The structure of the proposed framework for the generalized multiscale feature extraction.

1D operations. U-Net [20] was invented for image segmentation, concatenating feature maps from different levels to improve segmentation performance and it combines low-level detail information with high-level semantic information. It has achieved promising performance on various image segmentation problems. Based on CNN, the U-Net architecture composes of an encoder, a decoder, and skipped connections between the encoder and the decoder. The encoder is composed of several levels of convolutional operations, which extract increasingly abstract representations of the input data, and a down sampling block to reduce input complexity. The convolutional operations followed by max-pooling down sampling applied to encode the input image into feature representations at multiple different levels. The decoder block is also composed of several levels of convolutional operations, up sampling blocks, and concatenation blocks. The decoder block semantically projects the discriminative features learned by the encoder onto higher resolution image space to get a dense classification. Unlike the general usage of U-Net for image segmentation task, we found that the structure of the U-Net enables the network to consider the deep and the shallow features at the same time and improve the effectiveness of the feature extraction. Thus, it helps to reconstruct the features to satisfy the conditions we designed. Whereas Gaussian noise is input to the generator in the general GAN, we provide noise only in the form of dropout applied on several layers of the generator. The architecture of the generators is shown in Fig. 2. The generator consists of the encoder and the decoder which process the input data.

Even though the U-Net enables the network to consider the deep and the shallow features at the same time, the U-Net with deep levels of convolutional operations usually focuses on more localized features and the detail information of input can be lost at the higher feature levels. Therefore, the integration of different feature levels may lead to a better performance of the feature extraction. The three multiscale generators have a different number of encoder and decoder blocks: (1) three encoder–decoder blocks (G3), (2) four encoder–decoder blocks (G4), (3) five encoder–decoder blocks (G5). The proposed multiscale generators could use two encoder–decoder blocks (G2) and five encoder–decoder blocks (G5). However, the G2 structure is too shallow to extract the correct features and the G6 is too heavy while generating the same result as G5.

3.1.2. Discriminator

The proposed GAN was developed for purpose of domain adaptation from multiple sources. Generally, the discriminator is trained to distinguish generated and real data in GAN. However, our discriminator also has a functionality of domain discriminator which distinguishes the data domain. It is because the proposed framework for the generalized multiscale feature extraction aims to extract the features of bearing wear from multiple domains. In this study, the data domain implies what bearings and under what conditions they were collected. The adversarial training enables the generator to extract high-level features containing domain-unbiased information, and makes it hard for the discriminator to classify only real or fake (denoted as a degree) as well as source domain. To do so, the convolutional layer of discriminator extract the features of input (reconstructed features of real and fake data from the generator) with a Leaky ReLU, and then the two separated linear layers output the true degree and the domain classification, respectively (shown in Fig. 3).

3.1.3. Classifier

As mentioned in Section 1, it is necessary to determine the HS of the machinery for accurate RUL prediction, rather than a regression model is trained directly for RUL prediction, because the machine in a healthy state would not have any remarkable difference in the run-to-failure bearing wear. To improve the performance of the HS division and the RUL prediction, we employ the structure of the classifier network to the independent network in the proposed GAN structure. In Fig. 1, the first stage involves a generalized multiscale feature extraction G_{HS} for the HS division and the HS division model C_{RUL} with the CNN and the transformed NSP images from the extracted features. In the second stage, a generalized multiscale feature extraction model G_{RUL} for the RUL prediction is trained and a CNN–LSTM model C_{RUL} predicts the RUL with the transformed NSP images from the extracted features.

3.1.4. Loss functions and training procedure

In this study, it is assumed that the vibration signals are collected by two high-frequency vibration sensors placed horizontally and vertically on each bearing, and a number of N_{train} run-to-failure vibration data in the whole life cycle from different
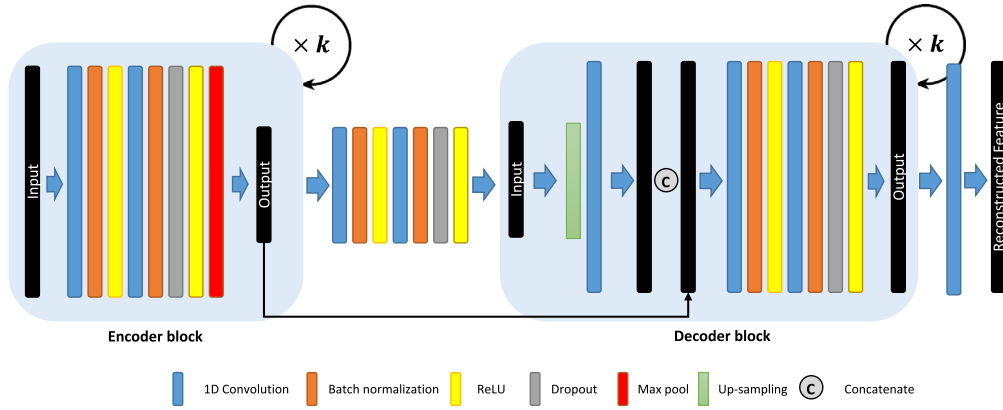
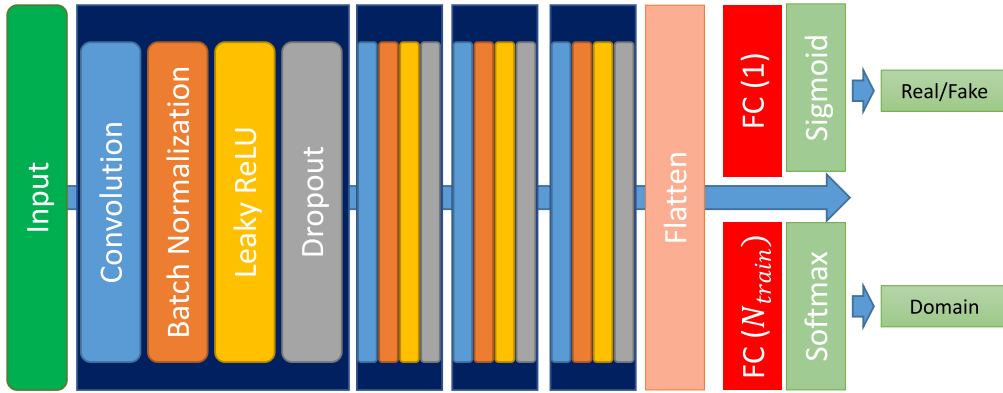**Fig. 2.** Structure of the generator based on U-Net.

