IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 5, 2019, accepted August 20, 2019, date of publication August 27, 2019, date of current version September 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937838

Data Simulation by Resampling–A Practical Data Augmentation Algorithm for Periodical Signal Analysis-Based Fault Diagnosis

TIANHAO HU, TANG TANG, AND MING CHEN
School of Mechanical Engineering, Tongji University, Shanghai 201804, China

Corresponding author: Tang Tang (tang.tang@tongji.edu.cn)

This work was supported in part by the Application of New Mode for Intelligent Manufacturing of Lifting Equipment for Large-scale Marine Engineering 2017, in part by the Project of Remote Operation and Maintenance Standards and Test Verification for Integrated Circuit Packaging Key Equipment 2018 in the Ministry of Industry and Information Technology, China, and in part by the Program for Young Excellent Talents in Tongji University, Shanghai, China under Grant 2016KJ020.

ABSTRACT In recent years, machine learning and deep learning based fault diagnosis methods have been studied, however, most of them remain at the experimental stage mainly because of two obstacles, briefly, a) inadequate faulty examples and b) various working conditions of industrial data. In this literature, a practical algorithm named Data Simulation by Resampling (DSR) is proposed for data augmentation to alleviate the two problems in fault diagnosis. In essence, as a form of Vicinal Risk Minimization (VRM), DSR utilizes a two-stage resampling operation to simulate vicinal examples in both time domain and frequency domain. By doing so, DSR can both increase the sample diversity and the quantity of training set, which regularizes machine learning and deep learning based methods to achieve a higher generalization performance. Our experiments verify the effectiveness of DSR and show the possibility of combining it with other augmentation algorithms.

INDEX TERMS Data augmentation, resampling, vicinal risk minimization, generalization.

I. INTRODUCTION

Fault diagnosis plays an important role in Prognostic and Health Management (PHM) of intelligent manufacturing. With the rise of artificial intelligence, machine learning and deep learning technologies have been utilized for different fault diagnosis applications [1]–[4]. Usually, these models are trained to minimize their average error over the training data, which is a learning rule called Empirical Risk Minimization (ERM). Based on ERM principle, an adequate and complete training set with a large quantity of examples is required to optimize models' parameters. Besides, the successful applications of ERM always rely on a universal hypothesis that, training data and testing data are expected to subject to the same distribution.

However, in real-world diagnostic scenarios, it is hard to meet the two demands. On the one hand, considering the production cost and machinery stability requirements, the distribution of industrial data is severely imbalanced [4], [5], where in most operating periods, machinery runs under a healthy condition. Although a large amount of data can be easily obtained through an online data acquisition system [3], actually effective faulty data is very rare. Besides, labelling work for massive data is also expensive. On the other hand, in real production processes, the generation mechanism of data is often non-stationary [6]. For instance, a piece of production equipment needs to operate alternately under 3 different set of process parameters, where load, rotating rate, temperature and other factors keep fluctuating and/or differ in different periods. Limited faulty data might not cover all the variations of working conditions. And this may lead to a bad generalization performance when applying trained models directly to testing scenarios.

An insufficient and incomplete training set really challenges the suitability of machine learning based models, especially those equipped with ERM principle. Strikingly, ERM is guaranteed as long as the number of training data is sufficient for model optimization [7]. When the model complexity is fixed, the less the training data is, the larger the generalization error boundary is [8]. Recent research [9] has shown that

The associate editor coordinating the review of this article and approving it for publication was Dong Wang.

several ERM based models (like large neural networks) tend to memorize (instead of generalizing from) training data. And when it comes to the evaluation of examples from testing distribution, a sharp decline may occur to those models just because of a slight discrepancy between training distribution and testing distribution [10]–[13].

An intuitive solution to this dilemma is *Data Augmentation* [14], where a model is trained on similar but different examples to the training data. Data Augmentation was initially formalized by the *Vicinal Risk Minimization* (VRM) principle [7]. In VRM, the concept of vicinity is utilized to describe the neighborhood around each training example, and what Data Augmentation does is to define the form of vicinal distribution, then virtual examples could be sampled from the vicinal distribution as to enlarge the original distribution of training domain. Taking image classification as an instance, a typical strategy is to utilize horizontal reflections, slight rotations, mild scaling and adding noise as a set of augmentation operations [14], and the distribution of virtual examples derived from those operations is defined as the vicinity of an original example. Despite the consistent contribution of Data Augmentation to generalization, however, for fault diagnosis, what is the vicinity of most industrial data (especially periodical vibration signals)?

A practical vibration analysis technology named Order Tracking [18] gives us inspiration. In Order Tracking, time-varying signals can be approximately resampled into a stationary pattern in the angular domain, thus to allow stationary frequency analysis based algorithms to be applied. The successful applications of Order Tracking indicate a hypothesis that *speed-related periodical vibrations share the same pattern in the angular domain despite different rotating speeds*. Therefore, intuitively, it is common to define the vicinity of an original (vibration) example as a set of signals from the same pattern but under different rotating speeds.

In this paper, a simple data augmentation algorithm named as Data Simulation by Resampling is proposed to simulate (virtual) vicinal examples from each original signal. In essence, it is a simulation process, where simulated signals can be obtained via a two-stage resampling process based on a pseudo rotating speed ratio from one training example. To fulfill the whole process, Fast Fourier Transform (FFT) and its inverse version (IFFT) are embedded in to acquire frequency spectra and simulated time signals. Besides, the influences of working load and environmental noise are also taken into consideration in DSR, which brings two extra parameters in DSR.

Despite the simplicity of the core philosophy, DSR allows a significant improvement of generalization performance on several different diagnosis models in, a) working condition transferring diagnostic scenarios as well as b) its few-shot counterpart (with extremely few training examples). Besides, a thorough set of ablation study experiments are also implemented, and the results validate the effectiveness of both DSR as well as its combination with a prior data augmentation algorithm (Mix-up [15]).

The rest of paper is organized as follows. In Section II, some prior works on fault diagnosis and data augmentation are reviewed. Section III describes the proposed algorithm in detail. Sequentially, experiments and the corresponding result analysis on several diagnosis scenarios, as well as the ablation studies are depicted in Section IV. Finally, the conclusion of this literature is made in Section V.

II. RELATED WORKS

A. FAULT DIAGNOSIS

In industry applications, most bearing fault diagnosis systems are based on traditional signal processing methods [1], [2] such as the FFT, Hilbert Transform (HT), Empirical Mode Decomposition (EMD) [16], Wigner-Ville Distribution (WVD), and Wavelet Transform (WT) [17]. In view of the varying rotating speed problem, Order Tracking has been proposed and has achieved a great success [18], [19]. Besides, traditional machine learning methods have also been applied for rotating machinery fault diagnosis, and some typical models developed with different modification strategies have been verified to be efficient, such as K-Nearest-Neighbor (KNN) [20], Support Vector Machine (SVM) [21], Decision Tree (DT) [22], Back Propagation Neural Network (BPNN) [23], Extreme Learning Machine (ELM) [24], Sparse Coding (SC) [25], etc.

In recent years, deep learning (DL) based algorithms have been studied for different fault diagnosis applications [26]–[38]. Convolution Neural Network (CNN) was first proposed by Lecun, and the classic LeNet-5 has been successfully modified for fault diagnosis [26] where 1-D raw time domain signals are folded into 2-D images via segmentation. Because of its unique topology structure, CNN has been utilized for most cases of time domain and time-frequency analysis based diagnosis scenarios. Sun *et al.* [27] utilized model transferring strategy that directly transfers parameters from trained BPNN to convolutional filter without fine-tuning. Experiments show the great adaptability of convolutional filters when processing raw time sequences. Jia *et al.* [5] proposed a Deep Normalized Convolutional Neural Network (DNCNN) in which normalization layers are utilized intended for imbalanced classification. Zhang *et al.* [28] proposed a new 1-D deep convolutional structure called Convolution Neural Networks with Training Interference (TICNN) for vibration signal under noisy environment and different working loads. They evaluated the effects of Dropout, Batch Normalization and noise injection on the model's performance. Inspired by the Second Generation Wavelet Transform (SGWT), Pan *et al.* [29] presented the Lifting-Net where the structure is constructed alternately by split layers, predict layers and update layers with different convolutional kernel size. The effectiveness of some other novel CNN-based structures have also been verified [30]–[31].

In most cases of frequency domain based diagnosis, Auto-Encoder (AE) models are usually utilized as an inference

mechanism. Jia *et al.* [32] presented a study of stacked AE with two-stage training processes for bearing diagnosis. Sun *et al.* [33] proposed a Sparse Auto-Encoder (SAE) based model for induction motor fault diagnosis. Tricks of de-noising coding and dropout are utilized to get a more robust feature representation. In [34], Wang *et al.* combined the Gaussian kernel function and stacked auto-encoder for fault diagnosis of the inter-shaft bearing of an aircraft engine. Except the literature mentioned above, some researchers have prompted the AE based deep structure for gearbox [35], [36] and hydraulic pump [37]. Besides, there are also a few attempts of shallow CNN structures being utilized for frequency domain based fault diagnosis [38].

