Computers in Industry 106 (2019) 85–93

Contents lists available at ScienceDirect

Computers in Industry

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

Generative adversarial networks for data augmentation in machine fault diagnosis

Siyu Shao[a], Pu Wang[a], Ruqiang Yan[a,b,*]

[a] School of Instrument Science and Engineering, Southeast University, Nanjing, 210096, China
[b] School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, 710049, China

ARTICLE INFO

Article history:
Received 20 September 2018
Received in revised form 13 December 2018
Accepted 2 January 2019
Available online 9 January 2019

Keywords:
Auxiliary classifier generative adversarial networks
Fault diagnosis
Data augmentation
Signal generation
Induction motor

ABSTRACT

Generative adversarial networks (GANs) have been proved to be able to produce artificial data that are alike the real data, and have been successfully applied to various image generation tasks as a useful tool for data augmentation. In this paper, we develop an auxiliary classifier GAN(ACGAN)-based framework to learn from mechanical sensor signals and generate realistic one-dimensional raw data. The proposed architecture contains two parts, generator and discriminator, and both of them are built by stacking one-dimensional convolution layers to learn local features from the original input. Such stacked structure is able to learn hierarchical representations through convolution operation and easy to train. Batch normalization is performed within generator to avoid the problem of gradient vanishing during training, and category labels are used as the auxiliary information in this framework to help train the model. The proposed approach is designed to produce realistic synthesized signals with labels and the generated signals can be used as augmented data for further applications in machine fault diagnosis. In order to evaluate the performance of the generative model, we introduce a set of assessment to evaluate the quality of generated samples, including statistical characteristics and experimental verification. Finally, induction motor vibration signal datasets are utilized to investigate the effectiveness of the proposed framework.

1. Introduction

Machine learning (ML), especially deep learning (DL), has been highly successful in recent applications of machine condition monitoring and fault diagnosis. Compared with conventional methods that rely on manually selected features, DL-based fault diagnosis framework has powerful feature learning ability which has the capability to learn hierarchical representations directly from the original sensor data via multiple hidden layers, and automatically select discriminative features that are beneficial for accurate working condition classification. Advances in deep learning provide an opportunity to improve performance in machine fault diagnosis tasks [1–4].

For DL-based methods, training data is an important factor that affects the performance of deep architecture. A deep neural network contains multiple hidden layers, and the number of free parameters that needs to be trained is enormous. In order to achieve accurate predictions, the deep architecture needs to be well-trained, and fully training a large network usually requires abundant balanced data. However, in practice, training samples among different machine states are usually unbalanced. For example, for a mechanical system under operation, it is working under normal circumstance most of the time and the collected sensor data that represent positive training samples are sufficient, while machine operating in fault is infrequent and corresponding collected samples are limited compared to positive samples. Therefore, there is an unbalance between positive training samples and fault samples. Suffered from limited training data, especially unbalanced data, training a deep model to achieve accurate prediction of machine conditions is relatively difficult.

Generative adversarial networks (GANs) offer an alternative when facing the issue of unbalanced data classification by generating data for minor classes. GANs have shown prominent capability in producing realistic looking data and have been successfully applied in image generation. It can be used for data augmentation by generating artificial data that are similar to original data and thereby enriching training dataset. Moreover, the generative architecture models data generation process and therefore helps to understand the original data distribution, providing a new perspective for creating predictive systems in machine fault diagnosis tasks.

* Corresponding author at: School of Instrument Science and Engineering, Southeast University, No. 2 Sipailou, Nanjing, 210096, China.
E-mail addresses: cathygx.sy@gmail.com (S. Shao), wangpuupup@outlook.com (P. Wang), ruqiang@seu.edu.cn (R. Yan).

https://doi.org/10.1016/j.compind.2019.01.001
0166-3615/

GAN is first introduced as a framework for generation of artificial images [5] and has the capability to produce convincing image samples. However, the original GAN model suffers from training instability and elaborated regularization is needed to realize desirable performance. Therefore, more researches have been carried out in regard to model stability and generation quality. Wasserstein GAN (WGAN) leverages the Wasserstein distance to form a new loss function which contributes to better model stability than the original, and it related loss function to the quality of generated images [6], and an improved training strategy for WGAN is proposed to insure stable training of various GAN architecture [7]. A structure which combined deep convolutional generative adversarial networks (DCGANs) with certain architectural constraints is designed to learn a hierarchy of representations from object parts to scenes for unsupervised learning [8]. Semi-supervised learning using GAN is introduced to produce class labels in discriminator network and improve generated samples quality [9]. Afterwards, a new variant GAN, called auxiliary classifier GAN (ACGAN), using label information is proposed to generate high resolution images and achieves desirable performance in classification tasks [10].

GANs and their variants have been proven to be effective in image generation and recent applications in generation of artificial audio [11] and electroencephalographic (EEG) brain signals [12] also show their potentials for generating time series data. However, limited work has been investigated in generating raw data of sensor signals. Furthermore, for evaluation of GANs, most of the existing researches rely on visual estimation of the sample quality, which is not appropriate for the scenarios of sensor signals. Therefore, this paper is a first attempt for using ACGAN architecture to generate mechanical sensor signals for data augmentation together with subsequent fault classification, and it also proposes an evaluation system to assess the quality of generated samples. The main contributions of this paper can be summarized as follows:

1) An ACGAN architecture based on one-dimensional convolutional layers is proposed here to learn features from limited training data and generate realistic sensor data, and the high quality generated samples can be used in further applications in machine fault diagnosis. One-dimensional convolutional neural network (1D-CNN) is adopted as the building block of both generator and discriminator to leverage its capability of learning local and hierarchical representations from raw data which are beneficial for classification tasks and allow for feature interpretability [13]. Category labels are added in both generator and discriminator to help accelerate model training and batch normalization technique is performed in generator to overcome the problem of gradient vanishing and therefore avoid overfitting.
2) In order to assess the performance of the generative model, we demonstrate a set of metrics to evaluate the quality of generated samples quantitatively and visually. In order to analyze the diversity of the generated samples and true data, both time domain statistical characteristics and frequency distribution are calculated, and for classification performance, experiments using generated samples to train and test classifier are carried out to verify the effectiveness of generated samples in fault diagnosis tasks.
3) Experimental results using induction motor dataset are reported here, demonstrating the data generation ability of the proposed framework and providing classification results using artificially augmented dataset. Different building strategies for generator are investigated for comparison.