Fig. 3. Structure of the discriminator.

bearings can be used to train the proposed DNNs. Suppose that $X_j = \{x_j^i\}_{i=1}^{n_j} \in \mathcal{R}^{N_{samples}}, j = 1, 2, \ldots, N_{train}$ denotes n_j sequential training samples from the jth bearing, where $N_{samples}$ is the dimension of the sample. In each bearing data, two samples exist: each from horizontal and vertical vibration sensors.

We aim to extract generalized prognostic features to improve the performance of the HS division and the RUL prediction. In the HS division, labeling of the target data is necessary for supervised learning to distinguish the features of the HS of a machine under different bearing degradation conditions. The training dataset can be divided into two health stages based on the time of acquisition. The initial part, corresponding to the nominal functioning in the entire vibration dataset, is labeled as healthy, and the last part of the sample duration, when the bearing is damaged, is labeled as unhealthy. This is on the assumption that degrading data of rotating machinery are obtained until the point of failure. This manner not only reduces the effort of labeling but also improves the prediction of HS division when no ground truth of HS is available (most of the open dataset has no information about HS). The corresponding HS labels are expressed as follows.

$$y_{HS_j}^i = \begin{cases} 0, & \text{if } i < n_j \times p \\ 1, & \text{if } i > n_j \times (1-p) \end{cases} \tag{3}$$

where $y_{HS_j}^i$ represents the HS label of x_j^i and p is the percentage of the total degradation process. The percentage of the total degradation process is used for labeling the first portion p of the total degradation process as healthy and the last portion p as unhealthy. The assumption is labeling a small portion of the run-to-failure dataset where the signals are clearly distinguishable into healthy and unhealthy when no labels for health stages are given. In the previous study [26], we found that the FPT

is not sensitive to p ranging from 0.05 to 0.5. $p = 0.05$ is used in the experiments in Section 4. By the CNN-based binary regression model, the HS and the FPT of the jth bearing, $T_{FPT j}$, can be determined.

Next, the RUL labels are expressed as follows.

$$y_{RUL_j}^i = \frac{n_j - i}{n_j - T_{FPT j}} \tag{4}$$

where $y_{RUL_j}^i$ denotes the RUL label of x_j^i.

The generator network is decomposed into two sub-generators: G_{HS} and G_{RUL}. The two sub-generators correspond to discriminators D_{HS} and D_{RUL} with the same structure. For the stability of the training procedure and the quality of the generated data, we apply the WGAN-GP to the objective function to guide the training process. The objective functions are defined as follows.

$$\mathbb{L}_D(x, d; \theta_D) = -\mathbb{E}_x[D_R(x)] + \mathbb{E}_x[D_R(G(x))] \\ + \alpha \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D_R(\hat{x})\|_2 - 1)^2] + \lambda_D \mathbb{L}_{CE}(D(x), d), \tag{5}$$

$$\mathbb{L}_G(x, d, y; \theta_G) = \mathbb{E}_x[D_R(G(x))] + \lambda_G \mathbb{L}_C(G(x), y) - \lambda_D \mathbb{L}_{CE}(D_D(x), d), \tag{6}$$

$$\mathbb{L}_C(x, y; \theta_C) = \begin{cases} \mathbb{L}_{C_{HS}}(G(x), y), & \text{for the HS division} \\ \mathbb{L}_{C_{RUL}}(G(x), y), & \text{for the RUL prediction} \end{cases} \tag{7}$$

where x is a sequential training sample, y is the HS label in the case of the HS division and the RUL label in the case of the RUL prediction, d is the associated label of the domain, λ_D is a hyperparameter that controls the effect of domain generalization, λ_G controls the relative importance of different loss terms, $\mathbb{L}_{CE}$ denotes the standard cross-entropy loss function, and θ_D, θ_G, and

θ_C are parameters of the discriminator, the generator, and the classifier, respectively. Both the real data and generated data are entered into the discriminator, while the data reality and the domain classification are the output of the discriminator D_R and D_D, respectively. Whereas the discriminator D is trained to minimize $\mathbb{L}_D$ for distinguishing the data from different domains and between the real and the generated data simultaneously, the generator G is trained to minimize $\mathbb{L}_G$. Furthermore, the classifier C, which is the CNN-based HS division C_{HS} or the LSTM-based RUL predictor C_{RUL}, is trained to classify the HS or predict the RUL. In other words, the generator reconstructs the generalized features to fool the discriminator and to improve the performance of the HS division and the RUL prediction. Additionally, by reconstructing features which are similar to the real data, feature values can be generated within a certain range, and the time-series properties can be maintained. Finally, the classifier losses for the HS division and the RUL prediction are defined as follows.

$$
\begin{aligned}
\mathbb{L}_{C_{HS}}(G(x), y) =& \mathbb{L}_{BCE}(G(x), y), \\
\mathbb{L}_{C_{RUL}}(G(x), y) =& \lambda_{MAE}\mathbb{L}_{MAE}(G(x), y) + \lambda_{RMSE}\mathbb{L}_{RMSE}(G(x), y) \\
& + \lambda_{MAPE}\mathbb{L}_{MAPE}(G(x), y)
\end{aligned}
\tag{8}
$$

where $\mathbb{L}_{BCE}$ denotes the standard binary cross-entropy loss, $\mathbb{L}_{MAE}$ is mean absolute error (MAE) in (9), $\mathbb{L}_{RMSE}$ is root mean square error (RMSE) in (10), and $\mathbb{L}_{MAPE}$ is mean absolute percentage error (MAPE) in (11).