However, despite the contributions of those models mentioned above, the contradiction between limited faulty samples and model complexity challenges most ERM based models and prevent them from being applied in actual diagnostics applications. In order to validate the effectiveness of the proposed data augmentation algorithm DSR, both ERM based (BPNN, CNNs) and several non-ERM based (SVM, KNN) models are utilized as comparison methods, and their implementation details are described in Section 4.1.

B. DATA AUGMENTATION

Nowadays, the successful applications of most ERM-based deep models cannot be separated from data augmentation. In the field of image recognition, it is common to use slight rotation, transition, cropping, scaling as data augmentation [8], [14], where its rationality lies at the fact that the semantic meaning of training examples does not change. In [39], Zhong *et al.* proposed the Random Erasing algorithm on training images to enhance the invariance of trained models. In the field of speech recognition, where speech signals are similar to the periodic vibration signals, noise is routinely injected into training signals with high Signal-to-Noise Ratio (SNR), as to improve the robustness and accuracy of models [40], [41]. And in fault diagnosis, one has also been used to improve the generalization performance of diagnosis models [28], [30]. Another practical augmentation method for fault diagnosis is slicing training samples with overlap [28], and extra acquisition and storage requirements are introduced to obtain primitive signals, which are expected to be longer than sliced samples.

In all cases mentioned above, substantial domain knowledge has been leveraged for the design of suitable augmentation approaches. And several domain-independent algorithms have also been developed mainly for network based models. The core philosophy is to replace the high confidence of softmax distribution caused by the hard label in ERM with a smooth one by regularizations, such as label smoothing [42], and/or adding a penalty term to cross-entropy loss [43]. Besides, Zhang *et al.* [15] proposed Mixup, which uses a simple convex combination of inputs and the corresponding labels given as follows for augmentation:

$$\begin{cases} x_{input} = \lambda x_i + (1-\lambda)\, x_j \\ y_{input} = \lambda y_i + (1-\lambda)\, y_j \end{cases}$$

where examples (x_i, y_i) and (x_j, y_j) are randomly drawn from the training set, and $\lambda \sim Beta\,(a, a)$ for $a \in (0, +\infty)$. We can easily find that $\lambda \in (0, 1)$ and a controls the strength of interpolation between feature-label pairs. The mixture of features and the corresponding labels contributes to smoothing the output distribution of models, and this method potentially satisfies the Lipschitz condition.

III. DATA SIMULATION BY RESAMPLING

A. MOTIVATIONS FROM ORDER TRACKING

Order Tracking is a classic vibration analysis method, which allows stationary frequency analysis based algorithms to be applied. It uses recorded information of rotating speeds [19] or estimated instantaneous phases [20] to resample vibration signals into the angular domain as to eliminate the interference of speed-variations.

Take Computed Order Tracking (COT) [18] as an instance, the whole process is illustrated in Fig.1. Conventionally, vibration signals are sampled at a constant increment of time (i.e. uniform Δt) and keyphasor pulses are sampled at a constant increment of rotating angle, COT uses the time intervals between pulses to resample vibration signals in the angular domain with constant angular increment (i.e. uniform $\Delta\theta$). And usually, quadratic fitting is utilized to determine the resampling coordinate along the time axis.

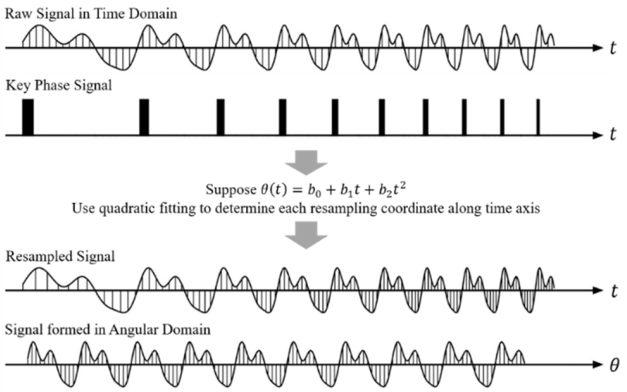


FIGURE 1. The resampling process of COT.

However, convinced by the effectiveness of Order Tracking, we usually ignore a basic hypothesis underlying it. That is, in the perspective of pattern recognition,

Assumption: *Signals under different speed-variations / speeds may satisfy the same pattern in the angular domain.*

Order Tracking use this assumption to find the invariant angular pattern of signals with speed-variations. And inversely, we can also utilize it to simulate virtual signals with various speeds / speed-variations from just the same pattern, as shown in Fig.2.

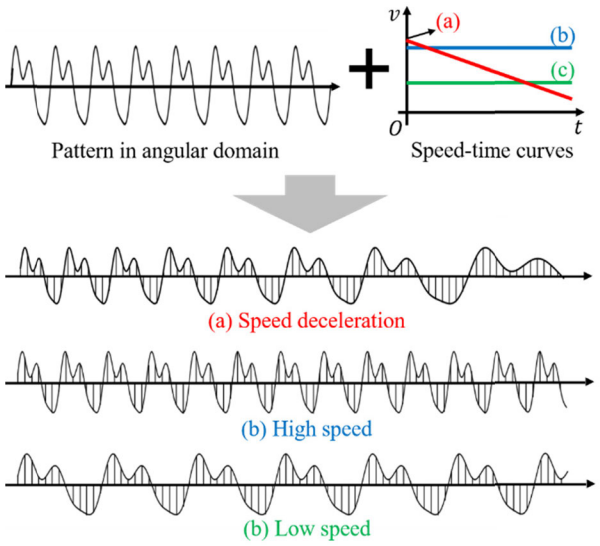


FIGURE 2. Signals under different speed-variations / speeds may satisfy the same pattern in the angular domain.

According to the idea above, in this paper, we design a two-stage resampling process to fulfill the framework of our methodology. And the Fast Fourier Transform is embedded in to obtain the frequency spectrum.

B. RESAMPLING IN TIME DOMAIN

Suppose the shaft rotates at a constant speed in a relatively short period of time, that means the relationship between rotating angle and speed is a linear function, where $\theta(v,t) = \theta_0 + 2\pi vt/60$. As illustrated in Fig.3, two signals with rotating speeds (here v_o and v_s respectively) but the same sampling interval (Δt) in the time domain are derived from the same angular pattern. The only difference between them is their different sampling intervals ($\Delta\theta$ and $\Delta\varphi$) in the angular domain.

Intuitively, we can easily obtain a simulated signal under a different rotating speed by resampling. Given an original signal with n sampling points, denoted as $\mathcal{A}^{(t)} = \{(\Delta t, a_1), (2\Delta t, a_2), \ldots, (n\Delta t, a_n)\}$, where $a_{1\ldots n}$ are the vibration amplitudes. Its rotating speed is v_o. It can be reformed in the angular domain as $\mathcal{A}^{(\theta)} = (\theta_1, a_1), (\theta_2, a_2), \ldots (\theta_n, a_n)$, where $\theta_i = \theta_0 + i \cdot 2\pi v_o \Delta t/60$. If we want to obtain a signal with rotating speed v_s, we can just resample $\mathcal{A}^{(\theta)}$ in the angular domain with a set of new coordinates, which can be computed as $\varphi_j = \theta_0 + j \cdot 2\pi v_s \Delta t/60$.

Moreover, the resampling process can be simplified to a more direct form, which represents another physical meaning. Further considering the following equation:

$$\varphi_j = \theta_0 + \frac{2\pi}{60} v_o \cdot j\Delta t = \theta_0 + \frac{2\pi}{60} v_o \cdot j\frac{v_s}{v_o}\Delta t \xrightarrow{def} \Delta\tau = \frac{v_s}{v_o}\Delta t$$

Suppose a pseudo speed rate denoted as $r = v_s/v_o$, thus the new time increment for resampling can be calculated as

$$\Delta\tau = r\Delta t \tag{1}$$

Therefore, as also illustrated in Fig.3, the resampling process can be directly done in the time domain with resampling coordinates $[\Delta\tau, 2\Delta\tau, \ldots N\Delta\tau]$. After that, we replace the increment of $\Delta\tau$ with the original increment Δt, and finally, the resampled signal can be reformed as:

$$\mathcal{A}_s^{(t)} = \{(\Delta t, b_1), (2\Delta t, b_2), \ldots, (N\Delta t, b_n)\} \tag{2}$$

where $b_{1\ldots n}$ represent the interpolation result during resampling.