The rest of the paper is organized as follows. Theoretical background about generative adversarial networks, training strategy and the proposed assessment method is introduced in Section II. In Section III, the whole framework based on ACGAN is illustrated in detail. Experimental verification and discussion of the results are given in Section IV. Finally, conclusions and future work are listed in Section V.

2. Theoretical background

2.1. Generative adversarial networks (GANs)

A regular GAN consists of two parts which are trained in opposition to each other, the generator G and the discriminator D, shown in Fig. 1(a). The main thought behind GAN is using adversarial networks to improve the quality of generated data. The generator is trained to produce realistic synthesized data $x_{generated} = G(z)$ from a random noise vector z, trying to fool the discriminator so that $x_{generated}$ would not be recognized as generated samples. The generated distribution is denoted as P_g. On the contrary, the discriminator takes both the real training data and fake samples generated from generator as the input, and then it is trained to distinguish between generated samples and real data. Discriminator outputs the probability that certain samples corresponds to possible data sources [10]. The loss function of discriminator L can be defined as:

$$\begin{aligned} L &= E_{x\sim P_{data}}[\log p(s = real|x_{real})] + E_{z\sim P(z)}[\log p(s = generated|x_{generated})] \\ &= E_{x\sim P_{data}}[\log(D(x)] + E_{z\sim P(z)}[\log(1 - D(G(z)))] \end{aligned} \quad (1)$$

Where P_{data} is the real data distribution, $P(z)$ is a prior distribution on noise vector z,$D(x)$ denotes the probability that x comes from the real data rather than generated data, $E_{x\sim P_{data}}$ represents the expectation of x from real data distribution P_{data}, and $E_{z\sim P(z)}$ is the expectation of z sampled from noise. For discriminator, the goal of training is to maximize the loss function which means maximizing the log-likelihood with correct sample source, while for generator, the training goal is to minimize the second term in Eq. (1) to confuse the discriminator. Therefore, the goal of the whole GAN architecture can be summarized as [14] :

$$Goal = arg \min_G \max_D L(G, D) \quad (2)$$

Updating model parameters based on the objective function is able to train GAN by stochastic gradient descent (SGD) and achieve proper pair of discriminator and generator.

2.2. Auxiliary classifier generative adversarial networks (ACGANs)

A variant architecture of the regular GAN is achieved by leveraging additional class labels for both discriminator and generator, shown in Fig. 1(b). Class conditional generation process

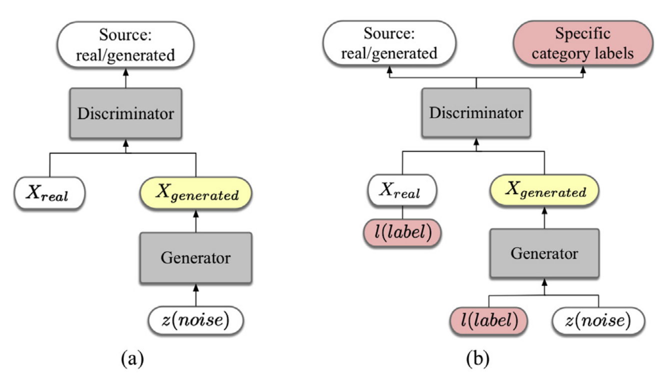

Fig. 1. Typical architecture of (a) regular GAN, and (b) Auxiliary classifier GAN.

is able to improve the quality of generated data [10]. In addition, the discriminator is combined with an auxiliary part to output specific class labels so that the improved discriminator can not only recognize data source, but also differentiate among various classes. This variant that combines class conditional architecture and auxiliary network for classification is called auxiliary classifier generative adversarial network (ACGAN).

Compared with regular GANs, ACGANs is able to generate high quality data and provide label information simultaneously. Mathematically, generator leverages both noise z and label l to produce artificial data samples $X_{generated} = G(z, l)$, and discriminator outputs probability on data sources and class labels. Therefore, the objective function contains two separate log-likelihoods corresponding to correct data source and correct class label, shown as:

$$L_{Source} = E_{x \sim P_{data}}[\log p(s = real|x_{real})] + E_{z \sim P(z)}[\log p(s = generated|x_{generated})] \tag{3}$$

$$L_{Class} = E_{x \sim P_{data}}[\log p(Class = c|x_{real})] + E_{z \sim P(z)}[\log p(Class = c|x_{generated})] \tag{4}$$

For discriminator, training goal is to maximize both log-likelihood $L_{Source} + L_{Class}$, while generator is trained to maximize $L_{Class} - L_{Source}$.

The improved architecture ACGAN can generate data samples with better quality owing to additional class label information, so that it is suitable for supervised classification tasks.

