A Cross Domain Feature Extraction Method for Bearing Fault diagnosis based on Balanced Distribution Adaptation

Jiawei Gu[1], Yanxue Wang[2,*]
School of Mechanical and Electrical Engineering
Guilin University of Electronic Technology
Guilin, China
jiawei__gu@126.com

***Abstract*—Traditional intelligent fault diagnosis techniques for rotating machines have two limitations: 1) Big data with fault information is not available in some cases; 2) The training and testing data are often drawn under discrepant distribution. Thus, transfer component analysis (TCA) has been designed to reduce the distance of marginal distribution between domains. The joint distribution adaptation (JDA) was proposed to simultaneously reduced the difference between the conditional distribution and marginal distribution in source or target domains. However, these two distributions are often treated equally in these existing methods, which will lead to poor performance in practical applications. Therefore, a cross-domain feature extraction method based on balanced distribution adaptation algorithm(BDA) has been proposed, which can adaptively utilize the importance of difference between marginal distribution and conditional distribution. It should be noted that several existing cross domain feature extraction methods can be treated as special cases of BDA. As a new method in the field of transfer learning, BDA is an effective cross-domain feature extraction method. The validity of the BDA algorithm has been successfully evaluated in the actual data set in this paper.**

Keywords-Fault diagnosis; Feature extraction; Cross domain; Balanced distribution adaptation; Transfer learning

I. INTRODUCTION

To date, according to the procedure of diagnosis framework, most of previous studies can be divided into two stages. In the first stage, the diagnosis framework mainly consists of three steps 1) data collection, 2) feature extraction and selection, and 3) fault classification [1-4]. In this framework, massive efforts have been devoted to manual feature extraction and selection. This process benefits from the extensive domain expertise captured by diagnosis specialist, but inevitably requires a large expenditure of labor and time. In the second stage, an adaptive feature learning based diagnostic framework with deep learning technology is emerging [5-8]. This framework provides an end-to-end learning process from input signals to output diagnosis labels. The training process, in which the error estimated by the upper classification layer is back-propagated to update the parameters of lower feature descriptor layers, further guarantees the co-adaptation of the whole network.

Despite its marvelous success, the above frameworks still limited by the lack of data and discrepant data distribution. In this paper, we will solve the problem from another aspect, using a cross-domain method called balanced distribution adaptation (BDA)[9]. Balanced distribution adaptation (BDA) is an algorithm to reduce the distance between conditional domain and marginal domain. It extracts some components from domains by using the so-called Maximum Mean Discrepancy (MMD) in replicated Kernel Hilbert Space (RKHS). In the subspace composed of these components, the data distribution of the domains are very close. Therefore, we can use new representation in the subspace to diagnose faults. Traditional machine learning methods can now train classification models in source domain, and then use them to test target domain.

We will make the following arrangements for the rest of this paper. The concept of BDA algorithm is arranged in the Section 2. In Section 3, a large number of experiments are carried out on CWRU bearing data. The conclusion of this paper are covered in Section 4.

II. RELEATED WORKS

In the Balanced Distribution Adaptation (BDA) algorithm, for a specific task, it adaptively adjusts the importance of the marginal distribution and conditional distribution according to the actual distribution. Specifically, In the BDA algorithm, the method of adjusting the importance of different distributions is to use the balance factor μ:

$$D(\mathcal{D}_s,\mathcal{D}_t) \approx (1-\mu)D\big(P(\boldsymbol{X}_s),P(\boldsymbol{X}_t)\big) + \mu D\left(P\big(y_s \mid \boldsymbol{X}_s\big),P\big(y_t \mid \boldsymbol{X}_t\big)\right) \quad (1)$$

where $\mu \in [0,1]$. If $\mu \longrightarrow 1$, it shows that the datasets are more diverse, so the importance of edge distribution needs to be focused on; If $\mu \longrightarrow 1$, we need to give more attention to the conditional distribution because the dataset is more similar. Therefore, due to the importance of each distribution, the balance factor μ adaptively adjusts the weights, thereby producing good results.

