A One-Class Generative Adversarial Detection Framework for Multifunctional Fault Diagnoses

Ziqiang Pu, Diego Cabrera, *Member, IEEE*, Yun Bai, *Member, IEEE*,
and Chuan Li, *Senior Member, IEEE*

***Abstract*—In this article, fault diagnosis is of great significance for system health maintenance. For real applications, diagnosis accuracy suffers from unbalanced data patterns, where normal data are usually abundant than anomaly ones, leading to tremendous diagnosis obstacles. Therefore, it is challenging to use only normal data for fault diagnosis under this imbalanced condition. In addition, a single fault diagnosis model can only conduct one fault diagnosis task in most of cases. Accordingly, a one-class generative adversarial detection (OCGAD) framework based on semisupervised learning is proposed to learn one-class latent knowledge for dealing with multiple semisupervised fault diagnosis tasks, i.e., fault detection using only normal knowledge learning, novelty detection from unknown conditional data, and fault classification with unlabeled data. A bi-directional generative adversarial network (Bi-GAN) is first trained with only normal data. A one-class support vector machine is then established using features exacted by Bi-GAN from signals acquired from an attitude sensor for multifunctional fault detection. The presented OCGAD model is validated using an industrial robot with experiments of three fault detection tasks. The results demonstrate that the present model has good performance for dealing with multiple semisupervised diagnosis problems.**

***Index Terms*—Fault diagnosis, latent knowledge, one-class generative adversarial detection (OCGAD), semisupervised learning.**

I. Introduction

As a branch of the fault diagnosis, anomaly detection [1] has been applied to handle the problem of finding outliers in data that do not conform to the expected behavior [2]. However, these outliers or anomalies are common to cause damage in the manufacturing systems, such as industrial robots. Condition monitoring systems are helpful in detecting anomalies and reducing maintenance costs [3]. Therefore, anomaly detection plays an important role in the industrial machinery fault diagnosis.

Different intelligent fault diagnosis [4] models have been reported by using support vector machine (SVM) [5], random forest [6], K-nearest neighbor [7], Bayesian network [8], autoencoder (AE) [9], and convolutional neural network (CNN) [10]. Cabrera *et al.* [11] proposed a group of long short-term memory models with time series dimensionality reduction for the compressor fault diagnosis. Han *et al.* [12] proposed a fault detection method using least square SVM with cross-validation optimization on chillers. Wang *et al.* [13] developed an ensemble extreme learning machine for the fault diagnosis of rotating machinery. Long *et al.* [14] proposed a sparse echo AE for the fault diagnosis of delta three-dimensional printers by using attitude data. Gong *et al.* [15] developed an improved CNN using global pooling technology with SVM for the fault diagnosis of the rotating machinery.

Besides supervised fault diagnosis models, there are still unsupervised or semisupervised diagnosis tasks, such as anomaly detection with normal data, novelty detection from unknown pattern data, and fault classification with unlabeled data. Otherwise, the anomaly detection only observes outliers without considering the type of classification. Some works using one-class SVM (OCSVM) [16], [17] have been reported for machinery anomaly detection. Fiore *et al.* [18] used a discriminative restricted Boltzmann machine integrated with generative models for anomaly detection. Chen *et al.* [19] presented an unsupervised convolutional variational AE for the anomaly detection of an industrial robot. As for the novelty detection [20], extra observations would be added for detecting outliers from unknown data. Amarbayasgalan *et al.* [21] developed a deep AE with a density-based cluster for the novelty detection with outlier detection datasets. Feng *et al.* [22] proposed a novelty detection approach based on curvelet transform, nonlinear principal component analysis, and SVM to indicator diagram diagnosis. For the classification with unlabeled data, unsupervised learning with clustering can solve this problem. Webtao *et al.* [23] developed a new fault diagnosis

Manuscript received February 28, 2021; revised May 24, 2021 and July 20, 2021; accepted August 17, 2021. Date of publication September 3, 2021; date of current version March 1, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 52175080, Grant 51775112, and Grant 71801044, in part by Chongqing Natural Science Foundation under Grant cstc2019jcyj-zdxmX0013, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2019B1515120095, and in part by Intelligent Manufacturing PHM Innovation Team Program under Grant 2018KCXTD029 and Grant TDYB2019010. *(Corresponding author: Ziqiang Pu.)*

Ziqiang Pu is with the College of Mechanical and Electrical Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China (e-mail: a66835@ualg.pt).

Diego Cabrera is with GIDTEC Group, Universidad Politécnica Salesiana, Cuenca 010105, Ecuador (e-mail: dcabrera@ups.edu.ec).

Yun Bai is with the Faculty of Science and Technology, University of Algarve, 8005-139 Faro, Portugal (e-mail: ybai@ualg.pt).

Chuan Li is with the College of Mechanical and Electrical Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China, and also with the Research Center for System Health Maintenance, Chongqing Technology and Business University, Chongqing 400067, China (e-mail: chuanli@ctbu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3108719.

Digital Object Identifier 10.1109/TIE.2021.3108719

method based on feature weighted fuzzy clustering model. Yu *et al.* [24] introduced a cluster-based feature extraction approach by employing a discrete wavelet transform and probabilistic neural network for the machine fault diagnosis.

The rest of this article is organized as follows. The motivation of this article and proposed multifunctional model are developed in Section II. In Section III, experiments on an industrial robot are introduced. Results and comparisons with different fault diagnosis methods are detailed in Section IV. Finally, Section V concludes this article.

II. Methodology

A. Theoretical Background

Most of clustering techniques were reported to detect known classes instead of unknown ones. Moreover, these fault diagnosis methods were developed to deal with single task instead of multiple tasks. In addition, it is a tough problem to obtain faulty data instead of normal ones [25]. Consequently, these intelligent diagnosis approaches cannot precisely conduct fault diagnosis tasks due to the shortage of faulty data. Thus, the fault diagnosis accuracy is deteriorated.

