Adversarial Domain Adaptation for Gear Crack Level Classification Under Variable Load

Dongdong Wei
Department of Mechanical Engineering
University of Alberta
Edmonton, Alberta, T6G 1H9, Canada
do1@ualberta.ca

Te Han
Department of Energy and Power Engineering
Tsinghua University
Beijing, 100084, China
hant15@mails.tsinghua.edu.cn

Fulei Chu
Department of Mechanical Engineering
Tsinghua University
Beijing, 100084, China
chufl@mail.tsinghua.edu.cn

Ming Jian Zuo*
Department of Mechanical Engineering
University of Alberta
Edmonton, Alberta, T6G 1H9, Canada
ming.zuo@ualberta.ca*

Abstract— **Traditional intelligent fault diagnosis assumes that the training and testing samples are drawn from the same distribution. This assumption does not hold when working condition changes, as variable working condition can make the training and the testing datasets have different distributions. A novel working condition might be encountered in the testing stage, and there will be no label available under that novel working condition. This paper studies domain adaptation for gear crack level diagnosis under variable loading. A new two-stage fault diagnostic method for variable load condition is developed based on adversarial training strategy and gradient reversal layer. Both labeled and unlabeled data are utilized to learn best model for the novel load condition. An experimental case study is carried out to demonstrate the effectiveness of the proposed method.**

Keywords-deep learning; transfer learning; domain adaptation; intelligent fault diagnosis; gear crack

I. Introduction

Gear tooth crack detection and assessment is very important for reliable operation of wind turbines. Recently, big data and intelligent models, such as support vector machine (SVM) and deep neural networks, are being extensively used to detect gear tooth crack. However, lack of labeled data and working condition changes (e.g. speed and load changes) are two key challenges in fault diagnosis of wind turbines.

Most intelligent models need labeled data as they need to be trained via supervised learning. For field wind turbines, it could be expensive or prohibitive to obtain labeled data for training. That is, we need to develop a way to train our intelligent models with unlabeled data. In lab experimental settings, we can obtain large amount of labeled data when the wind turbines are being tested for certain working conditions. However, only unlabeled data will be available when the wind turbines are put into operation. Its working condition will also be different from the experiment setting when exposed to natural wind. The problem we are focusing on in this paper is, given labeled data from some certain working conditions and unlabeled data from another working condition, how can we train a good intelligent model to perform well in that unlabeled working condition.

Transfer learning is one of the most promising technology for unlabeled fault diagnostic problems [11]. By regarding each working condition as a domain, we can fit the problem stated above into an unsupervised domain adaptation (a subset of transfer learning) framework. Unsupervised domain adaptation is to learn best deep neural network using a labeled source dataset (drawn from source domain) and an unlabeled target dataset (drawn from target domain). The model will be trained towards a good performance on a test dataset drawn from the target domain. In the context of machine fault diagnosis, we can expect the trained model to perform well when operating in a novel working condition that has only unlabeled data.

Several unsupervised domain adaptation methods have been developed for fault diagnostic problems. For example, Lu et al. [1] developed a Domain Adaptation for Fault Diagnosis (DAFD) model. This model is tested by both bearing fault diagnostic case and gearbox fault diagnostic case with load and speed changes, respectively. Han et al. [2] proposed a deep transfer network (DTN) that can adaptively diagnose wind turbine faults under wind speed changes. A more detailed literature review with discussions on different domain adaptation methods will be given in Section II.

This paper presents a gear crack level classification method based on deep neural networks and unsupervised domain adaptation. Training and testing data will be collected from two different load conditions. A load condition with labeled data available will be used as the source domain and another load condition with unlabeled data will be regarded as the target domain. To adapt these two domains, corresponding to the two load conditions, a two-stage training algorithm of neural networks will be studied and applied to solve a gear crack level classification task.

The remaining parts of this paper are organized as follows: Section II reviews reported literatures about domain adaptation

This research was supported by the Future Energy Systems under Canada First Research Excellent Fund [grant number FES-T14-T01, FES-T11-P01, FES-T14-P02], Natural Sciences and Engineering Research Council of Canada (Grant #RGPIN-2015-04897), and China Scholarship Council.

