Measurement 167 (2021) 108234

Contents lists available at ScienceDirect

Measurement

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

A threshold self-setting condition monitoring scheme for wind turbine generator bearings based on deep convolutional generative adversarial networks

Peng Chen[a], Yu Li[a], Kesheng Wang[a,*], Ming J. Zuo[b,a], P. Stephan Heyns[c], Stephan Baggeröhr[c]

[a] *Equipment Reliability Prognostics and Health Management Lab (ERPHM), School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China*

[b] *Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta T6G 1H9, Canada*

[c] *Center for Asset Integrity Management, Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, South Africa*

ARTICLE INFO

Article history:
Received 11 February 2020
Received in revised form 4 June 2020
Accepted 11 July 2020
Available online 21 July 2020

Keywords:
Wind turbine
Condition monitoring
Deep learning
Deep convolutional generative adversarial networks

ABSTRACT

Long-term reliable health condition monitoring (HCM) of a wind turbine is an essential method to avoid catastrophic failure results. Existing unsupervised learning methods, such as auto-encoder (AE) and de-noising auto-encoder (DAE) models, are utilized to the condition monitoring of wind turbines. The critical bottleneck of these models for monitoring is to determine a threshold for identifying different health conditions. Unfortunately, the threshold is usually set up with different kinds of calculation methods or even based on experience. Therefore, the uncertainty of the threshold will inevitably influence the accuracy of the monitoring process and may lead to misdiagnoses. To overcome this limitation, this research introduces a threshold self-setting HCM scheme, based on deep convolutional generative adversarial networks (DCGAN) and employed for defining a self-setting threshold to monitor wind turbine generator bearings. A threshold for HCM can be automatically created through the output of the G network in the DCGAN model, and the challenging problem of setting up a threshold can be solved. Besides, the use of Nash Equilibrium for training enables this scheme to become self-defined evaluators with a high level of consistency, without any human intervention and can be treated as a self-defined threshold, and it is a model self-tuning process. Furthermore, a sample discrepancy analysis based on the output of the G network is utilized so that a quantitative indicator of the fault severity in wind turbine generator bearings are provided. By tracking a real wind turbine dataset from the LU NAN wind farm in China, the effectiveness of the proposed scheme is verified.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Wind turbines (WTs) are large-scale and complex electromechanical systems. They are widely utilized in renewable energy applications. A wind turbine comprises hundreds of key parts, such as tower, rotor hub, transmission chain system (main shaft, gear drive system and bearings), spinner and blades assembly, generator frame, cooling cabinets, power electronics, etc.[1,2]. Particularly, rolling bearings are a critical part of the transmission chain in a wind turbine. Yet, they are often exposed to extreme environmental variables, such as frost and typhoon weather states. Such states make rolling bearings prone to suffer from frequent damages and various kinds of failures [3]. As a result, unexpected malfunctions would unavoidably make extra maintenance costs. To avoid catastrophic accidents and reduce operation and maintenance (O&M) costs, HCM can prevent failures by providing early alerts as well as enable better maintenance plans.

Recent researches [4–10] on HCM for rolling bearings can be categorized as two kinds of methodologies, namely model-based and data-driven based approaches. The measured vibration signals are regularly characterized by non-linear and non-stationary characteristics due to the operational conditions with varying loads and fluctuating speeds. As a result, it makes model-based methods and signal processing methods more challenging. Furthermore, these traditional model-based approaches are heavily dependent on a rich understanding of the behavioral characteristics of rolling bearings and human experience from real practice [11–13]. Furthermore, the model-based approaches are usually restricted with a specific part in rotating machinery. Thus it usually cannot be generalized and used for monitoring other parts. That may signifi-

[*] Corresponding author.
E-mail addresses: peng.kaysen.chen@gmail.com (P. Chen), keshengwang@uestc.edu.cn (K. Wang).

https://doi.org/10.1016/j.measurement.2020.108234
0263-2241/© 2020 Elsevier Ltd. All rights reserved.

cantly limit the generalization capability of the model-based approach in various applications.

Conversely, for data-driven approaches like machine learning techniques, they are not only having the capability of generalization but also have been proven to be effective in various applications of fault diagnosis and HCM [2,14–16]. However, the traditional machine learning methods, especially the supervised learning algorithms, also have its hindrances [17,18]. For instance, support vector machines (SVMs) and also support vector networks [19] are supervised learning models, which can be employed for both classification or regression issues. The SVMs is a binary classifier. Consequently, it does not allow good results for multi-class classifications or pair-wise classifications. Besides, SVMs cannot deal with massive amounts of industrial data that is because the computation is expensive and running slow [20]. A degradation-Hidden-Markov model [21] is proposed for evaluating the health condition status of wind turbine bearings under different stages. However, it can only accommodate small samples rather than a massive amount of industrial data.

In contrast to the traditional machine learning, the strengths of deep learning algorithms are listed as follows. (a) Learning features and recognizing fault automatically. (b) Learning a more complex structure from data due to the deep architecture. (c) Handcraft feature engineering is not necessary [22]. Numerous researches [23–26] in terms of deep learning algorithms are offered in the field of HCM. However, deep learning methods, especially the supervised deep learning algorithms, such as a convolutional neural network (CNN) [27] requires a sufficient amount of historical labeled failure data for training. Unfortunately, for industrial applications [28,29], it is often challenging to achieve a massive amount of labeled samples. That makes the supervised learning methods, to some extent, unrealistic. According to the above discussions, some hurdles of HCM for rolling bearings in WTs in terms of data-driven approach can be summarized as follows,

1. Large and massive amounts of data are featured with scarce labels.
2. Vibration signals are characterized by nonlinearity and nonstationarity caused by varying loads and fluctuating speeds.
3. The strategy of HCM requires to be reliable and on-line with less human intervention.

An unsupervised deep learning approach, namely an autoencoder (AE) model, is therefore introduced for wind turbine monitoring in which only the healthy data are needed for training [2,30,31]. An AE model can learn an underlying representation of the available health data in an unsupervised manner, and it may well describe the nonlinear features of the raw input data. Its hidden layer information can be further served as the raw input data to the decode network, which is trained to reconstruct the raw input. Consequently, the model is trained to minimize the reconstruction error (RE). Afterward, through RE, it may help to detect previously unseen patterns in the new data sequences, in other words, the RE may be treated as a threshold for condition monitoring purposes. Recently, some attempts [2,31] were made to utilize AE models for anomaly detection and blade breakage in WTs. Inspired by these works, Jiang et al. introduced a de-noising autoencoder (DAE) model to build a robust multivariate reconstruction model based on the original time-series data from multiple sensors. By analyzing the Supervisory Control and Data Acquisition (SCADA) data, a receiver operating characteristic (ROC) analysis [32] is utilized to evaluate RE and hence for monitoring. A proper threshold for trending RE is, however, not provided. In order to determine the threshold, an exponentially weighted moving average (EWMA) control chart [31,33] is applied to analyze RE distribution. To detect the anomalous condition of wind turbines, an assumption of the RE distribution, which satisfies an exponential distribution with a threshold based on extreme value theory, was proposed for monitoring purposes [34]. Meanwhile, the recent investigations shed light on the bearing fault diagnosis based on different domains. In [35], to accommodate the issue of lacking substantial labeled samples, a domain adaptive deep belief network is proposed, and it has been verified in the case of rolling bearing fault diagnosis. To remedy between the source domain and target domain, in [36], a novel knowledge transfer network with fluctuating operational condition adaption was proposed, and this model enables the capacity of fault diagnosis under varying operational conditions for rolling bearings. In [37], an intelligent fault diagnosis method based on generative adversarial nets combined with the convolutional neural network was proposed. This method is implemented via Wavelet transform, and then the time–frequency images are used to train WT-CNN-GAN model for producing new samples for classification. However, the core problem of the approaches, as mentioned above, is that the threshold is dependent on the RE distribution. Once the distribution of RE violates this assumption, the threshold setting is problematic, and thus, fault modes or malfunctions may not be identified. Besides, during the above threshold setting process, human intervention is unavoidable, and different threshold setting methods may give different results of threshold. Subjective uncertainties are inevitably introduced to the health monitoring process. Furthermore, owing to the noisy RE distribution, in real condition monitoring of wind turbines, a practical threshold for HCM based on AE or DAE model is an issue to be resolved.

To overcome the above drawbacks, a self-setting threshold scheme based on deep convolutional generative adversarial networks (DCGAN) is proposed in this paper. In this scheme, an unsupervised DCGAN is trained with measured healthy data. Once the DCGAN model is well trained, i.e., the model reaches a Nash equilibrium [38], a threshold for HCM can be automatically created through the output of the DCGAN model. Afterward, to quantitatively evaluate and track the fault severity of a wind turbine bearing with different measurements, a sample discrepancy method is developed for evaluating the health conditions. In this paper, a wind turbine dataset measured at LU NAN wind farm in China over about 2 years is utilized to validate the effectiveness of the proposed scheme by comparing with AE, DAE and vanilla generative adversarial networks (VGAN). The proposed scheme shows distinct advantages in HCM for wind turbine bearings.

A schematic depiction of the proposed scheme is depicted in Fig. 1. Furthermore, the performance of the scheme is compared with the AE and DAE based models. Primarily, the self-setting threshold is employed to substitute the Receiver operating characteristic (ROC) analysis method, EWMA control chart and extreme value theory based on exponential distribution. The rest of the paper is outlined as follows. Section 2 specifies a vanilla generative adversarial network (VGAN) model, which is the basis of the proposed scheme. Section 3 illustrates the proposed threshold self-setting HCM scheme based on a DCGAN model. Subsequently, a case study is discussed in detail in Section 4. Section 5 concludes the proposed scheme and gives a summary of the results. Finally, in A, the basis of auto-encoder and de-noising auto-encoder models are illustrated.