To migrate the data imbalance problem, most researchers used data augmentation techniques [26]–[28] to oversampling the dataset. Chawla *et al.* [29] used a synthetic minority oversampling technique (SMOTE) algorithm to randomly synthesis artificial data for solving the data imbalance problem, and He *et al.* [30] reported an improved SMOTE algorithm. Recently, a generative adversarial network (GAN), which was first proposed by Goodfellow *et al.* [31], is a novel way for generating data. GAN is consisting of a generator and a discriminator for adversarial learning. It was first applied in image recognition [32]–[34] and later widely used in industrial machinery for fault detection with imbalanced data [35]–[37]. Although there are some tools solving imbalance problems for diagnosis data, it is still a challenge to detect failures with only normal data. In the past, Donahue *et al.* [38] proposed a variant framework of GAN for adversarial feature learning named bi-directional GAN (Bi-GAN) including a generator, an encoder, and a discriminator. The encoder tries to map the original data into its latent space for capturing important features of original data while the generator tends to generate synthetic data from a random distribution. Thus, Bi-GAN can obtain both the encoder for data mapping and the generator for data generation through adversarial learning.

Considering multiple diagnosis tasks and data acquisition problems (only having enough normal data in the dataset), this article proposed a one-class generative adversarial detection (OCGAD) framework using only normal class to address the multifunctional fault diagnosis tasks. This multifunctional framework was combined with both Bi-GAN and OCSVM. The feasibility of the presented framework was verified by an industrial robot dataset. Main contributions of this work were as follows: first, a novel OCGAD framework was developed using only normal data for training to conduct fault diagnosis; and second, the proposed training strategy could deal with at least three fault diagnosis tasks including fault detection with only normal data, novelty detection from unknown conditional data, and fault classification of unlabeled data.

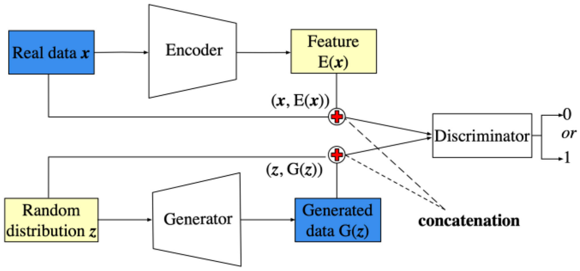


Fig. 1. Structure of a Bi-GAN.

B. Bi-GAN for Latent Knowledge Learning From Raw Data

GAN is an effective framework for learning complicated sample distribution to generate samples. This framework has a generator to produce samples from a random distribution, e.g., Gaussian distribution. At the same time, a discriminator is applied to distinguish whether an instance is from real or generated samples. The goal of the generator is to convince the discriminator by producing synthetic samples in which the generated distribution is closer to real distribution. GAN is optimized by an adversarial *min–max* game between the generator and the discriminator, given by

$$\begin{aligned}\min maxV\ (G,D) = E_{x\sim data}\ [\log\left(D\left(x\right)\right)] \\ + E_{x\sim P_G}\left[\log\left(1-D\left(G\left(z\right)\right)\right)\right] \quad (1)\end{aligned}$$

where x stands for real data, $G(z)$ is the synthetic data, $D(x)$ is the output of the discriminator, $D(G(z))$ is the output of the discriminator corresponding to the synthetic data as the input, $E_{x\sim \mathrm{data}}[\cdot]$ stands for the expectation of real distribution $P_x(x)$, and $E_{x\sim P_G}[\cdot]$ denotes as the expectation of random distribution $P_G(x)$.

With Kullback–Leibler divergence-based loss function [39] to measure the distance between real distribution $P_x(x)$ and generated distribution $P_G(x)$, GAN searches its hyper-parameter through stochastic gradient descent (SGD) [40]. Thus, a trained GAN can generate more samples for solving the data imbalance problem (samples in the faulty condition are more difficult to be acquired than those in the normal condition) in fault diagnosis. Besides, feature learning tools are performed in most of the ways to project the sample from high dimension into its low dimension. This kind of process can reduce samples' dimension, helping the model with fast convergence and low computational burden for training.

Considering the difficulty for faulty data collection where normal data are more common than faulty ones, it remains an open problem for the anomaly detection to classify the fault by using only normal data. In addition, with a tough problem for modeling using the aforementioned raw data, Bi-GAN model was applied by using only the normal data for training a mapping function to low dimensionality data for each class in this work. Fig. 1 illustrates the structure of a Bi-GAN. Different from the

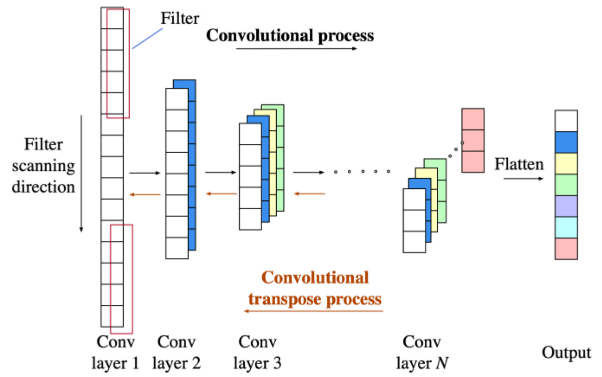


Fig. 2. Schematic of the convolutional scanning process.

traditional GAN where the discriminator only takes ***x*** or G(*z*) as input, the discriminator of Bi-GAN employs *x* and its feature space E(***x***) or *z* and the generated G(*z*) as the input. On the one hand, the real data *x* is given from real distribution *p*(*x*) and an encoder is used to project the data *x* into its feature space. On the other hand, the random distribution *z* is set as the input to generate synthetic data G(*z*). Note that the same color in Fig. 1 represents the same dimension. This conforms to the so-called "bi-direction". In addition, the joint (the red dashed line as shown in Fig. 1) means that these two types of data are concatenated for the input to the discriminator. Both the encoder and decoder are neural networks. Therefore, the architecture of the neural network is a vital point for model training.

GAN-based models often suffer from model collapse, i.e., the model cannot completely learn the distribution from the original data [41]. Hence, GAN-based models struggle with training. To enhance the training process, Radford *et al.* [42] proposed a deep convolutional GAN (DCGAN), and experiments demonstrated that the present DCGAN with CNN structure in networks was a powerful tool for unsupervised feature learning. It increased the convergence performance. Therefore, CNN structure was used in the Bi-GAN to develop the encoder, generator, and discriminator in this article.

The basic convolutional and transpose scanning process is shown in Fig. 2. At the first convolutional layer, the filter is moved from the top to the bottom. Second, the dot product is conducted in each scanning movement, which is called stride. For instance, with one stride the filter will move one block. After the dot product, the value in each block will sum up to a total value with biases that are represented as the input for the next convolutional layer. Finally, after several convolutional processes, all the acquired vectors are flattened into one vector which is the output of the convolutional process.