It is notable that, because of the lack of labels in target domain $\mathcal{D}_t$, the evaluate of the conditional distribution $P\big(y_t \mid \boldsymbol{X}_t\big)$ can not be carried out. Instead, class conditional distribution

National Natural Science Foundation of China (51875032, 51475098 and 61463010), Guangxi Natural Science Foundation (2016GXNSFFA380008).

978-1-7281-0860-5/19/$31.00 ©2019 IEEE

$P(\boldsymbol{X}_t \mid y_t)$ can be used to approximate $P(y_t \mid \boldsymbol{X}_t)$ by us. Because when the size of sample is large, $P(\boldsymbol{X}_t \mid y_t)$ and $P(y_t \mid \boldsymbol{X}_t)$ can be completely involved according to the sufficient statistics. To calculate $P(\boldsymbol{X}_t \mid y_t)$, we apply some predictions to $\mathcal{D}_t$ using some basic classifiers trained on $\mathcal{D}_s$ to obtain the soft labels for $\mathcal{D}_t$. We will refine the soft labels repeatedly because they may not be reliable.

The Maximum Mean Discrepancy (MMD) algorithm will be used by us to solve the difference in the marginal distribution of the conditional distribution in Eq. (1). The core of the MMD algorithm is to estimate the difference between the two distributions based on experience. The MMD algorithm has been widely used in many cross-domain learning algorithms, since the MMD algorithm does not require parameters. Therefore, Eq. (1) can be expressed as

$$D(\mathcal{D}_s, \mathcal{D}_t) \approx (1-\mu)\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_{s_i} - \frac{1}{m}\sum_{j=1}^{m}\boldsymbol{X}_{t_j}\right\|_{\mathcal{H}}^2 + \mu\sum_{c=1}^{C}\left\|\frac{1}{n_c}\sum_{\boldsymbol{X}_{s_i}\in\mathcal{D}_s^{(c)}}\boldsymbol{X}_{s_i} - \frac{1}{m_c}\sum_{\boldsymbol{X}_{t_j}\in\mathcal{D}_t^{(c)}}\boldsymbol{X}_{t_j}\right\|_{\mathcal{H}}^2 \tag{2}$$

In Eq.(2), H denotes the regenerative kernel Hilbert space (RKHS), and various class labels are denoted by c $\in \{1, 2, ..., C\}$, and the number of samples in the source and target domains is denoted as n , m, and the sample values of c in the source and target domains can be expressed as $\mathcal{D}_s^{(c)}$ and $\mathcal{D}_t^{(c)}$. At the same time, $n_c = \left|\mathcal{D}_s^{(c)}\right|$, $m_c = \left|\mathcal{D}_t^{(c)}\right|$ can be understood as the number of samples in $\mathcal{D}_s^{(c)}$ and $\mathcal{D}_t^{(c)}$. $\mathcal{D}_s^{(c)}$ represents the marginal distribution distance between domains, and the $\mathcal{D}_t^{(c)}$ is the conditional distribution distance.

Once again, matrixing and regularization of Eq.(1) can be further converted to:

$$\min \operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\left((1-\mu)\mathbf{M}_0 + \mu\textstyle\sum_{c=1}^{C}\mathbf{M}_c\right)\mathbf{X}^{\mathrm{T}}\mathbf{A}\right) + \lambda\|\mathbf{A}\|_F^2 \tag{3}$$
$$\text{s.t. } \mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A} = \mathrm{I}, \quad 0 \le \mu \le 1$$

From Eq.(3), two items can be explored: the adaptation of the balance factor of the marginal distribution and the conditional distribution, and the regularization term. There are two constraints in Eq.(3): first, the transformed data $(\mathbf{A}^{\mathrm{T}}\mathbf{X})$ and the properties inside the original data remain unchanged. Second, the range of the balance factor μ is finite and constitutes the second constraint.