2. Generative adversarial networks

2.1. Vanilla generative adversarial networks (VGAN) model

In this section, a state-of-the-art algorithm for generative adversarial networks is introduced, commonly referred to as VGAN [39]. VGAN is inspired by binomial zero-sum game theory [40]. It

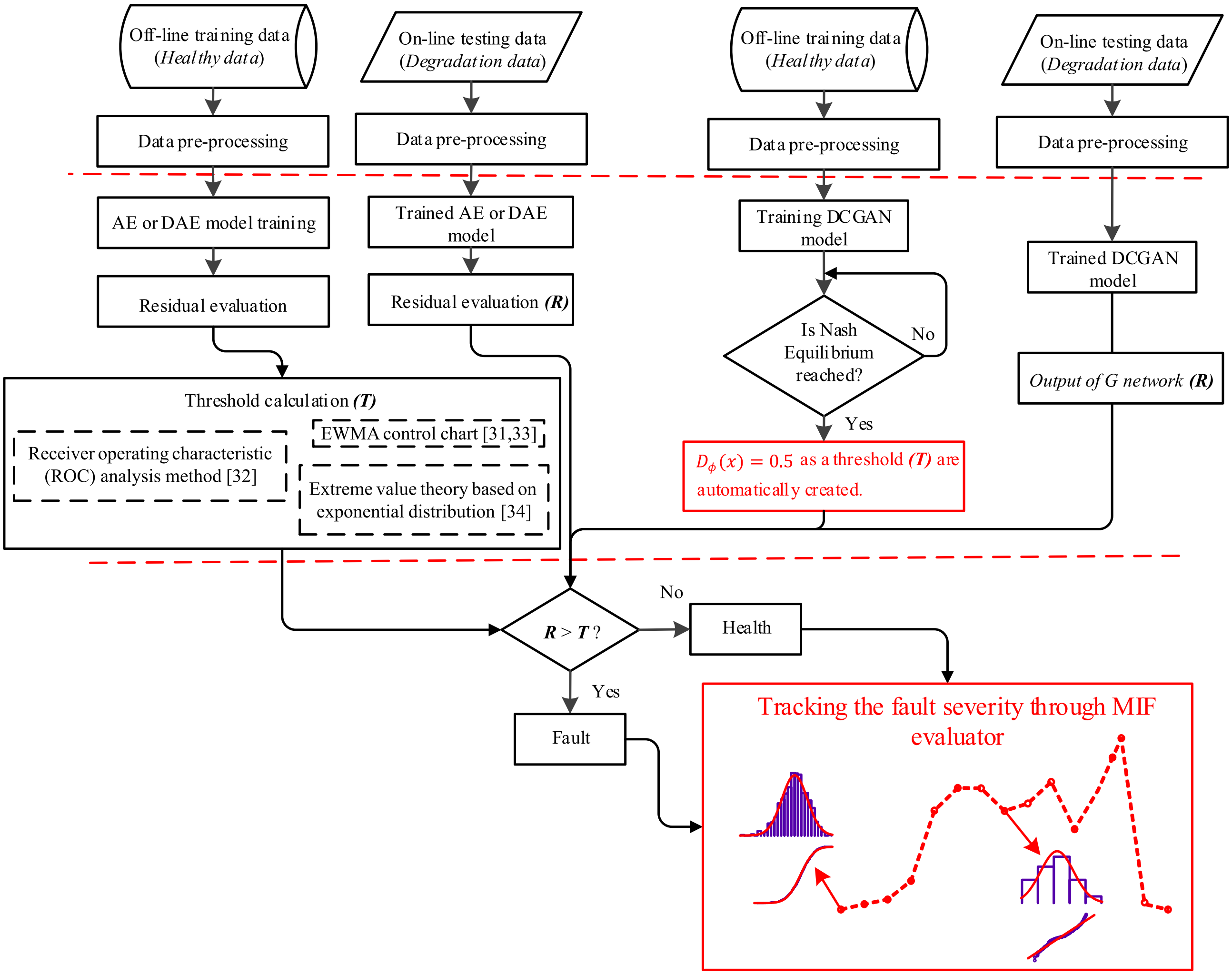
Off-line training data (Healthy data)
On-line testing data (Degradation data)
Off-line training data (Healthy data)
On-line testing data (Degradation data)
Data pre-processing
Data pre-processing
Data pre-processing
Data pre-processing
AE or DAE model training
Trained AE or DAE model
Training DCGAN model
Trained DCGAN model
Residual evaluation
Residual evaluation (R)
Is Nash Equilibrium reached?
No
Yes
Output of G network (R)
Threshold calculation (T)
EWMA control chart [31,33]
Receiver operating characteristic (ROC) analysis method [32]
Extreme value theory based on exponential distribution [34]
$D_\phi(x) = 0.5$ as a threshold (T) are automatically created.
R > T ?
No
Health
Yes
Fault
Tracking the fault severity through MIF evaluator

Fig. 1. Schematic description of health condition monitoring methods based on traditional methods and proposed method.

consists of two competing subnetworks, namely a G (generator) network, $\boldsymbol{G}_\theta$ and a D (discriminator) network, $\boldsymbol{D}_\phi$. Each opposing subnetwork is parameterized by a different set of weights θ and biases ϕ. By changing θ, the optimizer G minimizes the loss function $\mathcal{L}_{G_\theta(z)}(\theta)$ in Eq. (1). Likewise, by changing ϕ, the optimizer D minimizes its own loss function $\mathcal{L}_{D_\phi(x)}(\phi)$ in Eq. (2). The structure of VGAN is shown in Fig. 2.

$$\mathcal{L}_{G_\theta(z)}(\theta) = -\log\left(\frac{\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))}{1-\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))}\right) \tag{1}$$

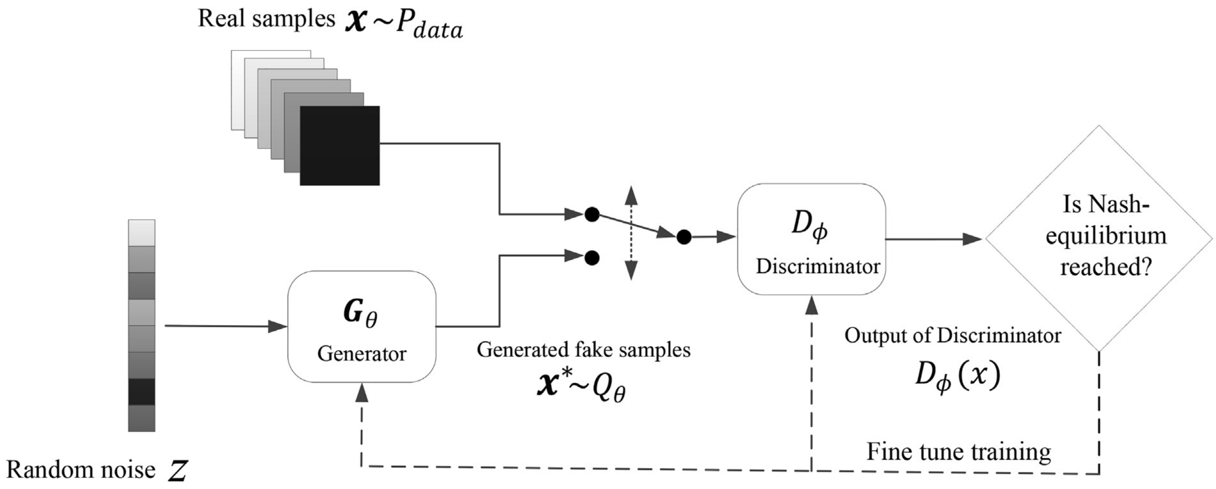

Fig. 2. Computational procedure and structure of VGAN.

$$\mathcal{L}_{D_\phi(x)}(\phi) = -\frac{1}{2}\log(\boldsymbol{D}(\boldsymbol{x})) - \frac{1}{2}\log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))) \quad (2)$$

In the training process, the goal of the discriminator is to determine whether the input samples are the real samples $\boldsymbol{x}$ or generated fake samples $\boldsymbol{x}^*$. The real samples follow the probability distribution of P_{data}. G tries to capture the underlying feature distribution of the real samples so that the generated fake samples could fool the discriminator, i.e., the produced fake samples would be considered as real samples by the discriminator. The training of the VGAN model is achieved by setting G against D in a min–max two-player game fashion, and it can be explained mathematically as an optimization process for function $V(D_\phi(x), G_\theta(\mathbf{z}))$.

Random noise $\boldsymbol{z}$ is input to G and it will generate fake samples x^* which follows a parameterized distribution Q_θ, i.e., $x^* \sim Q_\theta$. With the training progressing, the discrimination ability of D improved so that it could separate the samples from G or D subnetwork.

G or D updates its parameters by a fine-tuning process guided by the gradients derived from the cost function defined in Eq. (3). Through the mutual adversarial learning mechanism between the two subnetworks, G and D continuously confront each other and optimize themselves. Finally, when training is well completed, Nash-equilibrium is automatically reached. According to this game principle, the Nash-equilibrium reached a point that leaves G/D in no better or no worse situation, no matter what G/D opponent decides to do. At this stage, the continued adversarial training of both competing subnetworks can lead to the generation of high-quality samples, i.e., $Q_\theta \approx P_{data}$. D now is not sure of whether the samples are real or produced, then it means that the $D_\phi(x)$is close to 0.5.

$$\min_{G_\theta(z)} \max_{D_\phi(x)} V\ (D_\phi(x), G_\theta(\mathbf{z})) = \mathbb{E}_{x\sim p_{data}(x)}[\log\ D(x)] + \quad (3)$$
$$\mathbb{E}_{z\sim p_z(z)}[\log\ (1 - D(G(z)))]$$

3. A threshold self-setting HCM scheme based on DCGAN model

To begin with, in order to accommodate the non-stationary conditions and capture the underlying nonlinear and non-stationary features, a convolutional neural network, commonly referred to as ConvNet or CNN [41], are integrated into the original VGAN model. This model is named deep convolutional generative adversarial networks (DCGAN) [42].

Secondly, a threshold, i.e., $D_\phi(x) \approx 0.5$ is created by the well trained DCGAN model, and it is used as a criterion for setting up a threshold and monitoring input samples. Once a DCGAN model is trained well (or reaching Nash-equilibrium) based on healthy samples, a fault sample input will make $D_\phi(x)$ away from 0.5. So the trained DCGAN model can be treated as a self-defined evaluator for identifying the measured anomalous samples. More important, it should be noticed that, for calculating $D_\phi(x)$, there is no human intervention or manual setting up of a threshold for health monitoring. Instead, the DCGAN model will automatically train itself as an evaluator for monitoring purposes. The further away from $D_\phi(x) \approx 0.5$, the more severe the deviation of the input samples. The $D_\phi(x)$ can be used as the DCGAN metric in Section 4.2.3. Further, it is also noticed that $D_\phi(x)$ is not a single value, but a data sample with the same size as the input sample. Therefore, it is reasonable to transform the data sample $D_\phi(x)$ to be a fusion indicator. Thus, a sample discrepancy based method is adapted here to calculate a monitoring indicator function (MIF) based upon the output of DCGAN, $D_\phi(x)$, so that MIF could easily be used for health condition monitoring.

The corresponding application flow chart of the proposed scheme is shown in Fig. 3. The implementation procedures can be summarized as follows:

1. Raw data, such as a wind turbine generator bearings dataset, is collected by accelerometers. Afterward, the raw vibration signal is transformed into frequency domain signal through Fourier analysis. Finally, the frequency domain data is fed into the DCGAN model.
2. Via the adversarial learning mechanism, G and D subnetworks of DCGAN model are iteratively trained by alternating optimization until Nash equilibrium is reached.
3. The well trained DCGAN model is used as a self-defined evaluator, and the output $D_\phi(x)$ can be creatively used for anomalous detection or thresholding.
4. A MIF indicator based upon sample discrepancy analysis is further calculated with $D_\phi(x)$. Then, quantitative analysis or a fusion measure is established for health condition monitoring.

The detailed configuration of the DCGAN model used in this paper is explained in the following. G and D (shown in Fig. 2) are replaced by deep convolutional neural networks (DCNN), respectively. A visual representation of the G architectural topology is presented in Fig. 4 and the architectural topology of the transposed convolutional neural network is shown in Fig. 5. The G network is comprised of 4 transposed convolutional layers, which is summarized in Table 1. For this network, random noise is passed through a fully connected layer, with an output size selected and reshaped to ensure the produced samples fit the dimension of the real samples, which are then passed through four deconvolution layers. In [42], it is suggested that a Relu activation function is used in the generator for all layers except for the final layer (Sigmoid), which is determined by the scope of input. Therefore, a Relu activation function is applied to all layers, whereas the final layer passes through a sigmoid activation function. Thus, training data is normalized between [0, 1], before being presented to the discriminator. D consists of 4 convolutional layers with an increasing number of filters on each layer. The stride is kept low on the first and second layers and then increased on the rest of the layers with the value reaching 4 (see in Table 2). The fully connected neural network is utilized to transfer the output of discriminator with an output size of 4096 (second last layer) and 1 (last layer) respectively. The Leaky Relu activation function is utilized in all layers except the output [42]. This can potentially make the adversarial training of the architecture more stable and avoiding sparse gradients [42].