In perspective of the simplified process, resampling can be simply regarded as a stretch or compression operation for 1-D signals along the time axis. And it is important to emphasis that the dimension of a resampled signal is different from its original one, where $N \neq n$. And the boundary constraint should be satisfied that:

$$N\Delta\tau < n\Delta t \Longleftrightarrow N < \frac{n}{r} \tag{3}$$

In order to obtain features of simulated signals with the same dimension of original ones, another resampling is needed to be done, which is described in the next subsection.

C. LOAD INFLUENCE AND ENVIRONMENTAL NOISE

To the knowledge of several prior works [44], in fault diagnosis, the manifestation of vibration is affected not only by the rotating speed, but also by the load distribution around the defect point. A classic defect model [44] gives us an insight of this, Given a vibration signal $x(t)$ produced by rolling bearing with a single defect on the inner race, it can be represented as

$$x(t) = [d(t) \cdot q(t) \cdot a(t)] * h(t)$$

where $d(t), a(t), h(t)$ are the pattern of impulses, transfer function and impulse response of the low-pass filter, respectively; and $q(t)$ denotes the changes in load distribution when the shaft rotates. which is affected by the maximum load intensity q_0,a load coefficient term affected by the applied load.

Therefore, in the perspective of data augmentation, signals under different loads and rotating speeds can be commonly regarded as the vicinal examples of the original training signals. In addition, environmental noise as another factor [28], [30] should also be considered for the generation of vicinal signals. In order to integrate both load and noise influence, a pseudo load ratio coefficient l and a set of noise $\{\varepsilon_{1\ldots N}\}$ are introduced, and the resampled signal $\mathcal{A}_s'^{(t)}$ can be multiplied by l and then added by noise, where the result is calculated as:

$$A_s'^{(t)} = (\Delta t, lb_1 + \varepsilon_1), (2\Delta t, lb_2 + \varepsilon_2), \ldots, (N\Delta t, lb_N + \varepsilon_N) \tag{4}$$

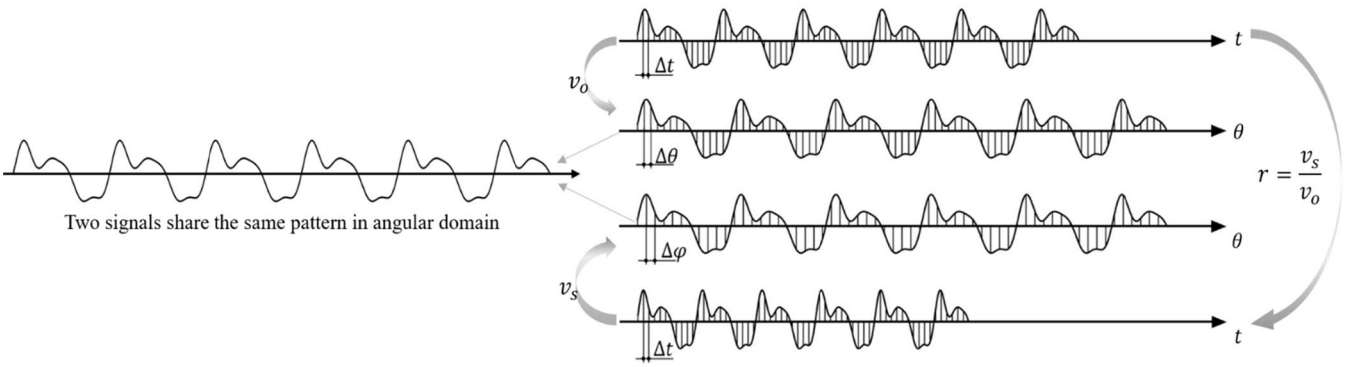


FIGURE 3. Illustration of two signals in the time domain with different rotating speeds and their sharing pattern in the angular domain. Here $\Delta\theta$ and $\Delta\varphi$ are the sampling intervals in angular domain, $\Delta\theta \neq \Delta\varphi$, Δt is the shared sampling interval in the time domain.

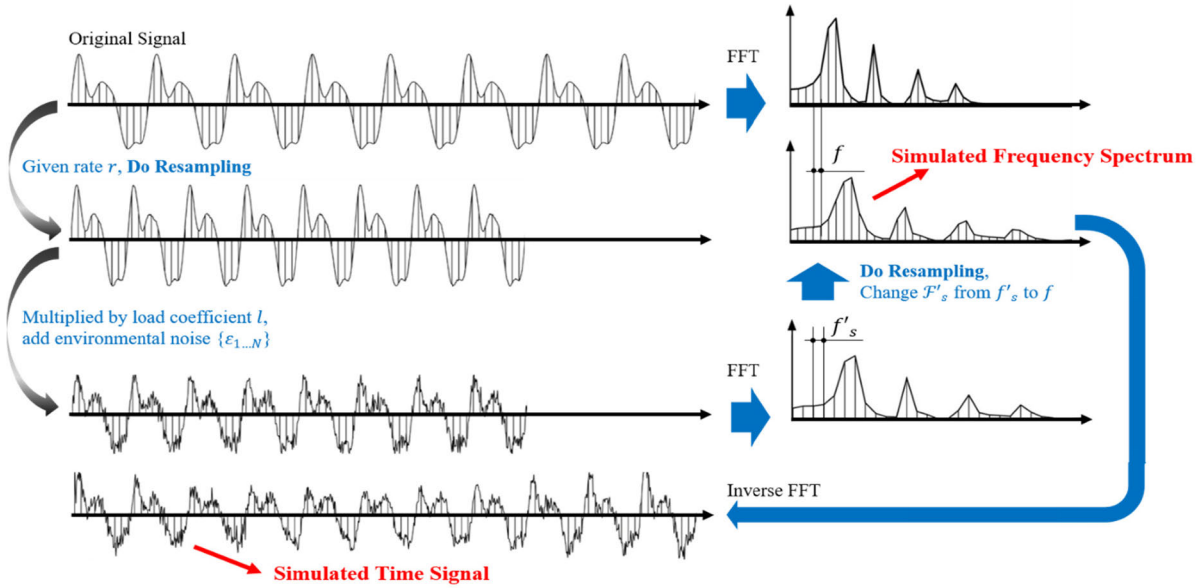


FIGURE 4. The whole process of the proposed DSR (Data Augmentation by Resampling). Resampling operations take place in both time domain and frequency domain, where the first resampling leads to dimension incompatibility (n≠N) and the second one solve the problem as well as keep the characteristics of the frequency spectrum of the resampled signal.

D. RESAMPLING IN FREQUENCY DOMAIN

By means of the FFT method, we can obtain the frequency spectra of $\mathcal{A}^{(t)}$ and $\mathcal{A}'^{(t)}_s$, denoted as $\mathcal{F}$ and $\mathcal{F}'_s$, which share the same maximum of frequency f_{max}, but differ in the frequency resolution of, denoted as f, f'_s, where:

$$(f_{max}, f, f'_s) = \left(\frac{1}{2\Delta t}, \frac{1}{n\Delta t}, \frac{1}{N\Delta t}\right) \tag{5}$$

It is necessary to align the dimension of both the time domain and frequency domain, while isomorphic data are the basic requirement for most machine learning based algorithms. Therefore, another resampling operation is utilized to transfer the resolution of $\mathcal{F}_s$ from f'_s to f. And the amplitude spectrum is selected as the simulated signal in the frequency domain.

After that, an Inverse FFT (IFFT) operation is adopted to cope with the simulated frequency spectrum. And the real-part of the complex result after IFFT is selected as the simulated signal in the time domain. Since we only select the real part of the IFFT result, in order to compensate for the lost energy (roughly half of the IFFT result), we multiply the real part by $\sqrt{2}$ to get the final simulated signal in the time domain. The whole process of DSR is shown in Fig.4.

It is important to emphasize the choice of basement signal used for interpolation in the second resampling process. There are two options, one is to interpolate directly on the complex result after FFT, and the other is to interpolate separately on both amplitude characteristics and phase characteristics of the complex result. We choose the first one on account of a less phase shift. Taking the average interpolation as an example, from Fig.5 we can observe that the interpolation solution on the complex result retains more phase characteristics of the intensive spectral line, the analysis of which plays an important role in traditional fault diagnosis methods.

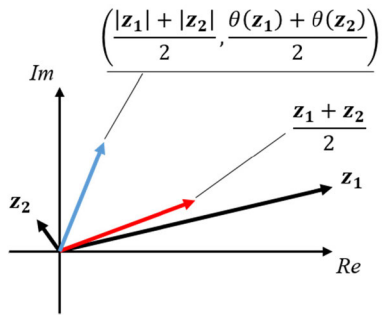


FIGURE 5. Different ways of interpolation. The red arrow represents the interpolation result based on the complex result $\mathbf{z_1}, \mathbf{z_2}$, and the blue one represents the interpolation result separately based on the amplitude and phase characteristic of $\mathbf{z_1}, \mathbf{z_2}$.