2.3. Model evaluation

The goal of using GAN architecture is generating convincing data samples to realize data augmentation, therefore, the quality of generated data samples is significant. It is important to evaluate the similarity between generated data and the real data [14]. Model evaluation for GAN architecture is still an open issue and several evaluation metrics are proposed to provide quantitative assessment, including inception score (IS) [15], the Frechet inception distance (FID) [16], and sliced Wasserstein distance (SWD) [17]. For conventional image generation tasks, it is also possible to conduct visual evaluation. However, for time series signals, especially mechanical sensor signals, it is not suitable to directly calculate the score using model pre-trained on natural images, and considering the inherent property of sensor data, time domain features and frequency distribution is able to reflect signal characteristics to some degree. Hence, the evaluation metrics is designed based on these statistical characteristics. In this paper, we demonstrate two types of approaches to assess the performance of generative model, statistical indicators and experimental verification.

For the statistical features, Euclidean distance (ED), Pearson correlation coefficient (PCC) and Kullback–Leibler divergence (K–L) are introduced here to evaluate the similarity between generated samples and the real training data. These metrics are calculated to investigate the capability that generator model the distribution of training data.

ED is a direct way to measure the distance between two samples and evaluate their similarity. For a set of generated samples, the average ED is calculated to measure the distance between generated sample distribution and the real distribution. PCC is a measure of the correlation between two variables and is used to evaluate the strength of linear correlation between them. K–L divergence (K–L) is a factor to evaluate the difference between two probability distributions.

Furthermore, we introduce a unique training and testing strategy to evaluate whether the generated samples are suitable as a training dataset independently in fault classification tasks. First, the real dataset is divided into two parts, training and testing datasets. Then, training dataset is used to train the generative model and a set of fake data is achieved. Then the dataset generated by the GAN is used as a training set to train a model, and the trained model is validated by testing data. The testing results including model loss and classification accuracy indicate the quality of the generated dataset and demonstrate the ability of the generated samples to be used for further practical applications.

3. System framework and model training

3.1. System framework design

We aim to establish a framework using ACGAN to generate high-quality artificial sensor signals for data augmentation and the generated data is able to supplement unbalanced dataset for further applications. In practice, unbalanced training dataset is not suitable for training model to perform fault classification, and the whole dataset is limited by the data from minor class. Thus, high quality data generated by GAN-based model is able to enrich the original dataset, which contributes to accurate fault diagnosis. This paper focuses on artificial data generation and samples evaluation.

The whole architecture contains two main parts: one is data generation based on generative adversarial network, the other is generated data assessment and application of augmented data, illustrated in Fig. 2.

Firstly, an improved ACGAN architecture is designed to generate sensor signals. The proposed ACGAN combines generator and discriminator together. The generator is able to generate samples from latent space with specific label and try to confuse the discriminator. The generated samples and real training data are mixed together before being sent to the discriminator. For each sample, discriminator is able to output two labels, one indicating whether the sample is real (output 1) or generated (output 0), and the other corresponding to specific category.

After samples generation, dataset has been augmented based on $X_{generated}$. The next step is to evaluate the quality of generated data and apply both generated data and real data to real problems. As demonstrated in Section 2, statistical characteristics and experimental verification strategy are performed to verify the effectiveness of the generated data. Generated data are used to train a model while test data are applied to validate, and classification accuracy is calculated to measure the similarity of generated samples with real data.

In detail, we leverage one-dimensional convolution operation to build generative adversarial network, as convolution operation has been proven to have the capability to learn local features and architecture built by several convolutional layers is able to learn hierarchical representations. Compared with unsupervised learning, class conditional neural networks contain auxiliary information of category labels, which contributes to model convergence and helps avoid model collapse. Batch normalization is added after 1D-CNN layer to avoid gradient vanishing during training process. Detailed information about the proposed generator architecture is shown in Fig. 3.

In generator, the input layer is merged from noise input and class input, and it contains two upsampling layers with size of 2. There are 2 layers of 1D-convolution operation followed by a batch normalization respectively with momentum 0.8. The first 1D-convolution has 16 feature maps with Rectified Linear Unit (ReLU) as activation function and the kernel size is 16, while the second 1D-conlution layer has only 1 feature map with kernel size 16,

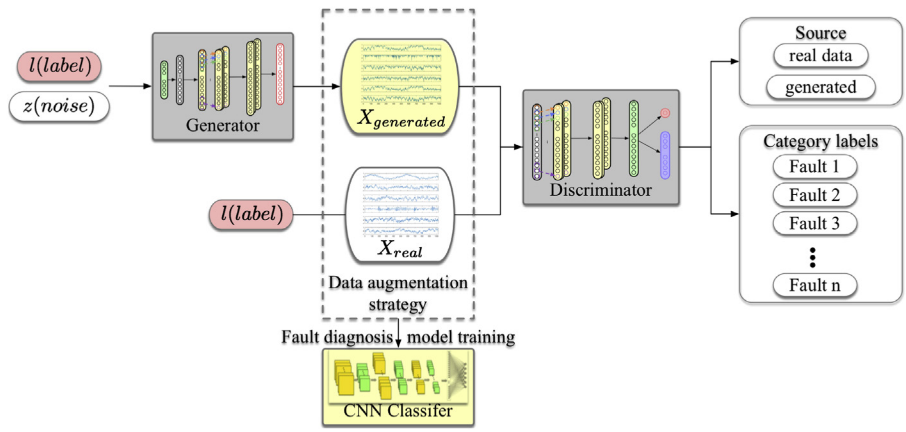

Fig. 2. Data augmentation strategy using ACGAN for fault diagnosis task.

using hyperbolic tangent as the activation function. The output of generator is one-dimensional data sample.

As for the discriminator, the input layer is followed by one 1D-convolution layer with 8 kernels using LeakyReLU as activation function, and the kernel size is 16. Another 1D-convolution layer with 16 kernels using LeakyReLU is added and followed by dropout with the probability of 0.5. Then the model layer is flattened and linked to one fully-connected layer with 0.5 dropout. Finally, two label prediction layers are added as the output layers.