4. Case study for analyzing the bearing in wind turbine

The proposed scheme for wind turbine bearing health condition monitoring is validated through measured vibration data from a real wind farm in LU NAN, China. The doubly-fed induction generator (DFIG) wind turbine system [43] is being monitored, and its configuration is shown in Fig. 7. This kind of wind turbine is an offshore 1.5-MW three-bladed horizontal axis system. Twelve accelerometers are installed at different locations of this wind turbine transmission line (see Figure 6). The measured signal for the studies in this paper is acquired from channel 05, and it is highlighted with the dotted square shown on the generator in Fig. 6. Its real mounting situation is also shown in Fig. 8. It can be seen that channel 05 is located at the input shaft of the generator with the shortest distance to the monitored bearing among all 12 accelerometers.

Many input rotational speeds are possible for a wind turbine due to the various kinds of operational conditions, such as variable stochastic wind speeds (see in Fig. 9), non-stationary loads and dynamic transmission torques [2]. It is always a question before implementing a health condition monitoring scheme of the real-world wind turbines. In order to have consistent monitoring with

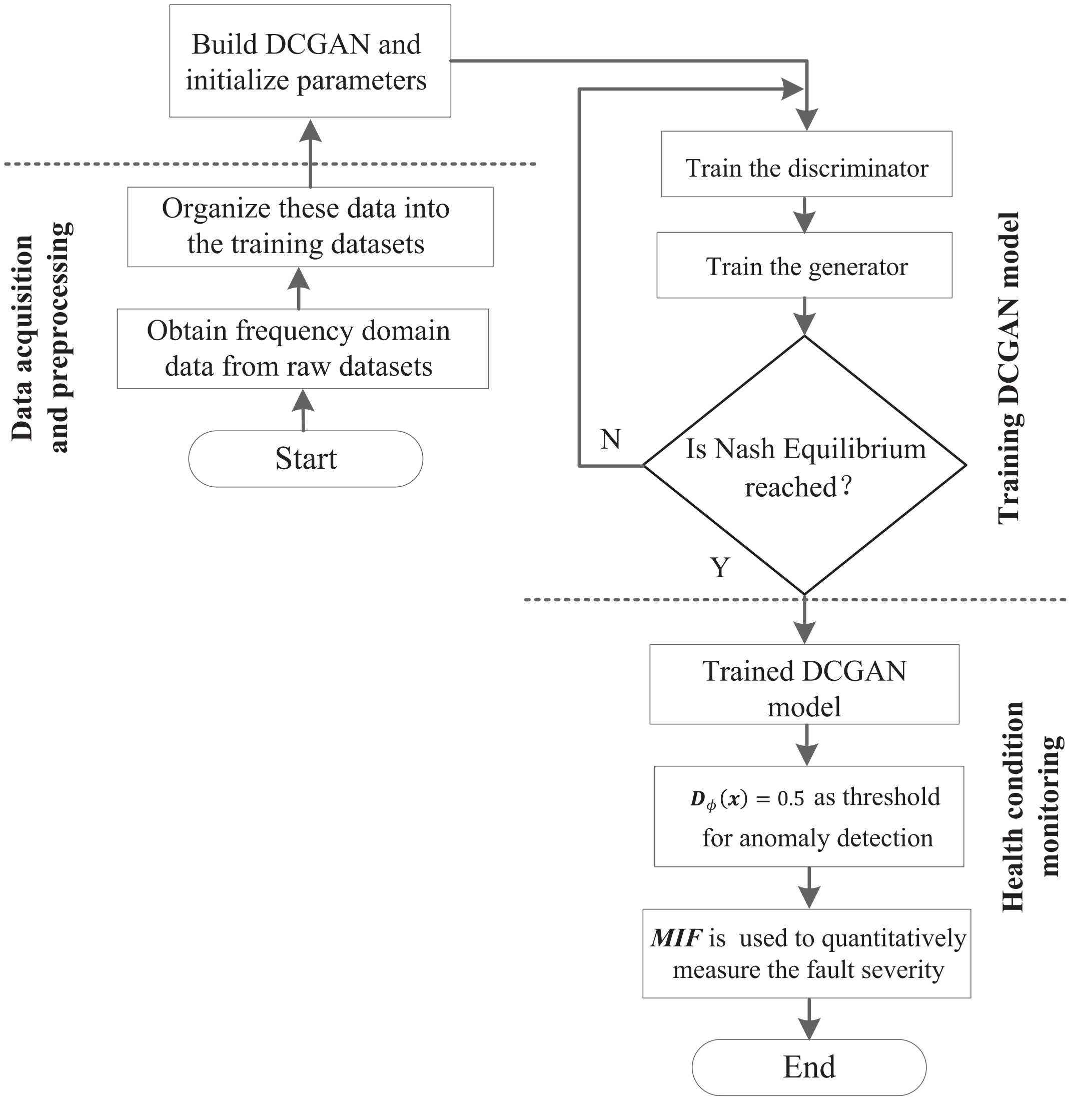
Build DCGAN and initialize parameters
Organize these data into the training datasets
Obtain frequency domain data from raw datasets
Start
Data acquisition and preprocessing
Train the discriminator
Train the generator
Is Nash Equilibrium reached?
N
Y
Training DCGAN model
Trained DCGAN model
$D_\phi(x) = 0.5$ as threshold for anomaly detection
MIF is used to quantitatively measure the fault severity
End
Health condition monitoring

Fig. 3. Flow chart of proposed new HCM scheme based on DCGAN.

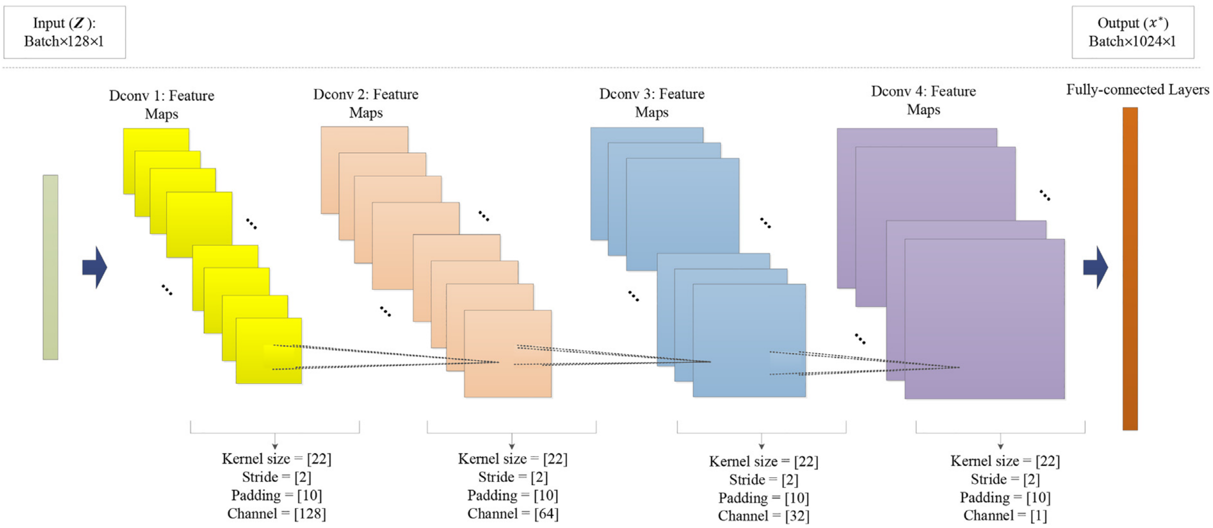

Fig. 4. A visual representation of G generator architecture.

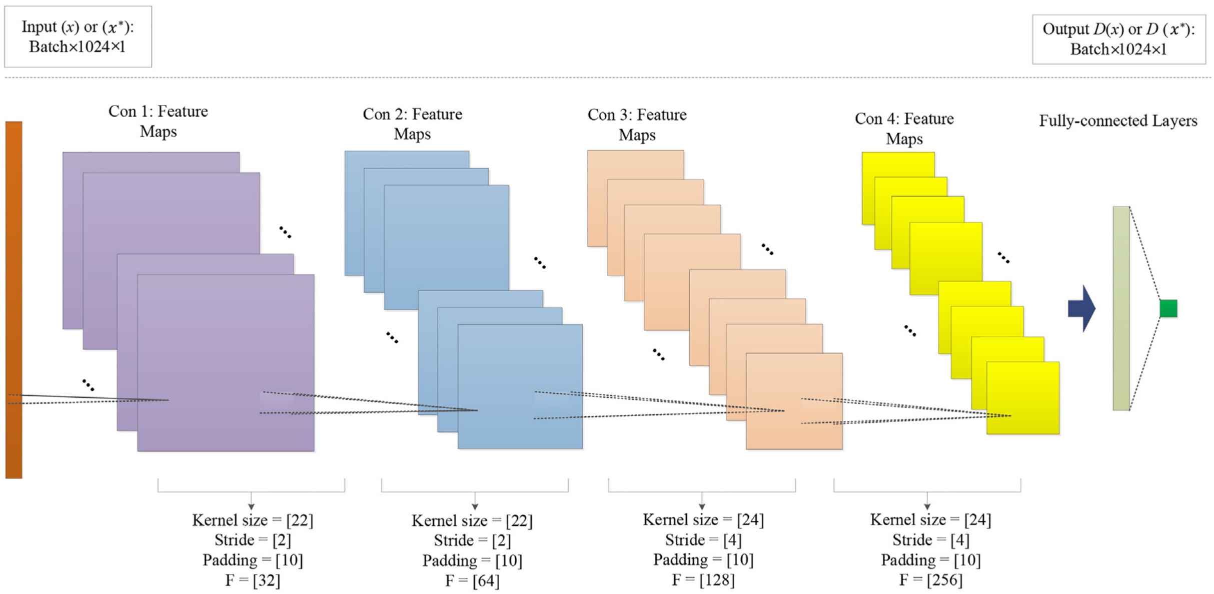

Fig. 5. A visual representation of D discriminator architecture.

Table 1
Generator architecture of DCGAN.

Layer	DCGAN Generator
Input	z = [128×1]
1st	$Deconv_{1\times 256\|\|Kernelsize=[22],Stride=[2],padding=[10]}(Relu)$
2nd	$Deconv_{256\times 128\|\|Kernelsize=[22],Stride=[2],padding=[10]}(Relu)$
3rd	$Deconv_{128\times 32\|\|Kernelsize=[22],Stride=[2],padding=[10]}(Relu)$
4th	$Deconv_{32\times 1\|\|Kernelsize=[22],Stride=[2],padding=[10]}(Relu)$
5th	$Dense_{(2048,1024)}(Sigmoid)$
Output	[1×1024]

Table 2
Discriminator architecture of DCGAN.