The 9th Asia-Pacific International Symposium on Advanced Reliability and
Maintenance Modeling (APARM 2020 – Vancouver)

978-1-7281-7102-9/20/$31.00 ©2020 IEEE

methods for machine fault diagnosis; Section III gives background knowledge domain adaptation and supervised training of neural networks; Section IV describes our crack level classification method; Section V presents a case study of gear crack level classification to demonstrate the effectiveness of the crack level classification method; Section VI concludes this paper.

II. Related works

Existing domain adaption methods fall into three types: sample processing, discrepancy based regularization, and adversarial training. Sample processing is to process the data samples before feeding them into an intelligent model. Typical methods are normalization and signal processing. The goal is to reduce the difference between the source and the target domain on the sample level. For examples, Zhang et al. [3] proposed to use the mean and variance of target domain data to normalize source domain data. This method is effective as their trained models have good performance under load changes. Wei et al. [4] rescaled the amplitudes of vibrational data based on the corresponding rotating speed curve, so that the impact of rotating speed change can be reduced. Regarding rotating speed changes, Rao and Zuo [5] proposed to use order spectrum instead of raw vibration signals as the input of deep neural networks. Adapting domains via sample processing can be effective but is highly dependent on the analyst's expert knowledge. The idea is similar to the traditional signal processing and shallow neural network pipeline (such as [6]). Its effectiveness relies on expert knowledge and has no guarantee when applied to other cases.

Discrepancy based regularization is one of the most popular ways of domain adaptation. This is done by minimizing the empirical risk of misclassifying source domain samples and the distributional discrepancy between the source and the target domains at the same time. A typical measure of this distributional discrepancy is Maximum Mean Discrepancy [7]. Discrepancy based regularization has been reported to be effective for both load changes and speed changes [1], [2], [8]–[10]. The success of this method largely depends on the design of network structure and the selection of hyper-parameters. If the network structure is poorly designed, the discrepancy of the features may fail to truly indicate the distributional difference between the source and the target. In addition, the added regularization term will change the convergence of the objective function. If the trade-off parameter is not optimized, we may not be able to train the network to converge, or the regularization term could fail to serve its purpose.

Adversarial training is very promising for domain adaptation. It has less hyper parameters to be tuned or structures to be designed, compared to the above two categories. A domain discriminator is introduced to recognize whether the input is from the source or the target domain and it will be trained against to the health condition classifier [11]. Fault discriminative but working condition invariant features can be learned this way. Successful applications of adversarial training for fault diagnosis are [11]–[15]. More explanations on adversarial training will be presented in Section III.B.

III. Preliminearies

A. Domain adaptation

For fault classification, two different working conditions may be considered as two different domains. The domain with labels for corresponding health conditions is called the source domain S and the one without label is the target domain T. From the source domain, we can draw a labeled source dataset $\chi_s = \{(x_{S_1}, y_{S_1}), (x_{S_2}, y_{S_2}), \dots, (x_{S_M}, y_{S_M})\}$, where x_{S_M} stands for a data sample, y_{S_M} is its corresponding label, and M is the number of source samples. An unlabeled target dataset with N labels $\chi_T = \{x_{T_1}, x_{T_2}, \dots, x_{T_N}\}$ can be drawn from the target domain. When the training and testing samples are from different domains, their distributions will be different. This problem challenges the performance of traditional machine learning algorithms that assume an identical distribution for both the training and testing dataset. Domain adaptation is to learn best model in terms of target domain test accuracy, given χ_S and χ_T. General approach of domain adaptation is to find good representations (features) for the input samples. The representations should be domain-invariant and have necessary information for fault classification. Training neural networks to get such domain-invariant features can be called as domain adaptive training.

B. Supervised learning of Neural Networks

For classification tasks, typical neural networks will have two parts: feature extractor $\mathcal{F}$ and classifier $\mathcal{C}$. The feature extractor transforms an input sample into a feature vector and then the classifier tries to recognize its corresponding label based on that feature vector. Conventional supervised training of neural networks is to minimize its classification error $\mathcal{L}_\mathcal{C}$ on the labeled training dataset. The optimization parameters are the weights and biases of the feature extractor and the classifier. This optimization can be formulated as follow:

$$\min_{\mathcal{C},\mathcal{F}} \mathcal{L}_\mathcal{C} \tag{1}$$

where $\mathcal{L}_\mathcal{C}$ is the classification error of classifier $\mathcal{C}$, which is measured by cross entropy function (see Eq. (2) in Section IV.A).