Generator is designed to generate data from latent vector z, which is sampled from a uniform distribution over the interval (-1,1), denoted as $U(-1,1)$. The outputs of generator form the fake dataset $X_{generated}$, and then they are sent to the discriminator together with real signal dataset. The discriminator is also built on 1D-CNN and the output layer adopts *sigmoid* function to predict

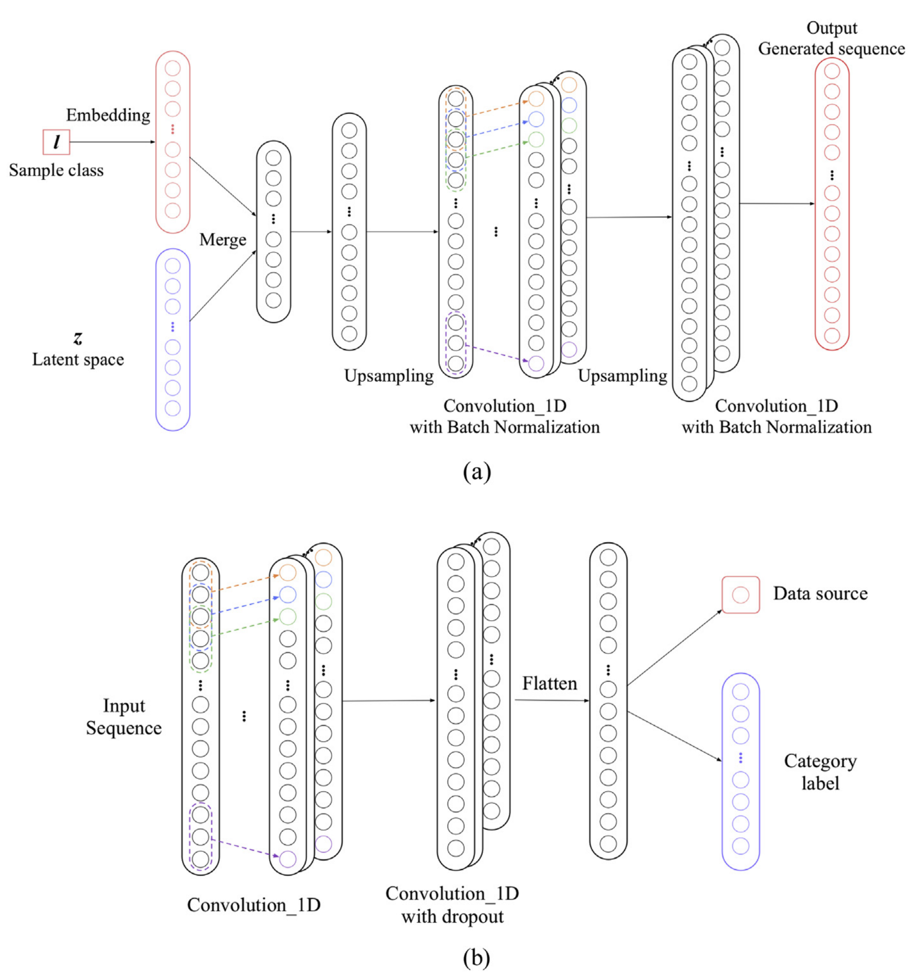

Fig. 3. Architecture of the proposed (a) generator and (b) discriminator.

sample source and adopts *softmax* function to predict specific labels.

3.2. Model training procedure

As for model training, based on loss function above, model parameters are updated iteratively. ADAM optimizer [18] is adopted for model training with learning rate 0.0001 for discriminator, and 0.0002 for generator. Within each training epoch, the training procedure can be denoted as 3 steps, shown in Fig. 4:

C1. The generator produces sequence samples from random noise of latent space with specific labels.

C2. Generated samples and real data samples are mixed together and sent to the discriminator. Based on loss function, discriminator is able to train using mixed data and labels and parameters in discriminator are able to be updated.

C3. After training the discriminator, the combined architecture starts to train. In this stage, the discriminator is set to be untrainable and parameters in it are frozen. During this step, only parameters in generator are able to be updated and the generator is trained to produce more realistic data samples. After training the combined structure, one epoch finished and training process goes on to start from ***C1*** again.

Multiple training iterations have been carried out to force the whole model to achieve a balance between discriminator and generator. After enough iteration, the losses from generator and discriminator are able to achieve the balance, called Nash Equilibrium, and generator can therefore produce realistic sensor data given certain labels.