Layer	DCGAN Discriminator
Input	x = [1024×1]
1st	$Conv_{1\times 32\|\|Kernelsize=[22],Stride=[2],padding=[10]}(LeakyRelu(0.2))$
2nd	$Conv_{32\times 64\|\|Kernelsize=[22],Stride=[2],padding=[10]}(LeakyRelu(0.2))$
3rd	$Conv_{64\times 128\|\|Kernelsize=[24],Stride=[4],padding=[10]}(LeakyRelu(0.2))$
4h	$Conv_{128\times 256\|\|Kernelsize=[24],Stride=[4],padding=[10]}(LeakyRelu(0.2))$
5th	$Dense_{(256\times 16,1)}(Sigmoid)$
Output	[1]

acceptable accuracy, in this scheme, it is sensible to consider the measured vibrations during the full working conditions. Vibrations at typical rotational speeds 1080 r/min were selected at a high priority for analysis.

The real monitoring conditions are as follows,

1. Due to the real monitoring schedule plan, a semi-monthly interval measurement is selected. The sampling frequency is 20,000 Hz, and 15 sets of data at random periods within one day are used for each analysis.
2. Due to the ever-changing weather conditions, the rotating speed could not always reach the selected speeds 1080 r/min. To ensure consistent health condition monitoring, the actual collection dates do deviate from the planned intervals.

4.1. Data pre-processing

The raw data x with $n = 2,156,000$ points, $\{\mathbf{x}_i\}_{i=1}^{n}$, is collected during each measurement. The training data was measured on May 31, 2015. To construct multiple samples for training, a sliding-window technique [2], is applied to produce time-series data. Specifically, the size of the sliding window and the stride are set at 1024 points and 512 points, respectively. Consequently, 1346 samples with 1024 points in each sample are produced. Then, the Fast Fourier transform (FFT) [44] is utilized to transform the time domain samples into the frequency domain. Finally, a normalized scheme shown in Eq. (4) is used to normalize the frequency domain data. Hence, the input data is first transformed into the frequency domain and then normalized to [0, 1]. The testing samples (from Oct. 31, 2015, to Apr. 19, 2016) are processed in the same manner.

$$x_{train} = \frac{(x - \min(x))}{\max(x) - \min(x)} \tag{4}$$

4.2. Case study

4.2.1. AE and DAE models

Once the training and testing dataset from the wind turbine is ready for analysis, the unsupervised learning models such as auto-encoder (AE) and de-noising auto-encoder (DAE) for health condition monitoring are also adopted for comparison. The models are implemented using machine learning PyTorch. The training models are implemented end-to-end on a GPU (Nvidia GTX 1080). For testing and validating the model, the model was run on a machine utilizing a 12-core CPU (Intel i7 8700 K) with 32G of DDR4 memory. The use of a GPU for the training of the models decreased the time significantly.

The first 15% (May 31, 2015) of the dataset is treated as a reference condition (healthy condition) and displayed in section I of Fig. 10. Reconstruction error (RE) which is explained in Appendix A is used for setting up the monitoring threshold. Through calculating RE (see Eq. (A.3)) for healthy data, a threshold (the dotted line) can be obtained as is shown in Figs. 10 and 11. The corresponding RE results through AE and DAE models can be depicted in Figs. 10 and 11.

In Fig. 10, the RE results are described as follows,

1. RE values of section I almost stay constant at 1.85μ except for two peak values, which may be caused by sudden disturbances,

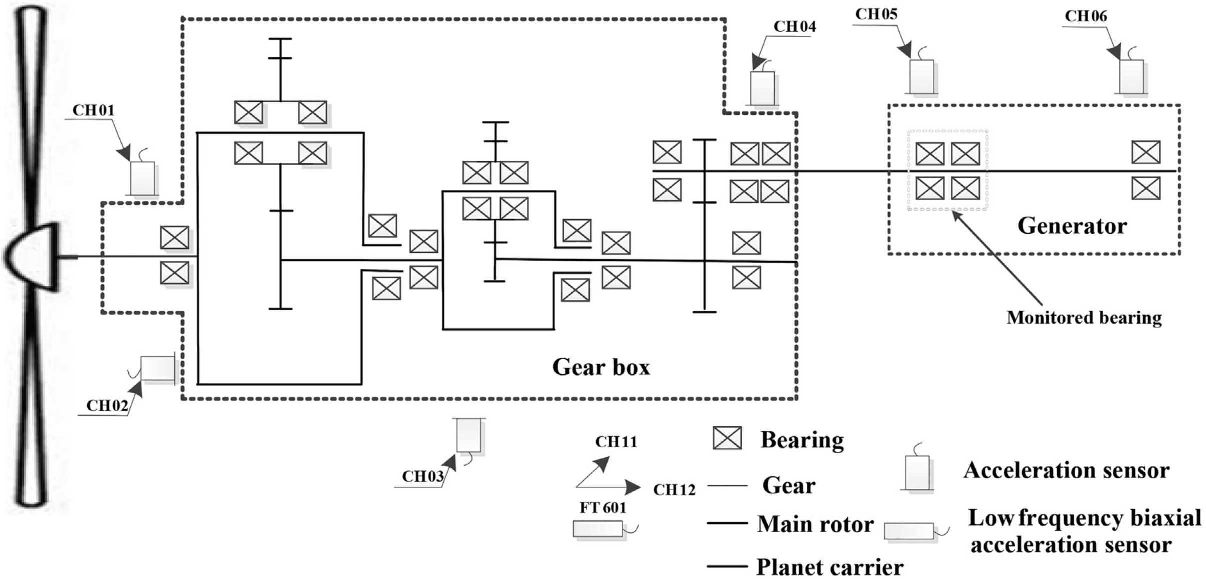

Fig. 6. The layout of wind turbine.

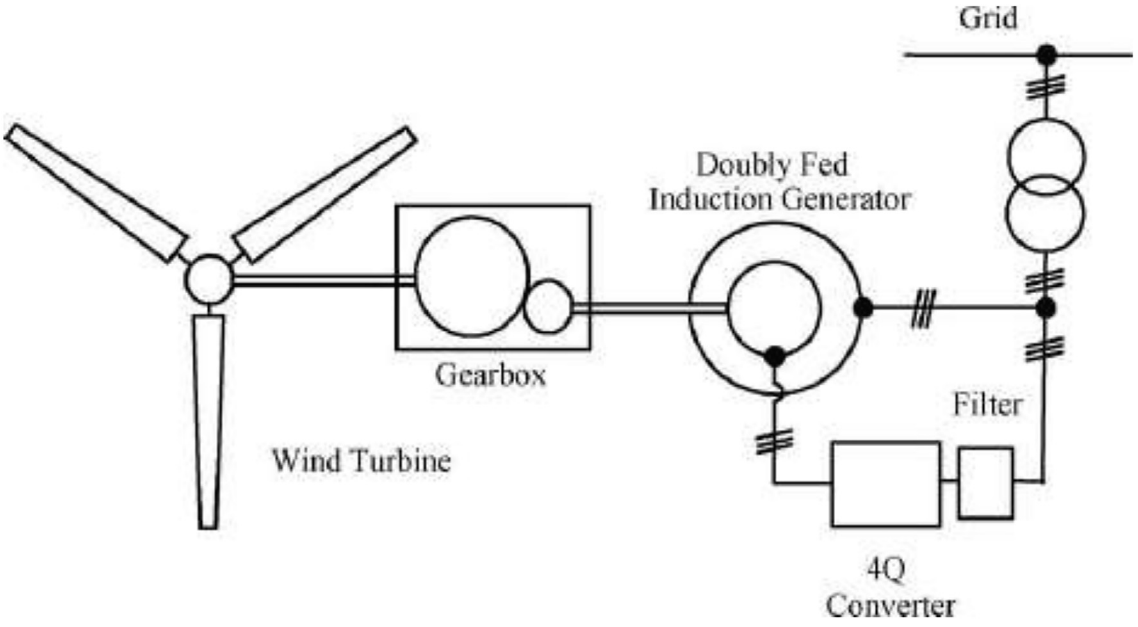

Fig. 7. Double fed induction generator wind turbine system.

and section I can be treated as a fairly good threshold or reference for monitoring.

2. Further, RE values from sections II and III both are above the threshold (dotted line). Data from section II was also measured during healthy conditions, while there is a distinct difference of RE values for sections I, II and III. That makes the setting up of the threshold problematic.
3. It shows that RE in section II fluctuates from 44.53μ to 122.20μ. Whereas, in section III, the RE value fluctuates over a wide range from 96.82μ to 493.21μ.

From the observations of Fig. 10, it may be observed that,

1. Due to the different RE results, the healthy state in section II and section III are both different from the reference condition (section I). That makes the set up a proper threshold for monitoring difficult.
2. The RE fluctuating range of section III is wider than section II. It indicates that the status of section III is deteriorating compared to section II.

Similarly, an investigation for HCM is implemented based on the DAE model (see Fig. 11). Due to the environmental noise being overwhelmed in a real wind farm, there is little impact on the DAE model by adding noise. Hence, the HCM result of the DAE model is almost the same as the AE model (see Fig. 11 and Fig. 10).

To further investigate the use of RE for health condition monitoring, the exponentially weighted moving average (EWMA) control chart, which is mentioned in [31,45] is implemented for

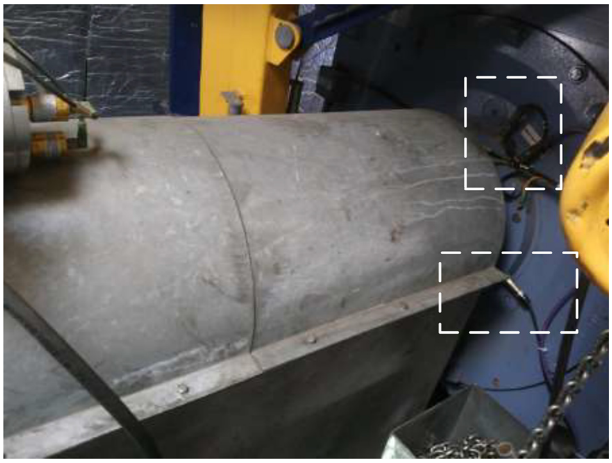

Fig. 8. Installed location of vibration accelerometers. (The accelerometers are mounted on outside of the generator in wind turbine horizontally and vertically, respectively.).

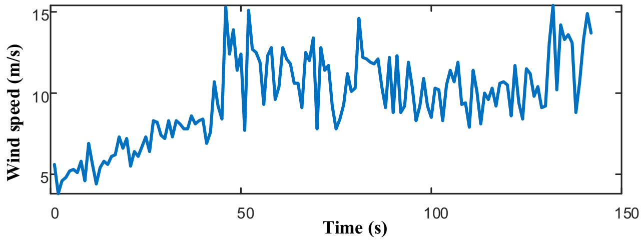

Fig. 9. Variable stochastic wind speeds.

comparison. In Fig. 12, the training data (see section I) is used to estimate a threshold based on the EWMA chart. The upper control limits (UCL) and lower control limits (LCL) [46] are calculated to be 30.56μ and 2.32μ, respectively. Compared to the original RE distribution (see Fig. 10), the EWMA control chart still treats section II as a faulty condition, which is not the case in the real situation.