Let the raw data in normal condition is $x = \{x^1, x^2, x^3, \ldots, x^m\} \in R^m$. The convolutional layer in the encoder involves all the inputs with convolutional filter kernels and then followed by the activation unit to generate the output for the next convolutional layer. This can be formulated by

$$x^{i'(l)} = \sum\nolimits_{i=1}^{m} \left(k^{ii'(l)} * x^{i(l-1)}\right)_{(k_s,s)} + b^{i'(l)} \qquad (2)$$

where k stands for the convolution filter kernel, $b^{i'(l)}$ is the lth bias, l is the number of layers in the network, $i = \{12, 3, \ldots, m\}$ is the index of the input dimensions, $i' = \{12, 3, \ldots, m'\}$ is the index of the output dimensions, "$*$" is the dot product, k_s is the kernel size of k, and s is the stride option that the kernel will move every *s* steps on the input data. In addition, the batch normalization technique is used in every layer for helping the model training shown as follows:

$$\bar{x}^{i'(l)} = \alpha \frac{x^{i(l)} - \mu}{\sqrt{\sigma^2 - \theta}} + \gamma \qquad (3)$$

where μ denotes mean, σ^2 is the variance of the input $\bar{x}^{i'(l)}$, α and γ are the scaling and shifting parameters, and θ is a constant close to zero. The batch normalization is applied to the inputs of each convolutional layer except the output layer of the encoder and generator to avoid adding extra variance to the output of CNN.

Generally, after building the model with CNN, the mapping relationship in the generator and encoder can be concluded as follows:

$$s = E(x) \qquad (4)$$

$$\tilde{x} = G(z) \qquad (5)$$

where $z \in R^n$ stands for random distribution, $\tilde{x} \in R^m$ denotes fake data, and $s \in R^n$ is the latent knowledge of the input x. It can be found that the encoder and the generator have symmetric structures. For the discriminator, it has the same architecture as the encoder except for the input and the output. The input for the discriminator *D* is defined as follows:

$$logit = \begin{cases} D(h) \\ D\left(\tilde{h}\right) \end{cases} \; h \; and \; \tilde{h} \in R^{m+n} \qquad (6)$$

where $h = (x, s)$ and $\tilde{h} = (z, \tilde{x})$ stand for the combined vectors. $logit$ stands for the output of the discriminator. It is a binary classification with output 1 or 0. Therefore, differing from (1), the objective function of Bi-GAN is recast as follows:

$$\min max V \;(G, D, E) = E_{x\sim data} \left[\log D(x, E(x))\right] + E_{x\sim P_G} \left[\log\left(1 - D(D(z), z)\right)\right]. \qquad (7)$$

Adam optimizer [43] is used to substitute the traditional SGD method for the optimization with learning rate $= 2\times10^{-4}$, and training steps = 1000. The optimized discriminator, generator, and encoder can be solved as follows:

$$\theta_E^*, \theta_G^*, \; \theta_D^* = \min max \, V(G, D, E) \qquad (8)$$

where θ_E^*, θ_G^*, and θ_D^* are the collections of optimized weights and bias in the encoder, generator, and discriminator, respectively.

For the sake of getting the low-dimensional data, the encoder is taken out with optimized weights and biases θ_G^* to project the original data (from both normal and faulty conditions) into its feature spaces. Finally, the obtained latent knowledge will be fed to OCSVM for anomaly detection. This will be illustrated in the following section.

C. OCSVM for Anomaly Detection With Latent Knowledge

The goal of SVM is to find an optimal hyperplane by minimizing an upper bound of the generalization error and maximizing the distance between the separating hyperplane and the data. Therefore, the objective of SVM is shown as follows:

$$min\ \frac{1}{2}||w||^2 + C\sum_{i\,=\,1}^{N}\xi_i \tag{9}$$

s.t.

$$\sum_{i,j\,=\,1}^{N} y_i\left(w^T k\left(s_i, s_j\right) + b\right) \geq 1 - \xi_i,\ > 0 \tag{10}$$

where s is the input data from (4), ξ_i is the slack variable that allows some data points to lie within the margin, and $C>0$ determines the tradeoff between maximizing the margin and the number of training data points within that margin. $k(s_i, s_j)$ denotes the kernel function that can be shown in the following:

$$k\ \left(s_i, s_j\right) = \exp\left(-\frac{\tau||s_i - s_j||^2}{2\sigma}\right) \tag{11}$$

where σ denotes a kernel parameter, $||s_i - s_j||^2$ is L_2 norm, and τ is a hyperparameter for kernel function. Unlike the classical SVM, OCSVM [44] is designed to find unusual events or clean database and to distinguish typical examples from the observation of input data. It builds a decision boundary for the hyperplane with the maximum margin between the normal data and outliers. Thus, the objective is recast as follows:

$$min\ \frac{1}{2}||w||^2 + \frac{1}{\nu N}\sum_{i\,=\,1}^{N}\xi_i - \rho \tag{12}$$

s.t.

$$\sum_{i,j\,=\,1}^{N} w^T k\left(s_i, s_j\right) \geq \rho - \xi_i \text{and} \sum_{i\,=\,1}^{N}\xi_i > 0 \tag{13}$$

where ν is the constrain variable for OCSVM. After getting the optimal w^* and ρ^* through Quadratic programming, the decision function rule for OCSVM can be written as follows:

$$f\ (x) = \ sgn\left(\sum_{i=1}^{N} w\phi\left(x_i\right) - \rho\right) = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\phi\left(x_i\right)\phi\left(x_j\right) - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\phi\left(x_i\right)\phi\left(x_j\right). \tag{14}$$

D. Model Development and Training Strategies

The proposed OCGAD model is related to both Bi-GAN and OCSVM. To illustrate, a dataset is split as follows:

$$\boldsymbol{x}\ = \boldsymbol{x^t}\ + \boldsymbol{x^T} \tag{15}$$

where $\boldsymbol{x^t} = \{x_1^t, x_2^t, x_3^t \ldots, x_n^t\}$ is the training set belonging to the class of $\{C_1^t, C_2^t, C_3^t, \ldots, C_n^t\}$ and $\boldsymbol{x^T} = \{x_1^T, x_2^T, x_3^T \ldots, x_n^T\}$ is the testing set belonging to $\{C_1^T, C_2^T, C_3^T, \ldots, C_n^T\}$. Besides, n stands for the class number.

