ISA Transactions 129 (2022) 540–554

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Practice article

A multi-branch redundant adversarial net for intelligent fault diagnosis of multiple components under drastically variable speeds

Zhen Shi [a], Xuan Liu [a], Jinglong Chen [a,*], Yanyang Zi [a], Zitong Zhou [b]

[a] State Key Laboratory for Manufacturing and Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, PR China
[b] Shaanxi Fast Gear Co., Ltd., Xi'an 710119, PR China

ARTICLE INFO

Article history:
Received 1 November 2021
Received in revised form 27 December 2021
Accepted 5 January 2022
Available online 19 January 2022

Keywords:
Generative adversarial network
Intelligent fault diagnosis
Multi-branch network
Redundant second generation wavelet transform
Varying speed

ABSTRACT

Intelligent fault diagnosis with small training samples plays an important role in the safety of mechanical equipment. However, affected by sharp speed variation, fault feature is extremely weak, which raises difficulty for fault diagnosis. The mutual coupling of multi-component fault features further increases the difficulty. Considering the ability of redundant second generation wavelet transform in non-stationary feature extraction, a multi-branch redundant adversarial net (RedundancyNet) is proposed to address the above issues. The Net consists of discriminator, the generator based on redundant reconstruction, and the classifier based on redundant decomposition. Firstly, through adversarial training process, the generator fuses multi-scale features to generate the signal with varying speeds, thereby expanding training data. Secondly, through layer-by-layer multi-resolution feature enhancement, the classifier boosts weak fault features of vibration signals at variable speeds. Finally, a multi-branch framework is proposed to realize multi-component fault location and damage identification. The proposed method is validated on two cases. The average classification accuracy in the two cases reach 97.14% and 98.33% respectively. However, other end-to-end intelligent fault diagnosis methods for varying speeds or small samples can only reach the highest classification accuracy of 95.14% in Case 1 and 93.59% in Case2, which is much less than RedundancyNet. The analysis results highlight the effectiveness of the net under drastically variable speeds and small faulty training samples. Besides, the proposed classifier is easy to understand, which reveals the process of feature learning and the extracted feature under varying speeds.

1. Introduction

Mechanical equipment, as the core part of modern industry, has become more and more intelligent and complicated with the development of technology. Once a failure occurs, it might result in significant economic losses and even disasters [1]. As a result, fault diagnosis is vital to ensure the safety of equipment in the modern industry. However, traditional fault diagnosis methods depend on expert experience and have limitations in processing big data [2]. Therefore, intelligent fault diagnosis (IFD) has become a research hotspot [3].

Vibration analysis has gotten the greatest attention among all IFD methodologies due to its advantages in revealing fault characteristics, low cost, and easy installation [4,5]. However, the majority of existing IFD methods based on vibration analysis are geared toward mechanical equipment that runs at constant speeds. In fact, on the one hand, a large number of critical machines operate under non-stationary speed. On the other hand, vibration signals at varying speeds frequently convey more defect information. Simultaneously, the fault features of mechanical equipment under varying speeds are time-varying [6], making feature extraction more difficult. As a result, the research of IFD technique under varying speeds has a broad prospect of engineering applications.

For IFD under variable speeds, scholars have proposed some solutions. These methods mainly include two categories. One is to employ a two-dimensional deep learning method for fault classification after turning vibration signal into time–frequency graph (TFG) [7]. The other is to utilize one-dimensional deep neural network to detect the fault of equipment after resampling vibration signals into angular-domain equal interval signals. To transform vibration signal into TFG, the former primarily utilizes traditional non-stationary signal processing methods such as synchroextracting transform [8], wavelet transform [9], and short-time Fourier transform (STFT) [10]. The latter employs order tracking to convert time-varying vibration signals into angular-domain stationary signals, and then realizes the fault diagnosis of mechanical equipment. For example, after transforming a speed-varying signal into a signal with stable impact interval, Xiang et al.

[*] Corresponding author.
E-mail address: jlstrive2008@mail.xjtu.edu.cn (J. Chen).

https://doi.org/10.1016/j.isatra.2022.01.011
0019-0578/© 2022 ISA. Published by Elsevier Ltd. All rights reserved.

effectively recognized twelve different types of bearing faults using a stacking auto-encoder [11]. Ma et al. used a combination of residual network and order tracking to successfully detect the fault of planetary gearbox under small speed fluctuations [12]. In addition, order tracking and deep neural network were also used by Rao et al. [13] to realize the fault diagnosis of gearbox under constant and variable speed conditions. Despite the fact that these methods have achieved a lot of success, there are the following drawbacks:

(1) The use of pre-processing affects the efficiency of these procedures [14] and leads in the loss of some essential information [15]. Furthermore, the selection of these pre-processing measure is based on expert judgment and does not achieve end-to-end intelligent fault identification.

(2) The success of these IFD approaches is subjected to the assumption that sufficient training data is available. In practice, however, faulty data is scarce, which will cause poor network generalization [16].

To solve the problem of small training samples, Scholars have suggested some preprocessing approaches for data augmentation. These methods, such as rotating [17], flipping [18], noise addition, and filtering [19], have played a major role in image processing due to their ease of implementation and ability to successfully reduce overfitting. Inspired by their success in the field of image processing, they have aroused the interest of researchers in applying them to the field of one-dimensional data augmentation [20]. For instance, Li et al. utilized time shifting to supplement gear pitting data [21]. To expand training samples for underwater target detection, Liu et al. [22] introduced time shifting and Gaussian noise addition. By adding noise, flipping and other techniques, Wang et al. effectively expanded seismic vibration data for recognizing moving ground target [23]. These methods provide possibility for the fault diagnosis of the lack of data. Nonetheless, these measures only increase generalization capacity to a limited extent. Furthermore, unlike images, the integrity of signal could be compromised during the process, which makes it difficult to evaluate the generated samples [24].

As another scheme for IFD of small training samples, generative adversarial network (GAN) [25] receives the more attention since it may considerably increase the classification performance [26] and better understand data through modeling data augmentation [27]. GAN could be categorized into three types according on whether labels are necessary throughout the process: unsupervised, semi-supervised [28], and supervised. Conditional GAN [29] and auxiliary classifier generative adversarial network (ACGAN) [30] are the two most used forms of supervised GAN models. ACGAN is preferred by IFD of mechanical equipment because it promotes generalization via supervised sample expansion [31]. For example, Zhang et al. used ACGAN to produce vibration signals for different types of faults, then merged generated signals with real signals to train classification network, thereby achieving IFD of small samples at constant speed [32]. Wan et al. also combined ACGAN and Wasserstein distance for IFD of liquid rocket engine under little training data and high speed [33]. Although the above ACGAN-based models could achieve IFD with a small amount of fault training samples, they are mostly predicated on the assumption of constant speed. As a result, it is difficult to apply these models to the intelligent fault diagnosis of small samples at varying speeds [34].

According to the above literature review, in terms of IFD with small training samples and sharp speed variations, the following two topics should be investigated further: (1) data expansion: a generation model that could produce vibration signals with speed-varying features should be established to expand training samples, thereby providing support for the generalization of classification network under small samples; (2) feature extraction: a model that could successfully extract weak speed-varying characteristics from vibration signal need to be established to achieve end-to-end IFD under drastically varying speeds [6].

To address these issues, a multi-branch redundant adversarial net (RedundancyNet) is proposed, including generator, discriminator, and classifier. Firstly, through fusing multi-scale features, the generator via redundant reconstruction generates the signals at varying speeds to expand training samples. Then, the convolutional neural network via redundant decomposition extracts fault features of vibration signal under drastically variable speeds through layer-by-layer multi-resolution feature enhancement. Finally, the multi-branch network is introduced to separate fault features in order to realize fault location and evaluation of multiple components.

The main contributions are summarized as follows:

(1) We propose a well-designed redundant decomposition classification network for intelligent fault diagnosis at drastically variable speeds without the need for sophisticated signal processing methods. The redundant decomposition parts of the classification network promote the represent ability of weak features caused by sharp speed variations through layer-by-layer multi-resolution feature augmentation.

(2) For the first time, we introduce a generator that generates one-dimensional vibration signal with variable speeds via redundant reconstruction. The redundant reconstruction sections in the generator improve the capacity to generate vibration signal at varying speeds by deeply integrating multi-scale features. The produced signals supplement training samples and improve the generalization performance of classification network.

(3) We investigate the feature learning process and decision foundation of the redundant decomposition network by visualizing the output vectors of each layer, demonstrating that the proposed network can learn the fault diagnosis knowledge of vibration signal at varying speeds.

The rest is organized as follows. Related work is stated in Section 2. Section 3 describes the details of RedundancyNet. Two case analyses and some comparisons are performed to evaluate the effectiveness of the proposed method in Section 4. The process of feature learning and other aspects of RedundancyNet are discussed in Section 5. At last, some conclusions are drawn in Section 6.

2. Related work

This paper pays attention to IFD based on the vibration signal under small samples and drastically variable speeds. Thus, in this section, related prior research work on end-to-end IFD under variable speeds is first reviewed. Then, the related work of generative adversarial network is introduced.

2.1. End-to-end IFD under variable speeds

The term “end-to-end IFD” refers to the direct input of the original vibration signal without the need to convert it into other formats such TFG or equal interval signals in angular domain. The benefit is that, on the one hand, the diagnosis efficiency is improved. On the other hand, the generalization of IFD models could be promoted since directly using vibration signals as input can lessen the reliance on expert experience [34]. However, compared to IFD at constant speeds, the end-to-end IFD at variable speeds is more difficult. This is because failure-induced non-stationary phenomenon is similar to the non-stationary phenomenon brought by the speed changes [13], resulting in a fault characteristic that is exceedingly faint. To solve the challenges posed by non-stationary conditions, Chen et al. constructed an adaptive fusing network of vibration and speed signals [35]. Despite this, the network must input both the vibration signal and speed signal as input at the same time. In actually, due to a lack of installation space, the rotation speeds of many devices cannot be collected. Furthermore, this also results in an increase in cost [36]. To get around this problem, some scholars have advocated using solely vibration signals for IFD under variable speeds. For example, An et al. presented a long short-term memory network that could enhance fault characteristics in the vibration signal while ignoring operating condition to realize end-to-end IFD under time-varying speeds [37]. Wang et al. developed a cascade network to enhance fault characteristics through fusing multi-scale features, thereby realizing end-to-end IFD of motor bearing under non-stationary conditions [38]. These studies provide the solutions for IFD under variable speeds. However, they are designed for small speed variations while they cannot effectively extract the extremely weak fault features caused by sharp speed variations. In addition, their feature extraction process is unexplainable, making their conclusions unconvincing. Therefore, a redundant decomposition classification network is proposed in this paper, which may not only increase the represent ability of weak features caused by drastically variable speeds, but also explain how it makes decisions.