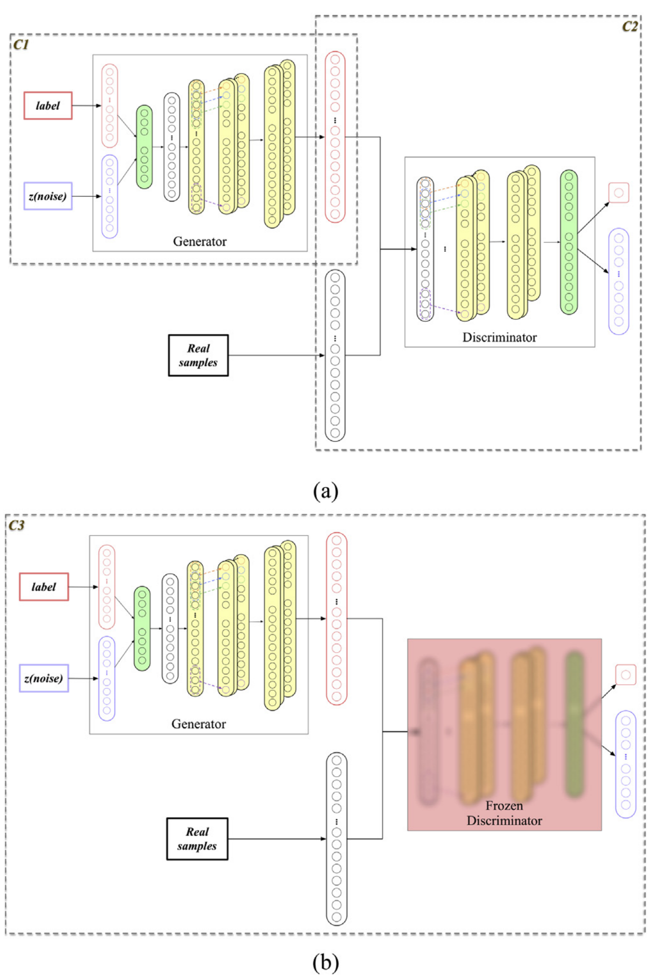

Fig. 4. Adversarial training process of the proposed method.

4. Experimental verification

4.1. Induction dataset and data preparation

In order to investigate the performance of the proposed framework in practice, experimental verification is carried out among induction motor fault simulator, shown in Fig.5. An acceleration sensor is installed on the machine fault simulator and vibration signals are acquired during operation. Six different working conditions are simulated here and corresponding vibration signals are collected as the training dataset. Detailed information about each working state are listed in Table 1 [19].

The sensor data are treated as the input of generative adversarial network, and a segment containing 4096 data points is regarded as one sample. Each working condition contains 300 separate samples and the whole dataset contains 1800 samples. The whole dataset is divided into two parts: training data and testing data. For each working condition, there are 200 time series samples for training and 100 samples for testing. Training dataset is used for training the ACGAN architecture and generated samples are achieved, while testing dataset is only used in testing procedure to validate the ACGAN model and to evaluate the quality of generated samples. Testing dataset is not involved in any training process.

4.2. Model training and data generation

The proposed architecture is utilized to generate realistic samples. In order to verify the generation ability of the proposed model, all of six kinds of sensor signals are generated. Training data together with corresponding labels are treated as the input of the discriminator in the proposed architecture, the generator samples from latent variable to model distribution of input data. Training epoch is set to be 100 and the number of generated samples for each working state is predefined as 1000. Experiments are carried out and achieve synthesized samples. Model performances are recorded, shown in Fig. 6. Generative model loss illustrates the loss in predicting sample source (real data or generated data), and classification loss and accuracy show performances of the model in predicting the specific category labels.

At the beginning, generator and discriminator start to go towards Nash Equilibrium but there is no decrease in classification loss which means model has not found the direction for optimal solution. After approximately 15 epochs, there is an obvious improvement in model classification accuracy and the generator and discriminator move to Nash Equilibrium and begin to perform stably. After approximately 60 epochs, model has been well-trained and the losses stay around Nash Equilibrium.

For comparison, we investigate several architectures in building generator, but the same discriminator built with 1D-CNN is used, including:

A Generator built by 3-layers fully connected neural network using backpropagation for generator training, denoted as NN.
B Generator contains 2-layers of 1D-Convolution operation without batch normalization, denoted as 1D-CNN.
C Generator built by 3-layers fully connected neural network with batch normalization, denoted as NN_BN.
D Our proposed framework, denoted as 1D-CNN_BN.

Hyper parameters are set the same, and the model performances are illustrated in Fig. 7. Model classification accuracy records

Table 1
Induction motor condition descriptions [19].

	Condition	Description
HEA	Normal motor	Healthy, no defect
SSTM	Stator winding defect	3 turns shorted in stator winding
UBM	Unbalanced rotor	Unbalance caused by 3 added washers on the rotor
RMAM	Defective bearing	Inner race defect bearing in the shaft end
BRB	Broken bar	Three broken rotor bars
BRM	Bowed rotor	Rotor bent in center 0.01"

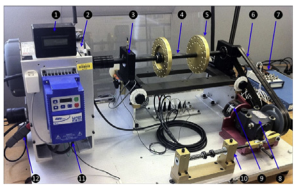

Fig. 5. Experimental setup: (1) Tachometer, (2) Induction Motor, (3) Bearing, (4) Shaft, (5) Load Disc, (6) Belt, (7) Data Acquisition Board, (8) Bevel gearbox, (9) Magnetic Load, (10) Reciprocating Mechanism, (11) Variable Speed Controller, (12) Current Probe [19].

the precision that the generative model provides the correct category labels.

From shown above, our proposed 1D-CNN_BN achieves the best performance in both classification accuracy and training speed. Generator built by 3-layers fully connected neural network is unable to accurately predict sample labels which means the model collapses and cannot generate reasonable samples. Neural network without batch normalization may be faced with the problem of overfitting, and the training procedure of the combined model may be easy to be dominated by the discriminator, where the generator model collapse. Neural network with batch normalization has much better performance than NN. Compared with 1D-CNN without batch normalization, our approach shows better model convergence rate. The experimental results show the effectiveness of our approach.

4.3. Data generation evaluation and further application

After training, generated samples from 1D-CNN_BN are achieved. Fig. 8 shows both the time domain waveform and frequency spectrum of real sensor data and generated samples among all six working conditions. From the figure shown, we can roughly recognize the similarity between real sensor data and generated samples.