For the training of Bi-GAN, the normal data $x_1^t \in \{C_1^t\}$ are sent into the model for unsupervised learning. A trained encoder is then taken out for mapping the data x_j^t and x_j^Tinto its feature space s_j^t and s_j^T ($j = 12,3,\ldots,n$) to get the low-dimensional latent knowledge based on (4).

Finally, these low-dimensional data are used in the OCGAD model for dealing with the following three fault diagnosis missions:

1) anomaly detection with only normal data;
2) novelty detection from unknown conditional data;
3) fault classification of unlabeled data, respectively.

To this end, different training strategies are suggested for three missions.

1) *Training strategy for the anomaly detection with only normal data*

To train OCGAD model for the anomaly detection with only normal data, $s_1^t \in \{C_1^t\}$ is first fed to OCSVM using (12) and (14) to learn a decision boundary given in (14). The testing set are subsequently evaluated as given by

$$\boldsymbol{s^T} \in \left\{C_1^T, C_F^T\right\} = \left\{C_1^T, C_2^T, \ldots, C_m^T, C_{m+1}^T, \ldots, C_n^T\right\} \tag{16}$$

where $\boldsymbol{s^T}$ is the low-dimensional testing dataset, C_F^T ($F>1$) stands for the fault classes. Therefore, the present model is capable of distinguishing faulty conditions from normal ones, which is represented by

$$h_1(Y_1|Y_2)\begin{cases} Y_1 = \{\hat{c}_1\} \\ Y_2 = \{\hat{c}_F\} & = \{\hat{c}_2, \hat{c}_3, \ldots, \hat{c}_n\}\end{cases} \tag{17}$$

where $h_1(Y_1|Y_2)$ represents the decision boundary to separate Y_1 and Y_2 with the first strategy, $Y_1 = \{\hat{c}_1\ \}$ denotes the estimated value of normal class, and $Y_2 = \{\hat{c}_F\}$ is the estimated value of the faulty class.

2) *Training strategy for the novelty detection from unknown conditional data*

Being different from the previous task, the novelty detection is capable of detecting extra class, which does not belong to the existing classes. For this reason, the input for the OCGAD model can be given by

$$s^t \in \left\{C_1^t, C_F^t\right\} = \left\{C_1^t, C_2^t, \ldots, C_m^t\right\}. \tag{18}$$

Upon finished the OCSVM training, the input for the testing dataset is same as (16). Therefore, the proposed model has the ability to separate the existing classes and novelty class, which is given by

$$h_2\ (Y_1 Y_2) = \begin{cases} Y_1 = \{\hat{c}_e\} = \{\hat{c}_1, \hat{c}_2, \hat{c}_3, \ldots, \hat{c}_m\} \\ Y_2 = \{\hat{c}_n\} = \{\hat{c}_{m+1}, \hat{c}_{m+2}, \ldots, \hat{c}_n\}\end{cases} \tag{19}$$

where $h_2(Y_1 Y_2)$ stands for the decision boundary to separate $Y_1$and $Y_2$with second strategy, $Y_1 = \{\hat{c}_e\}$ represents the estimated value of exist class, and $Y_2 = \{\hat{c}_n\}$ stands for the estimated value of novelty class.

1) *Training strategy for the fault type classification of unlabeled data*

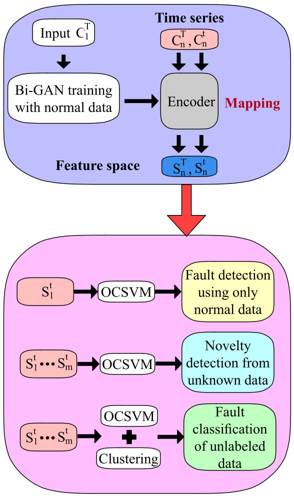


Fig. 3. Proposed OCGAD model for multifunctional fault diagnosis missions.

In response to the third task, the same training procedure and input given in (18) can be employed and perform clustering to classify unlabeled data. Euclidean distance is a good metric for measuring the distance between two classes. It is therefore applied to the clustering in this work. Thus, based on the second strategy, the detected classes are clustered through Euclidean distance. With this extra training strategy, the model is capable of not only distinguishing the existing class, but also separating the unlabeled class. This is given by

$$h_3\ (Y_1 Y_2) = \begin{cases} Y_1 = \{\hat{c}_e\} = \{\hat{c}_1, \hat{c}_2, \hat{c}_3, \ldots, \hat{c}_m\} \\ Y_2 = \{\hat{c}_m\} \\ Y_3 = \{\hat{c}_{m+1}\} \\ \ldots \\ Y_n = \{\hat{c}_n\} \end{cases} \tag{20}$$

where $Y_2 = \{\hat{c}_m\}$, $Y_3 = \{\hat{c}_{m+1}\}, \ldots, Y_n = \{\hat{c}_n\}$ stand for the estimated values of unlabeled class.

Therefore, following (17), (19), and (20), the proposed OCGAD model has the ability to realize multiple fault diagnosis tasks.

E. Application to Multifunctional Fault Diagnosis Missions

Having described all the modeling components and training strategies, the whole procedure of the proposed OCGAD model is illustrated in Fig. 3 and is specified as follows.

Step 1: Collect normal data C_1^t for training Bi-GAN model following (2)–(8).

Step 2: Take out the encoder from the trained Bi-GAN model and map the data $\{\mathrm{C}_n^T, \mathrm{C}_n^t\}$ into its feature space $\{\mathrm{S}_n^T, \mathrm{S}_n^t\}$ (*n* = 12, 3, …, *n*) defined in (4).

Step 3: Perform one of the multifunctional diagnosis tasks by

Substep 3.1: Train OCSVM by following (12) and (13) with normal data $\{\mathrm{S}_1^t\}$ to perform the anomaly detection. Go to Step 4.

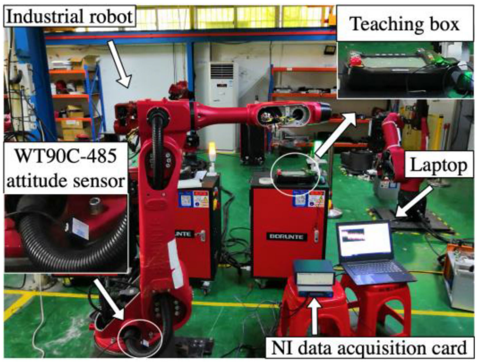


Fig. 4. Test-rig for the industrial robot fault diagnosis.