Three different reconstructed features are transformed into three channels of an NSP image by the following two steps. The first step is to compress the reconstructed two-channel features into nested clusters. Each channel feature is mapped onto the x- and y-axis and three different reconstructed features from the three multiscale generators are colored in red, green, and blue, respectively. To represent the intensity of each cluster from the reconstructed features, the mapped values count was translated into pixel intensity. In the second step, three scatter plots are aggregated together to form a single RGB image as shown in Fig. 1.

The training details for the proposed multiscale feature extraction method are summarized in Algorithm 1.

3.2. Health stage division and remaining useful life prediction with NSP

By reconstructing continuous raw vibration time-series data to multiscale features and transforming the features into NSP images, we can change the signal processing problem to an image classification problem and classify the images by using the CNN for the HS division model (CNN-HS) [26] and the CNN–LSTM for the RUL prediction model (CNN–LSTM-RUL). The structures of the proposed CNN-HS and CNN–LSTM-RUL are presented in Fig. 4.

The trained CNN-HS and the trained CNN–LSTM can classify the NSP of an external test dataset, which was not used in training but operated under the same conditions as the rest of the training datasets. For the HS division, the binary regression results of the whole run-to-failure are computed by the CNN-HS, wherein specifying a threshold value is not necessary. As the trained CNN-HS learns the differences in the features of the NSP for healthy and unhealthy data in the training datasets, it can recognize the degradation pattern of all of the data. After determining the FPT using the CNN-HS, extracting the multiscale features, and transforming the features into the NSP images, the CNN–LSTM can recognize the degradation patterns of all of the data from the FPT to the end time. The CNN in the CNN–LSTM perform feature extraction from the transformed NSP images, and the LSTM and the RUL prediction part calculates the percentage of the RUL.

Algorithm 1 Training procedure for generalized multiscale feature extraction using adversarial networks. We use default values of $\alpha = 10$, $\lambda_D = 1$, $\lambda_G = 50$

Require: Batch size m, Adam hyperparameters η, hyperparameter for λ_D, λ_G.

1: **Input:** $\{x_j^i\}_{i=1}^{n_j} \in \mathcal{R}^{N_{samples} \times 2}$, $j = 1, 2, ..., N_{train}$, y, d.
2: **for** number of training iterations **do**
3:      Sample $\{x_j^i\}_{i=1}^m$ a batch from the training dataset, corresponding HS or RUL label $\{y_j^i\}_{i=1}^m$, and corresponding domain label $\{d^i\}_{i=1}^m$.
4:      **for** $k =$ 3 to 5 **do**
5:          Update discriminator D by descending the gradient of (5) with G_k:
6:          $\theta_D \leftarrow \theta_D - \eta_D \nabla_{\theta_D} \mathbb{L}_D(x, d; \theta_D)$
7:          Update generator G_k by descending the gradient of (6):
8:          $\theta_{G_k} \leftarrow \theta_{G_k} - \eta_G \nabla_{\theta_{G_k}} \mathbb{L}_G(x, d, y; \theta_{G_k})$
9:          Update classifier C by descending the gradient of (7) and (8) with G_k:
10:          $\theta_C \leftarrow \theta_C - \eta_C \nabla_{\theta_C} \mathbb{L}_C(x, y; \theta_C)$
11: **for** $k = $ 3 to 5 **do**
12:      Sample $\{x_j^i\}_{i=1}^m$ a batch from the dataset
13:      Transform $G_k(x)$ into nested clusters
14: Merge three scatter plots to a single RGB image

---

4. Experimental results

4.1. Datasets and evaluation metrics

To evaluate the proposed method (denoted as GMFE) on the run-to-failure vibration dataset, two types of popular datasets were used: FEMTO [22] and XJTU-SY dataset [23]. The FEMTO dataset was collected by the PRONOSTIA test rig and has been available to the public since the IEEE PHM 2012 Prognostic Challenge (PHM 2012). The test rig mainly contains an asynchronous motor, a shaft, a speed controller, an assembly of two pulleys, and tested rolling ball bearings, which is shown in Fig. 5(a). The vibration data in the horizontal direction were investigated. The dataset was composed of 17 run-to-failure data in which a single bearing was tested (two columns of vibration data: horizontal and vertical). The sampling frequency of data acquisition is 25.6 KHz and the sampling time length was 0.1 s, which was recorded per 10 s. The 17 datasets are grouped into three sections by operating conditions that differ in rotation speed and radial load. Each dataset has a different run-to-failure time, requiring fault detection methods that are adaptable to time-varying operational conditions and environments. When the amplitude of the vibration data exceeds 20 g, the run-to-failure experiments were stopped and the bearing is considered defective. More details about the test rig and the datasets can be referred from [22]. The XJTU-SY dataset contains complete run-to-failure data of 15 rolling element bearings that were collected by the Xi'an Jiaotong University and the Changxing Sumyoung Technology [23]. The vibration signals were collected by two accelerometers placed at 90 degrees on the housing of the tested bearings, as shown in Fig. 5(b). The sampling frequency is 25.6 KHz and the sampling time length was 1.28 s, which was recorded per minute. Like the FEMTO dataset, the accelerated degradation bearings tests are stopped when the amplitude of the vibration data exceeds 20 g.