4.2.2. VGAN model

To illustrate the VGAN model and DCGAN model for health condition monitoring, a comparison with the generative data ability of the two models is made first. Based on the same datasets, the first 15% of the dataset (see section I in Fig. 10) as the reference condition is treated as healthy and used for training the VGAN again. Fig. 14 (a) and (b) visualize the loss functions of G (generator) as well as D (discriminator). It can be seen that after about 1200 iterations, the DCGAN model is stabilized, whereas with the same number of training epochs, the loss functions of VGAN shown in Fig. 13 (a) and (b), do not show a stable trend, i.e., the output of the loss functions are not stabilized. From Fig. 14 (a) and (b), it is clearly shown that the DCGAN outperforms the VGAN in terms of stability for the same number of iterations.

Furthermore, from Fig. 15 (a), especially the zoomed-in region, it is clear that the generated data sequence from the VGAN is not matching with the original data correctly. In DCGAN, however, the generated samples presented in Fig. 15 (b) are matching well with the real data. Thus, the DCGAN model can be used as a promising tool for monitoring the health conditions of wind turbine bearings.

4.2.3. DCGAN model

The proposed scheme based on the DCGAN model is now applied to monitor the health conditions of the wind turbine bearing. The datasets are split into training and similarly testing sets, as have done in Section 4.2.1. The training dataset containing solely healthy data sequences (May 31, 2015) are illustrated in section I of Fig. 16. In the training process of DCGAN, the hyper-parameters (see Eq. (3)) for training are evaluated empirically. Both subnetworks D and G are trained utilizing the RMSprop optimizer. For training of $D_\phi(x)$ [47], it provides meaningful learning curves that are useful for debugging and hyper-parameter exploration, and the optimizer is utilized with a learning rate of 2×10^{-6}. The same learning rate is also applied to train G. Dropout in both subnetworks is used with a probability of 0.2. Finally, adversarial training is done for up to 5000 epochs.

To illustrate the DCGAN metric ($D_\phi(x)$ in Nash equilibrium) in the DCGAN model, the probability density function (PDF) of the

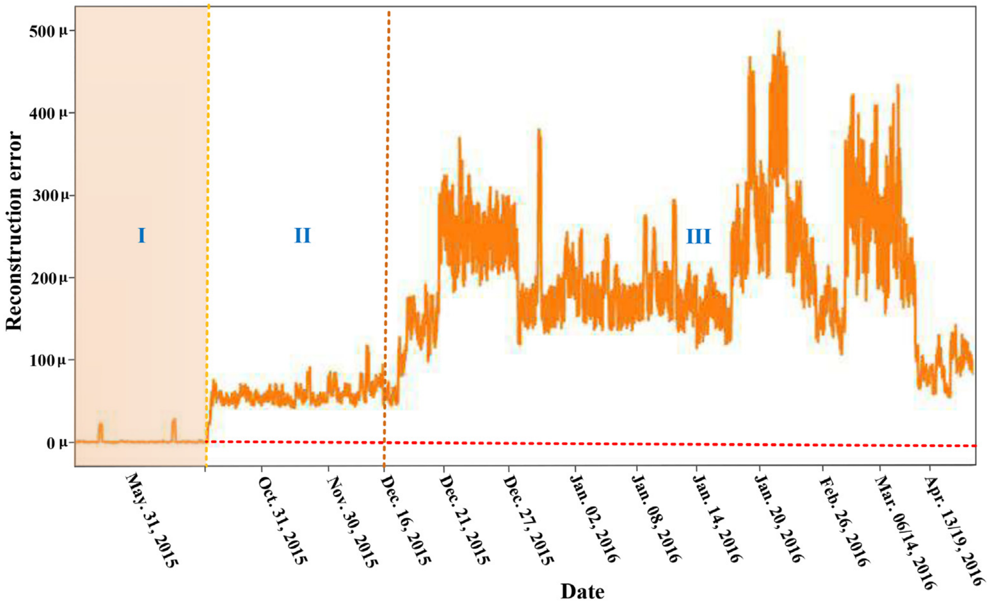

Fig. 10. Reconstruction error of the wind turbine dataset trended over almost 1 year. Section I with light yellow highlighted section represents the data used for training of the AE model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

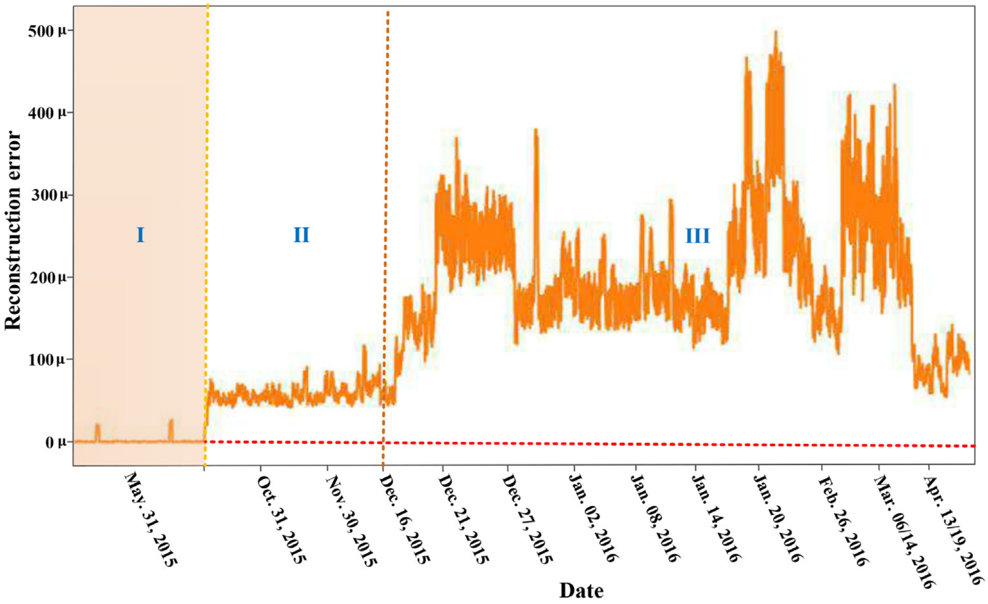

Fig. 11. Reconstruction error of the wind turbine dataset trended over almost 1 year. Section I with light yellow highlighted section represents the data used for training of the DAE model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

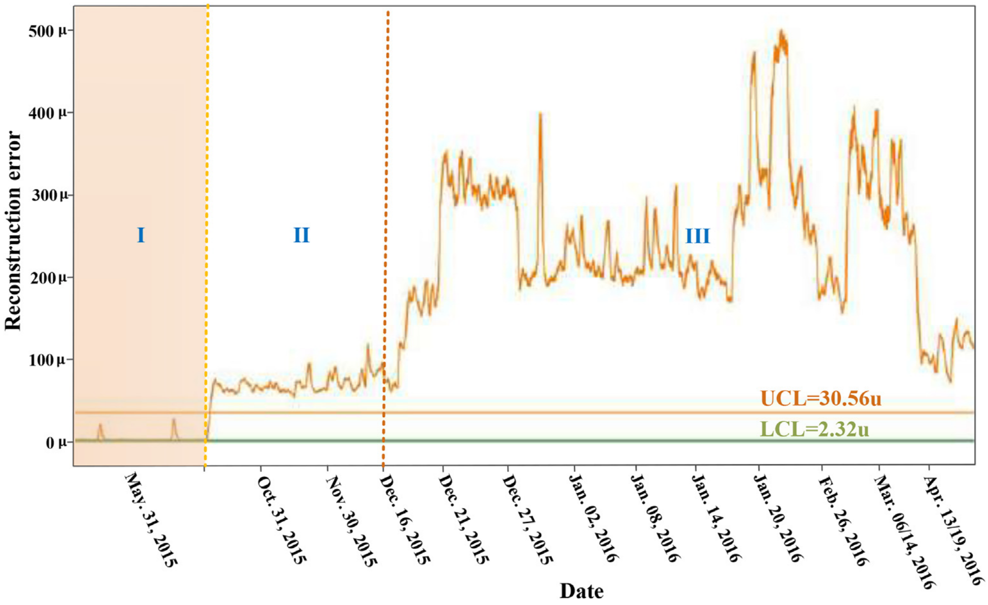

Fig. 12. EWMA control chart of wind turbine dataset trended over almost 1 year.

DCGAN metric $D_{\phi}(x)$ values of healthy and fault samples are investigated. As has been discussed in Section 3, the automatic setting of a threshold (Nash equilibrium) can be obtained by training on healthy samples (see section I in Fig. 16), and the resultant DCGAN metric can be treated as reference or threshold.

By normalizing the DCGAN metric, the sample discrepancies between health and faulty states are shown in Fig. 16. In Fig. 18 (a) (a healthy sample form May 31, 2015) and Fig. 19 (a) (a fault sample form Dec. 21, 2015), the probability density function (PDF) of DCGAN metric are displayed. The failure sample is selected in one stage of section III in Fig. 16. It can be seen that the estimated cumulative probability (see Fig. 18 (b)) and probability (see Fig. 19 (b)) are matched well with the original samples. From the estimated PDFs, the mean values can be calculated. They are 0.963 and 0.801, respectively. In Fig. 20, it can be observed that there is no overlap between the two samples, and the two states can easily be separated. Consequently, the DCGAN metric shows that its PDF can indeed separate the healthy and unhealthy samples and therefore monitor the health conditions of wind turbine bearing.

Examining Fig. 16 again, the distribution of the DCGAN metric can be described as follows,

1. The DCGAN metric value in section I stays constant. Specifically, its minimum and maximum values from 0.962 to 0.964.
2. The DCGAN metric value ranges from 0.937 to 0.971 in section II. Although there is still a difference from section I but it is small and can almost be treated as the same as section I.
3. Compared to sections I and II, DCGAN metric values in section III (from 0.77 to 0.92) fluctuate over a wider range.
4. A large drop occurred on Mar. 14, 2016 with a minimum value 0.08 in section IV.

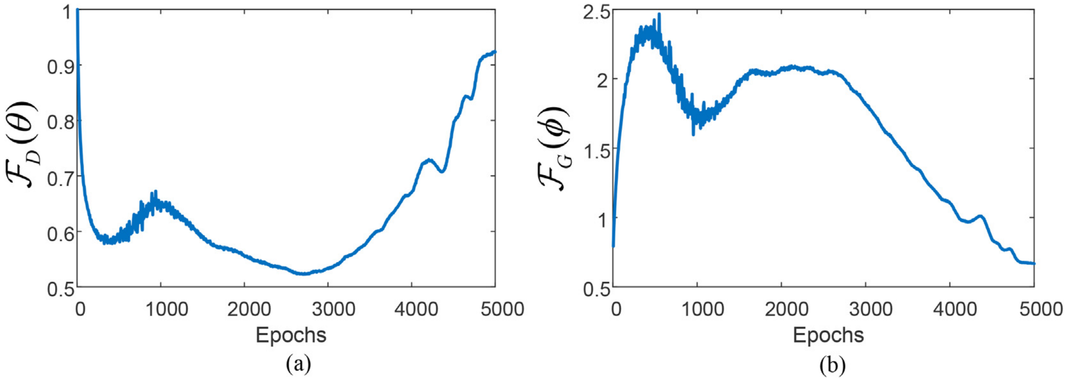

Fig. 13. VGAN model: (a) Loss function of discriminator (b) Loss function of generator.