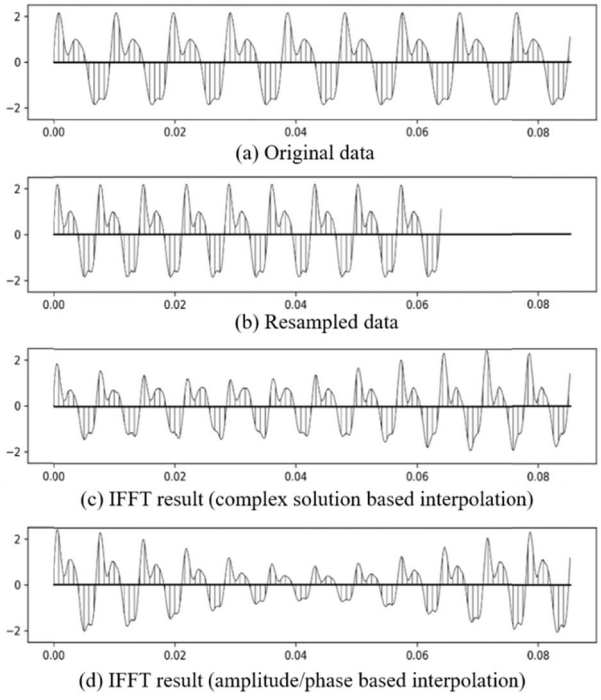


FIGURE 6. The comparison of (a) original signal, (b) resampled signal, (c)(d) two IFFT results based on different interpolation options.

This can be validated in Fig.6 that the IFFT result of the first option has a better performance than the one of the second option, where the first one has a lower deformation of vibration amplitudes in the time domain.

In a summary, in order to obtain a simulated frequency spectrum as well as the corresponding simulated time signal from the original signal $\mathcal{A}^{(t)}$, a pseudo speed ratio r, a pseudo load ratio l and a set of noise $\{\varepsilon_{1\ldots N}\}$ are required for the basic calculation. Empirically, we suppose those parameters subject the Gaussian distribution, where $(r, l, \varepsilon_{0\ldots N})$ are sampled from a joint distribution of three independent Gaussian distributions, where:

$$\begin{aligned}(r, l, \varepsilon_{0\ldots N}) &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}) \\ &= \mathcal{N}\left(1, \sigma_r^2\right) \cdot \mathcal{N}\left(1, \sigma_l^2\right) \cdot \mathcal{N}\left(\mathbf{0}, \sigma_\varepsilon^2 \boldsymbol{I}_N\right) \ldots\end{aligned} \tag{6}$$

Algorithm 1 Data Simulation by Resampling

Input: One original signal: $\mathcal{A}^{(t)}$ Parameters to generate pseudo coefficients: $\sigma_r^2, \sigma_l^2, \sigma_\varepsilon^2$

Output: A simulated frequency spectrum and a simulated time signal

Do:

1. Generate pseudo parameters $(r, l, \varepsilon_{0\ldots N})$ according to Eq.(6)
2. Resample the given original data $\mathcal{A}^{(t)}$ to $\mathcal{A}_s^{(t)}$
3. Add load influence and noise and compute $\mathcal{A}'^{(t)}_s$ according to Eq. (4)
4. Calculate the FFT result $\mathcal{F}'_s$ of $\mathcal{A}'^{(t)}_s$ by FFT.
5. Resample $\mathcal{F}'_s$ from resolution f'_s to f as a simulated frequency spectrum.
6. Calculate the IFFT result of resampled $\mathcal{F}'_s$
7. Select the real part of the IFFT result, and multiply it by $\sqrt{2}$ to obtain the simulated time signal

End

Besides, the generated r, l should subject to the boundary constrain, that is $r, l > 0$. The whole procedures are summarized as **Algorithm I**.

E. FORMALIZATION IN VICINAL RISK MINIMIZATION

In most diagnosis applications, the learning problem of diagnosis models can be summarized as searching a function $h \epsilon \mathcal{H}$ to describe the relationship between pattern $\boldsymbol{x}$ and target y. Giving a loss function $l\left(h\left(x\right), y\right)$ as the criterion, the minimization process of expected risk can be formulated as:

$$R\left(h\right) = \int l\left(h\left(x\right), y\right) dP(x, y)$$

where $P(\boldsymbol{x}, y)$ is the joint distribution of (x, y). But unfortunately, $P(x, y)$ is unknown in most situations. Instead, a set of examples, denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, can be relatively easy to be obtained, where $(x_i, y_i) \sim P$. And it is commonly to minimizes the empirical risk:

$$R_{ERM}\left(h\right) = \frac{1}{n} \sum_{i=1}^{n} l\left(h\left(x_i\right), y_i\right)$$

The empirical risk and the expected risk can be equivalent if and only if joint distribution P is approximated by the empirical distribution P_{ERM}, formed by Dirac function δ centered at each (x_i, y_i), where

$$dP_{ERM}\left(\boldsymbol{x}, y\right) = \frac{1}{n} \sum_{i=1}^{n} \delta\left(x = x_i\right) \delta\left(y = y_i\right)$$

However, when it comes to a model with quantities of parameters (i.e. $>$ n) such as a large neural network, it is trivial for model h to memorize limited training examples used for learning, which can lead to a sharp decline on performance as encountered with examples just outside the dense region of density $dP_{ERM}\left(x, y\right)$. This phenomenon is called, conventionally, ill-condition.

TABLE 1. Description of the dataset and diagnosis scenarios.

Conditions	Normal	Inner race fault (FI)			Outer race fault (FO)			Ball fault (FB)		
Fault Diameters/in.	/	0.007	0.014	0.021	0.007	0.014	0.021	0.007	0.014	0.021
Label	0	1	2	3	4	5	6	7	8	9
Dataset A (1HP) / B (2HP) / C (3HP)	Training set: 150/class, 1500 examples in total Validation set: 150/class, 1500 examples in total									
Diagnosis Scenarios	Dataset i (Training set) → Dataset j (Validation set), i, j ∈ {A, B, C}, i ≠ j									

It is natural to consider to improve the estimation $dP_{ERM}(x, y)$ of density $dP(x, y)$ by replacing the Dirac function $\delta(x = x_i)$ with a vicinal distribution centered at x_i, which are expected to provide a better support of the original distribution [7], [15]:

$$dP_{VRM}(x, y) = \frac{1}{n}\sum_{i=1}^{n}\delta(y = y_i)\, v(\tilde{x}|x_i)dx$$

where $v(\tilde{x}|x_i)$ as the vicinal distribution measures the probability density of the virtual pattern $\tilde{x}$ in the neighborhood about x_i. Particularly, Chapelle *et al.* [7] considered Gaussian vicinities $v(\tilde{\boldsymbol{x}}|\boldsymbol{x_i}) = \mathcal{N}(\boldsymbol{x_i}, \sigma_\varepsilon^2 \boldsymbol{I_N})$, which is equivalent to noise injection as augmentation [28], [30], [40], [41]. In this paper, the virtual example can be obtained by the simulation process of DSR, and the vicinity distribution of $\boldsymbol{x_i}$ can be denoted as:

$$\begin{aligned} v(\tilde{\boldsymbol{x}}|\boldsymbol{x_i}) &= P(simu(x_i) \mid r, l, \varepsilon) \\ (r, l, \varepsilon) &\sim \mathcal{N}\left(1, \sigma_r^2\right) \cdot \mathcal{N}\left(1, \sigma_l^2\right) \cdot \mathcal{N}\left(\boldsymbol{0}, \sigma_\varepsilon^2 \boldsymbol{I_N}\right) \end{aligned} \quad (7)$$

where function $simu(\cdot)$ represent the nonlinear mapping of the simulation process of DSR. The pair of parameter $\left(\sigma_r^2, \sigma_l^2, \sigma_\varepsilon^2\right)$ controls the manifestation of vicinity distributions. when $\sigma_r^2, \sigma_l^2 \to 0$, DSR degenerates into noise injection, and when $\sigma_r^2, \sigma_l^2, \sigma_\varepsilon^2 \to 0$, the vicinal risk degenerates into empirical risk.

F. LIMITATIONS OF DSR

In DSR, the second resampling procedure will inevitably cause phase shift. This shift has no effect on the amplitude spectrum, but it does lead to deformation (just like part of a beat phenomenon, see Fig.6(c)) in the virtual signals, which is analytically intractable. The deformation can bring in some unexpected patterns, leading to a harder optimization process, and/or even a wrong optimization direction for networks.

