Transform Based Subspace Interpolation for Unsupervised Domain Adaptation Applied to Machine Inspection

Kriti Kumar[†⋆], Angshul Majumdar[†], A Anil Kumar[⋆], M Girish Chandra[⋆]
[⋆] TCS Research, India.
[†] IIIT Delhi, New Delhi, India.
Email: {kriti.kumar, achannaanil.kumar, m.gchandra}@tcs.com, angshul@iiitd.ac.in

***Abstract*—In most practical application scenarios, several factors may introduce domain discrepancy between the train (source domain) and test (target domain) data. Domain adaptation techniques address this domain shift to ensure reliable inferencing. However, limited data and the unavailability of annotations of the target domain data pose an additional challenge for the adaptation task. Unlike divergence and adversarial learning-based techniques that are data-hungry, subspace modeling-based techniques are found more suitable for learning representations from limited data. This work presents a novel subspace interpolation-based method via transform learning for unsupervised domain adaptation. Transform learning framework has been used for subspace modeling that provides superior performance in terms of accuracy, computational complexity, and improved convergence over dictionaries. They model the subspace that links the source and target domain data and generates domain invariant features for cross-domain analysis. The potential of the proposed method is demonstrated using the challenging scenario of adaptation between different but related machines using two public datasets. Experimental results show the effectiveness of the proposed method compared to the state-of-the-art methods for machine diagnosis.**

***Index Terms*—Subspace modeling, Transform learning, Domain adaptation, Unsupervised learning, Machine fault diagnosis**

I. Introduction

With the rapid development of industrial data and IoT, prognostics and health management of industrial machines are becoming increasingly popular. It is an essential component in industry 4.0 as it helps to maximize throughput by outage prevention. Recently various intelligent fault diagnosis systems have come up that depend on deep representation learning techniques to ensure the reliable operation of machines. However, for these techniques to be effective, they need *huge amount of labeled data* for training. Additionally, they assume train and test data to follow *similar distribution*. These conditions cannot be assured in most practical application scenarios of machine inspection. Labeled data is usually scarce and challenging to collect (manual labeling is costly); moreover, faults are rare events. Domain discrepancy between the train (source domain) and test (target domain) data may arise due to various factors like change in the operating conditions that are affected by a change in speed, torque, sensor placement, bearing/gearbox specifications, working environment, etc. This domain discrepancy needs to be suitably addressed to ensure the reliable performance of the models.

Of late various deep learning-based techniques have been explored for adaptation that addresses the domain shift between the source and target domain for machine diagnosis. They provide domain invariant features for learning more general diagnosis by utilizing labeled samples from multiple domains. Due to the scarcity of labeled samples in such application scenarios, the focus has been on Unsupervised Domain Adaptation (UDA) techniques. Our current application focus is on bearing health monitoring since bearings are critical element for all rotating equipment in machines. They are often used under extreme loads making them vulnerable to damage. In literature, the two popular methods explored for unsupervised DA for bearing health monitoring are: (i) Divergence-based methods and (ii) Adversarial learning-based methods. These methods differ in the way they align the source and target domains for learning domain invariant feature representation.

Divergence-based methods like Joint Maximum Mean Discrepancy (JMMD) [1], Multi Kernels Maximum Mean Discrepancy (MK-MMD) [2], and CORrelation ALignment (CORAL) [3], extract domain invariant features by minimizing the respective divergence criteria between the source and target domain data distributions. On the other hand, adversarial learning-based methods like Domain Adversarial Neural Network (DANN) [4], and Conditional Domain Adversarial Network (CDAN) [5] minimize the distribution discrepancy of the two domains through an adversarial objective with respect to a domain discriminator for robust fault diagnosis. In general, adversarial learning-based methods are shown to perform better than divergence-based methods for machine fault diagnosis [6]. However, all these methods require huge amount of data and consider adaptation between different working conditions of the *same machine*. They do not consider adaptation between *different but related machines*, that is required in practice. For example, to transfer the knowledge acquired using the labeled data from one machine (lab setup or simulator) to different but related machines (industrial machines) for reliable diagnosis. Moreover, availability of *limited data* poses an additional challenge.

