Graph Assisted Unsupervised Domain Adaptation for Machine Fault Diagnosis

Naibedya Pattnaik*, Kriti Kumar*†, A Anil Kumar*, M Girish Chandra*

* Embedded Devices and Intelligent Systems, TCS Research, Bangalore, India.
† Indraprastha Institute of Information Technology Delhi, New Delhi, India.
Email: {p.naibedya, kriti.kumar, achannaanil.kumar, m.gchandra}@tcs.com

***Abstract*—Domain adaptation has emerged as a useful technology that leverages the knowledge learned from labeled data in the source domain to build an effective classifier for the target domain, given that the source and target data have different underlying distributions. Most of the existing techniques aim to address the marginal distribution discrepancy alone, ignoring the conditional distribution divergence that may exist between the two domains. In order to achieve good adaptation performance, both the marginal and conditional distribution of the source and target data need to be aligned. The problem becomes challenging when the data is limited, and no labels are available for the target domain data. To address this scenario, in this paper, we propose a novel formulation that utilizes the concepts of Class-wise Maximum Mean Discrepancy and label propagation over the graph for unsupervised domain adaptation. We have applied it to the challenging scenario of adaptation between physically different but related machines for fault diagnosis. Two publicly available bearing fault datasets have been employed for performance evaluation. The experimental results indicate the superior performance of the proposed method compared to the state-of-the-art methods showing efficient transfer learning capabilities for machine fault diagnosis.**

***Index Terms*—Domain adaptation, Maximum mean discrepancy, Graph total variation, Label propagation, Machine fault diagnosis**

I. Introduction

Classical machine learning algorithms assume that, the training and test data follow the same data distribution. However, in practice, this assumption does not always hold, which leads to deterioration in their performance [1]. Interestingly, Domain Adaptation (DA) has emerged as one of the upcoming methods to tackle this issue, where the training (source) and test (target) data can be from different distributions [2]. DA relies on leveraging the information learned from well studied source domain to improve the classification performance on the target domain. According to the availability of label information in the target domain, DA can be categorized as Unsupervised DA (UDA) where the target domain is completely unlabeled, and semi-supervised DA (SDA) where the target domain has limited labels [3].

Among all the existing DA methods, divergence and adversarial learning based techniques have been successfully applied in different applications [4]–[9]. Divergence based DA techniques map instances from both source and target domains to a common feature space to learn domain invariant features [4]–[7]. However, they fail to perform when a large distribution discrepancy exists between the two domains. Adversarial learning based DA methods are able to handle such a scenario, as they learn data translation between source and target domains by training a generator and discriminator network [8], [9]. However, these methods do not guarantee that class discriminability is preserved during the data translation [10]. Also, they require massive data for training, which may not be always available in many practical application scenarios.

Apart from the methods mentioned above, graph-based techniques have recently been used for DA, as graphs can capture the actual data manifolds effectively [11]. The existing techniques are based on Graph Convolutional Networks (GCN) [12], [13], Graph Signal Processing (GSP) [11], [14], and hybrid methods that utilize divergence method with graph [15] to learn domain invariant features. The work in [12] proposed an unsupervised Domain Adaptive Network Embedding (DANE) framework using GCN and adversarial network that learns transferable embeddings between the source and target domain. Another UDA method [13] utilized a dual GCN for local and global consistency for feature aggregation. Although popular, these methods ignore the property of graph structured data while carrying out classification [16]. To effectively exploit the underlying structure of the data, the work in [14] utilized the concepts of GSP for SDA. The method is based on aligning the Fourier bases of the graphs constructed using source and target domain data. The spectrum of the labels learned from the source graph is transferred to the target graph for DA. This work was extended by incorporating graph learning into the optimization formulation that aligns the spectrum of the graphs corresponding to the source and target data [11], which resulted in improved performance. Another work [15] proposed a Graph Adaptive Knowledge Transfer (GAKT) method that jointly optimized the domain invariant feature learning by weighted class-wise adaptation loss and label propagation over the graph. A joint graph is employed by augmenting source and target domain data to propagate the labels from known source to unknown target data. All the aforementioned methods mainly focus on computer vision related DA applications. However, the focus of our work is on time series data for the challenging adaptation scenario of machine inspection.