From a deeper interpretation, in Eq. (3), the two input matrices X_s and X_t are abbreviated as $\mathbf{X}$, and where A represents the transformation matrix, $\mathbf{I} \in \mathbb{R}^{(n+m)\times(n+m)}$ represents the identity matrix in Eq. (3), and $\mathbf{H}$ is the centered matrix, which can be understood as $\mathbf{H} = \mathbf{I} - (1/n)\mathbf{1}$. It has the same effect as the JDA algorithm. Both M_0 and M_c belong to the MMD matrix. We can construct it as follows:

$$(\mathbf{M}_0)_{ij} = \begin{cases} \frac{1}{n^2}, & \mathbf{X}_i, \mathbf{X}_j \in \mathcal{D}_s \\ \frac{1}{m^2}, & \mathbf{X}_i, \mathbf{X}_j \in \mathcal{D}_t \\ -\frac{1}{mn}, & \text{otherwise} \end{cases} \tag{4}$$

$$(\mathbf{M}_c)_{ij} = \begin{cases} \frac{1}{n_c^2}, & \mathbf{X}_i, \mathbf{X}_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{m_c^2}, & \mathbf{X}_i, \mathbf{X}_j \in \mathcal{D}_t^{(c)} \\ -\frac{1}{m_c n_c}, & \begin{cases}\mathbf{X}_i \in \mathcal{D}_s^{(c)}, \mathbf{X}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{X}_i \in \mathcal{D}_t^{(c)}, \mathbf{X}_j \in \mathcal{D}_s^{(c)}\end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Learning algorithm: It is known that the Lagrange multiplier is expressed as $\Phi = (\phi_1, \phi_2, \cdots, \phi_d)$, then the Lagrange formula of Eq. (3) can be changed to the following form:

$$L = \operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\left((1-\mu)\mathbf{M}_0 + \mu\sum_{c=1}^{C}\mathbf{M}_c\right)\mathbf{X}^{\mathrm{T}}\mathbf{A}\right) + \lambda\|\mathbf{A}\|_F^2 + \operatorname{tr}\left((\mathbf{I} - \mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A})\Phi\right) \tag{6}$$

In Eq. (6), the value of the derivative $\partial L / \partial \boldsymbol{A} = 0$ is set to 0, and we can derive the above formula into a generalized eigendecomposition problem.

$$\left(\mathbf{X}\left((1-\mu)\mathbf{M}_0 + \mu\sum_{c=1}^{C}\mathbf{M}_c\right)\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{I}\right)\mathbf{A} = \mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A}\Phi \tag{7}$$

Finally, we will analyze Eq. (7), which can get an optimal transformation matrix of the form **A**, in addition to its specific minimum eigenvector.

Estimation of μ: μ and λ are technically different, since μ is not a free parameter and the data distribution must be known before estimating μ. Due to this limitation, no effective estimation method has been obtained for μ. What we can do now is to roughly estimate its value by the performance of μ in real experiments. For practical applications, it is best to estimate μ in continuous cross-validation.

BDA is born out of JDA, but better than JDA is that BDA can adjust its weight according to the marginal distribution between domains and the importance of conditional distribution. From the essence of BDA, the following conclusions can be drawn: when the marginal distribution between multiple distributions is closer, the distance of conditional distribution dominates the transfer learning. But when calculating the conditional distribution, we use the class-like distribution in BDA, that is, $P(\boldsymbol{x} \mid y)$ and $P(y \mid \boldsymbol{x})$ are equivalent. This assumes from the side that the probability of a particular class in each domain is equivalent, but not in the real world. In this chapter, a more effective approximation of the conditional distribution of class imbalance problems is proposed by us.

$$\begin{aligned} &\| P(y_s \mid \boldsymbol{X}_s) - P(y_t \mid \boldsymbol{X}_t) \|_{\mathcal{H}}^2 \\ =&\| \tfrac{P(y_s)}{P(\boldsymbol{X}_s)} P(\boldsymbol{X}_s \mid y_s) - \tfrac{P(y_t)}{P(\boldsymbol{X}_t)} P(\boldsymbol{X}_t \mid y_t) \|_{\mathcal{H}}^2 \\ =&\| \alpha_s P(\boldsymbol{X}_s | y_s) - \alpha_t P(\boldsymbol{X}_t \mid y_t) \|_{\mathcal{H}}^2 \end{aligned} \tag{8}$$