2.2. Generative adversarial network

Generative adversarial network (GAN), presented by Goodfellow and his colleagues, consists of generator and discriminator [25]. The optimization of GAN is a minimax game determines that GAN will conduct an adversarial process until Nash equilibrium. The optimization function is [25]:

$$\min_G \max_D O(D,G) = \mathrm{E}_{x\sim P_r(x)}\left(\log\left(D(x)\right)\right) + \mathrm{E}_{z\sim P_z(z)}\left(\log\left(1-D\left(G(z)\right)\right)\right) \tag{1}$$

Where G and D stand for generator and discriminator, respectively; E and P represent the expectation and the probability distribution; r and z are real data and noise. The above equation shows that the training process of GAN aims to reduce the gap between generated data and real data, but the discriminator aims to increase the distance between generated data and real data. Thus, the objection function is as follows:

$$\min_{\theta_D} O(D) = -P_r(x)\log\left(D(x)\right) - P_g(x)\log\left(1-D(x)\right) \tag{2}$$

Where θ_D is the parameter of the discriminator; g is generated data. Through taking the derivative of objection function with respect to $D(x)$ and setting the result to 0, the optimal discriminator $\hat{D}(x)$ could be obtained [10]:

$$\hat{D}(x) = \frac{P_r(x)}{P_g(x)+P_r(x)} \tag{3}$$

Then, by bring $\hat{D}(x)$ into the optimization function of G, the function may be turned into [32]:

$$O(G) = \mathrm{E}_{x\sim P_r(x)}\left(\log\frac{P_r(x)}{1/2\cdot\left(P_r(x)+P_g(x)\right)}\right) + \mathrm{E}_{x\sim P_g(x)}\left(\log\frac{P_g(x)}{1/2\cdot\left(P_r(x)+P_g(x)\right)}\right) - 2\log 2 \tag{4}$$

Where the sum of the first two terms actually represents Jensen–Shannon divergence (JS). As a result, it could be written as [10]:

$$O(G) = 2JS\left(P_r \parallel P_g\right) - 2\log 2 \tag{5}$$

This denotes that when the discriminator is optimal, the training goal of the generator is to minimize the JS between generated distribution P_g and real distribution P_r. However, if there is no overlap between the two distributions or the overlap could be ignored, JS would be constant, which leads to the disappearance of the training gradient. Furthermore, as indicated in [39], the gradient is unstable owning to unreasonable distance. To address the difficulties, Wasserstein distance is introduced as the objective function of Wasserstein generative adversarial network (WGAN) [40].

$$W = \frac{1}{K}\sup_{\|D\|_L\le K} E_{x\sim P_g}\left(D(x)\right) - E_{z\sim P_z}\left(D\left(G(z)\right)\right) \tag{6}$$

Where W is the objective function of WGAN; K is the constant of $K-$ Lipschitz, which requires the weight parameters of the discriminator cannot exceed K. Despite this, due to the weight clipping, the training of WGAN is still difficult to converge. Thus, Wasserstein GAN with gradient penalty (WGAN-GP) was presented [41], which replaces the weight clipping with gradient penalty.

$$L(D) = E_{\widetilde{x}\sim P_g}\left(D(\widetilde{x})\right) - E_{x\sim P_r}\left(D(x)\right) + \lambda E_{\hat{x}\sim P_{\hat{x}}}\left(\left(\left\|\nabla_{\hat{x}}D\left(\hat{x}\right)\right\|_2 - 1\right)^2\right) \tag{7}$$

Where λ stands for penalty factor. The last term represents the gradient penalty. Moreover, $\hat{x}$ is the sample obtained by random interpolation between real data and generated data, expressed as $\hat{x} = \beta\widetilde{x} + (1-\beta)x$. $\beta \sim \mathrm{U}(0,1)$, and U is uniform distribution. According to the optimization function, the objective functions of generator and discriminator are [41]:

$$\min_{\theta_g} L_g = -E_{z\sim P_z}\left(D\left(G\left(z,\theta_g\right)\right)\right) \tag{8}$$

$$\min_{\theta_d} L_d = E_{\widetilde{x}\sim P_g}\left(D\left(\widetilde{x},\theta_d\right)\right) - E_{x\sim P_r}\left(D\left(x,\theta_d\right)\right) + \lambda E_{\hat{x}\sim P_{\hat{x}}}\left(\left(\left\|\nabla_{\hat{x}}D\left(\hat{x},\theta_d\right)\right\|_2 - 1\right)^2\right) \tag{9}$$

Where L_g and L_d are the loss functions of the generator and the discriminator; θ_g and θ_d are the parameters of the generator and the discriminator, respectively. Since WGAN-GP possesses the above-mentioned excellent performance, it is utilized as the basis of the proposed method.

3. Proposed method

The multi-branch redundant adversarial net (RedundancyNet) is presented for end-to-end IFD under small training samples and drastically variable speeds. RedundancyNet is introduced in four parts in this section. Firstly, the framework of the proposed RedundancyNet is stated in detail. Then, the relationship between redundant reconstruction and time-domain lifting architecture of redundant second generation wavelet reconstruction (RSGWR) is explored. Next, the principle that the proposed classifier could realize IFD under drastically variable speeds is theoretically explained. Moreover, the time complexity of RedundancyNet is also briefly analyzed. Finally, the learning algorithm of RedundancyNet is given.

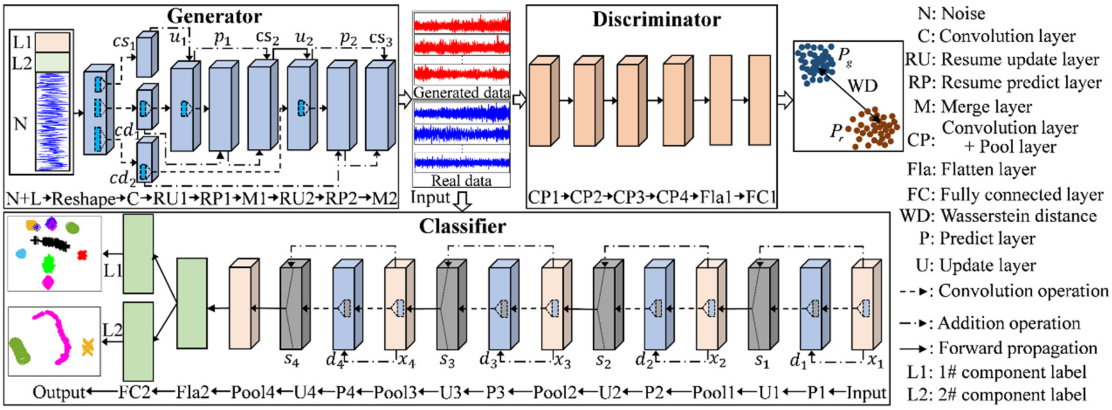

Fig. 1. Architecture of the proposed RedundancyNet.

3.1. Framework of RedundancyNet

The architecture of RedundancyNet is depicted in Fig. 1. The net contains generator, discriminator, and classifier. The generator and the classifier are both based on redundant second generation wavelet transform (RSGWT), so the net is named RedundancyNet. The purpose of generator is to increase the number of training samples. The job of discriminator is to close the gap between generated and real data, and the classifier is to undertake the task of classification. Their roles differ while they interact during training. For instance, the loss function of the generator and the loss function of the classifier simultaneously act on the training of generator, which could be view from Section 3.5.

The generator mainly includes two resume update layers (RU), two resume predict layers (RP), and two merge layers. The main learning process is a RU-RP-Merge loop. The input of generator is a combination of noise and labels, as illustrated below:

$$v = \left[z, y_{\mathrm{c}}^{(1)}, y_{\mathrm{c}}^{(2)}\right] \tag{10}$$

Where v represents the input of the generator; z is noise signal; c stands for the classifier; y is label; $y^{(1)}$ and $y^{(2)}$ represent the first and the second branch, respectively. In addition, it is noted that the input need to be reshaped to meet the requirement of convolutional operation.

The input of each loop is approximation signal and detail signal. Among them, approximation signal is the outcome of a large-scale convolution kernel, and detail signal is the result of a small-scale convolution kernel. The sizes of multi-scale convolutional kernels are shown in Table 1. Given approximation signal cs_k and detail signal cd_k as the input of kth loop, the output of RU is calculated as:

$$u_k = f\left(cs_k - cd_k * w_k^{\mathrm{ru}} + b_k^{\mathrm{ru}}\right) \tag{11}$$

Where k is kth loop; f is activation function of the generator and its type is Leaky ReLU function; cs_k and cd_k are the input of kth RU layer of the generator; ru stands for RU layer; w and b denote weight vector and bias vector, respectively; $*$ stands for convolutional operation. Then, given the output of RU and detail signal cd_k as the input of the following RP, the output of RP is given through the following equation:

$$p_k = f\left(cd_k + u_k * w_k^{\mathrm{rp}} + b_k^{\mathrm{rp}}\right) \tag{12}$$

Where rp denotes RP layer. Finally, the output of merge layer could be obtained through linear combination of u_k and p_k:

$$cs_{k+1} = \frac{u_k + p_k}{2} \tag{13}$$

Where cs_{k+1} is also the approximation input of $k+1$th loop in addition to being the output of kth loop.

Through multiple loops, the generated signals which are the output of last merge layer is obtained. As mentioned in the above process, the redundant reconstruction-based framework deeply integrates signals of different scales (approximation signal and detail signal), so that the generated signal contains rich fault features. Therefore, the generated signal could effectively expand the training samples under variable speeds, which is also why we propose the generator via RSGWR.

During training, the signals produced by the generator would be fed into both the discriminator and the classifier at the same time. The discriminator is to identify the distance between the generated sample and real sample. Compared with the classification task under drastically variable speeds, the task of the discriminator is straightforward, hence simply a one-dimensional convolutional neural network is used to enhance training efficiency. The discriminator consists of four convolutional layers, four pool layers, one flattened layer, and two full-connection layers. The output of each convolutional layer is calculated as follows:

$$O_j = \sigma\left(x_j^{\mathrm{d}} * w_j^{\mathrm{d}} + b_j^{\mathrm{d}}\right) \tag{14}$$

Where j is jth convolutional layer of the discriminator; O_j is the output of jth convolutional layer; x_j^{d} is the input of jth convolutional layer; w_j^{d} and b_j^{d} represent the weight vector and the bias vector of the discriminator; d stands for the discriminator; σ denotes nonlinear activation function, which is the ReLU function. A max pooling operation is performed after each convolutional layer to reduce output dimensionality, thereby reducing overfitting. After four convolution and pooling modules, the distance between generated sample and real sample is calculated using a flatten layer and two full-connection layers.

As the classifier, a one-dimensional convolutional neural network via redundant decomposition is proposed to remove interference information while enhancing fault features. The structure of the classifier is completely different with the discriminator, composed of four predict layers, four update layers, four pool layers, one flattened layer, and two fully connected layers. The main learning process of the classifier is a predict-update-pooling cycle.

Given x_i^{c} as the input of ith cycle of the classifier, the output of predict layer is calculated through the following equation:

$$d_i = \sigma\left(BN\left(x_i^{\mathrm{c}} - x_i^{\mathrm{c}} * w_i^{\mathrm{p}} - b_i^{\mathrm{p}}\right)\right) \tag{15}$$

Where i is ith cycle; σ denotes a ReLU-type nonlinear activation function; c represents the classifier; p denotes predict layer; BN stands for batch normalization, which is utilized to minimize condition-relevant information, thereby stabilizing the training process and speeding convergence [32]. The output d_i could be

Table 1
Main architecture of RedundancyNet.