In the experiment, 14 run-to-failure datasets under two different operating conditions and 10 run-to-failure datasets under two different operating conditions are used from the FEMTO dataset and the XJTU-SY dataset, respectively. The dataset information for each experiment condition is presented in Table 1.
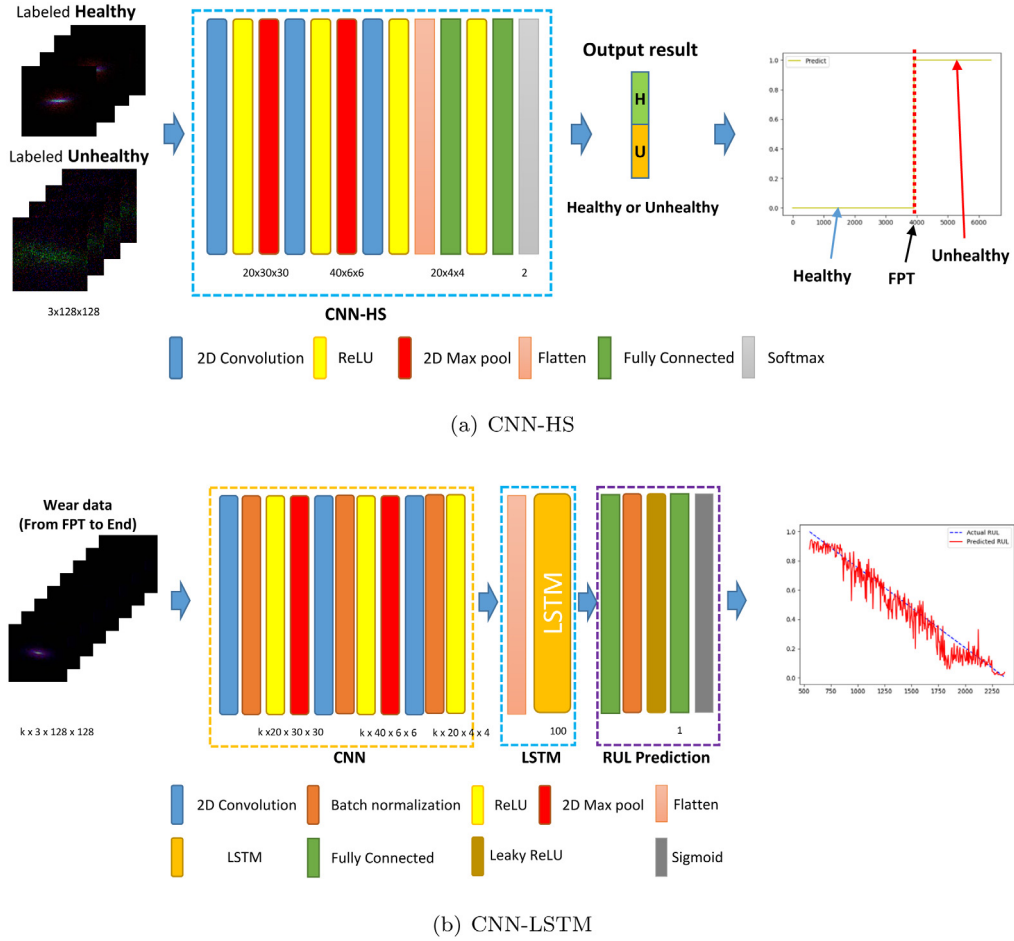
(a) CNN-HS

(b) CNN-LSTM

Fig. 4. Proposed deep neural network architecture, (a) the proposed CNN-HS, (b) the proposed CNN–LSTM-RUL.

Table 1
Dataset information.

Dataset	Condition	Load (N)	Speed (rpm)	The number of bearings
FEMTO	1	4000	1800	7
FEMTO	2	4200	1650	7
XJTU-SY	1	12000	2100	5
XJTU-SY	2	11000	2250	5

To evaluate and compare the performance of the proposed method with other methods, we adopted three evaluation metrics, which are used in various RUL prediction methods: MAE, RMSE, and MAPE which are calculated as follows.

$$MAE = \frac{1}{N_S} \sum_{i=1}^{N_S} |ActRUL_i - PreRUL_i| \qquad (9)$$

$$RMSE = \sqrt{\frac{1}{N_S} \sum_{i=1}^{N_S} (ActRUL_i - PreRUL_i)^2} \qquad (10)$$

$$MAPE = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{|ActRUL_i - PreRUL_i|}{ActRUL_i} \qquad (11)$$

where N_S is the number of testing samples, and $ActRUL_i$ and $PreRUL_i$ are the actual RUL and the predicted RUL values corresponding to the ith testing sample, respectively.

4.2. Implementation details

The experiments were all implemented using Python scripts in the PyTorch framework. Training procedures were conducted in the Linux system with four NVIDIA Quadro RTX 8000 GPUs. Our parameter settings for the HS division and the RUL prediction were the same except for λ_G: $\lambda_G = 50$ for the HS division and $\lambda_G = 1$, $\lambda_{MAE} = 100$, $\lambda_{RMSE} = 50$, $\lambda_{MAPE} = 20$ for the RUL prediction. Through various testing, it is observed that the parameters mentioned in the paper yield the best performance. We chose the Adam optimizer with a learning rate of $\eta_C = \eta_G = \eta_D = 1 \times 10^{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.99$. The batch size was 40 and the sequence lengths in the CNN–LSTM model for the FEMTO and the XJTU-SY datasets are 5 and 4, respectively.