Substep 3.2: Train OCSVM by following (12) and (13) with inputting known classes $\{\mathrm{S}_1^t, \mathrm{S}_2^t, \mathrm{S}_3^t, \ldots, \mathrm{S}_m^t\}$ [see (18)]. Perform the novelty detection task. Go to Step 4.

Substep 3.3: Following same procedure as Substep 3.2 for training OCSVM. Then, when unknown classes are detected as anomalies, perform clustering for the classification of these unlabeled class. Go to Step 4.

Step 4: Feed the testing data into the trained OCGAD model by using (16) to perform one of the fault diagnoses tasks.

Step 5: Fault diagnosis results of (17), (19), or (20) can be acquired related to Substeps 3.1, 3.2, or 3.3, respectively. End.

III. Experiments

A. Experimental Set-Up

The performance of the addressed OCGAD model was evaluated on an industrial robot as shown in Fig. 4. The experimental set-up contained a six-degree-of-freedom industrial robot, a WT901C-485C attitude sensor, a DELL laptop, a teaching box, and a data acquisition system. Experiments were carried out to diagnose the biggest transmission joint of the robot. The transmission joint to be diagnosed included a two-stage planetary gearbox and a fixed-axis gear shaft. The robot was driven by six motors, with the guidance from the teaching box for its movement. It should be noted that WT901C-485C is a low-cost attitude sensor collecting low-quality signals from the experimental set-up. The reason of using such a low-cost sensor is to highlight the performance of the present approach on compensating the shortcoming of the acquired low-quality raw data. The attitude sensor collected 3-axis vibration, angle, angular velocity, and magnitude signals. Therefore, 12 channels of raw data were obtained.

At the beginning of the experiment, the robot was at its resting place. Then, its axes were moved back and forth following a predefined locus. At last, the robot returned to the resting place. This series of dynamic movement formed one experiment process. In the next step, the faulty part (shown in Table I) was subsequently replaced by the corresponding counterpart of the previous experiment to restart the above movement for the next experiment. During the experiments, all the signals in all channels were collected by the data acquisition system.

TABLE I
HEALTH CONDITIONS

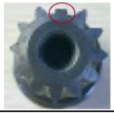
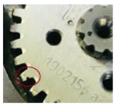
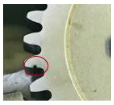
Pattern	C_1	C_2	C_3	C_4
Condition	Normal	Broken tooth in sun gear A	Cracking in planetary gear A	Cracking in planetary gear B
Fault position	N/A			

TABLE II
ARCHITECTURE SETTINGS FOR BI-GAN

Part	Input layer	Kernel size	Stride/ padding	Output shape	Batch normalization	Activation
G	Input	/	/	1 × 100	Decay 0.9 Epsilon 0.001	Leaky rectified linear unit (ReLU) with leaky rate 0.2
	Layer 1	2	2/valid	2 × 512		
	Layer 2	2	2/same	4 × 256		
	Layer 3	3	2/valid	9 × 128		
	Layer 4	3	2/valid	19 × 64		
	Layer 5	4	2/same	38 × 32		
	Layer 6	4	2/valid	78 × 16		
	Layer 7	2	2/same	156 × 8		
	Layer 8	4	2/same	312 × 4		
	Layer 9	3	2/valid	625 × 2		
	Layer 10	4	2/same	1250 × 1		
E	Input	/	/	1250 × 1	Decay 0.9 Epsilon 0.001	Leaky ReLU with leaky rate 0.2
	Layer 1	4	2/same	625 × 2		
	Layer 2	4	2/same	313 × 4		
	Layer 3	3	2/valid	156 × 8		
	Layer 4	4	2/ same	78 × 16		
	Layer 5	3	2/valid	38 × 32		
	Layer 6	4	2/same	19 × 64		
	Layer 7	3	2/valid	9 × 128		
	Layer 8	3	2/same	5 × 256		
	Layer 9	2	2/same	3 × 512		
	Layer 10	4	2/valid	1 × 100		
D	Input	/	/	1250 × 3	Decay 0.9 Epsilon 0.001	Leaky ReLU with leaky rate 0.2
	Layer 1	4	2/same	625 × 2		
	Layer 2	4	2/same	313 × 4		
	Layer 3	3	2/valid	156 × 8		
	Layer 4	4	2/same	78 × 16		
	Layer 5	3	2/valid	38 × 32		
	Layer 6	4	2/same	18 × 64		
	Layer 7	3	2/valid	9 × 128		
	Layer 8	3	2/same	5 × 256		
	Layer 9	2	2/same	3 × 512		
	Layer 10	3	2/valid	1 × 1		

In the cases with Bi-GAN approach, the architecture parts of generator (*G*), encoder (*E*), and discriminator (*D*) were all with the same configuration as described in Table II.

The acquired data were then analyzed for diagnosing its health condition by the laptop.

All data related to fault conditions described in Table I were collected under a sampling duration of 250 s with a sampling rate of 100 Hz, which is the maximum sampling rate for WT901C-485C attitude sensor. In this way, for each channel, 25 000 points of the signal were obtained for each fault pattern. To develop the dataset, the first 1250 points were selected from the raw data as the first sample. Then, an indicator was moved 10 points forth to find the second 1250 points to form the second sample. By repeating 2376 times, 2376 samples were obtained. In the experiments, each test was repeated 10 times. Among them, three times of experiments were randomly chosen for modeling. The first two experiments were used for training with 7128 samples, and the rest one for testing with 2376 samples.

The data were split in vibration channel, angular channel, angle channel, and magnitude channel to train OCGAD model as shown in Fig. 3. Trained encoders corresponding to four channels were then used to map these data into its low-dimensional space. All of 12 channels were subsequently reconstructed as one vector to be used in the OCGAD for three fault diagnosis missions, i.e.,

1) anomaly detection using only normal data for learning;
2) novelty detection from unknown conditional data;
3) fault classification of unlabeled data.