As illustrated in Section 2, statistical indicators are calculated to evaluate the quality of the generated samples, shown in Table 2. ED indicates the distance between generated samples and the real data, and the smaller number indicates more similarity. In the same way, K–LD shows the divergence between two distributions, thus bigger number shows worse performance. To the contrary, PCC shows the correlation between the generated samples and the real ones, and a high correlation over 0.8 means strong similarity among these samples. Based on these metrics, our proposed approach gets better performances among various generator architecture and the samples generated from our method are alike the real data.

Table 2
Statistical features with different generator architecture to evaluate generated samples.

MODEL	ED	PCC	K-LD
NN	0.3523	0.5899	0.3940
NN_BN	0.1487	0.8458	0.2011
1D-CNN	0.1204	0.8269	0.1581
1D-CNN_BN	0.0085	0.8664	0.1431

In order to further investigate the quality of generated samples, specific training strategy is carried out among generated datasets. The ACGAN model generates 6000 samples corresponding to 6 different working conditions, where each condition contains 1000 generated samples. These generated samples are then used to train a 2-layers CNN model to realize fault classification. After fully training CNN model, testing data that are real sensor signals are sent to the model to predict possible labels. The classification accuracy using this strategy implies the similarity between generated model and real sensor signals. In this experiment, we obtain highly-accurate classification accuracy 100% based on learning generated samples and testing in real data samples, which has shown that the generated samples is able to model the real data distribution.

To investigate the influence that the quantity of generated training samples has on model learning, we set various scenarios to test their performances. Classification accuracy and model loss are recorded during model learning, and performances on testing dataset in several selected scenarios are shown in Fig. 9. Different scenarios have different sizes of real data samples and generated data samples, and the detailed information about training data settings of each scenario are listed in Table 3. The classification accuracy of these scenarios after enough training iterations are also shown in Table 3.

Based on the results shown above, it is obvious that training data with larger scale has better performance in classification accuracy and model converge speed. When using mixed data including real data and generated data, model has better performances than only using generated data. For enough training data, model using generated data is able to achieve highly-accurate predictions.

Specifically, in order to investigate the effectiveness of ACGAN-based data augmentation strategy in helping improve fault diagnosis performances when the training data are limited, several experiments are designed. We simulate several circumstances where training data are limited in the form of various proportions of the available training data. In the meantime, different numbers of generated samples are used to give a comparison, as well as training without data augmentation. Classification results are shown in Table 4 under different settings.

From the results above, when training data are sufficient, there is a small improvement in classification accuracy using ACGAN-based data augmentation strategy. When the proportion of available training data drops to 25%, the improvement is obvious which verifies the effectiveness of the proposed method.

Furthermore, ACGAN-based data augmentation strategy is also helpful in dealing with unbalanced training data. Generated samples can be regarded as supplement samples to unbalanced training data, and the samples from minor category can be expanded by synthetic data. Finally, balanced training dataset is formed using real data and generated data. In order to investigate the performance of ACGAN-based model in unbalanced training dataset, we design several unbalanced datasets by reducing the number of training samples from one or more categories by 50%, and then ACGAN-based model is used to help generate balanced dataset. At last, both unbalanced original dataset and generated balanced dataset are utilized to train a 2-layer CNN separately, and testing data are used to investigate the model performances. In this study, the induction motor dataset contains six different working conditions which means the dataset has six categories, so we design 5 different unbalanced datasets which have 1–5 classes of