Algorithmically, we will approximate α_s and α_t on two domains (in a priori way). In order to achieve this goal, we further propose weighted equilibrium distribution adaptation (W-BDA), which can balance the proportion of categories in each different domain. Evaluate the difference in conditional distribution in the equation. $P(\boldsymbol{X}_s)$ and $P(\boldsymbol{X}_t)$ in Eq.(8) are marginal distributions, which are prerequisites for evaluating differences in conditional distribution. However, this assessment is not as simple as imagined. $P(\boldsymbol{X}_s)$ and $P(\boldsymbol{X}_t)$ can be fully adapted in the BDA algorithm, so in this section we will assume that they are changing and will not evaluate it. Next, each class is given a weight matrix $\mathbf{W}_c$:

$$(\mathbf{W}_c)_{ij} = \begin{cases} \frac{P(y_s^{(c)})}{n_c^2}, & \boldsymbol{X}_i, \boldsymbol{X}_j \in \mathcal{D}_s^{(c)} \\ \frac{P(y_t^{(c)})}{m_c^2}, & \boldsymbol{X}_i, \boldsymbol{X}_j \in \mathcal{D}_t^{(c)} \\ -\frac{\sqrt{P(y_s^{(c)})P(y_t^{(c)})}}{m_c n_c}, & \begin{matrix} \boldsymbol{X}_i \in \mathcal{D}_s^{(c)}, \boldsymbol{X}_j \in \mathcal{D}_t^{(c)} \\ \boldsymbol{X}_i \in \mathcal{D}_t^{(c)}, \boldsymbol{X}_j \in \mathcal{D}_s^{(c)} \end{matrix} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $P(y_s^{(c)})$ and $P(y_t^{(c)})$ denote the class prior on class c.

Bringing Eq.(9) into the BDA algorithm flow, the tracking optimization process of the W-BDA algorithm is obvious:

$$\min \operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\left((1-\mu)\mathbf{M}_0 + \mu\sum_{c=1}^{C}\mathbf{W}_c\right)\mathbf{X}^{\mathrm{T}}\mathbf{A}\right) + \lambda \parallel \mathbf{A} \parallel_F^2 \tag{10}$$
$$\text{s.t. } \boldsymbol{A}^T\boldsymbol{X}\boldsymbol{H}\boldsymbol{X}^T\boldsymbol{A} = I, \quad 0 \le \mu \le 1$$

Remark: From Eq. (4) and Eq. (9), it can be seen that the BDA algorithm and the W-BDA algorithm are very similar in nature. The biggest difference is that: 1) BDA considers the comparison one-sided, only the current class size will followed. W-BDA will be considered from front to back. 2) Eq. (9) provides a more accurate conditional distribution approximation than Eq. (4), especially when dealing with class imbalances.

The following shows the specific details of the BDA algorithm flow.

Algorithm 1 BDA: Balanced Distribution Adaptation

Input: Source and target feature matrix $\mathbf{X}_s$ and $\mathbf{X}_t$, source label vector y_s, #dimension d, balance factor μ, regularization parameter λ

Output: Transformation matrix $\mathbf{A}$ and classifier f

1: Train a base classifier on $\mathbf{X}_s$ and apply prediction on $\mathbf{X}_t$ to get its soft labels $\hat{y}_t$. Construct $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t]$, initialize $\mathbf{M}_0$ and $\mathbf{M}_c$ by Eq. (4) and (5) (or $\mathbf{W}_c$ using Eq. (9) for W-BDA)

2: **repeat**

3: Solve the eigendecomposition problem in Eq. (7) (or Eq. (10) for W-BDA) and use d smallest eigenvectors to build $\mathbf{A}$

4: Train a classifier f on $\{\mathbf{A}^{\mathrm{T}}\mathbf{X}_s, y_s\}$

5: Update the soft labels of $\mathcal{D}_t$: $\hat{y}_t = f(\mathbf{A}^{\mathrm{T}}\mathbf{X}_t)$

6: Update matrix $\mathbf{M}_c$ using Eq. (5) (or update $\mathbf{W}_c$ using Eq. (9) for W-BDA)

7: **until** Convergence

8: **return** Classifier

III. Performance Analysis Of Feature Classification Method Based On Bda Algorithm

In this section, we will apply the BDA algorithm to the actual fault classification process. The experimental object is the actual data set of the rolling bearing. Before the experiment, we will detail the data set information, and then explain the experimental steps. Finally, the final results of the experiment and the corresponding analysis will be explained.