In most practical applications of machine inspection, access to labeled data is difficult, as manual labeling is time consuming and inducing faults in machines is not economically viable.

Moreover, labeled data of every machine is not available. Thus, transferring the knowledge learned from labeled data of one machine (source) to a different but related machine (target) is important and required in practice. This is a challenging adaptation scenario since the data distribution of both domains is significantly different due to different working conditions, sampling frequency, location of sensor placement, etc.

To address these challenges, we propose a novel Graph Assisted Unsupervised Domain Adaptation (GA-UDA) method for machine fault diagnosis in this paper. It carries out domain adaptation in two stages. In the first stage, Class-wise MMD (CMMD) loss is minimized to transform the data from both domains to a shared feature space. In the second stage, the augmented transformed (projected) data from both domains are utilized to construct a joint graph. Subsequently, the labels of target domain data are estimated through label propagation over the joint graph. This work is similar in nature to the GAKT method [15]. However, unlike the fixed joint graph considered in GAKT learnt using raw sensor data, the proposed method iteratively updates the joint graph using transformed features of both domains obtained through the optimization formulation. This helps in addressing significant distribution shift between the two domains. We have evaluated our method for the challenging scenario of adaptation between different but related machines where the data from both domains is *limited*. Application to bearing fault diagnosis has been considered since bearings are critical elements of rotating machines that are more vulnerable to damage [17]. Experimental results are provided with Case Western Reserve University (CWRU) [18], and Paderborn University datasets [19]. The results show that GA-UDA method outperforms all the state-of-the-art methods, which indicates the effectiveness of the proposed method for the considered adaptation task.

The rest of the paper is organized as follows. Section II presents the problem statement, and a brief background of the CMMD and GTV used to formulate our proposed method. A detailed description of the proposed GA-UDA method is provided in Section III. Section IV presents the experimental results with details of the datasets and benchmark methods used for comparison. Finally, Section V presents the conclusion.

II. Problem Statement and Background

This section briefly presents the problem statement and the necessary details on the CMMD and GTV to help understand our proposed GA-UDA method.

A. Problem Statement

The focus of this work is on UDA. Let the labeled source domain S data be expressed as $\{X_s, Y_s\} = \{(x_{s_1}, y_{s_1}), ..., (x_{s_{n_s}}, y_{s_{n_s}})\}$, where $X_s \in R^{m \times n_s}$ denotes m features of n_s samples and $Y_s \in R^{n_s \times C}$ is the one-hot encoded labels with C number of classes. The unlabeled target domain T data be expressed as $\{X_t\} = \{x_{t_1}, ..., x_{t_{n_t}}\}$, where $X_t \in R^{m \times n_t}$ denotes m features of n_t samples. Given that distribution discrepancy exists between S and T, the task is to predict the labels of the target domain data X_t, assuming the feature and label space to be the same across both domains.

B. Class-wise Maximum Mean Discrepancy

Maximum Mean Discrepancy (MMD) is one of the popular techniques used to address the domain discrepancy between the S and T domains [15], [20]. It computes the deviation of sample means of two domains in the projected space. More formally, the MMD loss C_1 is expressed as [15]:

$$\begin{aligned} C_1(P_s, P_t) &= \| \frac{1}{n_s} \sum_{i=1}^{n_s} P_s^T x_{s_i} - \frac{1}{n_t} \sum_{j=1}^{n_t} P_t^T x_{t_j} \|_2^2 \\ &= \| \frac{P_s^T X_s \mathbf{1}_{n_s}}{n_s} - \frac{P_t^T X_t \mathbf{1}_{n_t}}{n_t} \|_2^2 \end{aligned} \quad (1)$$

where $P_s \in R^{m \times k}$ and $P_t \in R^{m \times k}$ are two projection matrices with $k < m$, x_{s_i} and x_{t_j} are the i^{th} and j^{th} sample of X_s and X_t, $\mathbf{1}_{n_s}$ and $\mathbf{1}_{n_t}$ are column vectors of all one of size n_s and n_t respectively. By reducing the deviation (loss), MMD tries to align the marginal distribution of the S and T data. However, it fails to address the conditional distribution discrepancy that may exist between the two domains [21].