Besides, the distribution of a simulated signal resembles its original one as shown in Fig.6(c), especially at both ends. Therefore, excessive use of simulated virtual examples may cause overfitting by similar distributions.

Third, since we can use DSR to simulate signals with a higher rotating speed, the compressed signal (after the first resampling procedure) struggles to have a frequency resolution as precise as the original one. Therefore, the pseudo speed ratio r cannot exceed a certain value, which is related to the cutoff frequency of the machinery and the sampling rate.

Finally, the selection of hyper-parameters $\left(\sigma_r^2, \sigma_l^2, \sigma_\varepsilon^2\right)$ also requires prior knowledge. For example, we need to know the range of the rotating speed fluctuations, the environmental signal-to-noise ratio, etc. to empirically determine those parameters.

IV. EXPERIMENTS

A. DATASET AND COMPARISON METHODS

In this paper, we evaluate the DSR on the Case Western Reserve University (CWRU) bearing database [45], which has several different working conditions of (1HP, 1772rpm), (2HP, 1750rpm), (3HP, 1730rpm), respectively. As shown in TABLE 1, 3 different faulty locations and corresponding with 3 different faulty diameters for each location are contained in this database. Combined with the normal condition, there are 10 classes in total for recognition. Besides, we organize 6 different diagnosis scenarios in TABLE 1, where dataset i can be regarded as the training domain while dataset j is the testing domain for model evaluation. During the testing period, statistics of only training domain are utilized as the referred statistics in batch normalization of neural network models.

It is very important to emphasis that any statistics information of testing domain can't be adopted for the training process of models. Considering the fact that during online condition monitoring, data is generated periodically single by single, it is not appropriate for real-time diagnosis model to use statistics of a batch of testing data to update several reference variables during training, like means and variances in batch normalization in neural networks. To ensure this, we only use training set in training domain for model optimization and validation set of testing domain for model evaluation. During the testing period, statistics of only training domain are utilized as the referred statistics in batch normalization of neural network models.

During the experiment, we use DSR and ERM to train several traditional and classic machine learning algorithms as well as deep learning structure based models. The implementation details are described in Table. 2. For neural network based BPNN, CNN, TICNN, DNCNN and LeNet-5, we train these models for 100 epochs. To guarantee their stable performance, Adam optimizer [46] is used for the first 90 epochs, and SGD is used for the last 10 epochs. Hyper parameters of these models are carefully searched only based on the validation set of training domain.

B. RESULTS AND ANALYSIS

We conduct the first experiment with fix parameters of DSR, which are set as $\left(\sigma_r^2, \sigma_l^2, \sigma_\varepsilon^2\right) = (0.02, 0.1, 0.02)$, where we roughly set the pseudo rate variance σ_r^2 as 0.02 according to

TABLE 2. Implementation details of the comparison methods.

Feature Domain Based	Method Name	Implementation Details
Frequency Domain	KNN	KD-tree is utilized for searching for K nearest samples.
	SVM	Gaussian kernel is used, and the multiclass strategy is set as one-against-one.
	BPNN	The number of hidden neural is searching within {100, 200, 400, 800, 1600, 3200}
	CNN [38]	Convolutional kernel number and size are set as 32@64x1. Full connection hidden layer contains 1024 nodes, other implementation details can be referred to [38]
Time Domain	LeNet-5 [26]	Network structure and implementation details can be referred to [26]
	DNCNN [5]	Network structure and implementation details can be referred to [5]
	TICNN [28]	Network structure and implementation details can be referred to [28]

TABLE 3. Classification accuracy ± standard deviation (%) on six diagnosis scenarios.

Diagnosis Scenarios			A->B	A->C	B->A	B->C	C->A	C->B	Average
Frequency Domain	KNN	NBE	79.49 ± 0.48	79.04 ± 0.72	72.92 ± 1.10	70.01 ± 1.44	73.37 ± 1.03	70.84 ± 0.70	74.28 ± 0.91
		DSR	97.64 ± 0.87	95.21 ± 1.09	96.13 ± 0.73	**99.63 ± 0.06**	**89.86 ± 0.79**	**97.24 ± 1.34**	**95.95 ± 0.81**
			18.15↑	16.17↑	23.21↑	29.62↑	16.49↑	26.4↑	21.67↑
	SVM	SRM	79.09 ± 0.97	79.95 ± 1.60	75.30 ± 0.64	76.60 ± 1.35	71.36 ± 0.50	70.02 ± 0.32	75.39 ± 0.90
		DSR	98.45 ± 0.36	93.59 ± 2.14	89.55 ± 0.53	98.71 ± 0.83	87.44 ± 1.78	89.63 ± 0.23	92.9 ± 0.98
			19.36↑	13.64↑	14.25↑	22.11↑	16.08↑	19.61↑	17.51↑
	BPNN	ERM	79.03 ± 3.94	77.48 ± 2.04	69.02 ± 4.20	72.38 ± 3.21	71.31 ± 3.31	68.42 ± 1.22	72.94 ± 2.99
		DSR	**99.93 ± 0.10**	**97.62 ± 2.97**	89.75 ± 1.74	**99.71 ± 0.31**	85.65 ± 3.43	88.84 ± 2.64	93.58 ± 1.87
			20.9↑	20.14↑	20.73↑	27.33↑	14.34↑	20.42↑	20.64↑
	CNN	ERM	79.03 ± 3.94	77.48 ± 2.04	69.02 ± 4.20	72.38 ± 3.21	71.31 ± 3.31	68.42 ± 1.22	72.94 ± 2.99
		DSR	98.77 ± 0.91	94.71 ± 2.90	89.72 ± 1.32	98.86 ± 1.80	79.69 ± 4.85	80.92 ± 2.12	90.45 ± 2.32
			19.74↑	17.23↑	20.7↑	26.48↑	8.38↑	12.5↑	17.51↑
Time Domain	TICNN	ERM	**99.92 ± 0.08**	**97.63 ± 1.82**	**96.73 ± 3.00**	97.72 ± 2.92	79.39 ± 2.31	84.68 ± 4.23	92.59 ± 2.39
		DSR	98.58 ± 0.60	97.07 ± 1.03	91.32 ± 1.55	97.17 ± 2.51	83.40 ± 2.96	89.53 ± 3.20	92.85 ± 1.98
			1.34↓	0.56↓	5.41↓	0.57↓	4.01↑	4.85↑	0.36↑
	LeNet-5	ERM	89.69 ± 4.30	92.67 ± 4.80	89.97 ± 5.70	85.19 ± 5.49	79.22 ± 9.47	88.09 ± 5.19	87.47 ± 5.83
		DSR	97.96 ± 1.06	94.06 ± 3.67	94.16 ± 2.36	97.78 ± 1.99	86.34 ± 1.98	89.70 ± 2.28	93.33 ± 2.14
			8.27↑	1.39↑	4.19↑	12.59↑	7.12↑	1.61↑	5.86↑
	DNCNN	ERM	99.99 ± 0.02	92.37 ± 4.98	87.14 ± 4.93	81.20 ± 9.43	76.50 ± 5.58	81.07 ± 3.10	86.38 ± 4.68
		DSR	95.84 ± 2.97	92.15 ± 4.10	89.18 ± 2.53	92.47 ± 4.86	79.56 ± 0.57	82.04 ± 1.17	88.54 ± 2.70
			4.15↓	0.22↓	2.04↑	11.27↑	3.06↑	0.97↑	2.16↑

the maximum fluctuation CWRU's rotating speed, and the other two are set empirically. For each example, 1024 continuous sample points in the time domain are randomly segmented from the original mat files. The simulated virtual examples are shown in Fig.7. And the first experiment result is shown in TABLE. 3, where the classification accuracy and the corresponding standard deviation are calculated under 10 repeated trials for the 6 diagnosis scenarios.

All the trained models perform well on the validation set of training domain (near to 100%), but as shown in TABLE 3, a collapse occurs when applying them directly to testing domain. Strikingly, we find that DSR boosts the generalization performance of those models' in most of the transferring scenarios, especially for frequency domain based models.

Amongst them, BPNN and CNN with DSR obtain 20.64% and 17.51% increases on the average accuracy. SVM with DSR gains 17.51% higher average accuracy than its original structural risk minimization (SRM) principle based version. In particular, KNN, which can be regarded as a neighbor based probability estimation (NBE) model, achieves the best and most stable performance with DSR, with 95.95% of the average accuracy and 0.81% of the corresponding standard deviation. This phenomenon fully verifies the effectiveness of our proposed DSR, where the vicinal examples generated by DSR improve the diversity of the original training set. Compared with Dirac distribution of ERM, and neighborhood based probability estimation, the vicinal distribution formulated by DSR is a better estimation of the true mechanism, which, thereby, increases the generalization ability of the diagnosis models.