A. Experimental Dataset and Data Processing

In the previous section, the specific derivation steps and points of attention of the BDA algorithm have been elaborated. Next, we will use it in practical examples, so that we can more easily understand its cross-domain feature extraction ability. Due to the limitations of traditional machine learning algorithms, we will use BDA to convert source and target domains to general purpose space before using machine learning algorithms for classification. After the conversion, the corresponding machine learning classification algorithm is performed. The process of the cross-domain classification method based on BDA algorithm is shown in Fig 1.

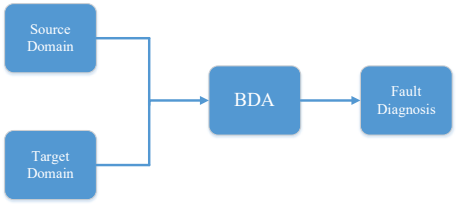

Figure 1. Process of the classification method based on BDA algorithm.

This time we will use the Case Western Reserve University (CWRU) Bearing Data Center [10] for bearing fault classification experiments. CWRU data is obtained from the end of the fan and drive where the accelerometer is installed. There

are two types of normal data and error data in the data set. These two categories are collected under different motor loads and range from 0 to 3 horsepower. Moreover, each set of fault data consists of three fault positions, namely the fault ball, the fault inner ring and the fault outer ring. In addition to the three fault locations, the fault size is divided into four levels, 0.007, 0.014, 0.021, and 0.028 inches.

In this experiment, the data of normal fault data at all locations, and the magnitude of the motor fault at zero load, is selected as the source domain data. We will define normal data as class 1. Ball fault data with a fault size of 0.007 inches will be defined as Class 2 source domain data. In this way, we will get a source domain consisting of 4 different categories of data. The inner race data and the outer race data are formed in the same manner as the third and fourth stages, respectively. The information of the defined data is represented in Table I.

Table I. Source Domain Data

No	Data	Motor Load	Fault Size
1	Normal Data	0	0.007
2	Ball Faulty Data	0	0.007
3	Inner Race Faulty Data	0	0.007
4	Outer Race Faulty Data	0	0.007

The target domain consists of normal data and fault data for all three locations with a fault size of 0.007 inches. In the experiment, we will use three bearing fault data under different loads to form three different target domains, which can more effectively evaluate the performance of the BDA algorithm. The target domain data is shown in Table II.

Table II. Target Domain Data

No	Data	Motor Load	Fault Size
1	Normal Data	1/2/3	0.007
2	Ball Race Faulty Data	1/2/3	0.007
3	Inner Race Faulty Data	1/2/3	0.007
4	Outer Race Faulty Data	1/2/3	0.007

According to the above method, we divide the source domain data into 4 categories, and the labels of the source domain data are all known. The data in the target domain can also be divided into 4 categories, but the target domain data labels are unknown. In this case we will classify the target domain data. From the data introduction, we can see that the data distribution of the source domain data and the target domain data is likely to be different, so the tried and tested machine learning algorithm will be very difficult to play a good role.

B. Traditional Machine Learning Methods

In this chapter, we apply different machine learning algorithms to the cross-domain feature extraction process and finally evaluate their performance. Here we will select three machine learning algorithms, including KNN, SVM, and Random Forest to verify the performance of the cross-domain feature extraction algorithm..