We address this scenario by employing subspace interpolation-based methods that have been successfully applied for feature augmentation in different adaptation tasks in the field of computer vision [7]–[10]. They seem to work for *limited data* scenarios. Unlike former methods, instead of minimizing the divergence, these methods align the two domains by generating discriminative features that are common (or invariant) across both the source and target domains, enabling cross-domain classification. They generate intermediate feature representations along a virtual path connecting the source and target domains. As opposed to the domain subspaces obtained using Principle Component Analysis (PCA) that may result in information loss, data-driven dictionaries (with non-orthogonal columns) have been employed that provide more flexibility to model and adapt the domain data [11]–[14]. Although data-driven dictionaries have proved to be quite successful in different applications, the approximate synthesis sparse coding algorithms can be computationally expensive [15]. Hence, Transform learning (TL) based techniques have gained more importance and have shown to be advantageous over dictionaries in terms of accuracy and complexity, and provide improved convergence [15], [16]. Especially in the image domain, transform learning methods produce state-of-the-art results [15], [17].

Motivated by the advantages of transform learning, in this work, we present a novel formulation employing Transform Learning based subspace interpolation for Unsupervised Domain Adaptation (TL-UDA) to address the challenging scenario of adaptation between *different but related machines*. To the best of our knowledge, this technique has not been explored so far in the literature. Unlike the work in [12] that uses dictionaries, this work uses transforms for modeling the sub-space that connects the source and target domain. Interpolated subspaces are learned using the source, intermediate, and target transforms that capture the domain shift between the source and target domain data. They provide a domain-invariant feature space for cross-domain analysis. Initial experimentation on different machine datasets demonstrates the superior performance of the proposed technique compared to the state-of-the-art techniques. The improvements achieved over the dictionary variant (Dictionary Learning for Unsupervised Domain Adaptation (DL-UDA) [12] are also highlighted, which indicates the efficacy of the proposed technique for adaptation tasks.

The paper is organized as follows. Section II presents a brief background on Transform Learning that forms the basis of our proposed formulation for domain adaptation. Subsequently, the problem definition and details of our proposed TL-UDA method are described in detail in Section III. Section IV provides the comparisons and results, with the conclusion in Section V.

II. Background on Transform Learning

Transform learning is an analysis approach for learning data representations. Given the data $\boldsymbol{X} \in \mathbb{R}^{L\times N}$ with L features and N samples, a transform $\boldsymbol{T} \in \mathbb{R}^{K\times L}$ of K atoms on its rows is learnt such that it produces the coefficients $\boldsymbol{Z} \in \mathbb{R}^{K\times N}$. The basic formulation is given as:

$$\boldsymbol{TX} = \boldsymbol{Z} \tag{1}$$

More formally, for learning sparse representations, the transform learning problem is formulated as [16]:

$$\min_{\boldsymbol{T},\boldsymbol{Z}} \|\boldsymbol{TX} - \boldsymbol{Z}\|_F^2 + \lambda(\|\boldsymbol{T}\|_F^2 - \log\det \boldsymbol{T}) + \mu\|\boldsymbol{Z}\|_0. \tag{2}$$

Here, the additional constraints on $\boldsymbol{T}$ and $\boldsymbol{Z}$ prevent trivial solutions, control the condition number of the learned transform, and ensure the computed coefficients are sparse, respectively.

The optimization problem in (2) is solved using alternating minimization technique to obtain the closed form updates of $\boldsymbol{Z}$ and $\boldsymbol{T}$ [15]. The sub-problem to solve for $\boldsymbol{Z}$ is expressed as:

$$\boldsymbol{Z} \leftarrow \min_{\boldsymbol{Z}} \|\boldsymbol{TX} - \boldsymbol{Z}\|_F^2 + \mu\|\boldsymbol{Z}\|_0 \tag{3}$$

The closed-form update is obtained by simple hard-thresholding expressed as:

$$\boldsymbol{Z} = (abs(\boldsymbol{TX}) \geq \mu).\ \boldsymbol{TX} \tag{4}$$

where the term in the bracket is hard thresholded against the value μ and '.' denotes the element-wise product. The sub-problem to solve for $\boldsymbol{T}$ is expressed as:

$$\boldsymbol{T} \leftarrow \min_{\boldsymbol{T}} \|\boldsymbol{TX} - \boldsymbol{Z}\|_F^2 + \lambda(\|\boldsymbol{T}\|_F^2 - \log\det \boldsymbol{T}). \tag{5}$$

Here, Cholesky decomposition is employed followed by singular value decomposition to compute the update for $\boldsymbol{T}$. It is expressed as: $\boldsymbol{X}\boldsymbol{X}^T + \lambda \boldsymbol{I} = \boldsymbol{L}\boldsymbol{L}^T$ and $\boldsymbol{L}^{-1}\boldsymbol{X}\boldsymbol{Z}^T = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$. This results in the following closed-form update for $\boldsymbol{T}$:

$$\hat{\boldsymbol{T}} = 0.5\boldsymbol{V}(\boldsymbol{S} + (\boldsymbol{S}^2 + 2\lambda\boldsymbol{I})^{1/2})\boldsymbol{U}^T\boldsymbol{L}^{-1}. \tag{6}$$

With this brief introduction, the proposed formulation for domain adaptation via transform learning is presented in the subsequent section.