IV. Method

In this study, we propose a two-stage domain adaptation method based on deep convolutional neural networks and adversarial training. The domain adaptation method uses a source-labeled dataset and a target-unlabeled dataset as the inputs, and the output is a deep convolutional neural network that maps vibration signals into corresponding health conditions (e.g. gear tooth crack level). In the following part of this section, we will first introduce our neural network structure and then explain the used adversarial training algorithm.

A. Neural Network structure

In this paper, we use a one-dimensional convolutional neural network (1DCNN) and our network structure is shown in Figure 1. In Figure 1, the types and parameters of each layer are marked. Convolutional layers are marked with 'Conv.' and the size of its convolutional kernels. Pooling layers are denoted with its ratios of downsampling. The flattening layer (marked with 'Flatten') is to convert its input which consists of tensors of arbitrary order into one-dimensional vectors . FC stands for fully connected layer and '->500' means its output vector is sized at 500×1.

The key parameters of the 1DCNN include the number of convolution layers, the size of the convolutional kernels, the downsampling ratio of the pooling layers, the number of convolutional kernels in each layer, the number of FC layers, and the output size of each FC layer. For the number of convolution layers, the size of convolutional kernels, and the downsampling ratio of the pooling layers, we follow the very first designed two-dimensional convolutional neural network [22] and convert it to one-dimensional. That is, two convolutional layers are used; the size of the convolutional kernels is 25×1 (converted from 5×5); the downsampling ratio of the pooling layers is 1/4. For the number of convolutional kernels, we follow the rule of thumb [11], [22] to increase this number with the depth of layer. We use 20 kernels in the first convolutional layer and 50 kernels in the second. For simplicity, a single FC layer is used in the feature extractor. It reduces the number of features from 2800 to 500. The 500 extracted features will then be sent into the domain discriminator and the classifier. In the domain discriminator, a FC layer is used to map the extracted features into a score vector with two elements for the two domains, while in the classifier, another FC will map the features into a score vector for each class. The output class label or domain tag will be the one that corresponds to the highest score in these two vectors. During training, cross entropy loss is used and it is calculated as [19]:

$$\mathcal{L} = -\sum_{(x,y)\in\mathcal{X}_S} \log \frac{exp(s[y])}{\sum_i exp(s[i])} \tag{2}$$

where x stands for a sample, y is the index of its corresponding class or domain, and vector s is the score vector of $\mathcal{C}$ or $\mathcal{D}$.

Note that the domain discriminator will only be used for adversarial training, but not traditional supervised training. Once the training is completed, only the feature extractor and classifier will be saved.

B. Two-stage domain adaptation

Traditional methods only have one supervised training stage that use source-labeled data only. It is done by solving the optimization problem in Eq. (1). Example fault diagnostic methods that are based on supervised learning include [20], [21], and [5].

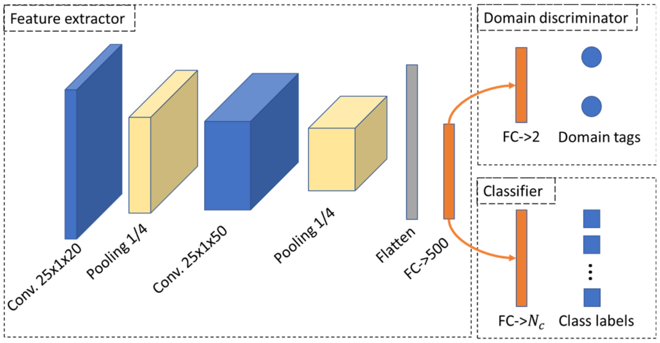

Fig. 1. Structure of the used neural network

The training process in this paper has two stages, i.e. pre-training (identical to the supervised training stage in traditional methods) and domain adaptive training. Figure 2 shows the schematic diagram of the traditional method and the two-stage domain adaptation in this paper. Firstly, the pre-training is to use a source-labeled dataset to train the feature extractor $\mathcal{F}$ and the classifier $\mathcal{C}$ via supervised learning algorithm. Then, the domain adaptive training starts with the pre-trained feature extractor, the pre-trained classifier, and a newly initialized domain discriminator $\mathcal{D}$. Both the source-labeled dataset and the target-unlabeled domain dataset will be used.