For time domain based diagnosis models, DSR also contributes to an improvement of generalization performance of

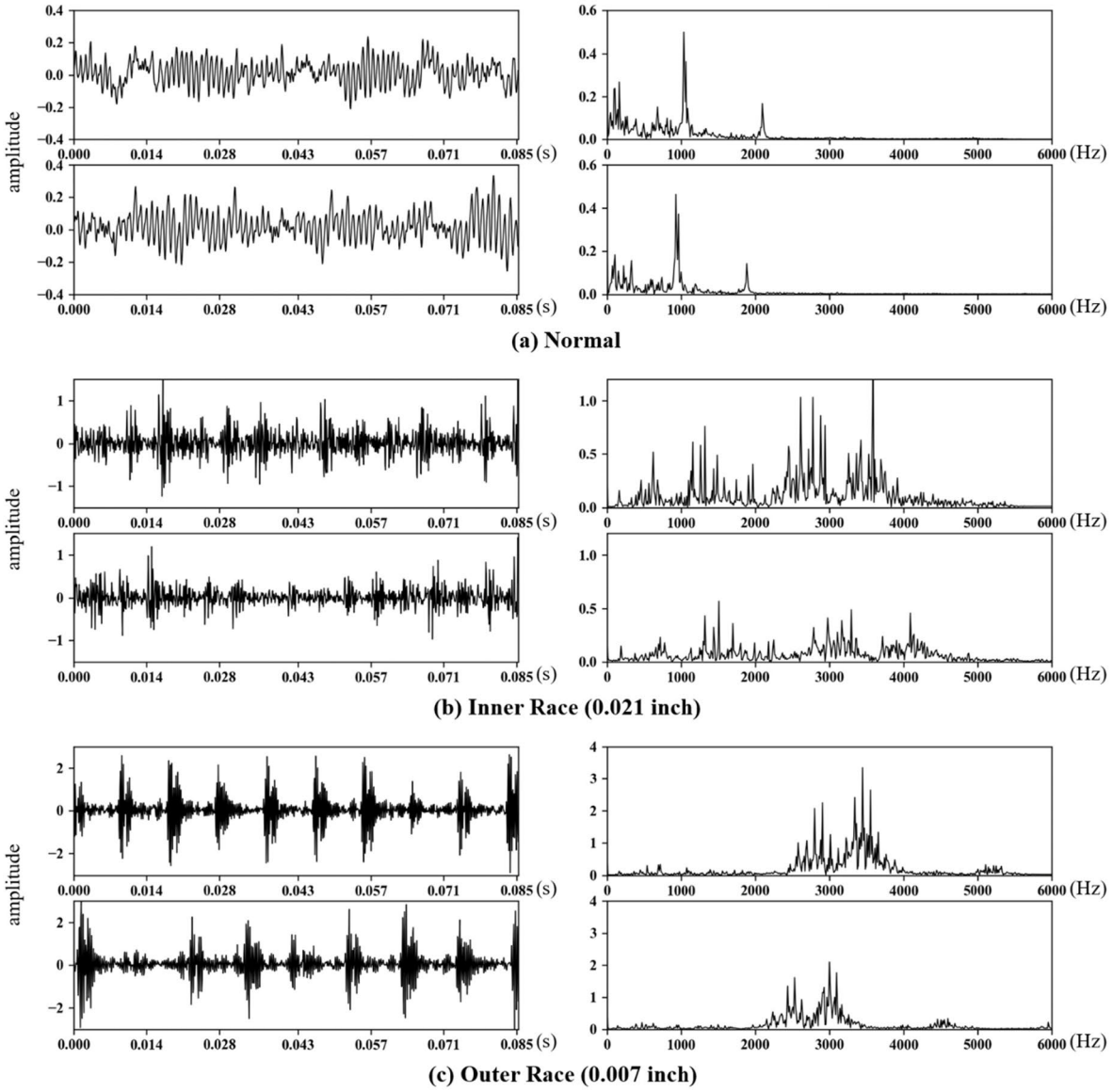


FIGURE 7. The comparison of the original signals and the simulated virtual examples on three classes: (a) normal, (b) inner race fault (0.021 inches), (c) outer race fault (0.07 inches). In each subFIG, the upper left and upper right are the original time sequence and its amplitude spectrum, the lower left and lower right are the virtual ones. Note that for better telling the difference, three pairs of examples with high speed ratios (r=0.89, 1.12, 0.88) are chosen here. Normally the difference between an original signal and its virtual one is not so obvious.

TICNN (0.36%), LeNet-5 (5.86%) and DNCNN (2.16%) but not as much as the one of frequency-based models. The modest increases for TICNN (the first 4 scenarios even decrease) and DNCNN (first 2 decrease) can mainly be attributed to the phase shift in simulated time signals caused by DSR. Specifically, in a relatively easy transferring scenario (i.e. A->B for DNCNN, B->A for TICNN) the unexpected deformation in the corresponding simulated time signals may result in errors in models' generalization process, which leads a decline on performance. But when it comes to more difficult scenarios (i.e. C->A), the vicinal examples generated by DSR can also contribute to models' generalization ability.

In addition, an experiment for few-shot learning scenarios is also conducted, where for the most extreme case, there are only 5 samples for each class utilized for training models. In this experiment, based on the few-shot training set (5/class), we use DSR to simulate vicinal examples to expand it to 2, 5 and 10 times bigger for comparison. The result is shown in TABLE 4, where we present the average accuracy on 6 few-shot diagnosis scenarios under 10 replicate trials.

The calculation in TABLE 4 strikingly shows the effectiveness of DSR for frequency domain based diagnosis models, where, for example, the accuracies of using DSR to expand the training set for 10 times (simulate 9 vicinal examples from

TABLE 4. Average classification accuracy ± average standard deviation on 6 few-shot diagnosis scenarios.

Feature Domain	Frequency Domain				Time Domain		
Method	KNN	SVM	BPNN	CNN	TICNN	LeNet-5	DNCNN
5/class	64.17 ± 4.09	34.10 ± 7.66	71.62 ± 3.65	76.23 ± 2.67	63.57 ± 6.02	63.92 ± 5.44	53.48 ± 6.78
10/class	71.11 ± 2.37	37.77 ± 4.82	71.75 ± 2.94	79.55 ± 2.01	79.01 ± 4.64	79.72 ± 4.40	73.47 ± 4.15
25/class	72.68 ± 1.53	40.17 ± 5.08	69.54 ± 1.70	77.12 ± 8.57	90.41 ± 2.54	85.86 ± 4.23	85.09 ± 4.35
50/class	73.68 ± 1.41	62.40 ± 3.63	71.55 ± 1.77	75.68 ± 6.18	**91.31 ± 4.02**	**88.86 ± 3.17**	**86.78 ± 3.85**
5 + 5 (DSR) → 10/class	77.11 ± 3.92	85.71 ± 4.56	86.26 ± 4.34	**88.88 ± 4.02**	72.67 ± 4.60	72.65 ± 3.90	67.03 ± 3.97
5 + 20 (DSR)→25/class	86.74 ± 3.25	87.93 ± 3.37	91.57 ± 2.50	87.76 ± 5.02	72.31 ± 4.84	71.81 ± 4.51	54.41 ± 4.47
5 + 45 (DSR)→50/class	**92.85 ± 1.76**	**89.73 ± 2.28**	**92.70 ± 1.93**	88.67 ± 3.41	68.50 ± 6.04	68.88 ± 4.55	44.22 ± 5.21

1 original example) overweigh the ones of directly adding extra 45 original examples per class.

As for time domain-based models, DSR can marginally improve the models' generalization performance with limited training examples (5 + 5 (DSR) → 10/class). However, the overuse of the simulated virtual examples in the time domain, especially for few-shot cases, can have an adverse impact on models' generalization performance. It is obvious that the accuracies of models decline gradually as the number of simulated signals increases. Inversely, adding raw training samples (if we have) is much more effective than using DSR. This phenomenon can be mainly owed to the following 3 reasons: a) first, the unexpected pattern brought in by the deformation may mislead the network to optimize in a wrong direction; b) second, as illustrated in Fig.7 (also in Fig.6), the distribution of a simulated signal along time axis resembles its corresponding original one. With more similar vicinal examples fed into deep models, severer overfitting occurs; and c) third, for signals under the same class, their representation in time domain can differ greatly with different time slot for segmentation; with more extra different time segmentations fed for optimization, deep models with plenty of parameters get better generalization performances.

C. ABLATION STUDIES

As a data augmentation algorithm, DSR mainly consists of three parts: main resampling procedures for different rotating speeds simulation, load influence simulation and noise injection. To compare the effect of each part, we conduct an experiment containing adding noise, multiplied by load coefficient, resampling as well as their combinations. Except noise injection [28] (as one part of DSR), we also utilize Mixup [15] as a comparison data augmentation method. Besides, we further discuss the combinations of Mixup and our proposed DSR.