To address this, Class-wise Maximum Mean Discrepancy (CMMD) has been proposed that computes the difference between sample means of two similar class data from different domains [22]. This method requires the knowledge of labels for both domains. Since the target domain data is unlabeled, in most works, pseudo labels are generated by applying the classifier trained on labeled source domain data to the target domain data [20]. The weighted CMMD loss, C_2 is expressed as [15]:

$$\begin{aligned} C_2(P_s,P_t,F_t) &= \sum_{c=1}^{C} \| \frac{1}{n_s^c} \sum_{i=1}^{n_s^c} P_s^T x_{s_i}^c - \frac{1}{n_t^c} \sum_{j=1}^{n_t} f_t^{(c,j)} P_t^T x_{t_j} \|_2^2 \\ &= \| P_s^T X_s Y_s N_s - P_t^T X_t F_t N_t \|_F^2 \end{aligned} \quad (2)$$

where $f_t^{(c,j)}$ denotes the probability of the j^{th} target domain data belonging to the c^{th} category are obtained from the target label predictions (pseudo labels F_t), and $\|.\|_F$ is the frobenius norm. n_s^c and n_t^c denote the number of samples in the c^{th} class for source and target domain respectively. N_s and N_t are diagonal matrices of size $C \times C$ with c^{th} diagonal elements as $\frac{1}{n_s^c}$ and $\frac{1}{n_t^c}$ respectively. Here, n_t^c is calculated as $\sum_{j=1}^{n_t} f_t^{(c,j)}$.

C. Label Propagation using Graph Total Variation

Graph signal processing has emerged as one of the most effective methods for signal modeling [23]. It involves a graph structure G and a graph signal F residing on that structure. The graph structure is represented as $G = \{V, E, W\}$, where V is the set of vertices and E is the set of edges connecting those vertices with weights specified in the weight matrix W. Given the data $X \in R^{m \times n}$ with m features of n samples, a graph of n vertices can be constructed using $W \in R^{n \times n}$ obtained from Gaussian kernel that is expressed as [23]:

$$W_{ij} = \exp(-\|x_i - x_j\|^2 / 2\sigma^2) \quad (3)$$

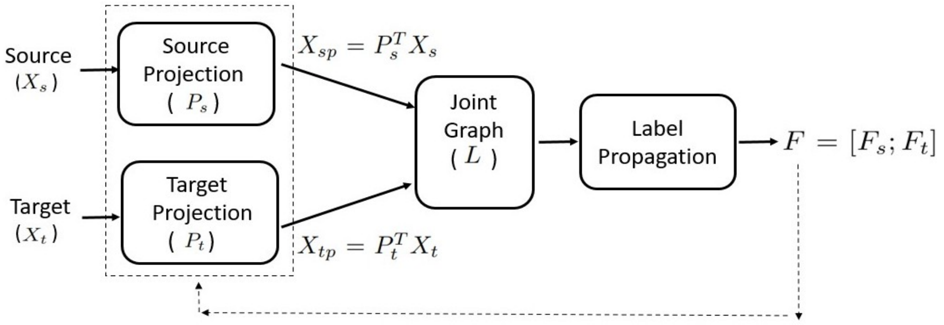

Fig. 1. Block diagram of the proposed Graph-Assisted Method for Unsupervised Domain Adaptation

where σ is the scaling factor, and x_i, x_j are the feature vectors at i^{th} and j^{th} vertices of the graph, respectively.

One of the important matrices associated with graphs is the graph Laplacian. The un-normalized graph Laplacian is expressed as $L = D - W \in R^{n \times n}$ where D is the degree matrix, which is a diagonal matrix whose diagonal entries are expressed as $D_{ii} = \sum_j W_{ij}$. The normalized graph Laplacian is expressed as $L_n = D^{-1/2}(D-W)D^{-1/2}$. The graph signal $(F : V \to R)$ is a function that takes real value at each vertex of G. The variation of this signal over the underlying graph structure is defined by Graph Total Variation (GTV), expressed as [11]:

$$F^T L F = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} (F(i) - F(j))^2 \tag{4}$$

where $F(i)$, $F(j)$ denotes the labels at i^{th} and j^{th} vertices of the graph, respectively.