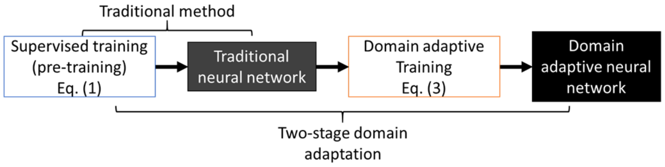
Traditional method
Supervised training (pre-training) Eq. (1)
Traditional neural network
Domain adaptive Training Eq. (3)
Domain adaptive neural network
Two-stage domain adaptation

Fig. 2. Shematic diagram of the proposed two-stage domain adaptation

Adversarial training strategy[16] will be used in the domain adaptive training stage. A domain discriminator $\mathcal{D}$ will be trained to determine whether a sample is from the source or the target domain. Instead of training the domain discriminator to be as accurate as possible, it is expected to fail to discriminate the features from the two domains. The key is to train good feature extractor that can extract fault-sensitive but domain-invariant features. It can be done by solving a minimax optimization that minimizing the error of the classifier $\mathcal{C}$ and maximizing the error of the domain discriminator simultaneously. Further, by introducing a gradient reversal layer[17], this minimax optimization can be solved as a simple minimization as follows:

$$\min_{\mathcal{C},\mathcal{D},\mathcal{F}} \mathcal{L}_{\mathcal{C}} - \mathcal{L}_{\mathcal{D}} \tag{3}$$

where $\mathcal{L}_{\mathcal{D}}$ is the error term for domain discriminator $\mathcal{D}$, and it is measured by cross entropy function (see Eq. (2) in Section IV.A).

The gradient reversal layer will be added in between the domain discriminator and the feature extractor during the domain adaptive training stage. It forwards the extracted features and reverses the sign of the backpropagated gradient from the domain discriminator. The mathematical formulation of forward calculation and backpropagation process in gradient reversal layer can be expressed as [17]:

$$R(z) = z \tag{4}$$

$$\frac{dR(z)}{dz} = -I \tag{5}$$

where z is the input vector with dimension n, $R(z)$ is the output vector with dimension n, and I is an $n \times n$ identity matrix. Once the training is completed, the domain discriminator and the gradient reversal layer will be discarded. Only the feature extractor and the classifier will be saved for testing.

To summarize this two-stage domain adaptation method, a convolutional neural network with the structure presented in

Section IV.A will be trained via the two-stage training explained in Section IV.B. Two datasets, one with label and the other has only unlabeled data, collected from two different loading conditions will be used for training. In the following section, a case study of gear crack level classification, under variable loading condition, will be presented to verify the effectiveness of this method.

V. EXPERIMENT

A gear crack level diagnostic case study is carried out to verify the effectiveness of our method. The task is to correctly map vibration signals into their corresponding gear crack levels, considering load changes.

The lab experiments were carried out in between September 2017 and June 2018, at the University of Alberta. The studied test rig is shown in Figure 3(a). Five different levels of crack were seeded in the objective gear shown in Figure 3(b). For parameters of the test rig and descriptions of the seeded cracks, please refer to [18]. We use the vibration signals from sensor 2, shown in Figure 3(c), in this study. The rotating speed of the drive motor was fixed at 10Hz, and the sampling frequency of data acquisition was set as 25600Hz. Two load conditions, Low for 3% and High for 8% of the rated load [18], are considered as the two domains to adapt. When one of these two domains is used as the source domain, the other one will be the target domain. The collected signals will first be downsampled to 12800Hz and then be sliced into signal segments (input data of our neural network) sized at 1024x1. For each load condition, each crack level, 1125 and 375 numbers of segments are used for training and testing, respectively.

As discussed in Section IV, we compare the traditional method and the two-stage domain adaptation shown in Figure 2. The neural network drawn in Figure 1 will first be pre-trained with the labeled source dataset, and then adaptively trained with both source-labeled samples and target-unlabeled samples. For pre-training, the learning rate is 0.0001, batch size is 50, and number of epochs is 30. For domain adaptive training, the learning rate is 0.00001, batch size is 50, and number of epochs is 24. The learning rate during domain adaptive training will be halved every 6 epochs to avoid overfitting.