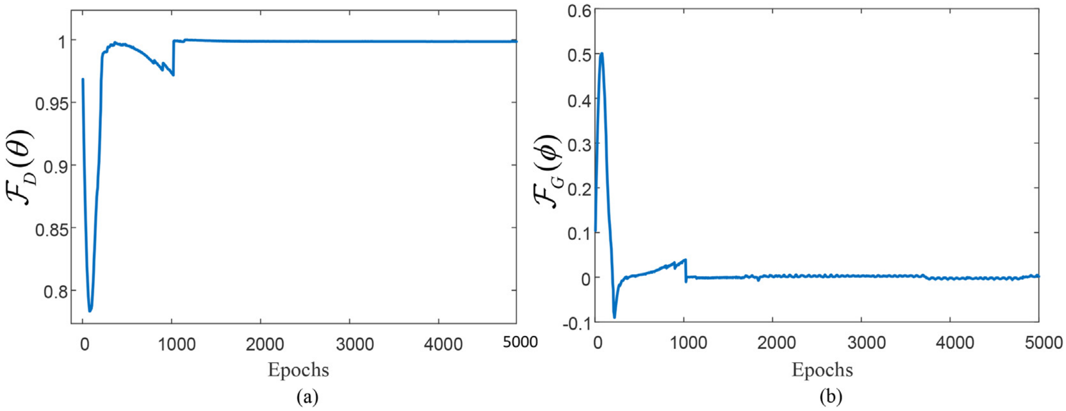

Fig. 14. DCGAN model: (a) Loss function of discriminator (b) Loss function of generator.

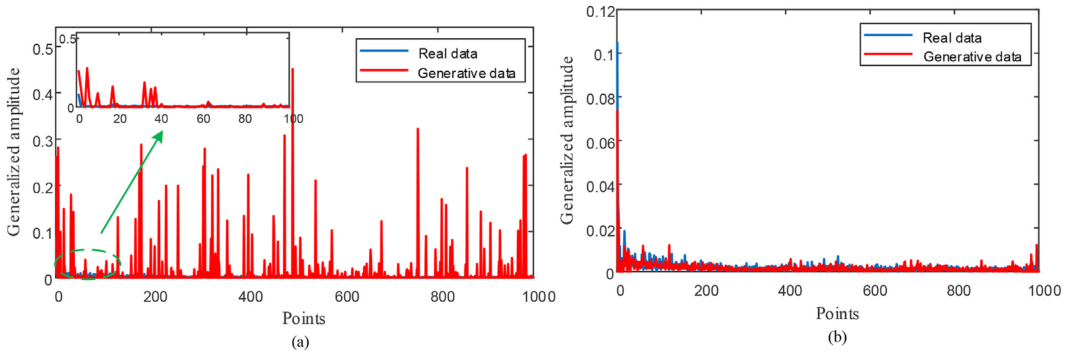

Fig. 15. (a) Comparison between generator network and real samples by VGAN (b) Comparison between generator network and real samples by DCGAN.

5. In section V, the DCGAN metric value ranges from 0.90 to 0.96. It returns back to almost 1.0 again after Apr. 2016.

From the observations in Fig. 16, it may be inferred that,

1. The DCGAN metric features almost are the same values in sections I and II. It indicates that the monitored bearing was essentially in similar health conditions, as was the case in reality. Compared with AE and DAE, DCGAN shows an improved ability to adapt to new data samples. That may be attributed to the fact that DCGAN is trained to generate data samples while AE and DAE are trained to represent given samples.
2. From the fluctuating DCGAN metric distribution in section III, it follows that the monitored bearing health condition is deteriorating. Especially, in section IV, the sudden drop indicates that a severe fault occurred in the wind turbine bearing.
3. A new bearing replacement was implemented (see in Fig. 17) by the maintaining engineers on Mar. 14, 2016. The DCGAN metric in section V returned to almost 1.0. It also proves that the monitored bearing returned to the healthy state after the replacement.

Further, comparing Fig. 10 and Fig. 11, the RE values of AE and DAE do not show an evident sudden change of values on Mar. 14, 2016. DCGAN metric in Fig. 16 indeed offers advantages in indicat-

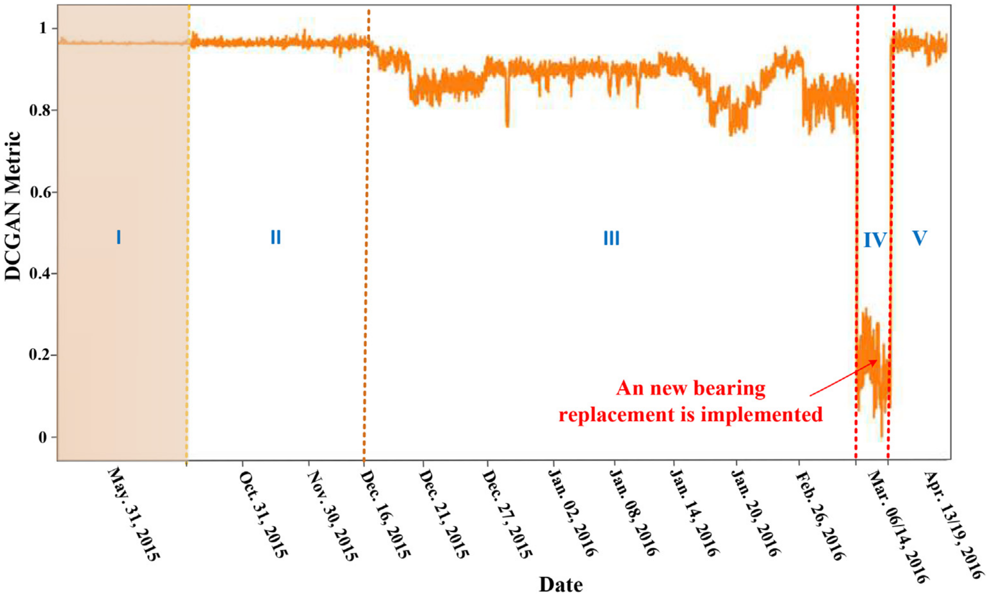

Fig. 16. DCGAN metric distribution of wind turbine bearing dataset (from the wind turbine in LU NAN wind farm) trended over almost 1 year. Section I with light yellow highlighted section represents the data used for training of the DCGAN model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

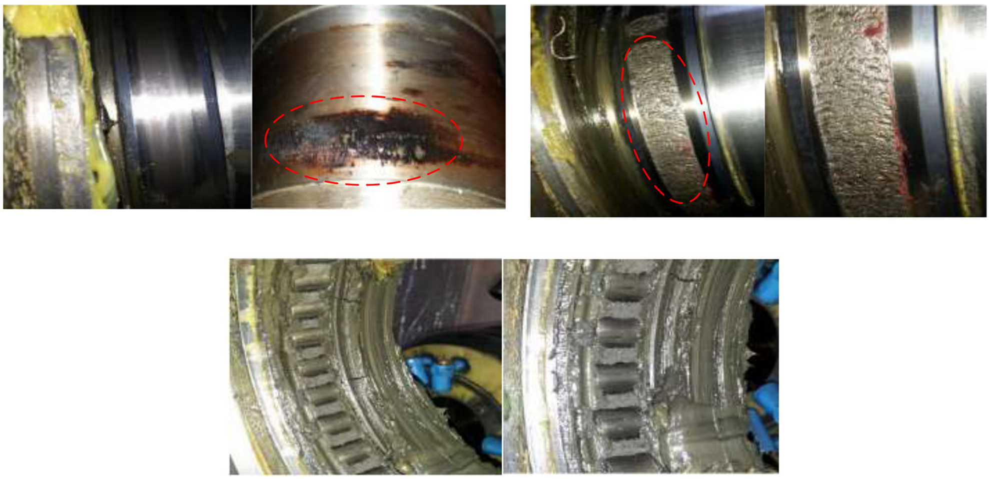

Fig. 17. Failures in inner race of rolling bearings (The regions of the dotted line are the defect with worn inner race).

ing the change of states. Meanwhile, the observations of Fig. 16 were validated by the replacement of bearing by the maintenance team. To highlight the advantages of the proposed threshold self-setting health condition monitoring scheme, a table (see in Table 3) summarizs the different methods for health monitoring of wind turbine bearing. The real bearing condition is shown in Fig. 17.

4.2.4. Tracking fault severity of wind turbine bearing

In the section, as mentioned above, 4.2.3, the failure measurements of the DCGAN metric renders the fluctuating result shown in Fig. 16. It can be seen as a qualitative evaluation or visual description. It is noted that the original output of the G network is a data set with several values. It is not convenient for condition monitoring purposes, thus for convenience sake, a single indicator monitoring indicator function (MIF) is introduced for monitoring.

The probability distribution of the DCGAN metric (PDM), is denoted as

$$S_{pdm} = \sum_{j=1}^{M} x^i_{pdm}, i = 1, 2, 3, \ldots, N; j = 1, 2, 3, \ldots, M \tag{5}$$

where x^i_{pdm} is the PDM sample with n points in each sample, and S_{pdm} is the datasets with M samples.

For a set of sample points $x^i_{pdm}, i = 1, 2, 3, \ldots, N$, the estimated PDF $p(x_{pdm})$ using the kernel density estimation [48] at point x_{pdm} is defined as

$$p(x_{pdm}) = \frac{1}{N\sigma}\sum_{i=1}^{N} K\left(\frac{x_{pdm}-x^i_{pdm}}{\sigma}\right) \tag{6}$$

where $K(\cdot)$ is the kernel function and σ is the bandwidth. The optimal value for σ is determined using the method discussed in [48]. A Gaussian kernel is utilized in this paper as follows,

$$K(g) = \frac{e^{\frac{g^2}{2}}}{\sqrt{2\pi}} \tag{7}$$

Table 3
Comparison of health condition monitoring performance by the different methods

Methods	AE	DAE	VGAN	DCGAN
Data generation ability	×	×	√	√
HCM threshold	RE[a]	RE[b]	VGAN metric	DCGAN metric[c]
Training/test samples	942/404	942/404	942/404	942/404
Validity for WT bearings	×	×	×	√

[a] Reconstruction error (RE) evaluations through receiver operating characteristic (ROC).

[b] Reconstruction error (RE) evaluations through EWMA control chart or Extreme value theory based on exponential distribution.

[c] DCGAN metric is the output of subnetwork Discriminator, i.e.,$D_\phi(x)$ in Nash equilibrium.

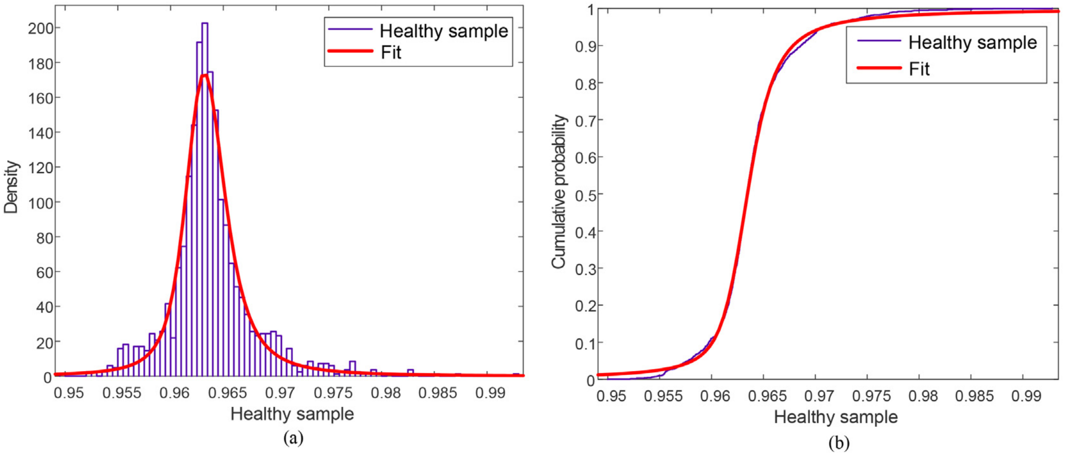

Fig. 18. Healthy sample based on the wind turbine: (a) Probability density function of DCGAN metric (b) Cumulative probability.