4.3. Results

Fig. 6 shows the examples of transformed NSP images on four example testing bearings from the FEMTO and the XJTU-SY datasets. The transformed NSP images can efficiently represent the health of machinery by extracting useful features from raw vibration data.

To evaluate the proposed method, while one bearing dataset is used for testing, the other bearings under the same operating conditions are adopted to train the DNNs. The experimental results are averaged by five trials to reduce the effect of model randomness. To address the advantage of the proposed method, we compare the proposed method with the multiscale CNN method (MCNN) by Li et al. [14], the deep adversarial neural networks
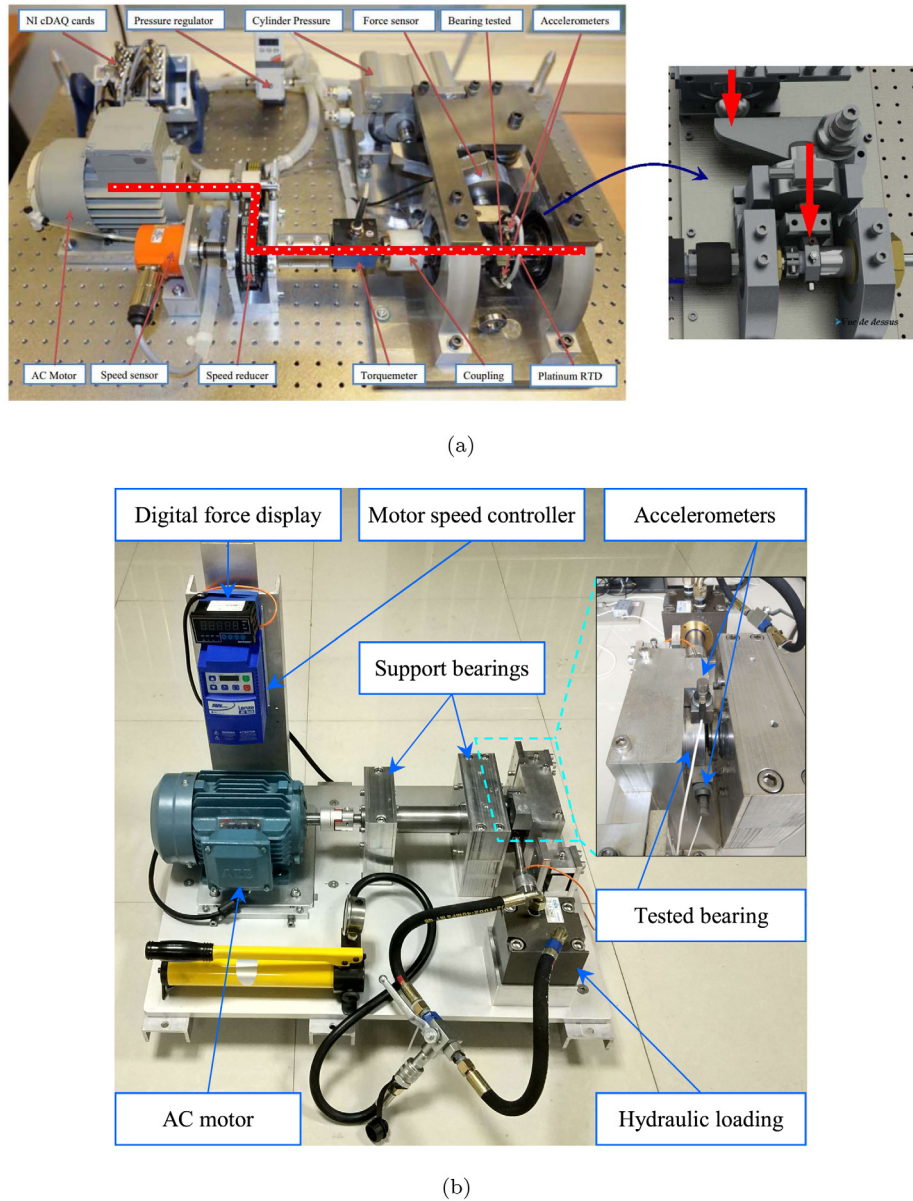
(a)

(b)

Fig. 5. Bearing test rigs (a) of the PRONOSTIA in the FEMTO dataset [22], (b) in the XJTU-SY dataset [23].

(DANN) by Li et al. [24], and the proposed CNN–LSTM method with the supervised HS division method [26]. We implemented MCNN to compare to the proposed method by following the detailed experimental setting in [14] because the XJTU-SY dataset was not used in [14]. Table 2 shows the quantitative evaluation results comparing to other methods by using three evaluation metrics. It is observed that the proposed method showed much lower prediction errors than others in terms of MAE and RMSE, while the MAPE of the proposed method is slightly higher than DANN [24]. When testing the various parameter setting, we found that more weight on MAE and RMSE than MAPE in the loss function yields the best performance of RUL prediction in all the ranges. If the weight on MAPE increases, the tendency of the predicted RUL decreases, and the overall performance is degraded. It is the reason why $\lambda_{MAPE} = 20$ is a lower value than $\lambda_{MAE} = 100$ and $\lambda_{RMSE} = 50$ in the best parameter setting for loss function.

To evaluate the effect of FPT and NSP, the ablation study is conducted. 'No FPT' is adopted and the degradation is considered to start at the beginning of machine operation to indicate the benefits of the HS division and FPT determination. The '1D-GAN'

method is implemented without merging the multiscale features by transforming into the NSP image and the proposed CNN–LSTM-RUL to show the advantage of the transformed NSP images and the CNN–LSTM-RUL. The proposed method compared to 'No FPT' and '1D-GAN' showed better results, indicating that the proposed HS division method is effective in capturing machine degradation at the beginning point and the CNN–LSTM-RUL with the transformed NSP images enhances the generalization ability on various datasets.