For the two compared methods and the two source-target pairs, the target domain test accuracies are shown in Table I. Also, CPU times (PowerEdge R730 Server at the University of Alberta, with 2 Intel Xeon E5-2630 v3 CPU cards) of the two methods are listed. In Table I, ‘Low-High’ means that Low (3%) load is the source domain and High (8%) load is the target domain, and ‘High-Low’ means High is the source and Low is the target. For each column, the displayed accuracies of our proposed method are the averages of 10 repeated runs, and all these runs start from the same pre-trained model, which gives the test accuracies of the traditional method.

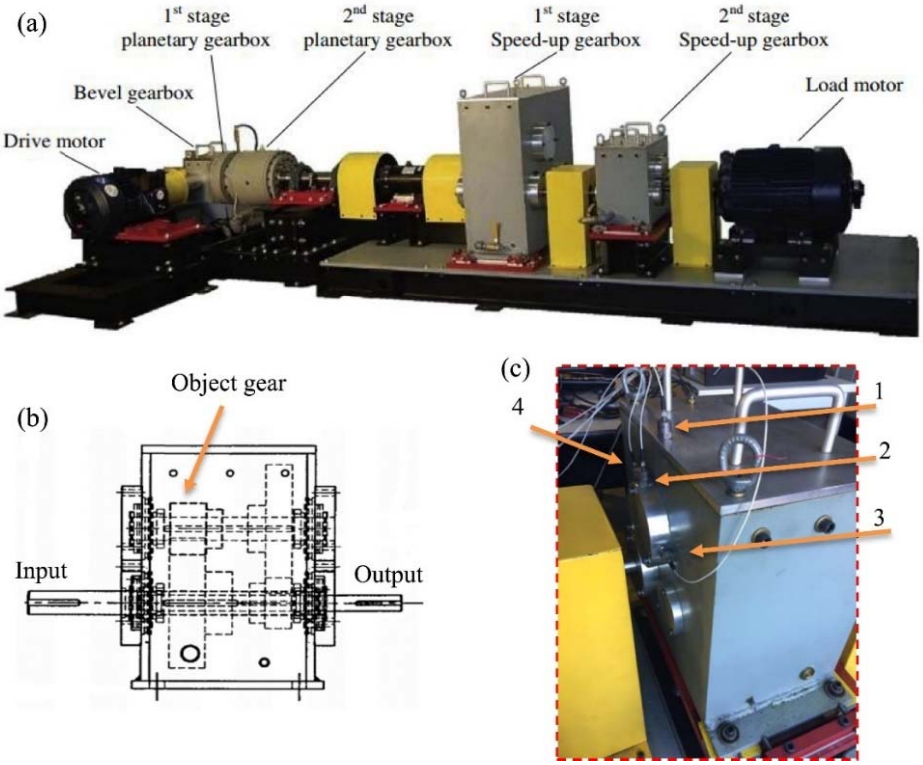
(a)
1st stage planetary gearbox
2nd stage planetary gearbox
1st stage Speed-up gearbox
2nd stage Speed-up gearbox
Load motor
Bevel gearbox
Drive motor
(b)
Object gear
Input
Output
(c)
1
2
3
4

Fig. 3. Experiment setup: (a) gearbox test rig, (b) schematic of the 2nd stage speed-up gearbox, (c) four sensor locations [18].

TABLE I. TEST ACCURACIES UNDER LOAD CHANGES

Method	L-H	H-L	CPU time	Number of parameters
Pre-training	0.7489	0.7231	527.35s	1,429,076
Two-stage domain adaptation	0.7618	0.7726	1,074.03s	1,430,078

From Table I we can see that the proposed two-stage domain adaptation method is effective as higher test accuracies can be obtained comparing to pre-training. The testing accuracy of the case when the source is Low and the target is High, i.e. Low-High, has increased from 0.7489 to 0.7618 (0.0129 increased). The accuracy of High-Low case has a larger increment (0.0495) comparing to Low-High. This indicates that, for our neural network, to diagnose gear crack level when the higher load condition is the target can be harder than to diagnose when the lower load condition is the target domain. For the tasks without load change, i.e. High-High and Low-Low, the testing accuracies of our model are 0.8524 and 0.8129, respectively. This also support that higher loading can make the diagnosis harder.