Layers	Input size	Output size	Parameters
C	1-[400 × 1]	16-[400 × 1]	Conv 32×1, 16×1, 8×1; Stride 1
RU1	16-[400 × 1]	16-[400 × 1]	Conv 16 × 1; Stride 1
RP1	16-[400 × 1]	16-[400 × 1]	Conv 16 × 1; Stride 1
RU2	16-[400 × 1]	16-[400 × 1]	Conv 8 × 1; Stride 1
RP2	16-[400 × 1]	16-[400 × 1]	Conv 8 × 1; Stride 1
CP1	1-[6400 × 1]	8-[1600 × 1]	Conv 16 × 1; Pool 2 × 1; Stride 2
CP2	8-[1600 × 1]	16-[400 × 1]	Conv 16 × 1; Pool 2 × 1; Stride 2
CP3	16-[400 × 1]	32-[100 × 1]	Conv 16 × 1; Pool 2 × 1; Stride 2
CP4	32-[100 × 1]	32-[25 × 1]	Conv 16 × 1; Pool 2 × 1; Stride 2
FC1	800	32	/
P1	1-[6400 × 1]	32-[6400 × 1]	Conv 8 × 1; Stride 1
U1	32-[6400 × 1]	32-[6400 × 1]	Conv 8 × 1; Stride 1
P2	32-[3200 × 1]	32-[3200 × 1]	Conv 16 × 1; Stride 1
U2	32-[3200 × 1]	32-[3200 × 1]	Conv 16 × 1; Stride 1
P3	32-[1600 × 1]	32-[1600 × 1]	Conv 32 × 1; Stride 1
U3	32-[1600 × 1]	32-[1600 × 1]	Conv 32 × 1; Stride 1
P4	32-[800 × 1]	32-[800 × 1]	Conv 32 × 1; Stride 1
U4	32-[800 × 1]	32-[800 × 1]	Conv 32 × 1; Stride 1
FC2	25600	32	/

understood as the deviation between the input x_i^c and the feature component $(x_i^c * w_i^p + b_i^p)$ extracted by the convolution process. As a result, d_i is also named as detail part, which contains more high-frequency noise. Then, given d_i and x_i^c as inputs to the following update layer, the output is given as:

$$s_i = \sigma\left(BN\left(x_i^c + d_i * w_i^u + b_i^u\right)\right) \tag{16}$$

Where u denotes update layer. The above Eq. (16) indicates that the result of the convolutional operation between the detail signal d_i and the filter of update layer is used to update the input signal x_i^c, thereby enhancing the fault characteristics in the output of update layer. Therefore, the output s_i contains more fault features and is referred to as approximation part.

Next, the output s_i of the update layer are passed into pooling layer for dimensionality reduction. And the output of pooling layer is the input of next cycle. In each cycle, the predict layer gets high-frequency detail signal, while the update layer obtains the low-frequency approximation signal, indicating that the learning process is a multi-resolution process. Of course, the output of each cycle is the result of dimensionality reduction of the approximation signal, which keeps the components holding fault characteristics while removing the information that is unnecessary to the problem (e.g. operating condition, noise and other interference). Through the above process, weak fault features caused by varying speeds could be fully extracted. This is why we suggest a redundant reconstruction architecture to achieve IFD under drastically variable speeds.

After numerous cycles, a multi-branch full-connection network is employed for multi-component fault diagnosis. Each branch has the same structure, which is composed of two fully connected layers. Constructing a multi-branch network serves two purposes: (1) the multi-branch network is able to separate multi-component fault features, thereby realizing fault location and fault evaluation. (2) compared to single-branch network, multi-branch network could improve generalization performance. This is since the multi-branch network can be view as a multi-task framework which shares the same input but different goals. Through parameter sharing of different tasks, the number of training samples for shared parameters is much larger than for single tasks, bringing better generalization performance for the network. Specific case descriptions could be found in Section 5. The detailed parameters of the proposed RedundancyNet are displayed in Table 1.

3.2. Relationship between redundant reconstruction and RSGWR

Compared to the vibration signals at steady speed, the notable feature of the signal at variable speeds is the time-varying fault characteristics. This necessitates not just the presence of fault information, but also the presence of speed-varying properties in the generated signal. At the same time, both should be fused together. Considering the benefits of RSGWR in signal fusion at various scales, it is possible to gain superior results when the reconstruction notion of RSGWR is applied to data generation. Based on this consideration, a generator via redundant reconstruction is presented for sample expansion at variable speeds. Of course, the relationship between redundant reconstruction and RSGWR should be explored to understand the advantages of redundant reconstruction framework.

First, RSGWR is briefly introduced. The reconstruction stage of RSGWG is composed of restoration update step and restoration predict step. The restoration update step refers to the use of approximation coefficient $cs_n^{(l-1)}$ and detail coefficient $cd_n^{(l-1)}$ to restore the sequence $u_n^{(l)}$ [42]:

$$u_n^{(l)} = cs_n^{(l-1)} - \sum_r wu_r^{(l)} cd_{n+r}^{(l-1)} \tag{17}$$

Where l represents lth layer of RSGWR; $u_n^{(l)}$ is the result obtained by restoration update step; $wu_r^{(l)}$ is the filter coefficient of lth restoration update step. Next, the restoration predict step uses the result of restoration update step $u_n^{(l)}$ and detail coefficient $cd_n^{(l-1)}$ to restore the sequence $p_n^{(l)}$ [42]:

$$p_n^{(l)} = cd_n^{(l-1)} + \sum_j wp_j^{(l)} u_{n+j}^{(l)} \tag{18}$$

Where $p_n^{(l)}$ is the result obtained by restoration predict step; $wp_j^{(l)}$ is the filter coefficient of lth restoration predict step. Finally, the reconstructed signal could be obtained by averaging the above two results [42]:

$$cs_n^{(l)} = \frac{u_n^{(l)} + p_n^{(l)}}{2} \tag{19}$$

Where $cs_n^{(l)}$ is new approximation coefficient.

It is noted that the last terms of Eqs. (17) and (18) are actually convolution operations. Therefore, it is easy to find the similarity between (11)–(13) and (17)–(19). Of course, there are some differences between them. For example, the filters of RSGWR must be pre-matched to the properties of the signal, whereas the kernels of the generator based on redundant reconstruction are learned from vibration signals. Therefore, the convolutional network based on redundant reconstruction is easier to adaptively acquire the abstract representation of the signal. In addition, Eqs. (11)–(12) also include a nonlinear activation function, which further improves the learning ability of the network in terms of nonlinear features. As a result, the learning mechanism of the generator could be summarized. Through deeply fusing multi-scale features (low-frequency features and high-frequency features), the redundant reconstruction-based framework of the generator ensures the integrity of fault information. At the same time, the generated signal is obtained on the basis of noise, thus again ensuring the diversity of the signal.

3.3. Interpretation of feature extraction with redundant decomposition

When the vibration signal is collected at a constant speed, the fault features vary in a linear manner. These features are simple to learn and could ensure a high level of generalization. When the signal is collected at a non-stationary speed, however, the failure-induced non-stationary phenomenon is similar to the non-stationary phenomenon brought by the speed changes, yielding very weak fault characteristics. The following two issues arise as a result of the poor fault characteristics: (1) It is difficult to learn the parameters of deep network due to weak fault features; (2) Choosing a large model to reduce experience errors would result in large generalization errors.

To solve the above issues, it is essential to design a network framework that could effectively enhance fault characteristics and eliminate interference components under non-stationary conditions. Inspired by the ability of redundant second generation wavelet decomposition (RSGWD), the classifier via redundant decomposition is firstly proposed. In order to explore the learning process of the classifier based on redundant decomposition, a brief description of RSGWD is given. The stage of RSGWD consists of predict step and update step. In the predict step, its output is the deviation between the input signal $s_n^{(l)}$ and its prediction sequence [42]:

$$d_n^{(l+1)} = s_n^{(l)} - \sum_j p_j^{(l)} s_{n+j}^{(l)} \tag{20}$$

Where l is lth layer of RSGWD; d_n represents the output of predict step, also called detail signal; p is the filter of predict step of RSGWD. Next, based on the detail signal $d_n^{(l+1)}$, update the input signal $s_n^{(l)}$ with the filter of update step [42]:

$$s_n^{(l+1)} = s_n^{(l)} + \sum_k u_k^{(l)} d_{n+k}^{(l+1)} \tag{21}$$

Where $s_n^{(l+1)}$ is the output of update step, also named approximation signal; u is the filter of update step of RSGWD. Through the predict step and the update step, RSGWD effectively removes the noise-containing detail signal, and better retains the approximation signal containing time-domain features. Of course, the last terms of (20) and (21) are actually a convolutional operation in essence as well.

Through the above description, the similarity between (15)–(16) and (20)–(21) could be easily found. As a result, their output is also similar. The output of update layer is high-frequency detail component, while the output of predict layer is low-frequency approximation component. It is a process of multi-resolution feature extraction. Through multiple predict-update cycles, low-frequency fault characteristics are fully enhanced, and the high-frequency interference information is completely removed. With obvious extracted fault characteristics, the deep learning model could easily find the classification boundaries for different fault types, thus reducing the generalization error. A detailed explanation of the learning mechanism of the classifier via redundant decomposition is given in Section 4. Of course, there are some differences between RSGWD and the proposed classifier network as well, such as the activation function. ReLU-type activation function is utilized in the paper. In fact, the activation function could be regarded as a denoising technique. This denoising technique would remove all the components below the threshold 0, while retaining the components above the threshold. Therefore, the use of the nonlinear activation function further improves the feature extraction capability of the network via redundant decomposition.

3.4. Time complexity

The complexity of a deep model is also an important indicator for assessing its performance. Therefore, the following will focus on the complexity of the proposed model. The complexity of a deep model is divided into two categories: time complexity and spatial complexity. Time complexity is a qualitative description of the running time of the model, and the spatial complexity is a measure of the temporary storage space of the model during the operation. Since researchers pay more attention to the efficiency of the model in real-work applications in industrial scenario [43], time complexity is discussed in the paper. The time complexity of the proposed RedundancyNet is composed of the generator via redundant reconstruction, the classifier via redundant decomposition, and the discriminator.

For the generator, its time complexity includes multiple RU-RP-Merge loops. The time complexity of each RU layer or RP layer is the same as that of convolutional operation. As a result, the time complexity of the generator is equal to that of multiple convolutional layers, shown as follows [43]:

$$G_{comp} \sim O\left(\sum_{i=1}^{I} M_i \cdot K_i \cdot D_i^{\mathrm{In}} \cdot D_i^{\mathrm{Out}}\right) \tag{22}$$

Where G_{comp} stands for the time complexity of the generator; i represents ith RU layer or RP layer; I is the total number of RU layers and RP layers; M is the size of output vector of each layer; K is the size of convolutional kernel; D^{In} is the number of input channels; D^{Out} is the number of output channels. The discriminator is a simple convolutional neural network, so its time complexity is given through the following [44]:

$$D_{comp} \sim O\left(k_1 \sum_{j=1}^{J} M_j \cdot K_j \cdot D_j^{\mathrm{In}} \cdot D_j^{\mathrm{Out}}\right) \tag{23}$$

Where D_{comp} is the time complexity of the discriminator; k_1 is a factor in the range of 1.05–1.1 [39], representing the time complexity caused by pooling layer and full-connection layer; j is jth convolutional layer; J is the number of convolutional layers.