Fig. 7 presents the RUL prediction results on four testing bearings in the two datasets. The degradation evaluations start when the FPTs of the testing bearings are detected. It can be observed that the degrading patterns are effectively captured by the proposed method and the errors between the actual RUL and the predicted RUL are quite small.

5. Conclusion

In this study, we have proposed generalized multiscale feature extraction and RUL prediction method using multiscale GAN. To
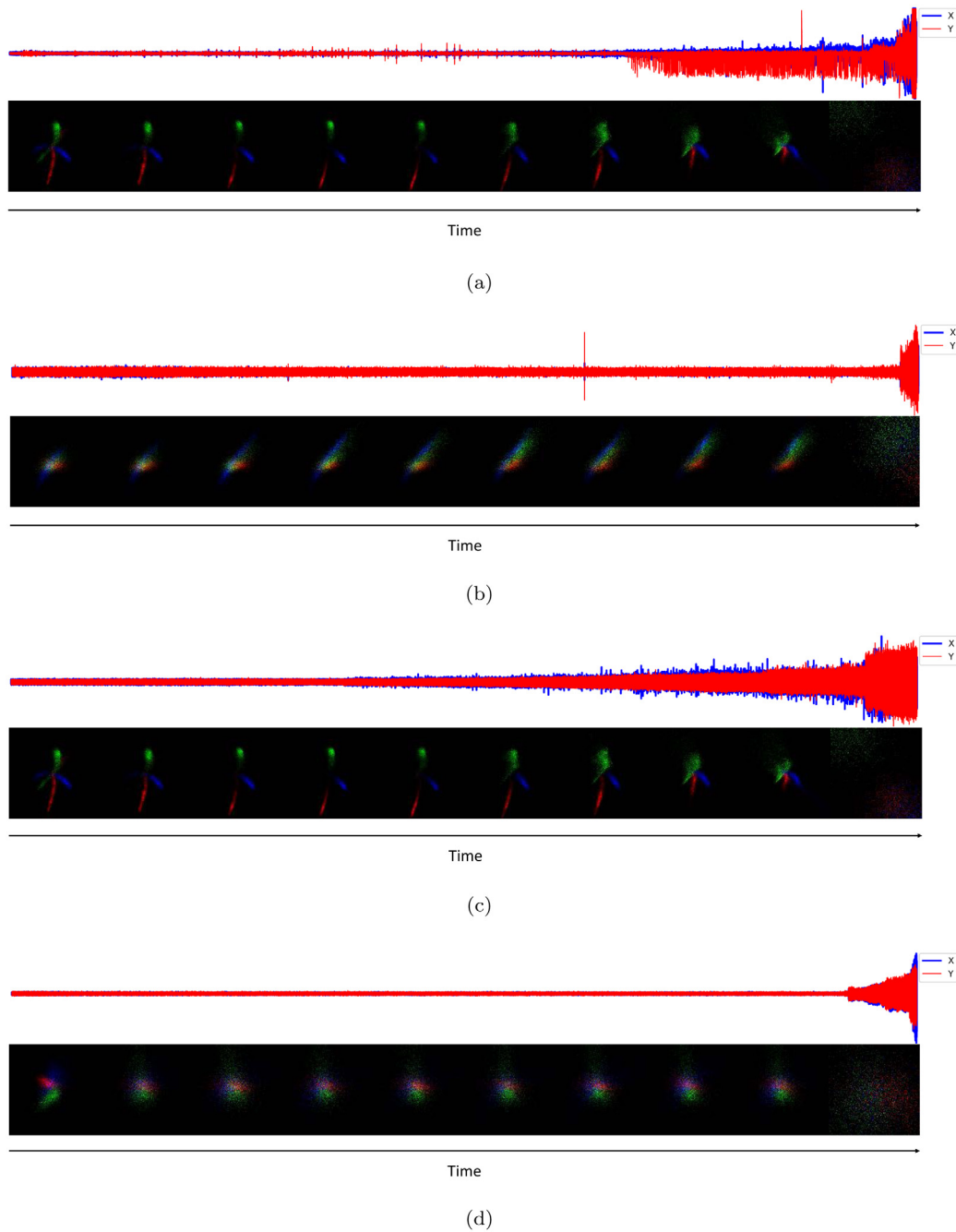
Fig. 6. Examples of transformed NSP images on four testing bearings. (a) Bearing 1–3 in the FEMTO dataset, (b) Bearing 2–6 in the FEMTO dataset, (c) Bearing 1–3 in the XJTU-SY dataset, (d) Bearing 2–1 in the XJTU-SY dataset.

extract generalized prognostic features, we designed a multiscale 1D U-Net architecture for the generator and adversarial learning procedure between the generator and the discriminator which contains the reality and domain discriminator. The adversarial learning procedure learns the distributions of multiple training data from different bearings and extracts domain-invariant generalized prognostic features. The proposed method does not require any domain knowledge and manual setting to extract generalized features and predict the RUL. A CNN-based binary regression model for the determination of the HS without any threshold and a CNN–LSTM model for the RUL prediction have been proposed, which can predict the RUL with fewer errors and higher prognosis accuracy than other existing methods. The experimental results indicated that the proposed method, where

CNN and CNN–LSTM are combined with NSP based on the reconstructed features of bearing wear, could capture the degrading patterns effectively and achieved better RUL prediction results compared to other methods.

To further improve the performance of the proposed method, more amounts training data are required. However, annotated data in the whole life cycle of bearings are hard to collect in the real industry fields and to define the healthy condition of bearings. Further related research works have to be conducted on unsupervised bearing degradation data. In the future, we plan to extend the proposed method to fault diagnosis and prognosis of other components in manufacturing environments and further develop the fault prediction model with attention mechanism and unsupervised transfer learning.