Specifically, the parameters for resampling, noise injection, and load influence are set the same as the first experiment. For Mixup, we follow [15] and set its parameter a=0.2 for Beta distribution. We have two modes for the combination of Mixup and DSR: a) for the Mode-1, we directly calculate the convex combination between random pair of an original training example and a simulated vicinal example (after DSR), where:

$$\begin{cases} x_{input} = \lambda x_i^{(origin)} + (1-\lambda)\, x_j^{(DSR)} \\ y_{input} = \lambda y_i^{(origin)} + (1-\lambda)\, y_j^{(DSR)} \end{cases} \tag{8}$$

and b) for Mode-2, we first concatenate the original examples and simulated vicinal examples, and then compute the convex combination between random pair of examples after concatenation, where:

$$\begin{cases} x_{input} = \lambda x_i^{(concat)} + (1-\lambda) x_j^{(concat)} \\ y_{input} = \lambda y_i^{(concat)} + (1-\lambda) y_j^{(concat)} \end{cases} \tag{9}$$

The experiment result is presented in TABLE 5.

From the ablation study experiment, we have the following observations. First, in the comparison of each part of DSR and their combination, resampling is the core part, where the main contribution of performance improvement of models owes to it. And by combined with noise injection and load influence, DSR can improve the stability of diagnosis models.

Second, in the comparison with noise injection and Mixup, DSR performs the best while the other two may cause generalization performance decline (expect of DNCNN with Mixup, an increase of 6.75%) under working condition transferring diagnosis scenarios, which verifies the adaptability and effectiveness of DSR as a specific data augmentation algorithm for periodic signal based diagnosis.

Third, the combination of DSR and Mixup can effectively improve the generalization ability and stability of models. But for TICNN and DNCNN, whose inputs are 1-D time signals, the direct convex combination of a random pair of a vicinal example (DSR) and an original example can cause a dramatic decline of model performance. This may be caused by the domain shift pattern and the unexpected deformation, which make the model hard to be optimized with the direct convex combination.

D. EFFECTIVENESS VISUALIZATION

Here we aim to demonstrate the effectiveness of the vicinal distribution formulated by the proposed DSR. And we choose

TABLE 5. Average classification accuracy ± average standard deviation for ablation studies.

Feature Domain	Frequency Domain				Time Domain		
Method	KNN	SVM	BPNN	CNN	TICNN	LeNet-5	DNCNN
ERM	74.28 ± 0.91	75.39 ± 0.90	72.94 ± 2.99	72.94 ± 2.99	92.59 ± 2.39	87.47 ± 5.83	86.38 ± 4.68
Noise Injection	74.43 ± 0.90	63.50 ± 0.79	71.43 ± 3.98	75.20 ± 7.45	81.70 ± 5.89	85.59 ± 10.4	86.70 ± 4.31
Load Influence	76.32 ± 1.01	64.23 ± 0.82	72.57 ± 2.42	76.46 ± 6.87	89.38 ± 4.59	85.68 ± 6.40	88.69 ± 3.58
Noise Injection + Load Influence	76.18 ± 1.02	64.55 ± 0.98	72.42 ± 1.90	78.12 ± 6.40	90.17 ± 5.96	86.84 ± 5.19	86.78 ± 4.21
Resampling	95.54 ± 1.04	88.35 ± 1.63	93.86 ± 2.21	90.81 ± 2.56	92.18± 3.57	92.57 ± 4.85	85.90 ± 3.95
Resampling + Noise Injection	95.77 ± 0.85	88.65 ± 1.42	93.52 ± 1.89	90.18 ± 2.89	92.04 ± 3.92	93.02 ± 5.46	87.43 ± 2.67
Resampling + Load Influence	94.75 ± 1.01	91.99 ± 0.87	93.34 ± 2.48	90.74 ± 3.37	91.90 ± 3.24	92.16 ± 4.73	87.11 ± 3.08
DSR	**95.95 ± 0.81**	**92.89 ± 0.98**	93.58 ± 1.86	90.44 ± 2.31	**92.85 ± 1.98**	93.33 ± 2.14	88.54 ± 2.70
Mixup (a=0.2)	-	-	72.50 ± 3.05	66.54 ± 3.66	90.84 ± 2.39	88.50 ± 2.69	**93.13 ± 2.37**
DSR + Mixup (Mode-1) (a=0.2)	-	-	94.36 ± 2.11	92.58 ± 1.59	28.99 ± 9.37	**93.71 ± 2.03**	31.41 ± 13.6
DSR + Mixup (Mode-2) (a=0.2)	-	-	**95.71 ± 0.84**	**94.73 ± 1.50**	90.54 ± 2.66	90.43 ± 1.91	92.26 ± 1.51

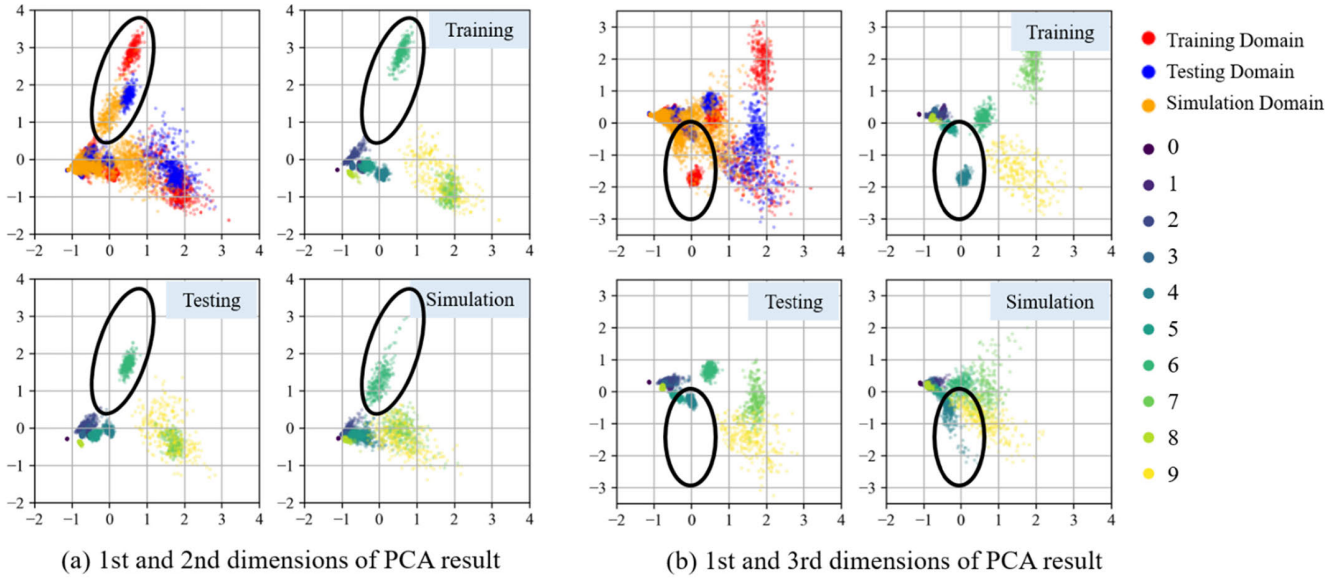


FIGURE 8. PCA visualization of the frequency spectrum of source domain signals, target domain signals and simulated signals based on source domain signals. The area in the ellipse box cites the distribution shift of the same fault in different domains.

the C->A diagnosis scenario, which is the hardest working condition transferring scenario, to conduct the experiment, and the parameters of DSR are set the same as the first experiment. Principle Component Analysis (PCA) is utilized to visualize the distributions of the training domain (C: 3HP), the vicinal domain (DSR) and the testing domain (A: 1HP).

The left-top parts Fig.8 (a) and (b) show the marginal distributions of three domains, and the residual parts show their condition distributions. From the figure, we can observe that compared with the distribution of the training domain. the vicinal distribution is a better estimation for the distribution of the testing domain. For example, in the ellipse boxes, the condition vicinal distribution is closer to the one of the testing domain, where there are more intersection regions between the two distributions. This phenomenon indicates the effectiveness of our proposed DSR as a data augmentation algorithm based on Vicinal Risk Minimization.

V. CONCLUSION

We mainly focus on the data augmentation for inadequate and incomplete training set for fault diagnosis, which is mainly caused by various working conditions and imbalanced distribution of industrial data. This leads us to study a specific augmentation algorithm for periodical signals. And in this paper, we have proposed DSR and its main procedures, and shown that DSR is a form of vicinal risk minimization, where the vicinal examples simulated by it are utilized for training. Throughout an extensive evaluation, we have

verified that DSR improves the generalization performance of diagnosis models, especially for frequency domain based models, on the benchmark CWRU database. Besides, we have further discussed the effect of combining DSR with other data-agnostic augmentation algorithms like Mixup.