B. Comparison Methods and Network Settings

To evaluate the effectiveness of the proposed OCGAD, 12 different state-of-the-art methods were employed for comparisons. These models included one-class learning of SVM (denoted as OCSVM); one-class learning of independent forest (denoted as OCiF); Bi-GAN with OCSVM in the 3-axis vibration channels (denoted as OCGAD1); Bi-GAN with OCSVM in the 3-axis angle channels (denoted as OCGAD2); Bi-GAN with OCSVM in the 3-axis angular velocity channels (denoted as OCGAD3); Bi-GAN with OCSVM in the 3-axis magnitude channels (denoted as OCGAD4); Bi-GAN with one-class independent forest (denoted as BGOiF); Bi-GAN with one-class independent forest in the 3-axis vibration channels (denoted as BGOiF1); Bi-GAN with one-class independent forest in the 3-axis angle channels (denoted as BGOiF2); Bi-GAN with one-class independent forest in the 3-axis angular velocity channels (denoted as BGOiF3); Bi-GAN with one-class independent forest in the 3-axis magnitude channels (denoted as BGOiF4), and DCAE with one-class OCSVM (denoted as OCDCAE), respectively.

IV. RESULTS AND DISCUSSION

In this section, the diagnosis results of different approaches introduced in Section III were given for the same industrial robot dataset. The three different missions were separately detailed in the following sections.

A. Fault Detection Using Only Normal Data for Learning

One of the challenging tasks was to employ only health data to monitor anomaly behavior. To crack this nut, only data collected from the normal condition C_1 were used to build a fault detection model of the industrial robot. Therefore, the first fault diagnosis case was to detect the outliers with only normal data for model training. While in the testing stage, a large amount of data (including C_1, C_2, C_3, and C_4 in this work) were randomly sent to the model for testing. When the process started, the model will give the normal data 1 while give −1 to the anomalies.

Fig. 5 shows the accuracy using the different approaches, which were listed in Section III. To simplify x labels of Fig. 5, it used numbers 1–13 for representing the candidate models. The results were 20.28% for OCSVM, 64.71% for OCiF, 97% for OCGAD, 55.32% for OCGAD1, 30.43% for OCGAD2, 94.18% for OCGAD3, 78.95% for OCGAD4, 25% for BGOiF, 25% for BGOiF1, 26.13% for BGOiF2, 26.67% for BGOiF3, 25.57% for BGOiF4, and 46.76% for OCDCAE, respectively. Therefore, the

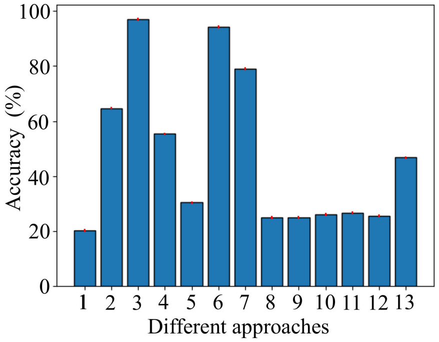


Fig. 5. Average accuracy for one normal class learning, where approaches 1–12 stand for OCSVM, OCiF, OCGAD, OCGAD1, OCGAD2, OCGAD3, OCGAD4, BGOiF, BGOiF1, BGOiF2, BGOiF3, BGOiF4, and OCDCAE, respectively.

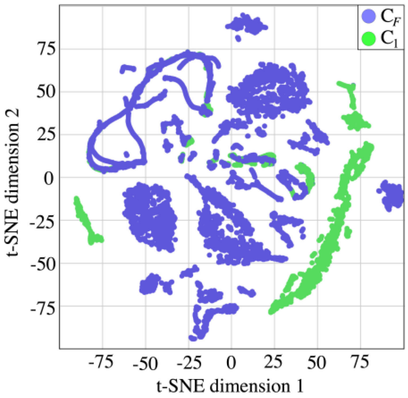


Fig. 6. Clustering performance of one normal class learning using t-SNE, where values of both axes were normalized between [−100, 100].

OCGAD model (the third histogram in Fig. 5) performed the best accuracy of 97.00% among all peer models.

Although diagnosis accuracy exhibited partial performance of the model, the clustering performance was another metric to evaluate its effectiveness. Fig. 6 displayed the clustering of the one normal class learning using t-distributed stochastic neighbor embedding (t-SNE) [45], where the green part was the normal data whereas the blue part was the error rate of the classification. It can be clearly shown that most of them were classified correctly.

B. Novelty Detection From Unknown Conditional Data

The second task was to detect novelty data from unknown conditional ones. This diagnosis could simulate the practical process of the complex operation of the industrial robot. Once the health monitoring system observed an outlier, it could also give the right response. To simulate this kind of process, it used the normal data (C_1) and one faulty data (C_4) for training the OCGAD. The goal of this mission was to send the testing samples of C_1, C_2, C_3, and C_4 separately and then detected whether these samples belong to C_1 and C_4. For the feature extraction tool used in Bi-GAN, only normal data were used for model training for this fault diagnosis experiment.

Fig. 7 plots the novelty detection result of OCSVM with accuracy 74.69%, OCiF with 41.42%, OCGAD with 95.74%, OCGAD1 with 36.65%, OCGAD2 with 35.98%, OCGAD3 with 96.88%, OCGAD4 with 97.03%, BGOiF with 33.43%, BGOiF1 with 33.33%, BGOiF2 with 34.01%, BGOiF3 with 38.01%, BGOiF4 with 33.96%, and OCDCAE with 34.15%, respectively. There was no doubt that OCGAD approach still came to the greatest performance among all comparison models. These results validated the powerfulness of the proposed OCGAD again when dealing with the novelty detection task for the robot fault diagnosis.

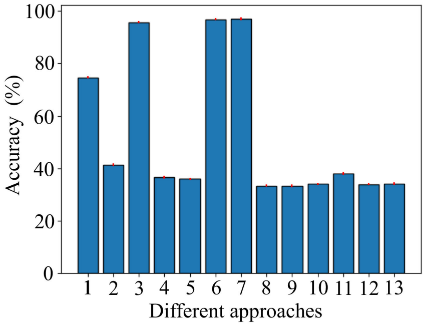


Fig. 7. Average accuracy of the novelty detection, where approaches 1–12 stand for OCSVM, OCiF, OCGAD, OCGAD1, OCGAD2, OCGAD3, OCGAD4, BGOiF, BGOiF1, BGOiF2, BGOiF3, BGOiF4, and OCDCAE, respectively.