Table 2
Evaluation of RUL prediction on the FEMTO and XJTU-SY datasets.

| Methods	Metric	FEMTO Condition1	FEMTO Condition2	XJTU-SY Condition 1	XJTU-SY Condition 2
MCNN [14]	MAE	0.1856 ± 0.0546	0.1738 ± 0.0521	0.2274 ± 0.0496	0.2063 ± 0.0427
	RMSE	0.2344 ± 0.0701	0.2105 ± 0.0557	0.2484 ± 0.0452	0.2302 ± 0.0319
	MAPE	0.8943 ± 0.4802	1.3567 ± 0.7913	1.1659 ± 0.3749	1.1802 ± 0.3945
DANN [24]	MAE	0.1739 ± 0.0542	0.1771 ± 0.0515	0.2588 ± 0.0499	0.2112 ± 0.0891
	RMSE	0.2063 ± 0.0603	0.2060 ± 0.0578	0.2966 ± 0.0587	0.2398 ± 0.0973
	MAPE	**0.4633 ± 0.0811**	**0.4970 ± 0.1200**	**0.4340 ± 0.0816**	**0.3872 ± 0.1535**
CNN-HS [26]	MAE	0.1697 ± 0.0267	0.2694 ± 0.0913	0.1919 ± 0.0469	0.1238 ± 0.0434
	RMSE	0.1959 ± 0.0297	0.2932 ± 0.0890	0.2172 ± 0.0505	0.1429 ± 0.0419
	MAPE	0.7983 ± 0.4179	1.2953 ± 0.9574	0.6574 ± 0.3538	0.6563 ± 0.4435
GMFE (ours)	MAE	**0.0906 ± 0.0237**	**0.1644 ± 0.0456**	**0.1620 ± 0.0476**	**0.1132 ± 0.0476**
	RMSE	**0.1053 ± 0.0222**	**0.1870 ± 0.0408**	**0.1795 ± 0.0488**	**0.1259 ± 0.0488**
	MAPE	0.4794 ± 0.0859	0.8515 ± 0.5413	0.5306 ± 0.1445	0.5721 ± 0.4197
No FPT	MAE	0.0678 ± 0.0212	0.2142 ± 0.0591	0.2541 ± 0.0802	0.2145 ± 0.0790
	RMSE	0.0842 ± 0.0274	0.2317 ± 0.0540	0.2647 ± 0.0770	0.2217 ± 0.0747
	MAPE	0.5860 ± 0.3584	1.3122 ± 1.0572	2.7105 ± 3.3264	0.9537 ± 0.6268
1D-GAN	MAE	0.1983 ± 0.0944	0.2879 ± 0.1059	0.2393 ± 0.1239	0.1295 ± 0.0664
	RMSE	0.2099 ± 0.0936	0.3143 ± 0.1098	0.2529 ± 0.1200	0.1427 ± 0.0650
	MAPE	0.8234 ± 0.4407	1.5198 ± 1.3135	0.6782 ± 0.2530	1.2928 ± 1.6532

(a)　(b)　(c)　(d)
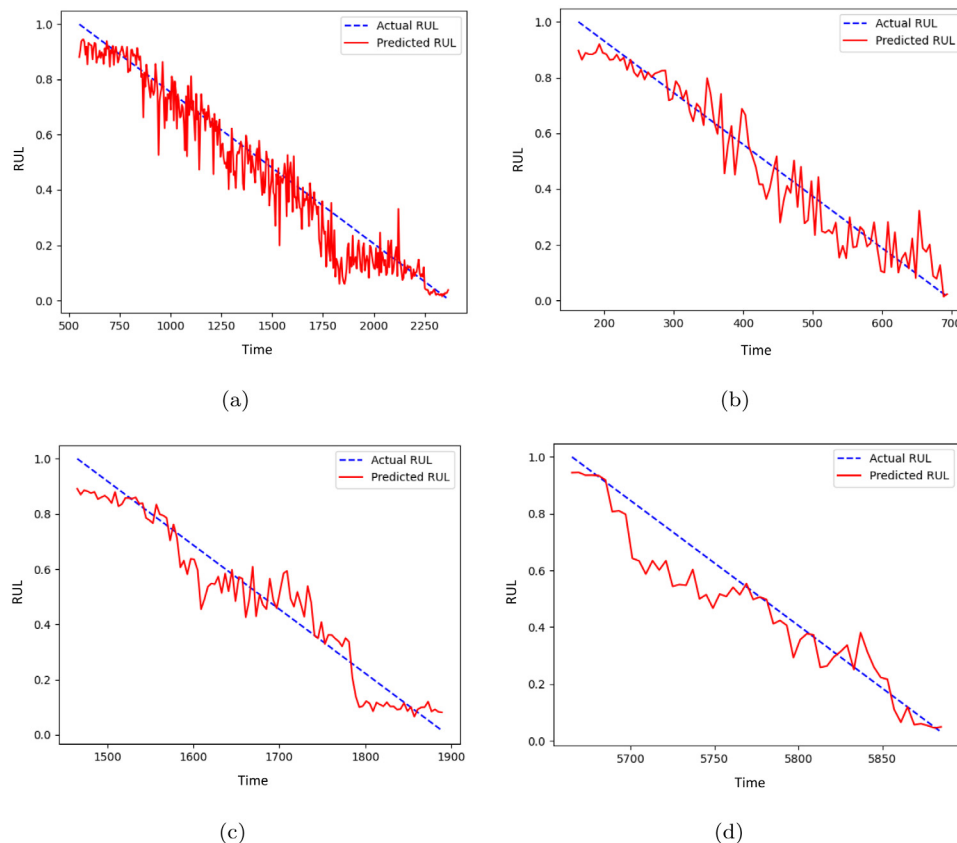
Fig. 7. RUL prediction results on four testing bearings. (a) Bearing 1–3 in the FEMTO dataset, (b) Bearing 2–6 in the FEMTO dataset, (c) Bearing 1–3 in the XJTU-SY dataset, (d) Bearing 2–1 in the XJTU-SY dataset.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