Most of the applications in GSP involve minimizing the GTV to ascertain that the graph signal is in agreement with the underlying graph structure [23]. Thereby ensuring a smooth transition of the signal over the graph. This term has been popularly used for label propagation, where data form the graph structure and labels are considered as the graph signal. If the data residing at the two vertices are similar, minimizing the GTV term enables the labels at those vertices to be similar as well. Using these concepts, the details of the proposed method for UDA are presented in the subsequent section.

III. Proposed GA-UDA Method

The flow diagram of the proposed GA-UDA method is presented in Fig. 1. Here, the two projection matrices P_s and P_t are learned from the source and target domain data X_s and X_t, respectively, by minimizing the weighted CMMD loss. Then, the projected data from both the domains, represented as $X_{sp} = P_s^T X_s \in R^{k \times n_s}$ and $X_{tp} = P_t^T X_t \in R^{k \times n_t}$ are utilized to construct a joint graph (G) of $n = n_s + n_t$ samples by augmenting X_{sp} and X_{tp} ($X_p = [X_{sp}; X_{tp}]$) using the Gaussian kernel in (3). The probabilistic target domain labels F_t are estimated through label propagation over G. The optimization formulations for learning P_s, P_t in the first stage, and F_t in the second stage are given below.

The projection matrices P_s and P_t are initialized using Principal Component Analysis (PCA) over X_s and X_t, respectively. Taking $P = [P_s, P_t]$, the projections P_s and P_t are learned by minimizing the following loss [15]:

$$\begin{aligned} P &= \min_{P^T \mathbf{S} P = I_k} \| P_s^T X_s \bar{Y}_s \bar{N}_s - P_t^T X_t \bar{F}_t \bar{N}_t \|_F^2 + \alpha \| P_s - P_t \|_{2,1} \\ &= \min_{P^T \mathbf{S} P = I_k} \text{tr}(P^T \mathbf{T} P) + \alpha \text{tr}(P^T \mathbf{M} P) \end{aligned} \tag{5}$$

where $\bar{Y}_s = [\mathbf{1}_{n_s}, Y_s]$, $\bar{F}_t = [\mathbf{1}_{n_t}, F_t]$, $\bar{N}_s$ = diag$(1/n_s, N_s)$ and $\bar{N}_t$ = diag$(1/n_t, N_t)$,

$$\mathbf{S} = \begin{bmatrix} X_s H_s X_s^T, & 0 \\ 0, & X_t H_t X_t^T \end{bmatrix}, \mathbf{M} = \begin{bmatrix} M, & -M \\ -M, & M \end{bmatrix},$$

$$\mathbf{T} = \begin{bmatrix} X_s \bar{Y}_s \bar{N}_s \bar{N}_s \bar{Y}_s^T X_s^T, & X_s \bar{Y}_s \bar{N}_s \bar{N}_t \bar{F}_t^T X_t^T \\ X_t \bar{F}_t \bar{N}_t \bar{N}_s \bar{Y}_s^T X_s^T, & X_t \bar{F}_t \bar{N}_t \bar{N}_t \bar{F}_t^T X_t^T \end{bmatrix}.$$

Here, H_s and H_t denote the centering matrices expressed as $I_{n_s} - \frac{1}{n_s} \mathbf{I}_{n_s}$ and $I_{n_t} - \frac{1}{n_t} \mathbf{I}_{n_t}$ respectively [20], where $\mathbf{I}_{n_s}$ and $\mathbf{I}_{n_t}$ are all one matrices of size n_s and n_t. $M \in R^{m \times m}$ is a diagonal matrix whose i^{th} diagonal entry $M_{ii} = 1/||G_i||_2$ if $G_i \neq 0$, otherwise $M_{ii} = 0$. G_i is the i^{th} row vector of $P_s - P_t$. While the first term in (5) is similar to (2), the second term is added to constrain the learned source and target projections to be similar with the help of hyperparameter α [24]. Here, $\|.\|_{2,1}$ stands for matrix $l_{2,1}$ norm. The optimization formulation in (5) is solved by formulating it as a generalized Eigen-decomposition problem: $(\mathbf{T} + \alpha \mathbf{M})\gamma = \beta \mathbf{S} \gamma$ [15]. Here, the eigenvectors (γ_i) are corresponding to minimum eigen values (β). We update subspace projections as $P = [\gamma_0, ..., \gamma_{p-1}]$.