For the classifier via redundant decomposition, the time complexity mainly comes from predict layers, update layers, pooling layers, full-connection layers. Compared to convolutional layer, the proposed update layer or predict layer only increases the parameters of batch normalization. Therefore, the time complexity of the classifier is calculated as [44]:

$$C_{comp} \sim O\left(k_2 \sum_{l=1}^{L} M_l \cdot K_l \cdot D_l^{\mathrm{In}} \cdot D_l^{\mathrm{Out}}\right) \tag{24}$$

Where C_{comp} is the time complexity of the classifier; k_2 is the factor which is caused by pooling layer, full-connection layer, and batch normalization; l is lth predict layer or update layer; L is the total number of predict layers and update layers. Of course, compared to convolutional operation, the time complexity of batch normalization is substantially low.

As a result, the convolutional operation still accounts for the majority of the time complexity of RedundancyNet. In addition, WGAN-GP-based intelligent fault diagnosis methods are generally composed of generator, discriminator and classifier, all of which are convolutional neural networks. This also suggests that the proposed RedundancyNet could obtain superior classification performance with a minor increase in time complexity. A detailed analysis of the calculation time is shown in Section 4.

3.5. Model training

The proposed RedundancyNet consists of generator, discriminator, and classifier. The training of RedundancyNet is an adversarial process between nominal generator (G) and nominal discriminator (D). G is also the generator of RedundancyNet. D is composed of the discriminator and the classifier. The specific flowchart of RedundancyNet is depicted in Fig. 2. The algorithm is shown in Table 2. The objective function contains two parts: the

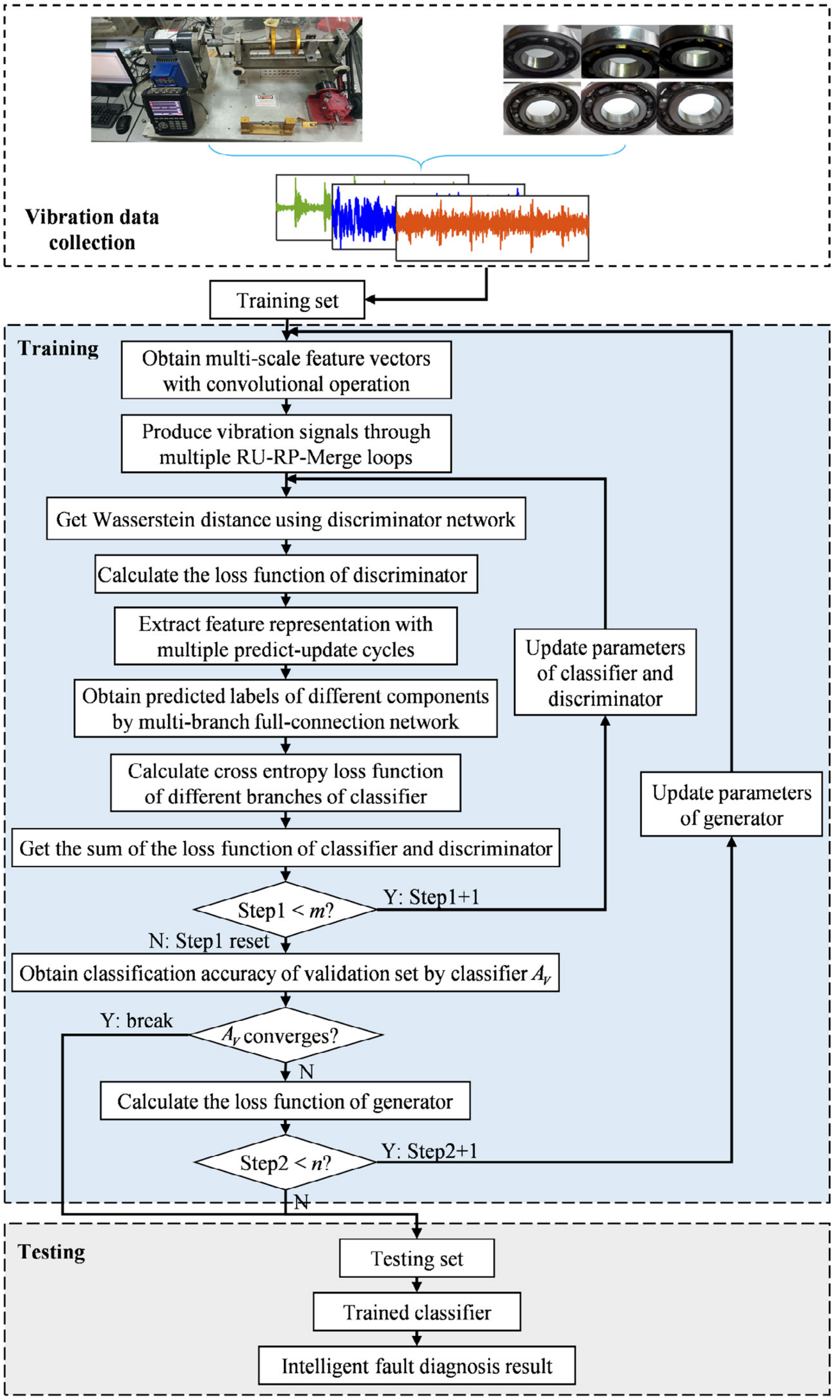
Vibration data collection
Training set
Training
Obtain multi-scale feature vectors with convolutional operation
Produce vibration signals through multiple RU-RP-Merge loops
Get Wasserstein distance using discriminator network
Calculate the loss function of discriminator
Extract feature representation with multiple predict-update cycles
Obtain predicted labels of different components by multi-branch full-connection network
Calculate cross entropy loss function of different branches of classifier
Get the sum of the loss function of classifier and discriminator
Update parameters of classifier and discriminator
Update parameters of generator
Step1 < m?
Y: Step1+1
N: Step1 reset
Obtain classification accuracy of validation set by classifier A_v
A_v converges?
Y: break
N
Calculate the loss function of generator
Step2 < n?
Y: Step2+1
N
Testing
Testing set
Trained classifier
Intelligent fault diagnosis result

Fig. 2. The specific flowchart of RedundancyNet.

loss of WGAN-GP and the cross entropy loss function of multi-branch classifier. Among them, the cross entropy loss function not only needs to train the parameters of the classifier, but also helps the training of the generator. Therefore, the loss function of G and D are described below.

$$L_G = L_g + L_{C_f^{(1)}} + L_{C_f^{(2)}} \tag{25}$$

$$L_D = L_d + L_{C_r^{(1)}} + L_{C_f^{(1)}} + L_{C_r^{(2)}} + L_{C_f^{(2)}} \tag{26}$$

Where L represents loss function; g and d are the generator and the discriminator of WGAN-GP, respectively; C denotes the classifier; f and r are generated data and real data; (1) and (2) stand for the first branch and the second branch of the classifier.

Training parameters are voluntarily updated through back propagation during the alternative optimization of G and D.

$$\theta_G = \theta_G - \eta_1 \cdot (\partial L_G / \partial \theta_G) \tag{27}$$

$$\theta_D = \theta_D - \eta_2 \cdot (\partial L_D / \partial \theta_D) \tag{28}$$

Where θ stands for training parameters; η_1 represents the learning rate of G; η_2 denotes the learning rate of D.

As a result, the training parameters of different parts are automatically updated through the following equations.

$$\theta_g = \theta_g - \eta_1 \left(\partial \left(L_g + L_{C_f^{(1)}} + L_{C_f^{(2)}} \right) / \partial \theta_g \right) \tag{29}$$

Table 2
The algorithm of RedundancyNet.

```
Algorithm: The training process of multi-branch redundant adversarial network ( RedundancyNet)
Require: Gradient penalty factor $\lambda = 10$; Learning rate of G 0.001; Learning rate of D 0.002; Batch size $N = 80$; Number of iteration steps $n = 3000$; The number of discrimination iterations $m = 5$ for per generation iteration.
Initialize: Parameters of generator, discriminator, and classifier, is initialized by a zero-mean standard normal distribution; Noise is initialized by uniform distribution.
Training: Alternative optimization of nominal discriminator and nominal generator.
for $i = 1, 2, \cdots, n$ do
    Sample a batch of noise samples $\{z_1, z_2, \cdots, z_N\}$ from noise data
    Sample a batch of labeled samples $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ from training set
    for $j = 1, 2, \cdots, m$ do
        Calculate the loss function of nominal discriminator by Equation (26)
            $L_D = L_d + L_{C_r^{(1)}} + L_{C_f^{(1)}} + L_{C_r^{(2)}} + L_{C_f^{(2)}}$
        Update the parameters of discriminator with Equation (30)
        Update the parameters of predict-update-pooling part of classifier by Equation (31)
            $\theta_{\hat{C}} = \theta_{\hat{C}} - \eta_2 \cdot \left( \partial \left( L_{C_r^{(1)}} + L_{C_r^{(2)}} + L_{C_f^{(1)}} + L_{C_f^{(2)}} \right) \middle/ \partial \theta_{\hat{C}} \right)$
        Update the parameters of different branches of classifier by Equations (32) and (33)
    end for
    Calculate the loss function of nominal generator by Equation (25)
        $L_G = L_g + L_{C_f^{(1)}} + L_{C_f^{(2)}}$
    Update the parameters of generator through Equation (29)
        $\theta_g = \theta_g - \eta_1 \left( \partial \left( L_g + L_{C_f^{(1)}} + L_{C_f^{(2)}} \right) \middle/ \partial \theta_g \right)$
    Calculate the classification accuracy of validation set; if converges: break; else: continue
end for
Testing: Fixed all of the parameters, and then classify testing set
```

$$\theta_d = \theta_d - \eta_2 \cdot \left(\partial L_d \middle/ \partial \theta_d \right) \tag{30}$$

$$\theta_{\hat{C}} = \theta_{\hat{C}} - \eta_2 \cdot \left(\partial \left(L_{C_r^{(1)}} + L_{C_r^{(2)}} + L_{C_f^{(1)}} + L_{C_f^{(2)}} \right) \middle/ \partial \theta_{\hat{C}} \right) \tag{31}$$

$$\theta_{\bar{C}^{(1)}} = \theta_{\bar{C}^{(1)}} - \eta_2 \cdot \left(\partial \left(L_{C_r^{(1)}} + L_{C_f^{(1)}} \right) \middle/ \partial \theta_{\bar{C}^{(1)}} \right) \tag{32}$$

$$\theta_{\bar{C}^{(2)}} = \theta_{\bar{C}^{(2)}} - \eta_2 \cdot \left(\partial \left(L_{C_r^{(2)}} + L_{C_f^{(2)}} \right) \middle/ \partial \theta_{\bar{C}^{(2)}} \right) \tag{33}$$

Where $\hat{C}$ is the part of multiple cycles of the classifier, that is, the framework shared by different tasks; $\overline{C}$ are the full-connection parts of the classifier.