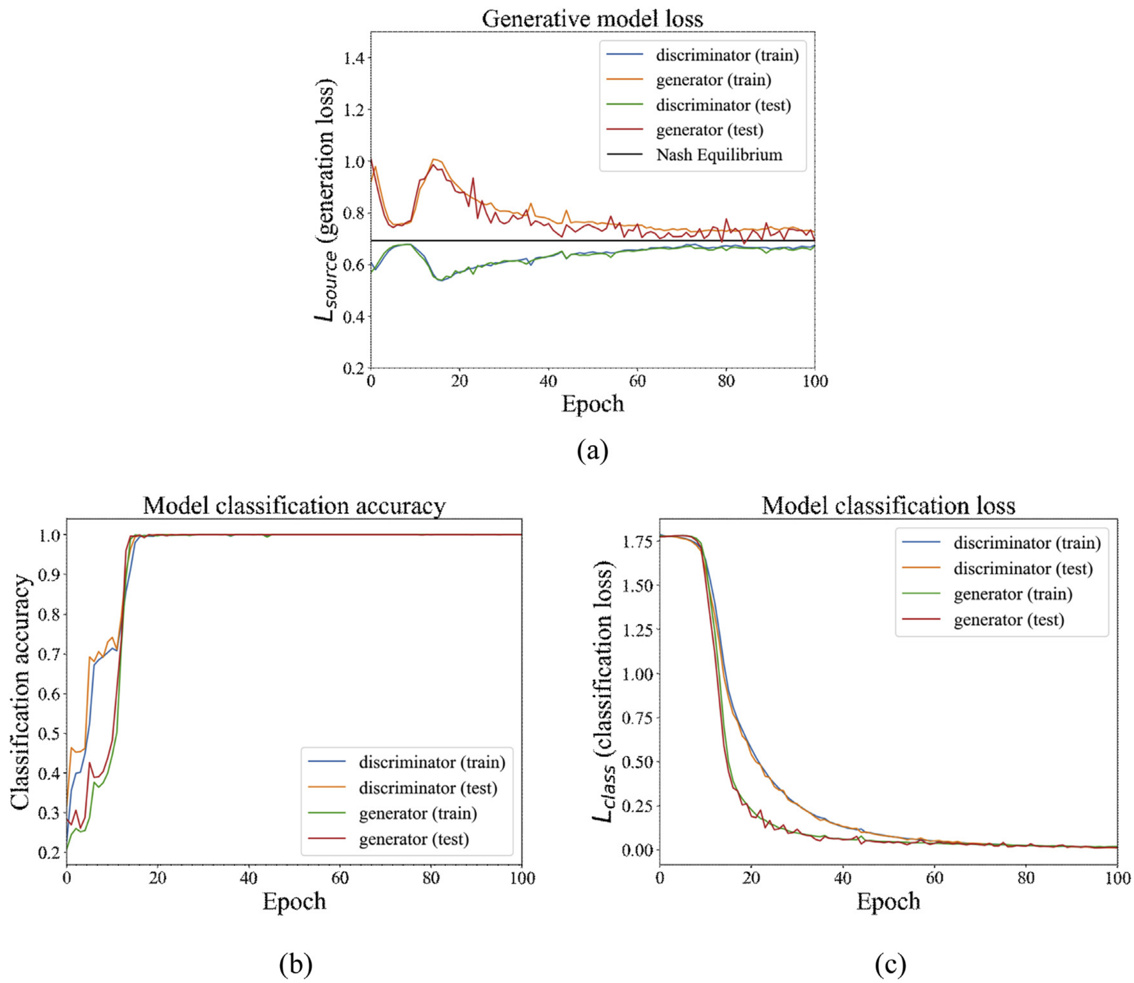

Fig. 6. 1D-CNN_BN model performance in (a) generative loss, (b) classification accuracy, and (c) classification loss.

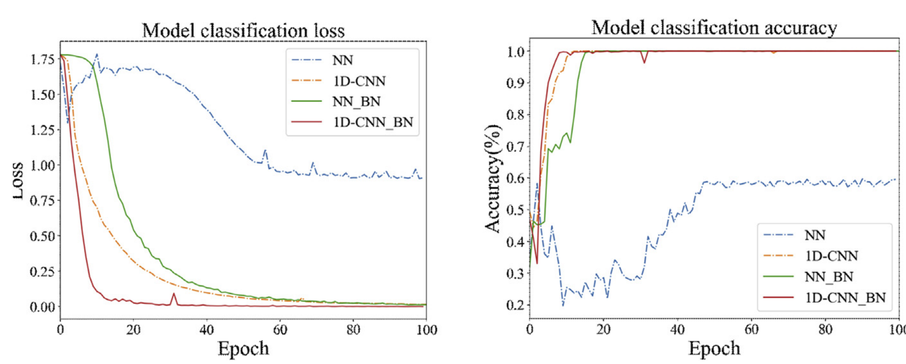

Fig. 7. Model performances between various generator architecture.

reduced samples, respectively. The minor class has only 100 samples while other classes contain 200 samples. In order to achieve balanced dataset, part of the unbalanced data are chosen to train the ACGAN model and generated samples for minor class are obtained. Then, samples of minor class are expanded to 200 by adding generated samples, where the synthetic dataset is balanced. Table 5 shows the various unbalanced datasets and their classification accuracies.

With unbalanced training dataset, deep model is not able to well learn the data distribution among input data. Using data augmentation is an alternative to supplement the minor classes so that the training dataset will be balanced among various classes. Classification results also verify the effectiveness among different unbalanced settings.

In addition, ACGAN-based data augmentation strategy starts an open issue in generating raw data samples in machine health monitoring. Investigation on building generative models and generating reasonable samples helps better understand sensor signal distribution and helps develop reliable predictive system in fault diagnosis and prognosis.

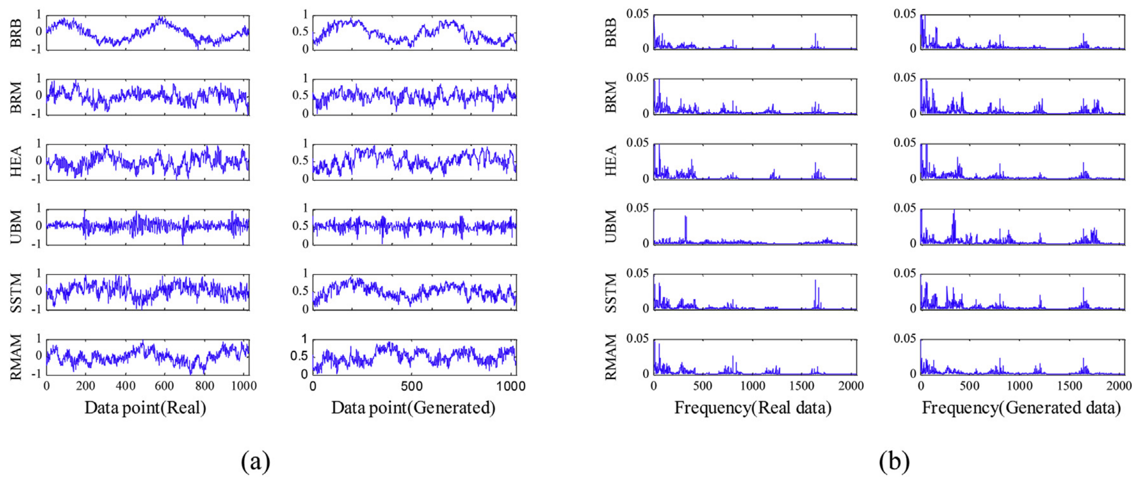

Fig. 8. Time domain waveform (a) and frequency spectrum (b) of 6 different working conditions from real sensor data and generated samples.