The main shortcoming of DSR lies at the limited improvement for time domain based models, which has been summarized in *Section.III.F*: a) the unexpected pattern caused by phase shift, b) the similar distribution along the time axis, c) high speed ratio r can result in a loss of the resolution precision of the simulated signals and d) prior knowledge about fluctuation range, signal-noise ratio, etc., are required. We do not yet find a practical solution to these problems.

In the future, we expect to study practical generalization mechanisms in diagnosis models, which can be combined with DSR and be utilized for online diagnosis under imbalanced circumstance.

REFERENCES

[1] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[2] M. Cerrada, R.-V. Sánchez, C. Li, F. Pacheco, D. Cabrera, J. V. de Oliveira, and R. E. Vásquez, "A review on data-driven fault severity assessment in rolling bearings," *Mech. Syst. Signal Process.*, vol. 99, pp. 169–196, Jan. 2018.

[3] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, vol. 5, pp. 17368–17380, 2017.

[4] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018.

[5] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mech. Syst. Signal Process.*, vol. 110, pp. 349–367, Sep. 2018.

[6] F. Wang, C. Liu, W. Su, Z. Xue, H. Li, and Q. Han, "Condition monitoring and fault diagnosis methods for low-speed and heavy-load slewing bearings: A literature review," *J. Vibroeng.*, vol. 19, no. 5, pp. 3429–3444, 2017.

[7] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Proc. NIPS*, 2000, pp. 416–422.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[9] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. ICLR*, 2017, pp. 1–15.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. ICLR*, 2014, pp. 1–10.

[11] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[12] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Jun. 2013, pp. 2200–2207.

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

[14] P. Simard, Y. LeCun, J. Denker, and B. Victorri, "Transformation invariance in pattern recognition—Tangent distance and tangent propagation," in *Neural Networks: Tricks of the Trade*. 1998.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," Apr. 2018, *arXiv:1710.09412*. [Online]. Available: https://arxiv.org/abs/1710.09412

[16] R. Li and D. He, "Rotational machine health monitoring and fault detection using EMD-based acoustic emission feature quantification," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 4, pp. 990–1001, Apr. 2012.

[17] Z. Huo, Y. Zhang, P. Francq, L. Shu, and J. Huang, "Incipient fault diagnosis of roller bearing using optimized wavelet transform based multi-speed vibration signatures," *IEEE Access*, vol. 5, pp. 19442–19456, 2017.

[18] K. R. Fyfe and E. D. S. Munck, "Analysis of computed order tracking," *Mech. Syst. Signal Process.*, vol. 11, no. 2, 1997, pp. 187–205, 1997.

[19] F. Bonnardot, M. El Badaoui, R. B. Randall, J. Danière, and F. Guillet, "Use of the acceleration signal of a gearbox in order to perform angular resampling (with limited speed fluctuation)," *Mech. Syst. Signal Process.*, vol. 19, no. 4, pp. 766–785, 2005.

[20] Y. Yinghua, S. Guoqiang, and S. Xiang, "Fault monitoring and classification of rotating machine based on PCA and KNN," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Shenyang, China, Jun. 2018, pp. 1795–1800.

[21] A. Wang, M. Sha, L. Liu, and M. Chu, "A new process industry fault diagnosis algorithm based on ensemble improved binary-tree SVM," *Chin. J. Electron.*, vol. 24, no. 2, pp. 258–262, Apr. 2015.

[22] W. Sun, J. Chen, and J. Li, "Decision tree and PCA-based fault diagnosis of rotating machinery," *Mech. Syst. Signal Process.*, vol. 21, no. 3, pp. 1300–1317, 2007.

[23] X. Zhao, X. Tang, J. Zhao, and Y. Zhang, "Fault diagnosis of asynchronous induction motor based on BP neural network," in *Proc. Int. Conf. Measuring Technol. Mechatronics Automat.*, Changsha City, China, Mar. 2010, pp. 236–239.

[24] Q. Tong, J. Cao, B. Han, X. Zhang, Z. Nie, J. Wang, Y. Lin, and W. Zhang, "A fault diagnosis approach for rolling element bearings based on RSGWPT-LCD bilayer screening and extreme learning machine," *IEEE Access*, vol. 5, pp. 5515–5530, 2017.

[25] S. Wang, I. Selesnick, G. Cai, Y. Feng, X. Sui, and X. Chen, "Nonconvex sparse regularization and convex optimization for bearing fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 65, no. 9, pp. 7332–7342, Sep. 2018.

[26] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

[27] W. Sun, R. Zhao, R. Yan, S. Shao, and X. Chen, "Convolutional discriminative feature learning for induction motor fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1350–1359, Jun. 2017.

[28] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mech. Syst. Signal Process.*, vol. 100, pp. 439–453, Feb. 2018.

[29] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "LiftingNet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4973–4982, Jun. 2018.

[30] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.

[31] J. Jiao, M. Zhao, J. Lin, and J. Zhao, "A multivariate encoder information based convolutional neural network for intelligent fault diagnosis of planetary gearboxes," *Knowl.-Based Syst.*, vol. 160, pp. 237–250, Nov. 2018.

[32] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vol. 72, pp. 303–315, May 2016.

[33] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, Jul. 2016.

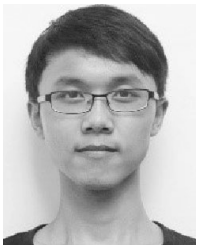
[34] F. Wang, B. Dun, G. Deng, H. Li, and Q. Han, "A deep neural network based on kernel function and auto-encoder for bearing fault diagnosis," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, Houston, TX, USA, May 2018, pp. 1–6.

[35] G. Jiang, H. He, P. Xie, and Y. Tang, "Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2391–2402, Sep. 2017.

[36] Y. Qi, C. Shen, D. Wang, J. Shi, X. Jiang, and Z. Zhu, "Stacked sparse autoencoder-based deep network for fault diagnosis of rotating machinery," *IEEE Access*, vol. 5, pp. 15066–15079, 2017.

[37] Z. Huijie, R. Ting, W. Xinqing, Z. You, and F. Husheng, "Fault diagnosis of hydraulic pump based on stacked autoencoders," in *Proc. 12th IEEE Int. Conf. Electron. Meas. Instrum. (ICEMI)*, Qingdao, China, Jul. 2015, pp. 58–62.

[38] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle, S. Van Hoecke, "Convolutional neural network based fault detection for rotating machinery," *J. Sound Vib.*, vol. 377, pp. 331–345, Sep. 2016.
[39] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," 2017, *arXiv:1708.04896*. [Online]. Available: https://arxiv.org/abs/1708.04896
[40] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*. [Online]. Available: https://arxiv.org/abs/1412.5567
[41] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. ICML*, 2016, pp. 173–182. [Online]. Available: https://arxiv.org/abs/1512.02595
[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015, *arXiv:1512.00567*. [Online]. Available: https://arxiv.org/abs/1512.00567
[43] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017, *arXiv:1701.06548*. [Online]. Available: https://arxiv.org/abs/1701.06548
[44] P. D. McFadden and J. D. Smith, "Model for the vibration produced by a single point defect in a rolling element bearing," *J. Sound Vib.*, vol. 96, no. 1, pp. 69–82, 1984.
[45] *Case Western Reserve University Bearing Data Center*. [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980



TIANHAO HU received the B.S. degree in mechanical engineering from the School of Mechanical Engineering, Tongji University, Shanghai, China, in 2017, where he is currently pursuing the M.S. degree in mechanical design and theory.

His current research interests include rotating machinery fault diagnosis problems and computer vision.

TANG TANG received the B.S. degree from Tongji University, Shanghai, China, in 2005, the M.S. degree from the Technische Universität München, Munich, Germany, in 2008, and the Ph.D. degree from the Technische Universität Dresden, Dresden, Germany, in 2015, all in electronic and information engineering.

Since 2015, he has been an Assistant Professor with the School of Mechanical Engineering, Tongji University, Shanghai. His current research interests include machinery condition monitoring, fault diagnosis, and deep learning.

MING CHEN received the B.S. degree in engineering machinery, the M.S. degree in mechanical design and theory, and the Ph.D. degree in mechanical manufacturing and automation from the School of Mechanical Engineering, Tongji University, Shanghai, China, in 1987, 1990, and 2006, respectively, where he is currently a Professor with the School of Mechanical Engineering and the Director of Industry 4.0-Smart Factory Lab.

His current research interests include structure and framework in industry 4.0, PLM in industrial big data, and development of MES system in the industrial sensor networks.