All hyperparameters of the proposed RedundancyNet are shown in Table 2. Among them, the gradient penalty factor and the number of discrimination iterations for per generation iteration is the default values in [41]. The other parameters are the optimal results selected according to the performance on validation set.

4. Case analysis

In this section, two cases whose rotating speeds are variable are analyzed to validate RedundancyNet. Some comparisons are also introduced to highlight the advantages of the net. All case studies are performed on a computer with i5-10400F CPU of 2.90 GHz processor parameters and GTX1650 GPU. Moreover, the RAM of the computer is 16.0 GB. All methods including RedundancyNet and comparison methods are constructed via Python 3.6 and the deep learning framework Tensorflow 1.8.

4.1. Methods for comparison

Since RedundancyNet is designed for end-to-end intelligent fault diagnosis with drastically variable speeds and small training samples, some state-of-the-art methods for fault diagnosis under varying speeds or small samples are used to verify its effectiveness. It should be mentioned that the parameters of these methods are all the optimal parameters selected according to validation set. The following are the details of these comparative methods:

One-dimensional vision ConvNet (VCN) was proposed by Wang et al. [45] for end-to-end fault diagnosis of bearing under complex operation conditions. VCN uses a multi-kernel framework to extract fault features. In VCN, the small convolution

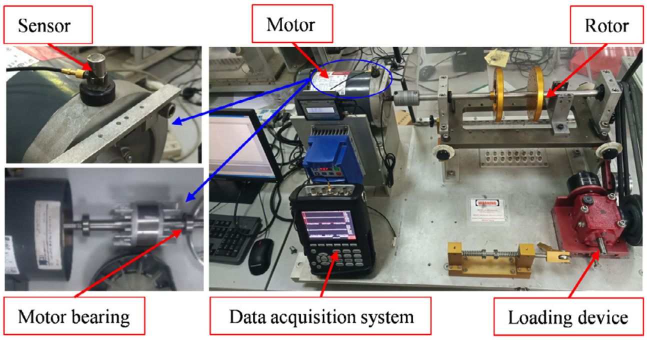

Fig. 3. Spectral Quest machinery fault simulation test bed.

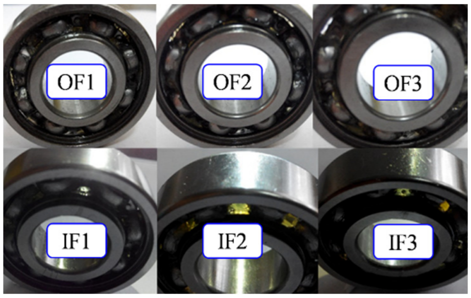

Fig. 4. Figure of manufactured failure bearings.

kernel is set as 3 × 1 and 5 × 1 in turn. The large convolutional kernel is 71×1, 74×1, 102×1, and 122 × 1, respectively. Furthermore, the number of channels has been adjusted to 32.

Liu et al. [6] presented a multi-scale residual network (MK-ResCNN) for motor bearing fault diagnosis under varying speeds, which uses multi-scale kernels to extract fault features. The sizes of different scale kernels are 8×1, 16×1, and 32 × 1.

Shao et al. [27] introduced an ACGAN-based method for surprised sample augmentation, which realizes fault diagnosis under small samples through sample generation. The size of convolutional kernel is 32 × 1. In addition, the dropout with a probability of 0.5 is substituted with one with a probability of 0.1 in the paper, because its classification performance is better. The remaining parameters are the same as those in [27].

The method via WGAN-GP [46] was proposed by Guo et al. for intelligent fault diagnosis of low-data input. The size of the convolutional kernels of generator and classifier is set at 32 × 1. The size of convolutional kernels of discriminator is 16 × 1. Batch size is chosen as 80. The other parameters are the same as [46].

Furthermore, a multi-task adversarial network (MRLAN) [10] is also used to compare with RedundancyNet owing that it was proposed for IFD under varying speeds and small samples. MRLAN uses two-dimensional convolutional neural network to extract fault features after transforming vibration signal into TFG. All parameters are the same as those in [10].

4.2. Case 1: SQ bearing experiment with varying speeds

To assess the effectiveness of RedundancyNet under small samples and varying speeds, an experiment is performed, which is the motor bearing experiment of Spectral Quest (SQ) machinery fault simulation bed. The bed is composed of a 1-hp motor, NSK6203 motor bearing, acceleration sensor, rotor, a loading device of 5 kg, and data acquisition system, displayed in Fig. 3. The measuring range and sensitivity of the sensor are ±100 g and 50 mV/g. The test bed accelerates from 0 to maximum speed of 2900r/min with 3s. In addition, the deceleration from 2900r/min to 0 also takes 3s. The above-mentioned process is repeated numerous times to complete data collection.

Six single-point failure bearings are machined besides of normal bearing (N). They belong to outer race minor failure (OF1), outer race medium failure (OF2), outer race severe failure (OF3), inner race minor failure (IF1), inner race medium failure (IF2), and inner race severe failure (IF3), shown in Fig. 4. The fault characteristic orders of outer race and inner race are 3.085 and 4.9147, respectively.

The vibration data are collected at a sampling frequency of 12.8 kHz. Sampling length is 0.5s for each sample. All signals are recorded under variable speeds. Besides, a validation set is developed to prevent overfitting by stopping the training early. The validation set has 20 samples for each category, whereas the test set contains 50 examples for each category. According to the amount of training samples currently utilized in the literature of IFD under small samples [10,32], a dataset with no more than 30 fault samples is considered a small sample set. Therefore, different numbers of samples (15, 20, and 30) for each fault of training set are selected to verify the effectiveness under small samples. And 50 normal samples are also chosen due to sufficient normal data.

For the sake of simplicity, only rotation speeds and vibration waveform of training set are given in Fig. 5. Since the samples are varying-speed, it is tough to diagnose fault through observing or using a traditional neural network.

RedundancyNet is firstly trained with 30 samples for each fault. After 1657 iterations, the model converged and test accuracy reached 97.14%. The classification results are shown in Fig. 6. From the scatterplot of the predicted labels in testing set, it could be viewed that distinct classes are essentially separated although there are occasional misclassifications or omissions. In particular, normal samples and faulty samples are thoroughly separated, which is important for avoiding serious safety accidents. Furthermore, the visualization results reveal that there is a significant disparity between normal type and various degrees of outer race failure. Of course, inner race faults of different degrees are relatively close, since that inner-race faults are more difficult to diagnose than others. In general, the cluster centers of different types are clear, which shows the excellent performance of RedundancyNet.

RedundancyNet is also trained on 15 and 20 samples. Some comparisons are also performed. The results are shown in Table 3, which indicates the amount of training samples has an impact on the classification outcomes. This is since the IFD methods under small training samples still need some data to ensure that real distribution can be learned. It is noted that the accuracy of MK-ResCNN achieves 93.43% when 30 faulty training samples are selected. The reason is that it is designed for fault diagnosis under varying speeds. However, since it targets at a large number of training samples, the accuracy does not exceed 95%. The accuracy of fault diagnosis methods for small samples is generally higher than other methods. Regardless of the quantity of training samples, RedundancyNet achieves the highest classification accuracy.

In addition, as shown in Table 4, the computation time of various methods on test set has been compared. MK-ResCNN takes the longest time since it performs multi-scale analysis to extract features so as to increase the complexity of the model. Due to multiple update-predict cycles, the classifier of RedundancyNet increases the amount of parameters, hence its test time is 0.35s longer than the WGAN-GP-based technique in [46]. However, its accuracy has increased by at least 2%. Moreover, MRLAN is also used to compare with RedundancyNet. Table 5 shows the test results after 30 samples of each class were used to train

Table 3
Test accuracy of different methods.

Case	Number of training samples	VCN	MK-ResCNN	[27]	[46]	RedundancyNet
Case 1	15	70.29%	80.00%	81.14%	83.43%	86.29%
	20	74.29%	82.86%	86.86%	91.14%	94.57%
	30	84.86%	93.43%	94.86%	95.14%	97.14%
Case 2	15	67.42%	63.64%	76.36%	83.18%	86.51%
	20	67.88%	74.39%	79.24%	88.03%	91.82%
	30	81.52%	87.73%	89.55%	93.79%	98.33%

Table 4
Test time of different methods.

Case	VCN	MK-ResCNN	[27]	[46]	RedundancyNet
Case 1	1.33 s	2.03 s	1.35 s	1.38 s	1.73 s
Case 2	1.47 s	2.80 s	1.71 s	1.71 s	2.05 s

Table 5
Test results of MRLAN after training with 30 samples.

Case	Test accuracy	Time		Total test time
		STFT	Classification network	
Case 1	99.43%	0.62 s	1.69 s	2.31 s
Case 2	98.93%	1.19 s	1.82 s	3.01 s

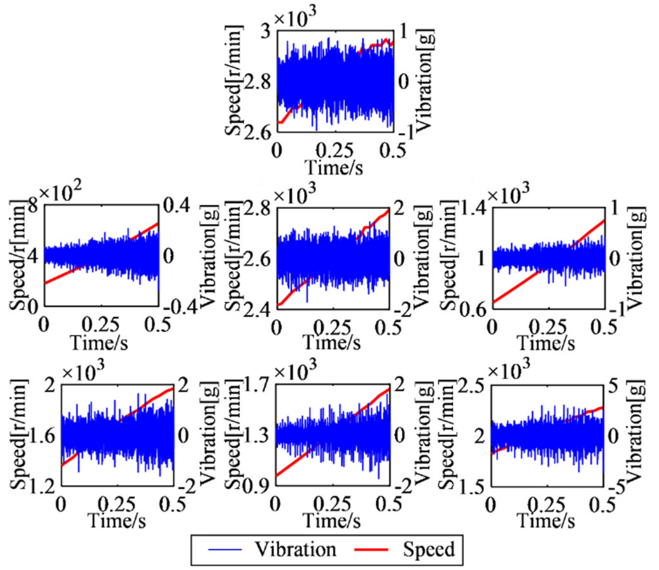

Fig. 5. Vibration waveform and rotation speed of training data from SQ test bed. From left to right are different levels (minor, medium, and severe). From top to down are different categories (normal, outer-race, and inner-race).