The pseudo labels F_t of the target domain data are updated through label propagation over the graph by minimizing the graph total variation in (4). Considering the augmented label matrix $F = [F_s; F_t] \in R^{n \times C}$, where $F_s = Y_s$, F_t is computed as:

$$\min_F \text{tr}(F^T L_n F) = \min_F \text{tr}(F^T \begin{bmatrix} L_{ss}, & L_{st} \\ L_{ts}, & L_{tt} \end{bmatrix} F), \text{ s.t. } F \geq 0 \tag{6}$$

where L_n is the normalized graph Laplacian for the graph G obtained using the augmented projected data X_p from the two domains using (3). Since L_n is symmetric, $L_{st} = L_{ts}^T$, solving for (6), results in the following closed form update $F_t = -L_{tt}^{-1} L_{ts} F_s$ [25]. Once F_t is computed, N_t is also updated.

We alternately solve for the two stages, iteratively till convergence is met, which is achieved when the loss in (6)

TABLE I
CLASSIFICATION RESULTS FOR $(CWRU \rightarrow Paderborn)$

Method	Acc	P	R	F1
JMMD	53.86	42.29	53.86	45.3
MK-MMD	36.25	35.57	36.25	33.42
CORAL	45.34	34.91	45.34	35.98
DANN	44.32	33.55	44.32	36.9
CDAN	44.66	31.3	44.66	34.92
GAKT	71.88	84.78	71.88	65.54
GA-UDA ($\sigma = 5, \alpha = 2$)	**90.9**	**92.93**	**90.9**	**90.67**

TABLE II
CLASSIFICATION RESULTS FOR $(Paderborn \rightarrow CWRU)$

Method	Acc	P	R	F1
JMMD	76.93	76.56	76.93	73.67
MK-MMD	82.73	88.22	82.73	81.74
CORAL	54.66	41.63	54.66	46.21
DANN	71.02	78.7	71.02	67.07
CDAN	80.68	75.11	80.68	76.12
GAKT	66.5	49.85	66.5	55.38
GA-UDA ($\sigma = 4, \alpha = 2$)	**93.29**	**94.73**	**93.29**	**93.1**

goes below an empirically computed threshold. Subsequently, the class label corresponding to the highest probability in the target predictions F_t, are considered as the final target labels Y_t.

IV. EXPERIMENTAL STUDY

This section briefly describes the bearing fault datasets and the benchmark methods used to analyze the performance of the proposed method. Subsequently, details of the experimental study and results are discussed.

A. Data Description

1) CWRU Dataset: This bearing dataset is collected by Case Western Reserve University (CWRU) [18]. It contains vibration data captured from the drive and fan end of the machine at a sampling frequency of 12 kHz. It has data for four different loading conditions (0, 1, 2, and 3 Horse Power (Hp)) with rotating speeds of 1797, 1772, 1750, and 1730 rpm, respectively. The data has four classes: Normal, Inner-race Fault (IF), Outer-race Fault (OF), and Bearing-race Fault (BF). Here, faults of different sizes (0.007, 0.014, 0.021 inches) are induced using electro-discharge machining (EDM).

2) Paderborn Dataset: This bearing dataset is collected from Paderborn University [19]. It contains vibration and stator current signals collected from a test rig consisting of a drive motor, a torque measurement shaft, the test modules, and a load motor. Data for both real and artificially damaged bearings are available with a sampling frequency of 64 kHz for two rotating speeds (900 and 1500 rpm) and loading torques (0.7 and 0.1 Nm). The data has three classes: Normal, Inner-race Fault (IF), and Outer-race Fault (OF). We have used only the vibration data in our analysis with faults introduced using EDM.