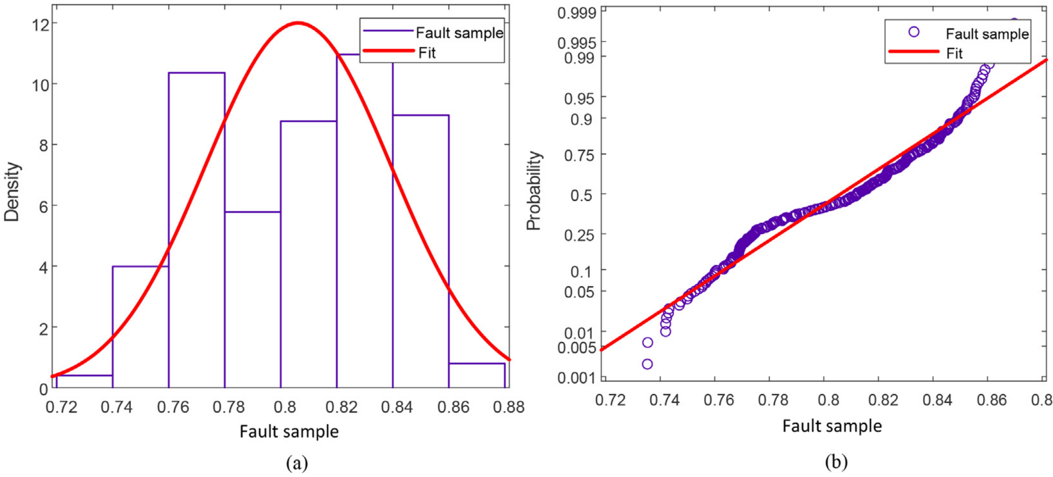

Fig. 19. Fault sample based on the wind turbine: (a) Probability density function of DCGAN metric (b) Probability.

The PDM sample based on the healthy condition is shown in Fig. 18 (a). Its cumulative distribution function is given as follows,

$$\int_{-\infty}^{+\infty} p(x_{pdm})dx_{pdm} = 1 \tag{8}$$

From Fig. 18 (b), it is clear that the estimated cumulative distribution function fits the original distribution well. Therefore, PDM follows a normal distribution $x_{pdm} \sim N(\mu_j, \sigma_j)$ with mean value μ_j and standard deviation σ_j. To avoid the influences of some strange peak values in a small failure dataset, the *three-sigma rule of thumb* is utilized to exclude outliers. Then, the valid value points between $\mu_j - 3\cdot\sigma_j$ and $\mu_j + 3\cdot\sigma_j$ are considered. Finally, the monitoring indicator function (MIF) can be calculated and utilized to track the fault severity.

$$\begin{aligned} MIF^j \; &= \sqrt{\tfrac{1}{N}\textstyle\sum_{i=1}^{N}(x_{pdm}^i - \mu_j)} - \sqrt{\tfrac{1}{N}\textstyle\sum_{i=1}^{N}(x_{pdm}^i - \mu_1)} \\ &i = 1,2,3,\ldots,N; j = 1,2,3,\ldots,M \\ &\mu_j - 3\cdot\sigma_j < x_{pdm}^i \leqslant \mu_j + 3\cdot\sigma_j \end{aligned} \tag{9}$$

where μ_1 is the mean value of PDM based on the healthy condition.

Fig. 21 shows the MIF results v.s. date, it is observed that the MIF can track the fault severity. A sudden increase of MIF at Mar. 14, 2016 is found, and its corresponding MIF is reached at almost 3.5×10^{-2}. This represents the worst condition of the wind turbine bearing. Afterward, bearing replacement is performed, and the MIF

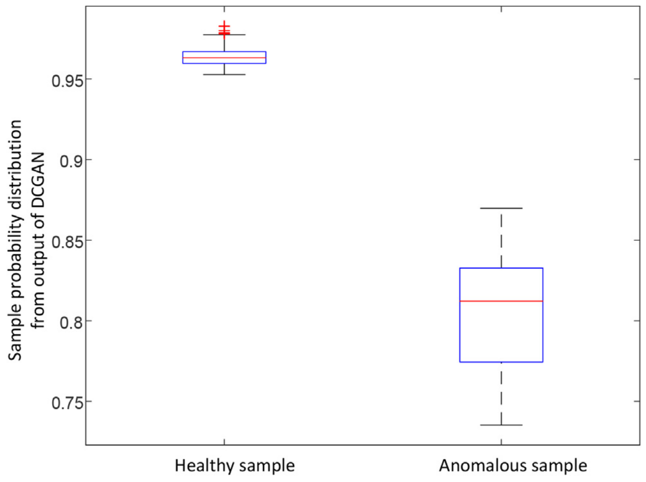

Fig. 20. Sample probability distribution from the output (DCGAN metric) of the DCGAN model.

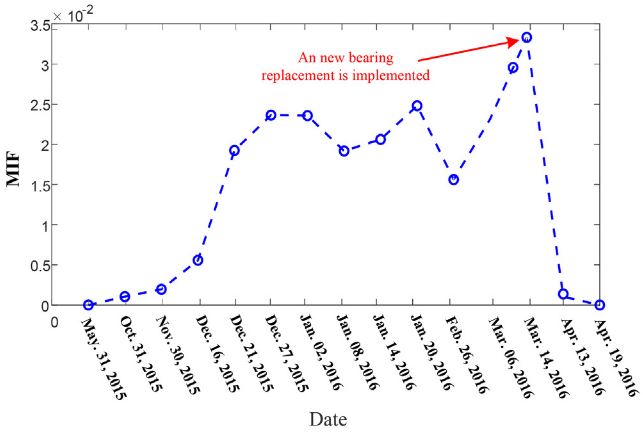

Fig. 21. Tracking the fault severity based on the wind turbine.

returned to 0 on Apr. 19, 2016. The result of the DCGAN metric in Fig. 16 is quantified and simplified by this proposed MIF.

5. Conclusion

For the AE based methods, the rule of thumb for training is to minimize the reconstruction error (RE) within a set of given samples. Instead of representing given samples, GAN based methods, on the other hand, generate new samples that allow these methods to accommodate more data sample variances with a stronger generalization ability. Furthermore, the use of Nash Equilibrium for training enables GAN based methods to become self-defined evaluators with a high level of consistency, with less human intervention. Hence, an attempt to make use of the self-defining ability of the DCGAN model is investigated. In the proposed method, the threshold value can be automatically created by the output of the G network in the DCGAN model. In other words, the determination of the threshold value for monitoring is shifted from subjective threshold calculation to the data itself. This advantage can be fully exploited for setting appropriate thresholds for health condition monitoring (HCM) purposes.

To this end, this paper develops a threshold self-setting HCM scheme based on a DCGAN model to monitor a wind turbine bearing. The issue of category imbalance can be solved mainly by the DCGAN's powerful capability in learning an underlying representation from the purely healthy data sequences. Furthermore, a monitoring indicator function is developed to measure the fault severity of the wind turbine bearing quantitatively. The industrial wind turbine data from real wind farm has successfully proved the effectiveness of the proposed scheme. For the future work, the massive amounts of data from multiple sensors (e.g., SCADA analysis) could be considered for wind turbine generator bearings in health condition monitoring and fault diagnostics.

CRediT authorship contribution statement

Peng Chen: Conceptualization, Data curation, Formal analysis, Methodology, Investigation, Software, Writing-Original draft. **Yu Li:** Visualization, Investigation, Validation. **Kesheng Wang:** Project administration, Resources, Supervision, Funding Acquisition. **Ming J. Zuo:** Supervision, Writing - review & editing. **P. Stephan Heyns:** Writing - review & editing. **Stephan Baggeröhr:** Writing - review & editing.

Declaration of Competing Interest

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

Acknowledgments

This research is supported by the National Key Research and Development Program of China (Project No. 2016YFB1200401 and 2017YFC0108401), Fundamental Research Funds for the Centered University (ZYGX2016J111), National Natural Science Foundation of China (51305067), and the Natural Sciences and Engineering Research Council of Canada (Grant #RGPIN-2015–04897).

We want to thank Mr. Binyuan Yang and Mr. Yu Wang, who are employed in Chengdu Forward Technology Corporation Limited for their assistance with the acquisition of wind turbine dataset. We would like to thank Mr. Binnyam Idris for the language editing of the manuscript. We also would like to express our gratitude to Mr. Wihan Booyse who are a director of artificial intelligence at Kriterion for rendering valuable comments that significantly enhanced this research.

Appendix A. Autoencoder (AE) model

The AE and DAE are used to compare with the proposed DCGAN scheme. Therefore, the two methods are introduced in the following.

An AE model [30] is usually treated as an unsupervised manner to learn the underlying representations from fault-free data. Generally, as depicted in Fig. A.22, an AE model is comprised of two parts, i.e., encoder network and decoder network. The encoder network transforms the raw input data from a high-dimensional space into an underlying representation with a low-dimensional space. Then, this latent representation further serves as input to the decoder network, which is trained to reconstruct the original input data. We are given the raw input data $\mathbf{x}^i$ from a dataset $\{x^i\}_{i=1}^{M}$, and each $\mathbf{x}^i$ has n dimensions. Typically, the encoder network transforms $\mathbf{x}^i$ to a hidden layer h^i, which can be mapped by a nonlinear function,

$$h^i = f(\mathbf{W} \cdot \mathbf{x}^i + \mathbf{b}) \tag{A.1}$$

where $f(.)$ is a nonlinear activation function, such as the sigmoid function. $\mathbf{W} \in \mathbb{R}^{d\times m}$is the weight matrix and $\mathbf{b} \in \mathbb{R}^{d}$ is the bias vector

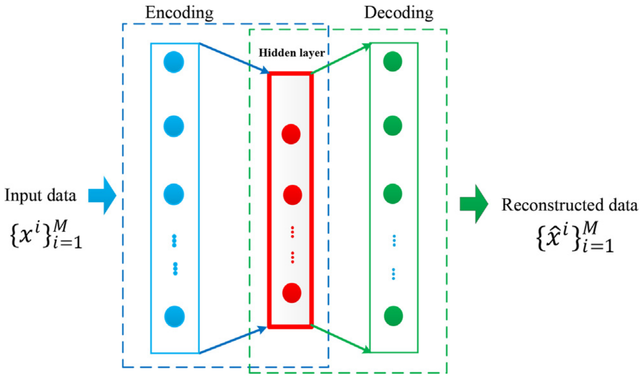

Fig. A.22. Schematic diagram of AE model.

which will be optimized in the encoding process with d nodes in the hidden layer.