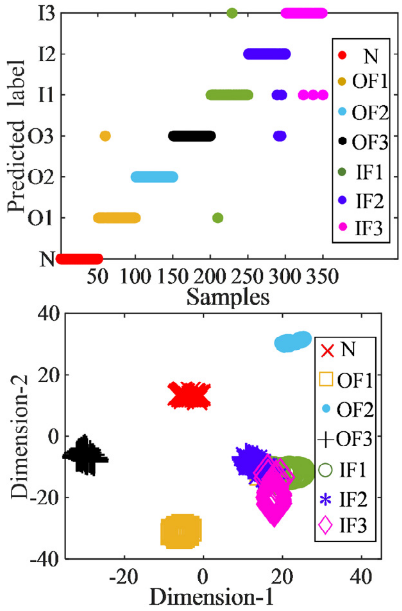

Fig. 6. Classification results of RedundancyNet in SQ. From top to down are scatterplot of classification labels and feature visualization by t-SNE.

it. It can be viewed that the test accuracy of MRLAN reaches 99.43%, which is attributable to the usage of STFT to enhance the fault characteristics of the input. While increasing the test accuracy, however, MRLAN relies on expert experience to choose the appropriate time–frequency analysis method. At the same time, its computing time is greatly increased. On the basis of an increase in accuracy of 2.29%, the test time has increased by 0.58s. As a consequence, the above analysis results suggest that the proposed method can improve the classification accuracy while not considerably reducing computation efficiency. This also demonstrates that RedundancyNet may be used for end-to-end IFD with small samples and variable speeds.

4.3. Case 2: pulley block experiment at drastically variable speeds

All of the data from SQ are collected at slowly variable speeds. A pulley block test rig is introduced to validate the effectiveness of RedundancyNet under drastically variable speeds and multi-component failure, displayed in Fig. 7. The rig consists of 1.5 kW servo motor, CoCo-80 data acquisition system, KISTLER accelerometer, and pulley block. Two springs are used as loading devices with a maximum load of 6.5 kg. On the pulley block, there are two test bearings: 1# test bearing and 2# test bearing. Despite the fact that both bearings work at varying speeds, the rotation speed of 2# bearing is 3/4 of that of 1# bearing. Only when one accelerometer is installed on the side plate of pulley block is vibration data from two test bearings recorded at a sampling frequency of 12.8 kHz. The whole working process of the experimental bench is that the motor drives the nut on the ball screw to move within a range of 1 m. It first accelerates from 0 to the highest speed in 0.1s. After working for a while, it slows down from the highest speed to 0 within 0.9s. Regardless of the maximum speed, the deceleration and acceleration time are fixed in addition to the movement distance of the nut. As a result, the data collected each time is extremely limited. For the reason, the data collecting has gone through multiple independent processes. There are four maximum speeds for each category: 1200r/min, 1500r/min, 2000r/min, and 2500r/min. For each speed range, 40-time independent data captures are made.

Eight 6202 bearings are used, whose health conditions are normal (N), outer race minor fault (OF1), outer race medium fault (OF2), outer race severe fault (OF3), inner race minor fault (IF1), inner race medium fault (IF2), inner race severe fault (IF3), and roller fault (RF). Conveniently, only four of them are illustrated in

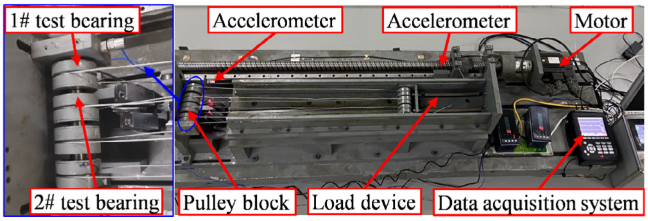

Fig. 7. Pulley block fault simulation test rig.

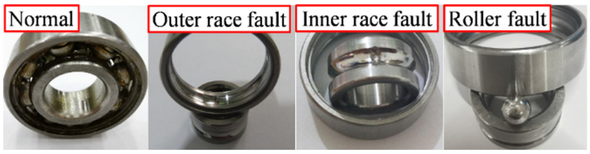

Fig. 8. Graph of test bearings of pulley block.

Table 6
The combination of different test bearings.

2#	1#							
	N	OF1	OF2	OF3	IF1	IF2	IF3	RF
N	N-N	O1-N	O2-N	O3-N	I1-N	I2-N	I3-N	R-N
IF3	×	×	×	O3-I3	×	×	×	R-I3
R	×	×	×	O3-R	×	×	×	×

Fig. 8. A total of eleven categories are produced, when any two of them are installed on 1# and 2# test points. Table 6 shows the detailed combination.

The case also includes training set, validation set, and test set. Those samples are cut from the collected data. The length of all samples is 0.5s. Compared with Case 1, the speed variation in Case 2 is more drastic, shown in Fig. 9. Test set contains 60 samples for each category, and validation set has 20 samples. The 50 normal samples are chosen for training set. And 15, 20, and 30 samples for each faulty category are selected as a part of training set, respectively.

Different from single-branch network, the test accuracy of multi-branch network is defined as the result of logical AND operation of the output of different branches. When 30 faulty training samples are used to train the net, the test accuracy achieves 98.33% after 2473 iterations. The accuracy of first branch is 98.33%, whereas the accuracy of second branch is 100%. The dimensionality of last layer of the classifier is reduced with t-SNE, shown in Fig. 10. The distances between distinct classes are relatively long, despite occasional misclassifications, as seen in the visualization result of the first branch. The core of each cluster is, of course, obvious. As a result, the various degrees and types of failures of 1# bearing are essentially separated. Furthermore, the visualization results of the second branch show that different faults of 2# bearing may be clearly recognized. There are no misclassifications or omissions. The above analysis indicates that the net could diagnose fault under the condition of not only drastically variable speeds but also multi-component failures.

Then, some other analyses and comparisons are implemented, the results of which are displayed in Table 3. As seen in the table, the classification accuracy of RedundancyNet in Case 2 is virtually identical to that in Case 1 for the same training samples, if not slightly better. Other comparison methods in Case 2, however, are far less accurate than Case 1. This demonstrates that sharp speed variations result in highly weak fault characteristics and significant generalization errors. At the same time, regardless of the amount of training samples used, the test accuracy of the suggested method is significantly higher than existing techniques.

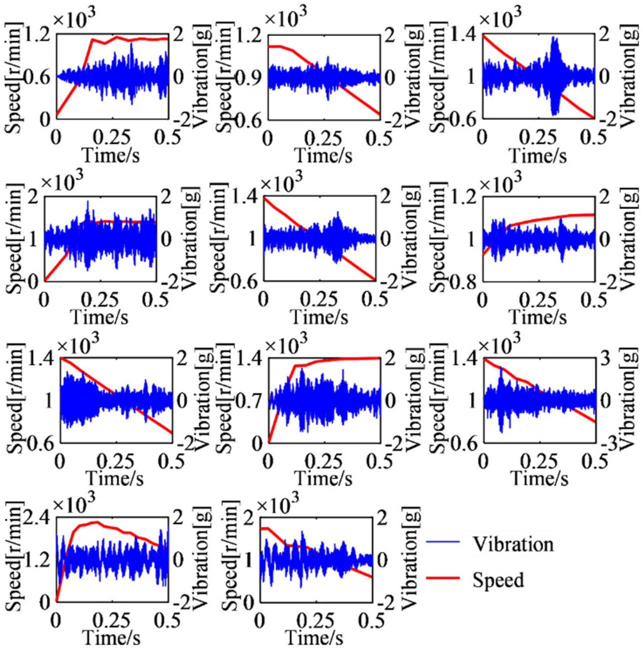

Fig. 9. Vibration waveform and rotation speed of training samples of Case 2. From left to right and from top to down are N-N, O1-N, O2-N, O3-N, I1-N, I2-N, I3-N, R-N, O3-I3, O3-R, and R-I3, respectively.

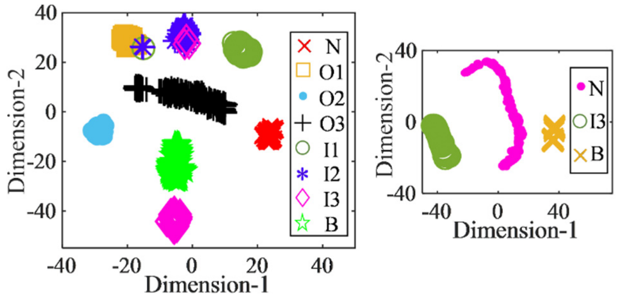

Fig. 10. Visualization of features of RedundancyNet in pulley block test rig. From left to right are the results of the first branch and the second branch, respectively.

Of course, as demonstrated in Table 4, the computing time is longer than WGAN-GP-based method in [46] and ACGAN-based method in [27] due to more parameters. Despite the fact that RedundancyNet takes 0.34s longer than WGAN-GP-based method, its test accuracy has improved by 2.33%, 3.79%, and 4.54% respectively under various training samples. Moreover, the proposed approach is compared to MRLAN. The test accuracy of both is similar. MRLAN is only 0.6% greater than RedundancyNet. However, the computing time is increased by 0.96s due to the need for STFT-based preprocessing. These analysis results imply the superiority of RedundancyNet under drastically variable speeds and small faulty training samples, besides of the condition of multi-component failures.

5. Discussion

5.1. Ablation analysis

The main components of the proposed method include a generator based on redundant reconstruction and a classifier based on redundant decomposition. To investigate the function of different parts, each one is ablated separately and then tested on

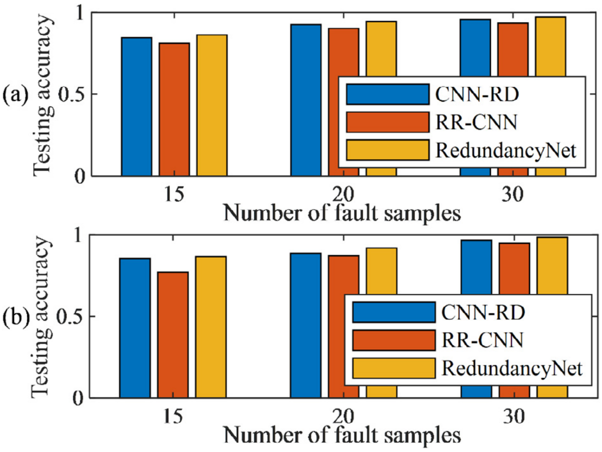

Fig. 11. Analysis results of ablation study. (a) The result on Case 1. (b) The result on Case 2.

two cases, yielding two models. The following are the two models:

(1) CNN-RD: This model also has three parts: generator, discriminator, and classifier. Its discriminator and classifier have the same structure and parameters as RedundancyNet. The generator of CNN-RD does not have merge layers. At the same time, resume update layer and resume predict layer are replaced by convolutional layers with the same parameters. In addition, all of the hyperparameters are identical to those used by RedundancyNet.

(2) RR-CNN: The model is also composed of a generator, a discriminator, and a classifier. The generator and discriminator of this model are identical to those of RedundancyNet. Unlike the classifier of RedundancyNet, the classifier of RR-CNN replaces four predict-update cycles with four convolutional layers. The rest of the structure and parameters are the same as RedundancyNet.

Both models are validated on Case 1 and Case 2, the results of which are shown in Fig. 11. As we can see, the testing accuracy of ablation models has dropped in both cases as compared to the proposed model. These results demonstrate that both the redundant reconstruction and redundant decomposition structures are critical to the success of RedundancyNet. When the redundant decomposition framework is absent, the test accuracy suffers more, indicating that the redundant decomposition structure plays a larger role. This is since, even if sufficient training samples are provided, extracting fault features from vibration data at sharp speed variations is challenging due to the speed-varying fault features of the data. As a result, the proposed redundant decomposition architecture is critical for IFD under varying speeds. Furthermore, the test accuracy of the two cases decreases when there is no redundant reconstruction structure. This shows that the proposed redundant reconstruction framework is effective and superior in the expansion of speed-varying vibration signal.