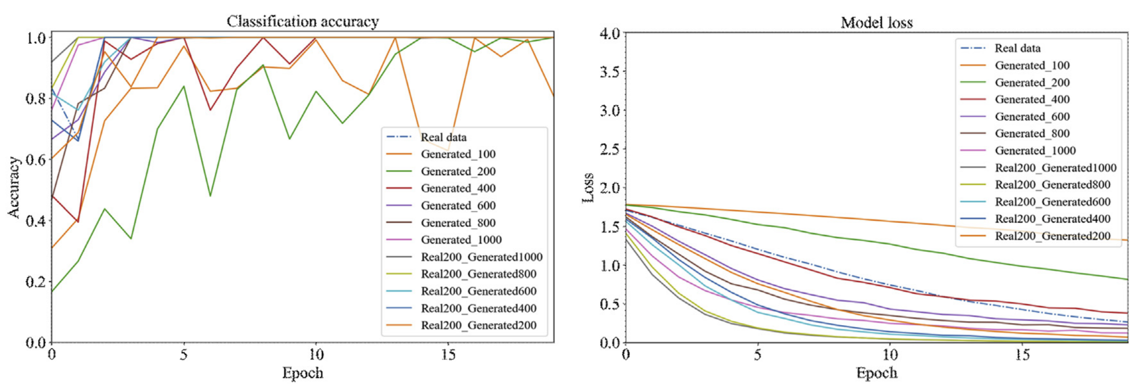

Fig. 9. Fault diagnosis model performances among different training data settings.

Table 3
Different training data settings using real data and generated data and the classification results.

	A	B	C	D	E	F	G	H	I	J	K	L
Real data (samples)	200	0	0	0	0	0	0	200	200	200	200	200
Generated data (samples)	0	100	200	400	600	800	1000	200	400	600	800	1000
Classification Accuracy (%)	99.80	95.33	99.50	99.83	99.83	100	100	99.83	99.83	100	100	100

Table 4
Classification accuracy (%) under various training settings.

ADDITIONAL GENERATED SAMPLES	PROPORTION OF AVAILABLE REAL DATA		
	25%	50%	100%
0	95.67	97.17	99.80
50	97.50	98.50	99.80
100	98.50	99.50	99.83
200	98.83	99.80	99.83
300	98.83	100	100

Table 5
Classification accuracy under different unbalanced dataset and generated dataset.

NUMBER OF CLASSES WITH REDUCED SAMPLES	UNBALANCED DATASET	GENERATED BALANCED DATASET
1	83.33%	99.83%
2	66.67%	99.50%
3	50.00%	99.33%
4	75.67%	99.33%
5	94.50%	99.16%

5. Conclusion

In this paper, we propose an ACGAN-based framework that has the capability to generate artificial raw data of mechanical sensor signals. Generated samples with high quality are used for data augmentation and are able to solve the problem of unbalanced data when training a deep architecture. In order to evaluate the quality of generated sensor data, statistical characteristics are calculated as the criterion to assess the difference between synthesized data and real sensor signals. For further investigation, we present a testing strategy to verify the effectiveness of generated samples by training classifier using generated samples and testing it by real data. Classification results have shown that the proposed framework is able to produce convincing sensor data, and the ACGAN-based method is able to serve as a data augmentation technique when dealing with unbalanced dataset. This paper attempts to create a predictive framework for machine fault diagnosis tasks, and modeling data generation process has the possibility to better learn and understand input data distribution, which contributes to subsequent classification tasks.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 51575102), Fundamental research Funds for the Central Universities of China, and Jiangsu Provincial Research Innovation Program for College Graduates of China (Grant No. KYLX16_0191).

References

[1] 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.
[2] W. Sun, R. Zhao, R. Yan, S. Shao, X. Chen, Convolutional discriminative feature learning for induction motor fault diagnosis, IEEE Trans. Industr. Inform. 13 (3) (2017) 1350–1359.
[3] M. Ma, C. Sun, X. Chen, Deep coupling autoencoder for fault diagnosis with multimodal sensory data, IEEE Trans. Industr. Inform. 14 (3) (2018) 1137–1145.
[4] S. Shao, S. McAleer, R. Yan, P. Baldi, Highly-accurate machine fault diagnosis using deep transfer learning, IEEE Trans. Industr. Inform. (2018).
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. (2014) 2672–2680.
[6] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint (2017) arXiv:1701.07875.
[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, Adv. Neural Inf. Process. Syst. (2017) 5767–5777.
[8] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint (2015) arXiv:1511.06434.
[9] A. Odena, Semi-supervised learning with generative adversarial networks, arXiv preprint (2016) arXiv:1606.01583.
[10] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, arXiv preprint (2016) arXiv:1610.09585.
[11] C. Donahue, J. McAuley, M. Puckette, Synthesizing audio with generative adversarial networks, arXiv preprint (2018) arXiv:1802.04208.
[12] K.G. Hartmann, R.T. Schirrmeister, T. Ball, EEG-GAN: generative adversarial networks for electroencephalographic (EEG) brain signals, arXiv preprint (2018) arXiv:1806.01875.
[13] P.J. Kindermans, K.T. Schütt, M. Alber, K.R. Müller, D. Erhan, B. Kim, S. Dähne, Learning how to explain neural networks: PatternNet and PatternAttribution, arXiv preprint (2017) arXiv:1705.05598.
[14] J.T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, arXiv preprint (2015) arXiv:1511.06390.
[15] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Adv. Neural Inf. Process. Syst. (2016) 2234–2242.
[16] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Adv. Neural Inf. Process. Syst. (2017) 6626–6637.
[17] G. Peyré, M. Cuturi, Computational Optimal Transport, (2017) (No. 2017-86).
[18] K. Diederik, B. Jimmy, ADAM: a method for stochastic optimization, arXiv preprint arXiv (2014) 1412.6980.
[19] X. Yang, R. Yan, R.X. Gao, Induction motor fault diagnosis using multiple class feature selection, in Proc, IEEE Int. Instrum. Meas. Technol. Conf. (2015) 256–260.