III. Transform Learning for Unsupervised Domain Adaptation

This section describes the problem definition. Subsequently, the proposed method for unsupervised domain adaptation via subspace interpolation employing TL method is described in detail.

A. Problem Definition

The problem focus is on unsupervised adaptation between the source domain S and target domain T where, the source and target data have different underlying data distribution. More formally, given the labeled source domain data $\boldsymbol{X}_s \in \mathbb{R}^{d\times n_s}$ with d features of n_s measurements associated with $\boldsymbol{Y}_s$ labels and the target domain data $\boldsymbol{X}_t \in \mathbb{R}^{d\times n_t}$ with d features of n_t measurements, the objective is to estimate the labels of $\boldsymbol{X}_t$ given the data distributions, $P(X_s) \neq P(X_t)$ but the feature and label space are same for both the domains.

B. Proposed Method TL-UDA

This formulation also follows the idea of subspace interpolation for domain adaptation presented in [12], but uses data-driven transforms instead of dictionaries to learn the source S, intermediate and target T domains. Starting with source domain transform $\boldsymbol{T}_0 \in \mathbb{R}^{k \times d}$ associated with the source domain data $\boldsymbol{X}_s$ with k atoms, a set of intermediate transforms $\boldsymbol{T}_m, m \in [1, M-1]$ (intermediate domains) are learned by transforming the target data $\boldsymbol{X}_t$, iteratively in the direction to reduce the residue on the target data till we reach $\boldsymbol{T}_M$ that best represents the target domain data $\boldsymbol{X}_t$. Fig. 1 presents the block diagram of TL-UDA that shows the different subspaces modeled by different transforms obtained by interpolation on the target data.

This method employs a training phase for learning the virtual path that connects S and T domains in terms of intermediate transforms that capture the domain shift between the two domains. This helps in generating domain invariant features for classification. Later, this mapping is utilized in the test phase for generating domain-invariant features from the target data for estimating target labels. More details on the two phases are presented in the following.

1) Training Phase: First the source domain transform $\boldsymbol{T}_0$ for the source data $\boldsymbol{X}_s$ is learnt by solving:

$$\min_{\boldsymbol{T}_0, \boldsymbol{Z}_0} \|\boldsymbol{T}_0 \boldsymbol{X}_s - \boldsymbol{Z}_0\|_F^2 + \lambda(\|\boldsymbol{T}_0\|_F^2 - \log \det \boldsymbol{T}_0) + \mu \|\boldsymbol{Z}_0\|_0 \tag{7}$$

The above expression is similar to (2). The source transform $\boldsymbol{T}_0$ and the coefficients $\boldsymbol{Z}_0$ are obtained using the standard updates (3) to (6).

Considering M subspaces with $m \in [0, M]$, for each m, the m^{th} domain transform $\boldsymbol{T}_m$ is applied on the target data X_t to generate the coefficients $\boldsymbol{Z}_m$ following the update of (3) and the residue $\boldsymbol{J}_m$ is computed using the following:

$$\boldsymbol{Z}_m \leftarrow \min_{\boldsymbol{Z}_m} \|\boldsymbol{T}_m \boldsymbol{X}_t - \boldsymbol{Z}_m\|_F^2 + \mu \|\boldsymbol{Z}_m\|_0 \tag{8}$$

$$\boldsymbol{J}_m = \boldsymbol{T}_m \boldsymbol{X}_t - \boldsymbol{Z}_m \tag{9}$$

The new transform $\boldsymbol{T}_{m+1}$ is computed by estimating $\boldsymbol{\Delta T}_m$ that represent the adjustment in the transform atoms between the transforms of $\boldsymbol{T}_{m+1}$ and $\boldsymbol{T}_m$ that helps in reducing the residue $\boldsymbol{J}_m$. $\boldsymbol{\Delta T}_m$ is learnt using by solving:

$$\boldsymbol{\Delta T}_m \leftarrow \min_{\boldsymbol{\Delta T}_m} \|\boldsymbol{\Delta T}_m \boldsymbol{J}_m - \boldsymbol{Z}_m\|_F^2 + \lambda(\|\boldsymbol{\Delta T}_m\|_F^2 - \log \det \boldsymbol{\Delta T}_m) \tag{10}$$

The above sub-problem has a form similar to (5), hence we follow the same update for computing $\boldsymbol{\Delta T}_m$ by appropriately changing the input from $\boldsymbol{X}$ to $\boldsymbol{J}_m$. Note here the first term reduces the residue and the second term discourages abrupt changes in the transforms of adjacent domains.