5.2. Understanding of learning process of the classifier

Known from the above analyses, high accuracy has been achieved under variable speed and small samples, except for multi-component failure, whereas it does not explain how the classifier learns these fault features.

In order to make sense the learning process, taking OF3 of Case 1 as an example, the outputs of predict layers and update layers are shown in Fig. 12. Output signal of predict layers and update layers get more and more sparse as these layers are closer to flattened layer. Comparing their last two layers, the signal from update layer is similar to approximation part, while that from predict layer is similar to detail part. That indicates that predict layers tend to obtain local components, and update layers prefer global components. It is a multi-resolution process. The signals from the first three predict layers and update layers are similar since the learning process is redundant as mentioned in Sectioned 3. Through the layer-by-layer multi-resolution process, the fault feature is enhanced and noise is reduced. As a result, the learning process of these layers of the net is similar to the decomposing process of RSGWD.

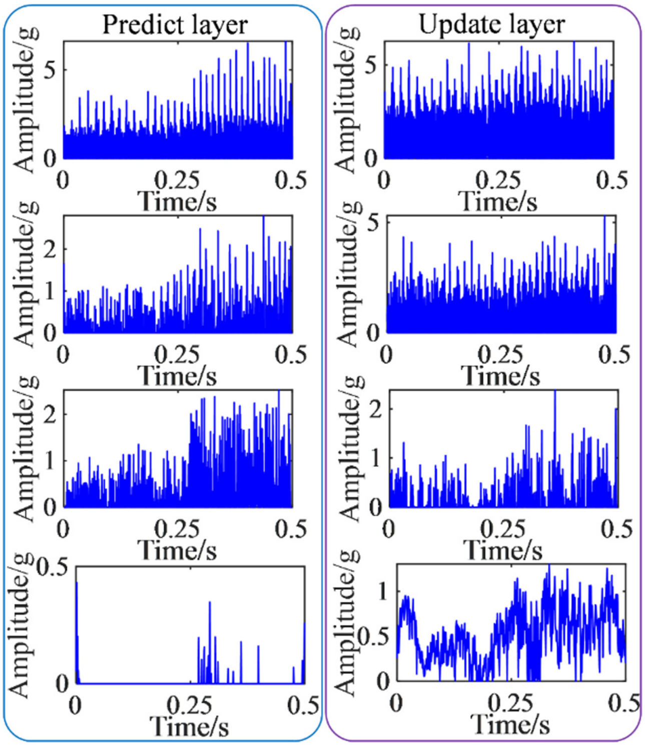

Fig. 12. Graphs of the outputs of different predict layers and update layers from OF3 of Case 1. Left graphs belong to predict layer, and right graphs belong to update layer. From top to down are the first, second, third, and fourth predict layer or update layer.

Although the above analyses illustrate that the learning process is layer-by-layer multi-resolution feature enhancement, it is a qualitative analysis. The Fourier spectrums of angular-domain resampling signal of those layers are used to further explore the learning mechanism. As shown in Fig. 13, along with close to flattened layer, high-frequency energy is lower and lower, and only outer-race fault characteristic order is observed in the order spectrum of the last update layer. Furthermore, there only exists noise in the last predict layer. It confirms that the detail part and approximation part are extracted respectively by predict layer and update layer. Therefore, the last update layer outputs the components that are closer to fault feature. And the process is a layer-by-layer multi-resolution feature enhancement similar to RSGWD. At the same time, irrelevant information such as noise is eliminated. Of course, it could be viewed that the extracted feature is fault characteristic order from the above analysis.

Therefore, redundant decomposition could capture weak fault feature and eliminate noise interference caused by varying speeds through the above process. Simultaneously, the learned fault feature is easy to understand.

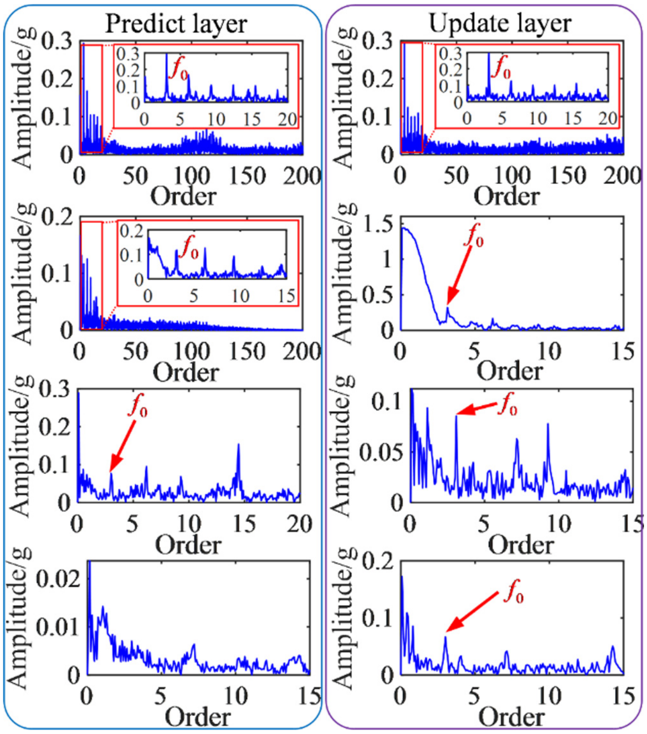

Fig. 13. Angular-domain spectrograms of the outputs of different predict layers and update layers from OF3 of Case 1. Left graphs are from predict layer, and right graphs are from update layer. From top to down are the first, the second, the third, and the fourth predict layer or update layer.

5.3. Benefits of generative adversarial framework

In order to improve the classification accuracy under small training samples, a generator via redundant reconstruction is included in RedundancyNet. The generator is to generate the data at varying speeds to expand samples through learning the features of real data. Many indicators have been applied to evaluate the generation performance of the generator [26]. However, the vibration signals analyzed in the paper are under varying speeds. At present, no index has been found to evaluate the similarity between different signals with varying speeds. As a result, the generator and the classifier are trained together to improve the classification accuracy. To evaluate the superiority of GAN, we use two approaches. One is human observation. The other is reverse evaluation through classification results.

The fault characteristic frequency cluster is important information of vibration signal with variable speeds in frequency domain, which is also a phenomenon of spectrum ambiguity. It can be viewed that the low-frequency fault characteristic frequency cluster is very similar between real data and generated data, shown in Fig. 14. As a result, fault feature is learned through generative adversarial framework. And training samples are effectively augmented.

Next, we evaluate generative adversarial framework through classification results. For this purpose, a network (RDN) whose structure is the same as that of the classifier of RedundancyNet is used to compare with RedundancyNet. As shown in Table 7, the accuracy of RDN is always lower than RedundancyNet. Although the accuracy of RDN reaches 95.71% in Case 1, the accuracy drops faster as the number of training samples decreases. Thus, GAN is important for the promotion of fault diagnosis performance, which expands the number of training samples through learning fault feature.

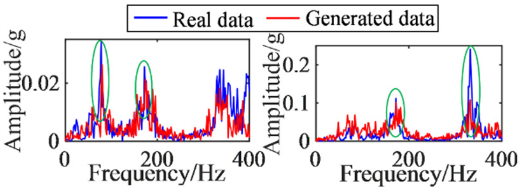

Fig. 14. Fourier spectrum of real sample and generated sample. From left to right belong to OF3 and IF3, respectively.

Table 7
Classification accuracy of RDN in Case 1 and Case 2.

Number of training samples	Accuracy (%)		Number of training samples	Accuracy (%)	
	Case 1	Case 2		Case 1	Case 2
20	85.71	78.18	30	95.71	90.61

Table 8
Classification accuracy of Single-Branch Network in Case 2.

Number of training samples	Test accuracy (%)	Number of training samples	Test accuracy (%)
20	88.94	30	97.12

5.4. Advantages of the multi-branch network

Due to the complex structure, multi-component failures often occur in the equipment together. There are two solutions to the problem. One is to directly classify all categories with single-branch architecture after ignoring the correlation of different components. The other is to use different branch for different component to diagnose fault. To evaluate which solution is more suitable for multi-component classification, a single-branch network whose structure is the same as RedundancyNet is applied except that it uses single-branch fully connected layer. The accuracy of single-branch network is shown in Table 8. Comparing the results of the single-branch network and the RedundancyNet, it is easily observed that the accuracy of the single-branch network is always inferior to that of multi-branch network no matter how the number of training samples changes.

This is due to the fact that the multi-branch network functions as a multi-task learning framework. There are both parallels and variances in the fault features of different components. The term “similarity” refers to the fact that they all need to reduce the noise of the signal before extracting fault characteristic order and its harmonics. At the same time, the vibration energy of different components varies due to differences in transmission paths and rotation speeds. As a result, some task parameters might be shared, but unique jobs have their own branches. One of the most significant advantages of this arrangement is that it provides more training examples for the shared parameters, hence enhancing the generalization performance of the classifier. Moreover, fault location and fault assessment can be realized together through separating multi-component fault features. Consequently, multi-branch architecture is more suitable for classification of multi-component failures, and it is applied in RedundancyNet.

5.5. Advantages and disadvantages of RedundancyNet

The advantages of the proposed RedundancyNet are summarized as follows:

(1) The proposed classification network provides a good solution for end-to-end IFD under variable speeds. From the experimental results of two cases, particularly the results

in Case 2, the proposed method can obtain a very high diagnostic accuracy under drastically variable speeds.

(2) The proposed generation network is a suitable fit for expanding vibration data at variable speed. As a result, the proposed RedundancyNet maintains a clear advantage over the existing methods in the case of small training samples.

(3) The suggested RedundancyNet combines the feature learning ability of a convolutional neural network with the interpretability of RSGWT, so that the feature extraction process of the proposed method is interpretable.

Although the proposed method has the aforementioned benefits, further study is needed to improve them. First, the suggested technique might take longer to compute than existing end-to-end IFD methods with small training samples. This is due to the fact that the classifier consists of multiple predict-update cycles, resulting in a rise in the amount of parameters. Changing the convolutional kernel of each cycle to a form comparable to the basis function of RSGWT could be one approach [47]. This solution could reduce the number of cycles through improving the feature extraction capacity of each cycle, thereby enhancing the computational efficiency.

In addition, due to the time-varying characteristics caused by variable speeds, the quality of generated samples at variable speed is indirectly assessed through their performance on the classification network. However, quantitative evaluation is more useful to find identifying high-quality generated samples and hence boosting classification performance with small training samples. Creating similarity index between generated samples and real samples at variable speeds might be a viable way to achieve this aim [32].