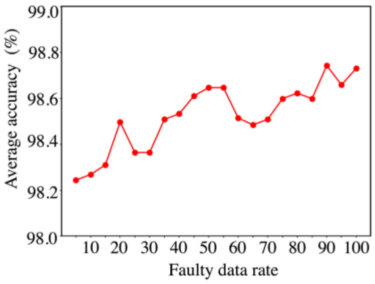


Fig. 8. Average accuracy with different proportions of faulty data.

To test the sensitivity of the faulty data, their proportion was changed from 5% to 100%. That is, the training samples were set as 4762 × i% ($i = 5, 10, 15, \ldots, 100$) and the testing samples remained the same. As plotted in Fig. 8, the accuracy oscillated only from 98.23% to 98.75%, illustrating that it was robust for the novelty detection.

Fig. 9 exhibits the clustering results of the novelty detection using t-SNE tool, where the green part was normal data whereas the blue part stood for novelty ones. It was clear that most of them are separated correctly.

C. Faulty Classification of Unlabeled Data

The third task was to perform fault classification of unlabeled data, which were not included for the model training. Compare with novelty detection from unknown conditional data, these two tasks had the same procedure in the training stage. The difference was that the third task was to test the ability for classifying unknown classes in the testing stage. In this task, unlabeled data (labels of the data were removed) were used to train the model. The method for separating the unlabeled data was performed through Euclidean distance clustering for OCGAD. The training

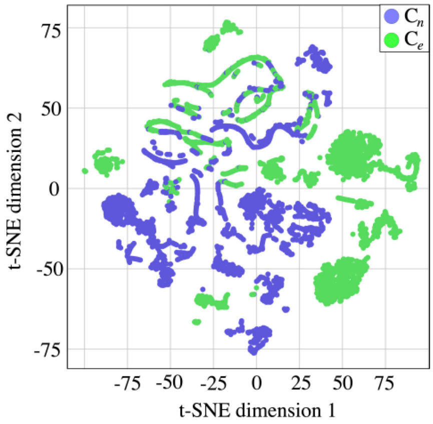


Fig. 9. Clustering performance of the novel detection using t-SNE, where values of both axes were normalized between [−100, 100].

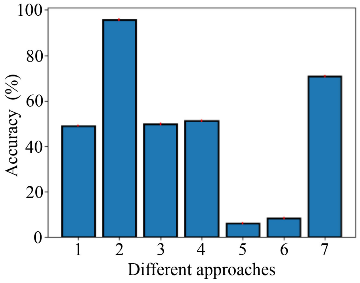


Fig. 10. Average accuracy of classification with unlabeled data, where approaches 1–6 stand for ED-ts, ED, ED1, ED2, ED3, ED4, and ED5, respectively.

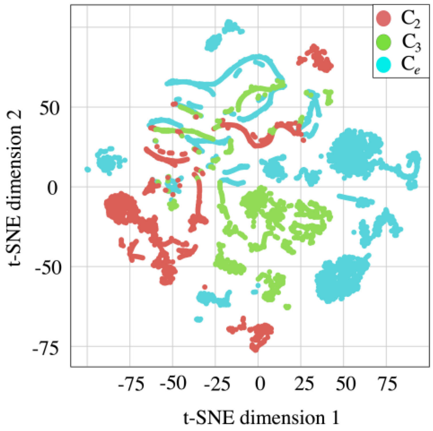


Fig. 11. Classification performance of the unlabeled data using t-SNE, where values of both axes were normalized between [−100, 100].

stage and test stage were the same as previous subsection. The difference was that the acquired outlier data did not belong to C_1 or C_4 for the classification in an unsupervised way. The comparison models were thus changed as OCSVM with Euclidean distance (ED-ts); OCGAD with Euclidean distance clustering (ED); OCGAD1 with Euclidean distance clustering (ED1); OCGAD2 with Euclidean distance clustering (ED2); OCGAD3 with Euclidean distance clustering (ED3); OCGAD4 with Euclidean distance clustering (ED4); and OCDCAE with Euclidean distance clustering (ED5), respectively.

As shown in Fig. 10, the fault type classification results were 49.09% for ED-ts, 95.74% for ED, 49.87% for ED1, 51.24% for ED2, 6.03% for ED3, 8.22% for ED4, and 70.79% for ED5, respectively. These results indicated that the proposed approach (ED) had the best performance when classifying faulty types from unlabeled data compared with the peer methods.

Besides, the clustering performance of the fault classification with an unlabeled class was taken into consideration. The result was plotted in Fig. 11. It was clearly that the model can mostly classify the unlabeled faulty data with this strategy.

V. Conclusion

In this article, a novel OCGAD framework using only normal data for model training was reported for dealing with multifunctional fault diagnosis missions. This multifunctional model was developed through three main steps. First, Bi-GAN was trained with only normal data. Second, encoders were taken out from the trained Bi-GAN for mapping the data to acquire its latent knowledge. Finally, OCSVM was employed to perform multiple fault diagnoses with latent knowledge. With different training strategies, the presented OCGAD framework was capable of anomaly detection with only normal data, novelty detection from unknown conditional data, and fault classification with unlabeled data. The proposed OCGAD was evaluated by using an experimental set-up for the fault diagnosis of the industrial robot. Comparison and discussion demonstrated that the addressed framework can be used as a semisupervised tool for multifunctional fault diagnoses of the industrial robot.

References

[1] L. Fang, Y. Li, Z. Liu, and C. Yin, “A practical model based on anomaly detection for protecting medical IoT control services against external attacks,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4260–4269, Jun. 2020.

[2] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, “Applications of machine learning to machine fault diagnosis: A review and roadmap,” *Mech. Syst. Signal Process.*, vol. 138, 2020, Art. no. 106587.

[3] W. Chen, “Intelligent manufacturing production line data monitoring system for industrial internet of things,” *Comput. Commun.*, vol. 151, pp. 31–41, 2020.

[4] Z. X. Hu, Y. Wang, M. F Ge, and J. Liu, “Data-driven fault diagnosis method based on compressed sensing and improved multiscale network,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3216–3225, Apr. 2020.

[5] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[6] A. Liaw and M. Wiener, “Classification and regression by random forest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[7] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 4, pp. 580–585, Jul./Aug. 1985.

[8] I. Kononenko, “Semi-naive Bayesian classifier,” in *Proc. Eur. Work. Session Mach. Learn.*, 1991, pp. 206–219.