Subsequently, the new transform $\boldsymbol{T}_{m+1}$ is obtained as:

$$\boldsymbol{T}_{m+1} = \boldsymbol{T}_m + \eta \boldsymbol{\Delta T}_m \tag{11}$$

where η is introduced to ensure a smooth transition between the transforms of neighbouring domains.

Now the new transform $\boldsymbol{T}_{m+1}$ associated with the next intermediate domain is used on the target data to compute the residue in the feature space. This process continues iteratively till $\boldsymbol{\Delta T}_m \leq \tau$ (threshold), suggesting the learnt intermediate domain transforms fully absorb the domain shift between S and T domains. The last obtained transform $\boldsymbol{T}_M$ is considered as the target transform that completely represents the target data. Kindly note we consider normalization for all the learned transforms, where the rows of all the transforms are normalized to unit norm. The pseudocode of the proposed TL-UDA method is summarized in Algorithm 1.

Algorithm 1 Subspace Interpolation using TL for UDA (TL-UDA)

```
Input: X_s, X_t
Parameters: λ, μ, τ, η, k (number of transform atoms)
Initialization: Set transform T_0 to random matrix with real
  numbers between 0 and 1 drawn from a uniform distribution,
  m = 0
Compute source transform T_0 and Z_0 with X_s using (7).
do
    Transform X_t with T_m using (8).
    Compute the residue J_m using (9).
    Estimate the adjustment in transform atoms ΔT_m using (10).
    Update the transform T_{m+1} using (11).
    Normalize each row of the transform to unit norm.
    m = m + 1
while (ΔT_m ≥ τ)
Output: {T_m}_{m=0}^M (source, intermediate and target transforms)
```

Once the transition path between the two domains is learnt, invariant sparse codes (coefficients) are applied across the source, intermediate, and target transforms ($\{\boldsymbol{T}_m\}_{m=0}^{M}$) to form new features for classification. The new feature space is given as: $[(\boldsymbol{T}_0^{-1} Z)^T, (\boldsymbol{T}_1^{-1} Z)^T, .., (\boldsymbol{T}_M^{-1} Z)^T]$ where $Z \in \mathbb{R}^k$ are the sparse codes generated either by transforming source data $\boldsymbol{X}_s$ with $\boldsymbol{T}_0$ (i.e., $\boldsymbol{Z}_0$) or transforming target data $\boldsymbol{X}_t$ with $\boldsymbol{T}_M$ (i.e., $\boldsymbol{Z}_M$). Since the labels $\boldsymbol{Y}_s$ are known only for the source data $\boldsymbol{X}_s$, the classifier is trained using features obtained by applying $\boldsymbol{Z}_0$ across the source, intermediate and target transforms. Please note, here we have used an SVM classifier but in general any suitable classifier can be employed.

2) Test Phase: To estimate the labels $\boldsymbol{Y}_t^{test}$ associated with the test target data $\boldsymbol{X}_t^{test}$, first $\boldsymbol{Z}_M^{test}$ is computed by applying $\boldsymbol{T}_M$ on $\boldsymbol{X}_t^{test}$. Subsequently, features are computed by applying $\boldsymbol{Z}_M^{test}$ across the source, intermediate and target transforms. These test features are fed to the classifier learnt in the training phase to estimate the test target labels.

IV. Experiments and Results

In this work, two publicly available bearing datasets, namely CWRU and Paderborn, are considered for the performance evaluation of the proposed method. More details on the datasets, benchmark methods used for comparison, and experimental results are presented in the subsequent sections.