As can be seen from the above description, the label of the source domain is known, and the label of the target domain is unknown, so we will use the data and labels of the source domain to train the classifier. After training the classifier, the data of the target domain is loaded into the classifier for testing. We will conduct three sets of experiments. In the first set of experiments, we used the source domain data to train the classifier, then tested the data in target domain 1, and the second and third groups of experiments and so on. The experimental results are shown in Table III.

Table III. Experimental Results of Traditional Machine Learning methods

Accuracy/%	KNN	SVM	Random Forest
Target Domain 1	17.26	23.65	30.49
Target Domain 2	13.15	20.83	26.58
Target Domain 3	**21.77**	**27.33**	**32.46**

C. Feature Extraction based on BDA Algorithm

Similar to the traditional intelligent fault diagnosis experiment, the source domain data is first used to train the classifier, and then the data of the target domain will be loaded into the classifier for testing. At the same time, the source and target domains will be converted to the same subdomain, thanks to the BDA algorithm. The experimental results are shown in Table IV.

Table IV. Experimental Results of BDA Algorithm

Accuracy/%	KNN	SVM	Random Forest
Target Domain 1	28.47	37.73	48.86
Target Domain 2	26.83	35.61	43.15
Target Domain 3	**34.86**	**41.92**	**54.86**

D. Experimental Results Analysis

The above results are the results of the classification of the traditional machine learning algorithm and the BDA algorithm. Since we only validated the validity of the BDA algorithm in this experiment, there is no feature extracted from the original data. Therefore, the accuracy rate is not as high as after feature extraction. Traditional machine learning algorithms cannot reduce the distribution distance between different domains, so the performance will be worse when classifying. In contrast, the BDA algorithm can convert the source and target domains into the same subdomain, thereby reducing the difference between distance and distribution. In summary, when we perform cross-domain feature classification, we can first load the data into the BDA algorithm and then classify it to get better results.

IV. CONCLUSION

In this paper, we innovatively proposed a cross-domain feature classification method based on BDA algorithm. As the most advanced cross-domain feature extraction algorithm, balanced distribution adaptive can effectively reduce the distance between different domains. And verified on the actual bearing fault data set. The experimental results show that the cross-domain feature classification algorithm solves the domain imbalance problem more effectively..

ACKNOWLEDGMENT

This research was partially supported by National Natural Science Foundation of China (No. 51875032, 51475098 and 61463010), Guangxi Natural Science Foundation (No. 2016GXNSFFA380008).

REFERENCES

[1] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches," IEEE Transactions on Industrial Electronics, vol. 62, no. 6, pp. 3757-3767, 2015.

[2] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part II: Fault Diagnosis with Knowledge-Based and Hybrid/Active Approaches.pdf," IEEE Transactions on Industrial Electronics, vol. 62, no. 6, pp. 3768-3774, 2015.

[3] C. Liu, D. Jiang, and W. Yang, "Global geometric similarity scheme for feature selection in fault diagnosis," Expert Systems with Applications, vol. 41, no. 8, pp. 3585-3595, 2014.

[4] Z. Shen, X. Chen, X. Zhang, and Z. He, "A novel intelligent gear fault diagnosis model based on EMD and multi-class TSVM," Measurement, vol. 45, no. 1, pp. 30-40, 2012.

[5] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," Mechanical Systems and Signal Processing, vol. 72-73, pp. 303-315, 2016.

[6] O. Costilla-Reyes, P. Scully, and K. B. Ozanyan, "Deep Neural Networks for Learning Spatio-Temporal Features From Tomography Sensors," IEEE Transactions on Industrial Electronics, vol. 65, no. 1, pp. 645-653, 2018.

[7] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Deep Residual Networks With Dynamically Weighted Wavelet Coefficients for Fault Diagnosis of Planetary Gearboxes," IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4290-4300, 2018.

[8] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network," IEEE Transactions on Industrial Electronics, vol. 65, no. 3, pp. 2727-2736, 2018.

[9] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced Distribution Adaptation for Transfer Learning," presented at the 2017 IEEE International Conference on Data Mining (ICDM), 2017.

[10] Bearing data Centre, Case Western Reserve University, Available: http://csegroups.case.edu/bearingdatacenter/home