6. Conclusion

In this paper, a multi-branch redundant adversarial net, named RedundancyNet, is proposed. The net is for fault diagnosis of raw signals under drastically variable speeds and multi-component failure. The net can also be used under the condition of small training samples. Two cases and some state-of-the-art methods are utilized to verify RedundancyNet. The test accuracy of the proposed method on Case 1 and Case 2 is 97.14% and 98.33%, respectively, when 30 fault samples are used for training. In Case 1, the classification accuracy of those comparison methods for varying speeds only reaches 93.43%, while it only reaches 87.83% in Case 2. Although these comparison methods for small samples outperform the comparison methods for varying speeds in terms of classification accuracy, they can only achieve 95.14% in Case 1 and 93.79% in Case 2. Simultaneously, compared to these methods, the calculation time of the proposed method does not dramatically increase. As a result of these comparisons, the proposed method offers a large increase in test accuracy with only a minor increase in calculation time. The proposed method is also compared to MRLAN based on time–frequency graph. The test accuracy of MRLAN is nearly identical to that of the present approach on Case 2, whereas its calculation time is significantly longer. These analysis results prove the effectiveness and superiority of the presented approach in the field of intelligent fault diagnosis under drastically variable speeds and small training samples. In addition, the classifier based on RSGWD is easy to understand. Its learning process is layer-by-layer multi-resolution feature enhancement. Some attractions of RedundancyNet are shown as follows:

(1) For the first time, the redundant reconstruction-based generator is proposed to generate the signal with varying speeds through deeply fusing multi-scale fault features. The generator ensures that the classification network could have excellent generalization performance under sharp speed variations and small training samples.

(2) The classifier via redundant decomposition is presented to enhance non-stationary fault features and eliminate irrelevant interference information through layer-by-layer multi-resolution feature decomposition. Compared with other classification networks for varying speeds, the suggested classifier not only realizes superior diagnosis accuracy at drastically variable speeds, but also explains its own feature learning process.

(3) Multi-branch network is introduced to realize fault location and fault assessment of multiple component. In addition, multi-branch network, in comparison to the single-branch network, improves the generalization performance of the classification network by sharing parameters.

However, there are still many tasks to improve the current work. For instance, the similarity index of speed-varying signal should be research to evaluate generation framework besides of the use of classification results.

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.

Data availability statement

The authors declare that all the data used in this research was generated during the work. Moreover, the SQ data in Case 1 has been made public online: https://drive.google.com/file/d/1ld6Qv95zYdql0HhEvET5tA-44gLA-G1_/view?usp=sharing.

Acknowledgments

The authors would like to sincerely thank all the anonymous reviewers for the valuable comments that greatly helped to improve the manuscript.

This research is supported financially by the National Natural Science Foundation of China (No. 52175119, No. 91960106, No. U1933101), National Key Research and Development Program of China (No. 2019YFF0302204), China Postdoctoral Science Foundation (No. 2020T130509, No. 2018M631145), Scientific research and technology development in Liuzhou (No. 2021AAA0112) and Fundamental Research Funds for the Central Universities (No. XZY022020007, No. XZY022021006).

References

[1] Weldcherkos T, Salau AO, Ashagrie A. Modeling and design of an automatic generation control for hydropower plants using Neuro-Fuzzy controller. Energy Rep 2021;7:6626–37.

[2] Han H, Wang H, Liu Z, Wang J. Intelligent vibration signal denoising method based on non-local fully convolutional neural network for rolling bearings. ISA Trans 2021;1–11.

[3] Kumar P, Hati AS. Deep convolutional neural network based on adaptive gradient optimizer for fault detection in SCIM. ISA Trans 2021;111:350–9.

[4] Pan J, Zi Y, Chen J, Zhou Z, Wang B. Liftingnet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification. IEEE Trans Ind Electron 2018;65:4973–82.

[5] Chen Y, Zuo MJ. A sparse multivariate time series model-based fault detection method for gearboxes under variable speed condition. Mech Syst Signal Process 2022;167:108539.

[6] Liu R, Wang F, Yang B, Qin SJ. Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions. IEEE Trans Ind Inf 2020;16:3797–806.

[7] Wen L, Li X, Gao L. A transfer convolutional neural network for fault diagnosis based on ResNet-50. Neural Comput Appl 2020;32:6111–24.

[8] Shang J, Lin TR. Varying speed bearing fault diagnosis based on synchroextracting transform and deep residual network. In: 2020 Asia-pacific international symposium on advanced reliability and maintenance modeling (APARM), 2020. p. 1-5.

[9] Chen R, Huang X, Yang L, Xu X, Zhang X, Zhang Y. Intelligent fault diagnosis method of planetary gearboxes based on convolution neural network and discrete wavelet transform. Comput Ind 2019;106:48–59.

[10] Shi Z, Chen J, Zi Y, Zhou Z. A novel multitask adversarial network via redundant lifting for multicomponent intelligent fault detection under sharp speed variation. IEEE Trans Instrum Meas 2021;70:1–10.

[11] Xiang Z, Zhang X, Zhang W, Xia X. Fault diagnosis of rolling bearing under fluctuating speed and variable load based on TCO spectrum and stacking auto-encoder. Measurement 2019;138:162–74.

[12] Ma S, Chu F, Han Q. Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions. Mech Syst Signal Process 2019;127:190–201.

[13] Rao M, Zuo MJ. A new strategy for rotating machinery fault diagnosis under varying speed conditions based on deep neural networks and order tracking. In: 17th IEEE international conference on machine learning and applications, 2018. p. 1214-8.

[14] Hoang DT, Kang HJ. A survey on deep learning based bearing fault diagnosis. Neurocomputing 2019;335:327–35.

[15] Zhang C, Feng J, Hu C, Liu Z, Cheng L, Zhou Y. An intelligent fault diagnosis method of rolling bearing under variable working loads using 1-D stacked dilated convolutional neural network. IEEE Access 2020;8:63027–42.

[16] Lashgari E, Liang D, Maoz U. Data augmentation for deep-learning-based electroencephalography. J Neurosci Methods 2020;346:108885.

[17] Koulali R, Zaidani H, Zaim M. Image classification approach using machine learning and an industrial hadoop based data pipeline. Big Data Res 2021;24:100184.

[18] Kusrini K, Suputa S, Setyanto A, Agastya IMA, Priantoro H, Chandramouli K, et al. Data augmentation for automated pest classification in Mango farms. Comput Electron Agric 2020;179:105842.

[19] Wang Z, Du L, Mao J, Liu B, Yang D. SAR target detection based on SSD with data augmentation and transfer learning. IEEE Geosci Remote Sens Lett 2019;16:150–4.

[20] Janssen LAL, Arteaga IL. Data processing and augmentation of acoustic array signals for fault detection with machine learning. J Sound Vib 2020;483:115483.

[21] Li X, Li J, Qu Y, He D. Semi-supervised gear fault diagnosis using raw vibration signal based on deep learning. Chin J Aeronaut 2020;33:418–26.

[22] Liu F, Shen T, Luo Z, Zhao D, Guo S. Underwater target recognition using convolutional recurrent neural networks with 3-D Mel-spectrogram and data augmentation. Appl Acoust 2021;178:107989.

[23] Wang Y, Cheng X, Zhou P, Li B, Yuan X. Convolutional neural network-based moving ground target classification using raw seismic waveforms as input. IEEE Sens J 2019;19:5751–9.

[24] Oh C, Han S, Jeong J. Time-series data augmentation based on interpolation. Procedia Comput Sci 2020;175:64–71.

[25] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Adv Neural Inf Process Syst 2014;27:2672–80.

[26] Xu K, Li S, Li R, Lu J, Li X, Zeng M. Domain adaptation network with double adversarial mechanism for intelligent fault diagnosis. Appl Sci-Basel 2021;11:7983.

[27] Shao S, Wang P, Yan R. Generative adversarial networks for data augmentation in machine fault diagnosis. Comput Ind 2019;106:85–93.

[28] Odena A. Semi-supervised learning with generative adversarial networks. 2016, arXiv preprint, arXiv:1606.01583.

[29] Mirza M, Osindero S. Conditional generative adversarial nets. Comput Sci 2014;2672–80.

[30] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier GANs. 2017, arXiv preprint, arXiv:1610.09585.

[31] Lei Y, Yang B, Jiang X, Jia F, Li N, Nandi AK. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mech Syst Signal Process 2020;138:106587.

[32] Zhang T, Chen J, Li F, Pan T, He S. A small sample focused intelligent fault diagnosis scheme of machines via multi-modules learning with gradient penalized generative adversarial networks. IEEE Trans Ind Electron 2021;68:10130–41.

[33] Wan W, He S, Chen J, Li A, Feng Y. QSCGAN: An un-supervised quick self-attention convolutional GAN for LRE bearing fault diagnosis under limited label-lacked data. IEEE Trans Instrum Meas 2021;70:1–16.

[34] Liu S, Chen J, He S, Shi Z, Zhou Z. Subspace network with shared representation learning for intelligent fault diagnosis of machine under speed transient conditions with few samples. ISA Trans 2021;1–14.

[35] Chen P, Li Y, Wang K, Zuo MJ. An automatic speed adaption neural network model for planetary gearbox fault diagnosis. Measurement 2021;171:108784.

[36] Choudhury MD, Hong L, Dhupia JS. A novel tacholess order analysis method for bearings operating under time-varying speed conditions. Measurement 2021;186:110127.

[37] An Z, Li S, Wang J, Jiang X. A novel bearing intelligent fault diagnosis framework under time-varying working conditions using recurrent neural network. ISA Trans 2020;100:155–70.

[38] Wang F, Liu R, Hu Q, Chen X. Cascade convolutional neural network with progressive optimization for motor fault diagnosis under nonstationary conditions. IEEE Trans Ind Inf 2021;17:2511–21.

[39] Xiong X, Jiang H, Li X, Niu M. A wasserstein gradient-penalty generative adversarial network with deep auto-encoder for bearing intelligent fault diagnosis. Meas Sci Technol 2020;31:045006.

[40] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. 2017, p. 30, arXiv preprint, arXiv:1701.07875.

[41] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of wasserstein GANs. In: Advances in neural information processing systems. Vol. 30, 2017.

[42] Zhou R, Bao W, Li N, Huang X, Yu D. Mechanical equipment fault diagnosis based on redundant second generation wavelet packet transform. Digit Signal Process 2010;20:276–88.

[43] He K, Sun J. Ieee, convolutional neural networks at constrained time cost. In: IEEE conference on computer vision and pattern recognition (CVPR). Boston, MA; 2015, p. 5353–60.

[44] Han T, Liu C, Wu L, Sarkar S, Jiang D. An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems. Mech Syst Signal Process 2019;117:170–87.

[45] Wang Y, Ding X, Zeng Q, Wang L, Shao Y. Intelligent rolling bearing fault diagnosis via vision ConvNet. IEEE Sens J 2021;21:6600–9.

[46] Gao X, Deng F, Yue X. Data augmentation in fault diagnosis based on the wasserstein generative adversarial network with gradient penalty. Neurocomputing 2020;396:487–94.

[47] Li T, Zhao Z, Sun C, Cheng L, Chen X, Yan R, et al. Waveletkernelnet: An interpretable deep neural network for industrial intelligent diagnosis. IEEE Trans Syst Man Cybern 2021;1–11.