A. Dataset Description

1) CWRU Dataset: This dataset contains vibration data acquired from the drive and fan end of the machine for bearing

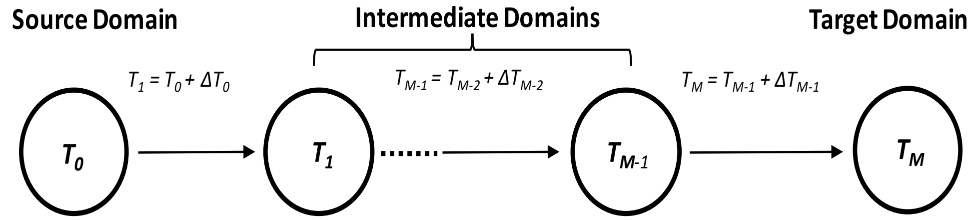

Fig. 1: Domain Adaptation via Interpolation over M Subspaces using TL

health monitoring. The experimental setup is described in [18]. It has data for four different working conditions (loading torques 0, 1, 2, and 3 Hp with speeds of 1797, 1772, 1750, and 1730 rpm, respectively) collected at a sampling frequency of 12 kHz. It contains both healthy and faulty bearing data with faults like Inner-race Fault (IF), Outer-race Fault (OF), and Bearing-race Fault (BF) of different sizes (0.007, 0.014, 0.021 inches).

2) Paderborn Dataset: This dataset contains current and vibration data acquired from a test rig consisting of a drive motor, a torque measurement shaft, the test modules, and a load motor [19]. It has data for two rotating speeds (900 and 1500 rpm) and two loading torques (0.7 and 0.1 Nm), collected at a sampling frequency of 64 kHz. It contains healthy and faulty bearing data with faults like Inner-race Fault (IF) and Outer-race Fault (OF).

B. Benchmark Methods

Five state-of-the-art methods for UDA focusing on bearing fault diagnosis are considered for evaluating the performance of our proposed method. Three of them utilize the divergence loss in terms of Joint Maximum Mean Discrepancy (JMMD), Multi Kernel Maximum Mean Discrepancy (MK-MMD), and CORrelation ALignment (CORAL), respectively. The other two make use of adversarial learning-based networks, namely, Domain Adversarial Neural Network (DANN), and Conditional Domain Adversarial Network (CDAN). The implementation of all these methods given in [6] is considered that employs the same deep CNN backbone and bottleneck architecture for all the methods. Additionally, comparison with the dictionary variant (DL-UDA) [12] is presented to demonstrate the enhancement gained with the transform version for adaptation tasks. Since the improvement obtained with adaptation is well demonstrated in the literature over methods without adaptation, here we present the results obtained with adaptation alone. We experimented with both raw data and domain-specific features as input to the different methods. We observed that features turned out to be more effective than using raw data directly for the challenging *limited data* scenario of adaptation between *different but related machines*. Domain-specific features combined with representation learning capability of the different methods resulted in effective adaptation. Hence, the subsequent section presents the results obtained with domain-specific features as input to all the methods.

C. Results and Discussion

Here we present the results for adaptation between CWRU (0 Hp loading torque and 0.007 inch fault size, drive end) and Paderborn (900 rpm and 0.7 loading torque) datasets. Note that the bearing specifications of both datasets and loading conditions are different, making it a challenging scenario for adaptation. Paderborn data is downsampled to 12kHz, and the same number of data samples as CWRU data are used for adaptation. Three class classification is considered namely, Healthy, Inner-race Fault (IF), and Outer-race Fault (OF).

For both datasets, vibration data is pre-processed by splitting into non-overlapping windows of 1024 samples, resulting in 351 samples for each dataset. Five time domain features relevant to the bearing fault detection, namely RMS, variance, maximum, kurtosis, and peak-to-peak value [20], are extracted from each data sample. Experiments are carried out using these features as input to all the different methods. Data is randomly split into train-test with 50% samples taken for training and the remaining considered for testing. Precision (P), Recall (R), F1-score (F1), and Accuracy (Acc) metrics are used for performance evaluation. Tables I and II present the results in %, averaged over five random train-test splits of the data for $CWRU \rightarrow Paderborn$ and $Paderborn \rightarrow CWRU$ adaptation, respectively. The best-performing method is highlighted in bold. The number of atoms (k) of the dictionaries/transforms for the respective methods and other hyperparameters are tuned using grid search. The optimal values for TL-UDA method are $k = 5$, $\lambda = 1$, $\mu = 0.1$, $\eta = 0.04$, $\gamma = 0.1$. The number of atoms k considered for the DL and TL-based methods are mentioned in the Tables. For the datasets considered in this work, convergence for DL-UDA method was achieved with $M = 3$ subspaces; hence the results for TL-UDA are reported for the same number of subspaces for a fair comparison.