The decoder network attempts to transform the corresponding hidden layer to a reconstructed vector $\hat{\mathbf{x}}^i$. This reconstructed vector $\hat{\mathbf{x}}^i$ is determined by,

$$\hat{\mathbf{x}}^i = g(\mathbf{W}' \cdot h^i + \mathbf{c}) \tag{A.2}$$

where the parameters $\mathbf{W}' \in \mathbb{R}^{d\times m}$and $\mathbf{c} \in \mathbb{R}^d$ are the weight matrix and the bias vector respectively. $g(.)$ is also a nonlinear function (typically sigmoid function). In this model, the weight matrix $\mathbf{W}'$ is usually selected as $\mathbf{W}' = \mathbf{W}^T$, which is refereed to as the tied weights for better learning performance [49].

To reconstruct the original data as closely as possible, parameter sets from the encoder and the decoder are optimized. It can be realized by minimizing the reconstruction error (RE) $L(\mathbf{x}^i, \hat{\mathbf{x}}^i)$, where $L(\mathbf{x}^i, \hat{\mathbf{x}}^i)$ is a loss function used for evaluating the discrepancy between $\mathbf{x}^i$ and $\hat{\mathbf{x}}^i$ [50]. This parameter optimization process can be implemented with stochastic gradient decent (SGD) or second-order gradient algorithms, such as limited-memory Broyden-Feletcher-Goldfarb-Shanno(L-BFGS) [51]. To achieve a stable and fast training process, the algorithm of L-BFGS is usually utilized. The minimizing reconstruction error is shown below

$$\begin{aligned} RE &= \underset{\Theta}{minimize} \frac{1}{M}\sum_{i=1}^{M} L(\mathbf{x}^i, \hat{\mathbf{x}}^i) \\ &= \underset{\Theta}{minimize} \frac{1}{M}\sum_{i=1}^{M} L(\mathbf{x}^i, g(\mathbf{W}' \cdot h^i + \mathbf{c})) \\ &= \underset{\Theta}{minimize} \frac{1}{M}\sum_{i=1}^{M} L(\mathbf{x}^i, g(\mathbf{W}^T \cdot f(\mathbf{W} \cdot \mathbf{x}^i + \mathbf{b}) + \mathbf{c})) \end{aligned} \tag{A.3}$$

where $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{c})$, and the loss function of $L(\mathbf{x}^i, \hat{\mathbf{x}}^i)$ can be calculated as,

$$L(\mathbf{x}^i, \hat{\mathbf{x}}^i) = \| \mathbf{x}^i - \hat{\mathbf{x}}^i \| \tag{A.4}$$

For industry examples, to adapt the complex noisy circumstances and enforce algorithm robustness, denoising autoencoder model (DAE) [49] is developed. The key task of the DAE model is to transform the original data $\mathbf{x}^i$ to a corrupted data $\tilde{\mathbf{x}}^i$. For each $\tilde{\mathbf{x}}^i$, some values of $\mathbf{x}^i$ are set to be 0 and others remain unchanged [49].

References

[1] W. Qiao, D. Lu, A survey on wind turbine condition monitoring and fault diagnosis Part I: Components and subsystems, IEEE Trans. Industr. Electron. 62 (2015) 6536–6545.

[2] G. Jiang, P. Xie, H. He, J. Yan, Wind turbine fault detection using a denoising autoencoder with temporal information, IEEE/ASME Trans. Mechatron. 23 (2018) 89–100.

[3] G.M.J. Herbert, S. Iniyan, E. Sreevalsan, S. Rajapandian, A review of wind energy technologies, Renew. Sustain. Energy Rev. 11 (2007) 1117–1145.

[4] B. Lu, Y. Li, X. Wu, Z. Yang, A review of recent advances in wind turbine condition monitoring and fault diagnosis, in: 2009 IEEE Power Electronics and Machines in Wind Applications, IEEE, pp. 1–7.

[5] Z. Hameed, Y.S. Hong, Y.M. Cho, S.H. Ahn, C.K. Song, Condition monitoring and fault detection of wind turbines and related algorithms: a review, Renew. Sustain. Energy Rev. 13 (2009) 1–39.

[6] C.C. Ciang, J.-R. Lee, H.-J. Bang, Structural health monitoring for a wind turbine system: a review of damage detection methods, Meas. Sci. Technol. 19 (2008), 122001.

[7] K.S. Wang, P.S. Heyns, Application of computed order tracking Vold-Kalman filtering and emd in rotating machine vibration, Mech. Syst. Signal Process. 25 (2011) 416–430.

[8] P. Chen, K. Wang, K. Feng, Application of order-tracking holospectrum to cracked rotor fault diagnostics under nonstationary conditions, in: Prognostics and System Health Management Conference (PHM-Chengdu), IEEE, 2016, pp. 1–6.

[9] P. Chen, K. Wang, M.J. Zuo, A generalized synchroextracting transform for fast and strong frequency modulated signal analysis, Cond. Monit. Diagnostic Eng. Manage. (2018) 189.

[10] P. Chen, K. Wang, M.J. Zuo, D. Wei, An ameliorated synchroextracting transform based on upgraded local instantaneous frequency approximation, Measurement 106953 (2019).

[11] J.P. Assendorp, Deep learning for anomaly detection in multivariate time series data, Ph.D. thesis, Hochschule für Angewandte Wissenschaften Hamburg, 2017.

[12] J. Wang, Y. Liang, Y. Zheng, R.X. Gao, F. Zhang, An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples, Renewable Energy 145 (2020) 642–650.

[13] E. Hart, A. Turnbull, J. Feuchtwang, D. McMillan, E. Golysheva, R. Elliott, Wind turbine main-bearing loading and wind field characteristics, Wind Energy (2019).

[14] L. Wen, X. Li, L. Gao, Y. Zhang, A new convolutional neural network-based data-driven fault diagnosis method, IEEE Trans. Industr. Electron. 65 (2017) 5990–5998.

[15] G. Jiang, H. He, P. Xie, Y. Tang, Stacked multilevel-denoising autoencoders: a new representation learning approach for wind turbine gearbox fault diagnosis, IEEE Trans. Instrum. Meas. 66 (2017) 2391–2402.

[16] Z. Liu, L. Zhang, A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings, Measurement 107002 (2019).

[17] J.A. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing 48 (2002) 85–105.

[18] A. Statnikov, L. Wang, C.F. Aliferis, A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification, BMC Bioinformatics 9 (2008) 319.

[19] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (1998) 18–28.

[20] S. Karamizadeh, S.M. Abdullah, M. Halimi, J. Shayan, M. javad Rajabi, Advantage and drawback of support vector machine functionality, in: 2014 International Conference on Computer, Communications, and Control Technology (I4CT), IEEE, pp. 63–65.

[21] J. Li, X. Zhang, X. Zhou, L. Lu, Reliability assessment of wind turbine bearing based on the degradation-hidden-Markov model, Renewable Energy 132 (2019) 1076–1087.

[22] R. Liu, B. Yang, E. Zio, X. Chen, Artificial intelligence for fault diagnosis of rotating machinery: a review, Mech. Syst. Signal Process. 108 (2018) 33–47.

[23] H. Oh, J.H. Jung, B.C. Jeon, B.D. Youn, Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis, IEEE Trans. Industr. Electron. 65 (2018) 3539–3549.

[24] O. Janssens, R. Van de Walle, M. Loccufier, S. Van Hoecke, Deep learning for infrared thermal image based machine health monitoring, IEEE/ASME Trans. Mechatron. 23 (2017) 151–159.

[25] M.H. Rafiei, H. Adeli, A novel unsupervised deep learning model for global and local health condition assessment of structures, Eng. Struct. 156 (2018) 598–607.

[26] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R.X. Gao, Deep learning and its applications to machine health monitoring, Mech. Syst. Signal Process. 115 (2019) 213–237.

[27] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: International Conference on Neural Information Processing Systems, pp. 1097–1105.

[28] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: review of methods and applications, Expert Syst. Appl. 73 (2017) 220–239.

[29] S. Vluymans, A. Fernández, Y. Saeys, C. Cornelis, F. Herrera, Dynamic affinity-based classification of multi-class imbalanced data with one-versus-one decomposition: a fuzzy rough set approach, Knowl. Inf. Syst. (2017) 1–30.

[30] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.

[31] L. Wang, Z. Zhang, J. Xu, R. Liu, Wind turbine blade breakage monitoring with deep autoencoders, IEEE Trans. Smart Grid (2016) 2824–2833.

[32] F. Cheng, Q. Zhang, Y. Tian, X. Zhang, Maximizing receiver operating characteristics convex hull via dynamic reference point-based multi-objective evolutionary algorithm, Appl. Soft Comput. 86 (2020), 105896.

[33] B. Zaman, M.H. Lee, M. Riaz, et al., An improved process monitoring by mixed multivariate memory control charts: an application in wind turbine field, Comput. Indust. Eng. 142 (2020), 106343.

[34] H. Zhao, H. Liu, W. Hu, X. Yan, Anomaly detection and fault analysis of wind turbine components based on deep learning network, Renewable Energy 127 (2018) 825–834.

[35] C. Che, H. Wang, X. Ni, Q. Fu, Domain adaptive deep belief network for rolling bearing fault diagnosis, Comput. Indust. Eng. 106427 (2020).

[36] P. Chen, Y. Li, K. Wang, M.J. Zuo, A novel knowledge transfer network with fluctuating operational condition adaptation for bearing fault pattern recognition, Measurement 107739 (2020).

[37] P. Liang, C. Deng, J. Wu, Z. Yang, Intelligent fault diagnosis of rotating machinery via wavelet transform, generative adversarial nets and convolutional neural network, Measurement 107768 (2020).

[38] Y. Pang, G. Hu, Distributed nash equilibrium seeking with limited cost function knowledge via a consensus-based gradient-free method, IEEE Trans. Autom. Control (2020).

[39] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, arXiv:1406.2661 [cs, stat], 2014. arXiv:1406.2661.

[40] J. Nash, Non-cooperative games, Ann. Math. 54 (1951) 286–295.

[41] J. Acquarelli, T. van Laarhoven, J. Gerretzen, T.N. Tran, L.M. Buydens, E. Marchiori, Convolutional neural networks for vibrational spectroscopic data analysis, Analytica chimica acta 954 (2017) 22–31.

[42] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv:1511.06434 [cs], 2015. arXiv: 1511.06434.

[43] L. Xu, P. Cartwright, Direct active and reactive power control of DFIG for wind energy generation, IEEE Trans. Energy Convers. 21 (2006) 750–758.

[44] K.-i. Kanatani, Fast Fourier transform, in: Particle characterization in technology, CRC Press, 2018, pp. 31–50.

[45] C.-W. Lu, M.R. Reynolds Jr., EWMA control charts for monitoring the mean of autocorrelated processes, J. Qual. Technol. 31 (1999) 166–188.

[46] S.S. Prabhu, G.C. Runger, Designing a multivariate EWMA control chart, J. Qual. Technol. 29 (1997) 8–15.

[47] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, arXiv:1701.07875 [cs, stat] (2017). arXiv: 1701.07875.

[48] P.-E.P. Odiowei, Y. Cao, Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations, IEEE Trans. Industr. Inf. 6 (2009) 36–45.

[49] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in, Proceedings of the 25th International Conference on Machine Learning, ACM (2008) 1096–1103.

[50] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[51] Q.V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, A.Y. Ng, On optimization methods for deep learning, Proceedings of the 28th International Conference on International Conference on Machine Learning, Omnipress (2011) 265–272.