[9] 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.

[10] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach,” *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[11] D. Cabrera *et al.*, “Bayesian approach and time series dimensionality reduction to LSTM-based model-building for fault diagnosis of a reciprocating compressor,” *Neurocomputing*, vol. 380, pp. 51–66, 2020.

[12] H. Han, X. Cui, Y. Fan, and H. Qing, “Least squares support vector machine (LS-SVM)-based chiller fault diagnosis using fault indicative features,” *Appl. Thermal Eng.*, vol. 154, pp. 540–547, 2019.

[13] X. B. Wang, X. Zhang, and Z. Li, "Ensemble extreme learning machines for compound-fault diagnosis of rotating machinery," *Knowl.-Based Syst.*, vol. 188, 2020, Art. no. 105012.

[14] J. Long, Z. Sun, C. Li, Y. Hong, Y. Bai, and S. Zhang, "A novel sparse echo autoencoder network for data-driven fault diagnosis of delta 3-D printers," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 683–692, Mar. 2020.

[15] W. Gong *et al.*, "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion," *Sensors*, vol. 19, no. 7, 2019, Art. no. 1693.

[16] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, 2016.

[17] T. Ergen and S. Kozat, "A novel distributed anomaly detection algorithm based on support vector machines," *Digit. Signal Process.*, vol. 99, 2020, Art. no. 02657.

[18] U. Fiore, F. Palmieri, A. Castiglione, and A. Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[19] T. Chen, X. Liu, B. Xia, W. Wei, and Y. Lai, "Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder," *IEEE Access*, vol. 8, pp. 47072–47081, 2020.

[20] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, 2014.

[21] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu, "Unsupervised novelty detection using deep autoencoders with density based clustering," *Appl. Sci.*, vol. 8, no. 9, 2018, Art. no. 1468.

[22] K. Feng, Z. Jiang, W. He, and B. Ma, "A recognition and novelty detection approach based on curvelet transform, nonlinear PCA and SVM with application to indicator diagram diagnosis," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12721–12729, 2011.

[23] S. Wentao, L. Changhou, and Z. Dan, "Bearing fault diagnosis based on feature weighted FCM cluster analysis," in *Proc. IEEE Int. Conf. Comput. Sci. Softw. Eng.*, 2008, pp. 518–521.

[24] G. Yu, C. Li, and S. Kamarthi, "Machine fault diagnosis using a cluster-based wavelet feature extraction and probabilistic neural networks," *Int. J. Adv. Manuf. Technol.*, vol. 42, no. 1/2, pp. 145–151, 2009.

[25] W. Zhang, X. Li, X. D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, 2020, Art. no. 107377.

[26] X. Li, W. Zhang, Q. Ding, and J. Q. Sun, "Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation," *J. Intell. Manuf.*, vol. 31, no. 2, pp. 433–452, 2020.

[27] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *J. Amer. Statist. Assoc.*, vol. 82, no. 398, pp. 528–540, 1987.

[28] D. A. Van Dyk and X. L. Meng, "The art of data augmentation," *J. Comput. Graph. Statist.*, vol. 10, no. 1, pp. 1–50, 2001.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[30] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[31] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[32] P. Isola, J. Y. Zhu, T. Zhou, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[33] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[34] J. Y. Zhu, T. Park, P. Isola, and A. A. Efro, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[35] Q. Liu, G. Ma, and C. Cheng, "Data fusion generative adversarial network for multi-class imbalanced fault diagnosis of rotating machinery," *IEEE Access*, vol. 8, pp. 70111–70124, 2020.

[36] Y. Xie and T. Zhang, "Imbalanced learning for fault diagnosis problem of rotating machinery based on generative adversarial networks," in *IEEE 37th Chin. Control Conf.*, 2018, pp. 6017–6022.

[37] D. Cabrera *et al.*, "Generative adversarial networks selection approach for extremely imbalanced fault diagnosis of reciprocating machinery," *IEEE Access*, vol. 7, pp. 70643–70653, 2019.

[38] J. Donahue, P. Krähenbühl, and T. Darrel, "Adversarial feature learning," 2017, *arXiv1605.09782*.

[39] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[40] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, 2010, pp. 177–186.

[41] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[42] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016, *arXiv:1511.06434*.

[43] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.

[44] B. Schölkopf, R. C. Williamson, A. J. Smola, S. John, and P. John, "Support vector method for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 582–588.

[45] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

Ziqiang Pu received the bachelor's degree in mechanical manufacturing and automation from Chongqing Technology and Business University, Chongqing, China, in 2016, and the master's degree in mechanical engineering from Yamaguchi University, Yamaguchi, Japan, in 2019. He is currently toward the Ph.D. degree in informatics engineering with the University of Algarve, Faro, Portugal.

He is a Visiting Student with Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include mechanical fault diagnosis and intelligent systems.

Diego Cabrera received the B.Sc. degree in electronic engineering from the Universidad Politecnica Salesiana, Cuenca, Ecuador, in 2012, and the M.Sc. degree in logic, computation, and artificial intelligence and the Ph.D. degree in computer science from Seville University, Seville, Spain, in 2014 and 2018, respectively.

He is currently a Professor of Engineering with the Universidad Politecnica Salesiana. His research interests include intelligent systems and data-driven modeling.

Yun Bai (Member, IEEE) received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2014.

He has been successively a Postdoctoral Fellow with the South China University of Technology, Guangzhou, China, and the University of Algarve, Faro, Portugal. He is currently a Visiting Researcher with the University of Algarve. He is also an Associate Professor with the Chongqing Technology and Business University, Chongqing. His current research interests include intelligent system management, modeling, and forecasting.

Chuan Li (Senior Member, IEEE) received the Ph.D. degree in industrial engineering from Chongqing University, Chongqing, China, in 2007.

He was a Postdoctoral Fellow with the University of Ottawa, Ottawa, Canada; a Research Professor with Korea University, Seoul, South Korea; a Senior Research Associate with the City University of Kowloon Tong, Hong Kong; and a Prometeo with Universidad Politecnica Salesiana, Cuenca, Ecuador, successively. He is a Graduate Supervisor with Zhengzhou University of Light Industry, Zhengzhou, China, and a Professor with Chongqing Technology and Business University, Chongqing. His research interests include PHM and intelligent systems.