B. Benchmark Methods

The proposed method is compared against five state-of-the-art UDA methods for bearing fault diagnosis and the Graph based DA (GAKT) [15] method for performance evaluation. The UDA methods for bearing fault diagnosis include mapping-based methods like Joint Maximum Mean Discrepancy (JMMD), Multi Kernels Maximum Mean Discrepancy (MK-MMD), CORrelation ALignment (CORAL), and adversarial learning based methods like Domain Adversarial Neural Network (DANN) and Conditional Domain Adversarial Network (CDAN). They have been successfully used for adaptation between different working conditions of the same machine. In this work, they are evaluated for the difficult scenario of adaptation between physically different but related machines for bearing fault diagnosis. These methods are implemented following the formulation given in [26] that considers the same deep CNN backbone and bottleneck architecture for all the methods.

C. Experiments and Results

This work considers a challenging adaptation scenario where the source and target data belong to physically different but related machines. Here, adaptation is considered between CWRU and Paderborn datasets for bearing fault detection and classification. Note that the bearing specifications, sampling frequency, and working conditions are different for both datasets, making it a challenging adaptation scenario. CWRU data with 0 Hp motor torque and 0.007 inch fault size, collected from the drive end, and Paderborn data with 900 rpm and 0.7 loading torque have been utilized for our experimentation. The Paderborn dataset is downsampled to 12 kHz to match the sampling frequency of CWRU dataset. The raw data is pre-processed by taking a sliding window of 1024 length, which results in 351 samples for each dataset. Five relevant time domain features, namely RMS, variance, data peak, kurtosis, and peak to peak are extracted from the raw data. They are well-studied features for bearing fault diagnosis that carry class discriminative information [27]. For a fair comparison, these features are fed as input to all the methods, and a three class classification problem is considered: Normal, IF, and OF.

The performance for all the methods is assessed using Accuracy (Acc), Precision (P), Recall (R), and F1 score (F1). To simulate a data limited scenario, experimentation is carried out considering 50 % train-test split. The average results obtained using five randomly generated train-test sets are summarized in the tables, with the best performing method highlighted in bold. Table I and II provide the classification results for CWRU → Paderborn and Paderborn → CWRU, respectively, where $S \rightarrow T$ denotes adaptation from source to the target domain. Note the optimal value of the hyperparameters σ and α for GA-UDA method are obtained using grid search and are mentioned in the tables.

For the case of $CWRU \rightarrow Paderborn$, Table I shows that the mapping and adversarial learning based DA methods do not perform well for limited data scenarios. Even with domain-specific features as input, they fail to learn discriminative representations from the data. On the other hand, for the $Paderborn \rightarrow CWRU$ case, the performance of these methods is comparatively better. However, for both cases, the proposed GA-UDA performs better than all the other methods. From both Table I and II, we can observe that GA-UDA significantly improves over the GAKT method with ≈ 19 % and 27 % increase in accuracy, respectively. Note the distribution shift between the sensor data of the two domains is significant for the challenging adaptation scenario considered here. In the GAKT method, since the graph is constructed using raw sensor data from both domains, large domain discrepancy results in two disjoint sub-graphs corresponding to the source and target data, which affects the label propagation leading to poor performance. Unlike the static graph in GAKT, the graph in GA-UDA is updated iteratively using the transformed features learned through the optimization formulation till convergence is met. This allows the proposed method to effectively handle the distribution shift between the two domains, thereby providing more reliable adaptation results.

V. Conclusion

This paper presents a novel formulation employing Class-wise Maximum Mean Discrepancy and label propagation on the graph for unsupervised domain adaptation. The method has been evaluated for bearing fault diagnosis using publicly available datasets for domain adaptation. The proposed method provides superior performance compared to the benchmark methods for the challenging data-limited scenario of adaptation between different but related machines. The experimental results demonstrate the applicability of the proposed method for domain adaptation.

It is important to note that this method is generic and can be applied to other application domains. In future, we will focus on establishing the versatility of the proposition by experimenting with data sets of different domains.

References

[1] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia, "A brief review of domain adaptation," *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

[2] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, "Universal domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2720–2729.