TABLE I: Classification Results for $CWRU \rightarrow Paderborn$

Method	P	R	F1	Acc
JMMD	42.29	53.86	45.3	53.86
MK-MMD	35.57	36.25	33.42	36.25
CORAL	34.91	45.34	35.98	45.34
DANN	33.55	44.32	36.9	44.32
CDAN	31.3	44.66	34.92	44.66
DL-UDA($k = 10$)	82.28	84.25	78.95	84.19
TL-UDA($k = 5$)	**92.73**	**92.31**	**92.19**	**92.23**

TABLE II: Classification Results for $Paderborn \rightarrow CWRU$

Method	**P**	**R**	**F1**	**Acc**
JMMD	76.56	76.93	73.67	76.93
MK-MMD	88.22	82.73	81.74	82.73
CORAL	41.63	54.66	46.21	54.66
DANN	78.7	71.02	67.07	71.02
CDAN	75.11	80.68	76.12	80.68
DL-UDA($k = 10$)	89.77	84.63	83.49	86.34
TL-UDA($k = 5$)	**93.16**	**90.48**	**90.12**	**90.51**

From both tables, one can observe that the deep learning-based benchmark methods do not perform well for the considered adaptation scenario. This is because more data is required for them to learn meaningful representations. Moreover, the accuracy of these methods also depends on the backbone model used for implementation [6]; hence, the results for the two adaptation cases are not consistent. While they perform well for $Paderborn \rightarrow CWRU$, the results obtained for $CWRU \rightarrow Paderborn$ case are poor. On the other hand, the subspace-based DL and TL methods demonstrate a superior performance compared to other benchmarks for both cases. We observe that with domain features, representation learning is more effective for data limited scenarios. Moreover, with small-sized transforms ($k = 5$, square transform), TL-UDA performs better than DL-UDA ($k = 10$, overcomplete dictionary) for both cases of adaptation. This demonstrates the computational advantage of using transforms over dictionaries. The initial results obtained are promising. They indicate the potential and applicability of the proposed TL-UDA method for the challenging adaptation scenario in machine inspection space.

V. Conclusion

The paper presents a novel transform learning-based subspace interpolation method to link the source and target domain data for unsupervised adaptation. Transforms are learned for the source, intermediate, and target domains to generate a shared feature space for robust classification. The details on the formulation and closed-form updates are presented. For data limited scenario, domain-specific features combined with representation learning using transforms allows meaningful mapping to be learnt for effective adaptation. Experimental results obtained for machine diagnosis using publicly available bearing datasets demonstrate the effectiveness of the proposed TL-DA for unsupervised DA. The proposed method shows improved performance over all the benchmark methods, including the dictionary variant, suggesting its applicability to real-life applications.

In the future, we plan to address more complex adaptation scenarios like adaptation from reference datasets collected through lab setups or simulators to real industrial machines. Please note here that we have considered machine data for adaptation, but the method is generic and can cater to data from different application scenarios.

References

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

[2] X. Li, W. Zhang, Q. Ding, and J. Sun, "Multi-layer domain adaptation method for rolling bearing fault diagnosis," *Signal processing*, vol. 157, pp. 180–197, 2019.

[3] X. Wang, H. He, and L. 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.

[4] H. S. Farahani, A. Fatehi, and M. A. 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)*, 2020, pp. 1132–1138.

[5] X. Yu, Z. Zhao, X. Zhang, C. Sun, B. Gong, R. Yan, and X. Chen, "Conditional adversarial domain adaptation with discrimination embedding for locomotive fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[6] Z. Zhao, Q. Zhang, X. Yu, C. Sun, S. Wang, R. Yan, and X. 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.

[7] R. Gopalan, Ruonan Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *2011 International Conference on Computer Vision*, 2011, pp. 999–1006.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[9] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 769–776.

[10] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2481–2488.

[11] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Berlin, Heidelberg, 2012, pp. 631–645, Springer Berlin Heidelberg.

[12] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 692–699.

[13] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

[14] B. Yang, A. J. Ma, and P. C. Yuen, "Domain-shared group-sparse dictionary learning for unsupervised domain adaptation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, S. A. McIlraith and K. Q. Weinberger, Eds. 2018, pp. 7453–7460, AAAI Press.

[15] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learning part i: Algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 625–636, 2015.

[16] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, 2013.

[17] J. Maggu and A. Majumdar, "Kernel transform learning," *Pattern Recognition Letters*, vol. 98, pp. 117 – 122, 2017.

[18] W. A. Smith and R. 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] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. 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] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539–1548, 2018.