It is important to visualize how the neural network had learnt to classify crack levels. In fact, applying a convolutional layer to an input signal is to apply different filters on that signal to obtain different filtered output signals. We follow [23] to visualize the filters in the first convolutional layer learned for the Low-High case (Figure 4). From Figure 4, we can see that the convolutional neural network had learned to formulate many band-pass filters (e.g. row #2, column #3, #4, and #5) and also a few low-pass filters (e.g. row #4, column #3 and #5) for this crack level identification task.

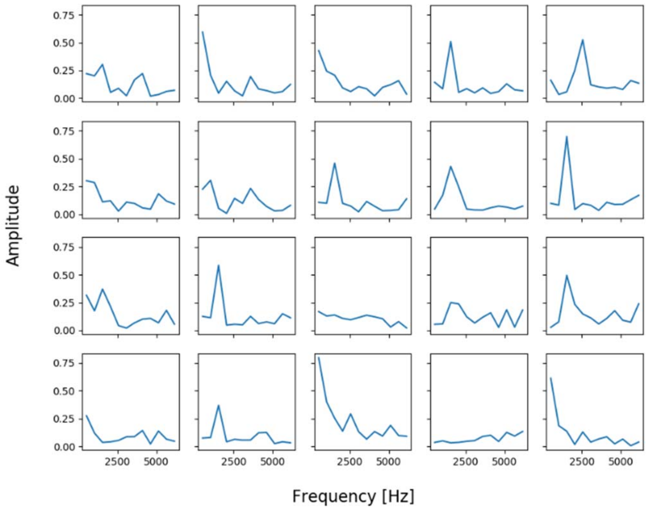

Figure 4. Amplitude spectrums of the learned 1st layer convolutional filters for the Low-High task

Trade-offs behind the higher accuracies of our two-stage method include the cost of CPU time and memory for training. Traditional method (pre-training) only needs 527.35 seconds for 30 epochs of parameter (weights and biases) updating, while additional 546.68 seconds are needed for 24 epochs of adversarial training. During domain adaptive training, larger memory is needed for additional 1002 parameters (1000 weights and 2 biases) and their gradients from the domain discriminator. Nevertheless, the time and memory for classifying a sample are the same for both methods as their output neural networks have identical structures. The time cost for the adaptive training is not an issue in real applications. In practice, traditional models can be put online first. Once there are new unlabeled data coming in, we can carry on the training off-line, and then upgrade the online model. The efficiency of establishing useful traditional models will not be affected by the prolonged training time.

VI. Conclusion

This study presents a two-stage domain adaptation method based on adversarial training. The results show that this method can effectively boost the diagnostic performance of neural networks under load changes. Evidence also show that the diagnostic performance of neural networks is susceptible to certain load conditions. Higher loading may lead to worse diagnostic performance than lower loadings.

The limitation of this work includes that the proposed method does not work when the source and the target have different fault levels. Besides, this method is based on a large amount of data. Its performance is not tested when the number of training samples are insufficient.

To further improve the adaptability of intelligent diagnostic models, relationships between different fault levels can be considered. For example, level 3 crack can be identified if a neural network is given the data of level 2 and level 4 cracks and designed to learn features that describe crack progressions. In addition, data augmentation methods can be developed to address the issues of insufficient data. The design of the neural network structure is also worth further investigations.

References

[1] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, “Deep Model Based Domain Adaptation for Fault Diagnosis,” IEEE Transactions on Industrial Electronics, vol. 64, no. 3, pp. 2296–2305, Mar. 2017, doi: 10.1109/TIE.2016.2627020.

[2] T. Han, C. Liu, W. Yang, and D. Jiang, “Deep Transfer Network with Joint Distribution Adaptation: A New Intelligent Fault Diagnosis Framework for Industry Application,” arXiv:1804.07265 [cs, stat], Apr. 2018.

[3] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, “A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals,” Sensors, vol. 17, no. 2, p. 425, Feb. 2017, doi: 10.3390/s17020425.

[4] D. Wei, K. Wang, S. Heyns, and M. J. Zuo, “Convolutional Neural Networks for Fault Diagnosis Using Rotating Speed Normalized Vibration,” in Advances in Condition Monitoring of Machinery in Non-Stationary Operations, 2019, pp. 67–76.