References

[1] G.Z. Vachtsevanos, Intelligent Fault Diagnosis and Prognosis for Engineering Systems, John Wiley & Sons, 2006.
[2] N. Li, Y. Lei, J. Lin, S.X. Ding, An improved exponential model for predicting remaining useful life of rolling element bearings, IEEE Trans. Ind. Electron. 62 (12) (2015) 7762–7773.
[3] Y. Qian, R. Yan, R.X. Gao, A multi-time scale approach to remaining useful life prediction in rolling bearing, Mech. Syst. Signal Process. 83 (2017) 549–567.
[4] M. Pecht, J. Gu, Physics-of-failure-based prognostics for electronic products, Trans. Inst. Meas. Control 31 (3–4) (2009) 309–322.

[5] X. Guo, L. Chen, C. Shen, Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis, Measurement 93 (2016) 490–502.

[6] A.P. Ompusunggu, J.-M. Papy, S. Vandenplas, Kalman-filtering-based prognostics for automatic transmission clutches, IEEE/ASME Trans. Mechatronics 21 (1) (2015) 419–430.

[7] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, N. Zerhouni, Particle filter-based prognostics: Review, discussion and perspectives, Mech. Syst. Signal Process. 72 (2016) 2–31.

[8] C. Chen, G. Vachtsevanos, M.E. Orchard, Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach, Mech. Syst. Signal Process. 28 (2012) 597–607.

[9] L. Ren, J. Cui, Y. Sun, X. Cheng, Multi-bearing remaining useful life collaborative prediction: A deep learning approach, J. Manuf. Syst. 43 (2017) 248–256.

[10] L. Guo, N. Li, F. Jia, Y. Lei, J. Lin, A recurrent neural network based health indicator for remaining useful life prediction of bearings, Neurocomputing 240 (2017) 98–109.

[11] L. Ren, Y. Sun, J. Cui, L. Zhang, Bearing remaining useful life prediction based on deep autoencoder and deep neural networks, J. Manuf. Syst. 48 (2018) 71–77.

[12] A.Z. Hinchi, M. Tkiouat, Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network, Procedia Comput. Sci. 127 (2018) 123–132.

[13] J. Zhu, N. Chen, W. Peng, Estimation of bearing remaining useful life based on multiscale convolutional neural network, IEEE Trans. Ind. Electron. 66 (4) (2018) 3208–3216.

[14] X. Li, W. Zhang, Q. Ding, Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction, Reliab. Eng. Syst. Saf. 182 (2019) 208–218.

[15] L. Ren, X. Cheng, X. Wang, J. Cui, L. Zhang, Multi-scale dense gate recurrent unit networks for bearing remaining useful life prediction, Future Gener. Comput. Syst. 94 (2019) 601–609.

[16] Y. Qin, D. Chen, S. Xiang, C. Zhu, Gated dual attention unit neural networks for remaining useful life prediction of rolling bearings, IEEE Trans. Ind. Inf. (2020).

[17] P.R.d.O. da Costa, A. Akcay, Y. Zhang, U. Kaymak, Remaining useful lifetime prediction via deep domain adaptation, Reliab. Eng. Syst. Saf. 195 (2020) 106682.

[18] M. Ragab, Z. Chen, M. Wu, C.K. Kwoh, X. Li, Adversarial transfer learning for machine remaining useful life prediction, in: 2020 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2020, pp. 1–7.

[19] M. Ragab, Z. Chen, M. Wu, C.S. Foo, C.K. Kwoh, R. Yan, X. Li, Contrastive adversarial domain adaptation for machine remaining useful life prediction, IEEE Trans. Ind. Electron. (2020).

[20] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[21] S. Suh, H. Lee, J. Jo, P. Lukowicz, Y.O. Lee, Generative oversampling method for imbalanced data on bearing fault detection and diagnosis, Appl. Sci. 9 (4) (2019) 746.

[22] K. Javed, R. Gouriveau, N. Zerhouni, P. Nectoux, A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling, in: 2013 IEEE Conference on Prognostics and Health Management (PHM), IEEE, 2013, pp. 1–7.

[23] B. Wang, Y. Lei, N. Li, N. Li, A hybrid prognostics approach for estimating remaining useful life of rolling element bearings, IEEE Trans. Reliab. (2018).

[24] X. Li, W. Zhang, H. Ma, Z. Luo, X. Li, Data alignments in machinery remaining useful life prediction using deep adversarial neural networks, Knowl.-Based Syst. (2020) 105843.

[25] E. Principi, D. Rossetti, S. Squartini, F. Piazza, Unsupervised electric motor fault detection by using deep autoencoders, IEEE/CAA J. Autom. Sin. 6 (2) (2019) 441–451.

[26] S. Suh, J. Jang, S. Won, M.S. Jha, Y.O. Lee, Supervised health stage prediction using convolutional neural networks for bearing wear, Sensors 20 (20) (2020) 5846.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, 2017, pp. 5767–5777.

[29] S. Suh, H. Lee, P. Lukowicz, Y.O. Lee, CEGAN: Classification enhancement generative adversarial networks for unraveling data imbalance problems, Neural Netw. 133 (2021) 69–86.