[3] Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang, "Semi-supervised models are strong unsupervised domain adaptation learners," *arXiv preprint arXiv:2106.00417*, 2021.

[4] Djebril Mekhazni, Amran Bhuiyan, George Ekladious, and Eric Granger, "Unsupervised domain adaptation in the dissimilarity space for person re-identification," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*. Springer, 2020, pp. 159–174.

[5] Xiang Li, Wei Zhang, Qian Ding, and Jian-Qiao Sun, "Multi-layer domain adaptation method for rolling bearing fault diagnosis," *Signal processing*, vol. 157, pp. 180–197, 2019.

[6] Xincheng Cao, Binqiang Chen, and Nianyin Zeng, "A deep domain adaption model with multi-task networks for planetary gearbox fault diagnosis," *Neurocomputing*, vol. 409, pp. 173–190, 2020.

[7] Xiaoxia Wang, Haibo He, and Lusi Li, "A hierarchical deep domain adaptation approach for fault diagnosis of power plant thermal system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5139–5148, 2019.

[8] Hossein Shahabadi Farahani, Alireza Fatehi, and Mahdi Aliyari Shoorehdeli, "On the application of domain adversarial neural network to fault detection and isolation in power plants," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 1132–1138.

[9] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018.

[10] Jinyang Jiao, Jing Lin, Ming Zhao, Kaixuan Liang, and Chuancang Ding, "Cycle-consistent adversarial adaptation network and its application to machine fault diagnosis," *Neural Networks*, vol. 145, pp. 331–341, 2022.

[11] Elif Vural, "Domain adaptation on graphs by learning graph topologies: theoretical analysis and an algorithm," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 3, pp. 1619–1635, 2019.

[12] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin, "Dane: Domain adaptive network embedding," *arXiv preprint arXiv:1906.00684*, 2019.

[13] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu, "Unsupervised domain adaptive graph convolutional networks," in *Proceedings of The Web Conference 2020*, 2020, pp. 1457–1467.

[14] Mehmet Pilancı and Elif Vural, "Domain adaptation on graphs by learning aligned graph bases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 587–600, 2020.

[15] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–52.

[16] Ruichu Cai, Fengzhu Wu, Zijian Li, Pengfei Wei, Lingling Yi, and Kun Zhang, "Graph domain adaptation: A generative view," *arXiv preprint arXiv:2106.07482*, 2021.

[17] Pankaj Gupta and Mohan K Pradhan, "Fault detection analysis in rolling element bearing: A review," *Materials Today: Proceedings*, vol. 4, no. 2, pp. 2085–2094, 2017.

[18] Wade A Smith and Robert B Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mechanical systems and signal processing*, vol. 64, pp. 100–131, 2015.

[19] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification," in *PHM Society European Conference*, 2016, vol. 3.

[20] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[21] Wei Wang, Baopu Li, Shuhui Yang, Jing Sun, Zhengming Ding, Junyang Chen, Xiao Dong, Zhihui Wang, and Haojie Li, "A unified joint maximum mean discrepancy for domain adaptation," *arXiv preprint arXiv:2101.09979*, 2021.

[22] Wei Wang, Haojie Li, Zhengming Ding, and Zhihui Wang, "Rethink maximum mean discrepancy for domain adaptation," *arXiv preprint arXiv:2007.00689*, 2020.

[23] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[24] Jing Zhang, Wanqing Li, and Philip Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1859–1867.

[25] Lichen Wang, Zhengming Ding, and Yun Fu, "Adaptive graph guided embedding for multi-label annotation," in *IJCAI*, 2018.

[26] Zhibin Zhao, Qiyang Zhang, Xiaolei Yu, Chuang Sun, Shibin Wang, Ruqiang Yan, and Xuefeng Chen, "Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–28, 2021.

[27] Saleem Riaz, Hassan Elahi, Kashif Javaid, and Tufail Shahzad, "Vibration feature extraction and analysis for fault diagnosis of rotating machinery-a literature survey," *Asia Pacific Journal of Multidisciplinary Research*, vol. 5, no. 1, pp. 103–110, 2017.