[5] M. Rao and M. J. Zuo, “A New Strategy for Rotating Machinery Fault Diagnosis Under Varying Speed Conditions Based on Deep Neural Networks and Order Tracking,” in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1214–1218, doi: 10.1109/ICMLA.2018.00197.

[6] H. Li, Y. Zhang, and H. Zheng, “Gear fault detection and diagnosis under speed-up condition based on order cepstrum and radial basis function neural network,” J Mech Sci Technol, vol. 23, no. 10, pp. 2780–2789, Oct. 2009, doi: 10.1007/s12206-009-0730-8.

[7] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A Kernel Two-Sample Test,” Journal of Machine Learning Research, vol. 13, no. Mar, pp. 723–773, 2012.

[8] L. Wen, L. Gao, and X. Li, “A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 1–9, 2018, doi: 10.1109/TSMC.2017.2754287.

[9] Z. Tong, W. Li, B. Zhang, and M. Zhang, “Bearing Fault Diagnosis Based on Domain Adaptation Using Transferable Features under Different Working Conditions,” Shock and Vibration, 2018. [Online]. Available: https://www.hindawi.com/journals/sv/2018/6714520/abs/. [Accessed: 26-Oct-2018].

[10] B. Zhang, W. Li, X. Li, and S. Ng, “Intelligent Fault Diagnosis Under Varying Working Conditions Based on Domain Adaptive Convolutional Neural Networks,” IEEE Access, vol. 6, pp. 66367–66384, 2018, doi: 10.1109/ACCESS.2018.2878491.

[11] T. Han, C. Liu, W. Yang, and D. Jiang, “A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults,” Knowledge-Based Systems, vol. 165, pp. 474–487, Feb. 2019, doi: 10.1016/j.knosys.2018.12.019.

[12] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, “Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines With Unlabeled Data,” IEEE Transactions on Industrial Electronics, vol. 66, no. 9, pp. 7316–7325, Sep. 2019, doi: 10.1109/TIE.2018.2877090.

[13] X. Li, W. Zhang, and Q. Ding, “Cross-Domain Fault Diagnosis of Rolling Element Bearings Using Deep Generative Neural Networks,” IEEE Transactions on Industrial Electronics, pp. 1–1, 2018, doi: 10.1109/TIE.2018.2868023.

[14] H. Liu, J. Zhou, Y. Xu, Y. Zheng, X. Peng, and W. Jiang, “Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks,” Neurocomputing, vol. 315, pp. 412–424, Nov. 2018, doi: 10.1016/j.neucom.2018.07.034.

[15] Z. Wang, J. Wang, and Y. Wang, “An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition,” Neurocomputing, vol. 310, pp. 213–222, Oct. 2018, doi: 10.1016/j.neucom.2018.05.024.

[16] I. J. Goodfellow et al., “Generative Adversarial Networks,” arXiv:1406.2661 [cs, stat], Jun. 2014.

[17] Y. Ganin and V. Lempitsky, “Unsupervised Domain Adaptation by Backpropagation,” arXiv:1409.7495 [cs, stat], Sep. 2014.

[18] Y. Chen, X. Liang, and M. J. Zuo, “An improved singular value decomposition-based method for gear tooth crack detection and severity assessment,” Journal of Sound and Vibration, vol. 468, p. 115068, Mar. 2020, doi: 10.1016/j.jsv.2019.115068.

[19] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[20] D. Peng, H. Wang, Z. Liu, W. Zhang, M. J. Zuo and J. Chen, "Multi-branch and Multi-scale CNN for Fault Diagnosis of Wheelset Bearings under Strong Noise and Variable Load Condition," in IEEE Transactions on Industrial Informatics.

[21] R. Liu, F. Wang, B. Yang and S. J. Qin, "Multi-scale Kernel based Residual Convolutional Neural Network for Motor Fault Diagnosis Under Non-stationary Conditions," in IEEE Transactions on Industrial Informatics.

[22] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[23] F. Jia, Y. Lei, N. Lu, and S. Xing, “Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization,” Mechanical Systems and Signal Processing, vol. 110, pp. 349–367, 2018.