Automated domain adaptation for bearings fault detection and classification

Fadi KARKAFI[1,2,*], Amani RAAD[3], Dany ABBOUD[1], Yosra MARNISSI[2], Guillaume DOQUET[2], Mohammed EL BADAOUI[2]

[1]Univ Lyon, INSA-Lyon, LVA, EA677, F-69621 Villeurbanne, France
[2]Safran Tech, Rue des Jeunes Bois – Châteaufort 78772 Magny-les-Hameaux, France
[3]Université Libanaise Faculté de Génie I, Tripoli, Liban
[4]Univ Lyon, UJM St-Etienne, LASPI, EA3059, F-42023 Saint-Etienne, France

*fadi.karkafi@insa-lyon.fr

Abstract

Effective fault classification of mechanical components based on vibration signals requires robust domain adaptation methods, particularly in scenarios where complete data on all fault types is unavailable. The challenge lies in monitoring the health of components with variable characteristics, such as speed, load, and torque, where the lack of information on their current state renders target domain labels unavailable. Maximum Mean Discrepancy (MMD) is a widely used loss function in domain adaptation; however, its popular Gaussian kernel necessitates prior knowledge of the target domain data, posing a limitation in domain adaptation problems. To address this challenge, this paper proposes a novel approach that aims to achieve two key goals: (1) accurate fault identification and (2) automatic tuning of the MMD Gaussian kernel to adapt to different domains. The proposed approach consists of two stages. Firstly, a static representation of the standard deviation is identified using the Pascal Triangle that allows the reconstruction of a Gaussian kernel accurately. In the second stage, a dynamic parameter is computed by considering the difference between the source and target features distribution, extracted from the desired layers of the considered model. The proposed approach is evaluated by the classification of bearings health state, using the Case Western Reserve University (CWRU) signals as the source dataset and Jiangnan University (JNU) signals as the target one. The model employed in this evaluation is a 2D Convolutional Neural Network (CNN) model designed to process 2D reshaped signals. The desired layers for adaptation were identified as the fully connected layers of the network. The results demonstrate that the proposed method outperforms traditional MMD with a manually tuned Gaussian kernel, as well as other domain adaptation methods such as Correlation Alignment (CORAL) and Wasserstein distance that do not require any parameter tuning, making it a valuable contribution to the field of intelligent fault classification based on vibration signals.

1 Introduction

Mechanical components, such as bearings, play a critical role in various industries and compact environments like aircraft engines, where monitoring their health is essential for ensuring machine and engine safety and reliability [1]. Vibration signal analysis has emerged as one of the most common methods for monitoring the health of bearings during operation [2]. However, the vibration signatures of bearings are influenced by factors such as speed, load, torque, and other operating conditions, making it challenging to accurately diagnose different fault types based on vibration signals. To overcome this challenge, robust domain adaptation methods are required for intelligent fault diagnosis based on vibration signals that can effectively handle the variability in operating conditions. Domain adaptation is the process of adapting a model trained on a source data domain to perform well on a new target domain, even when the distribution of the target data may differ from the source domain [3], [4], [5]. In the case of bearings fault classification, obtaining complete data on all fault types in the target domain is often not feasible, making it difficult to

acquire target domain labels. Maximum Mean Discrepancy (MMD) is a widely used discrepancy metric in domain adaptation that has shown promising results in various fields [6], [7], [8], [9]. However, using MMD presents a challenge in tuning the parameter of the chosen kernel, such as the variance for the Gaussian kernel, which is used to measure the distance between the distributions of the source and target domains. This requirement of prior knowledge for the target domain data contradicts the nature of the domain adaptation problem. To address the limitations of existing methods, this paper proposes a novel approach for automated deep domain adaptation specifically applied to bearings fault classification. The approach focuses on estimating the hyper-parameter of the MMD Gaussian kernel in two stages. In the first stage, a static representation of the standard deviation is determined using the Pascal Triangle, enabling accurate reconstruction of a Gaussian kernel for uniform change of discrepancies. In the second stage, a dynamic parameter is introduced, extracted from the desired layers of the model, to weigh the static representation in response to distribution changes during iterations. The remainder of the paper is organized as follows: Section 2 provides a detailed description of the preliminary concepts and related work in domain adaptation, including popular methods for estimating distribution discrepancies. Section 3 presents the proposed two-stage methodology, also considering the possibility of higher order moments. Section 4 presents the experimental results, encompassing dataset description, pre-processing techniques, and evaluation metrics. Finally, Section 5 concludes the paper by summarizing the findings and outlining potential directions with such approach.

2 Preliminaries

2.1 Transfer Learning

To provide a clear understanding of the problem at hand, certain concepts related to the Transfer Learning (TL) are first introduced. A domain is defined by $D = \{\chi, P(X)\}$; where χ is the feature space, $P(X)$ is the marginal probability distribution and $X \in \chi$. Two domains D_{source} and D_{target} are considered to have different distributions if $P_{source}(X_{source}) \neq P_{target}(X_{target})$ [9]. A task is defined by $T = \{y, f(X)\}$; where y is the label space, $f(X) = P(Y \mid X)$ is the conditional probability distribution and $Y \in y$. In the context of fault diagnosis, specifically in bearings, TL is applied as a domain adaptation problem where $X_s = X_t$ and $P_s(X_s) \neq P_t(X_t)$ such that both datasets revolve around the same context (i.e. bearing defects) but considering different conditions (angular speed, load, torque, etc…) [10]. Since the data in the source and target domains are collected under different working conditions, a shift in the distributions occurs, leading to a degradation in classification performance, as illustrated in Figure 1. In domain adaptation, the availability of a source domain dataset with labels $\left(x_i^s, y_i^s\right)_{i=1}^{n_s}$ and a target domain dataset without labels $\left(x_i^t\right)_{i=1}^{n_t}$ is assumed, where n_s and n_t represent respectively the number of samples in the source and target domain. Therefore, the target conditional distribution deviation is not considered due to the absence of the target labels while the samples from the source domain should be sufficient to construct an accurate source classifier. Thus, the primary objective of the domain adaptation is to reduce the distribution discrepancy between the source and target datasets during or after training by adapting the classifier learned from the source domain to effectively separate the target samples. To achieve this goal, various methods have been proposed that aim to minimize the difference in distribution between the source and target domains, such as Correlation Alignment, Wasserstein Distance, Maximum Mean Discrepancy, …

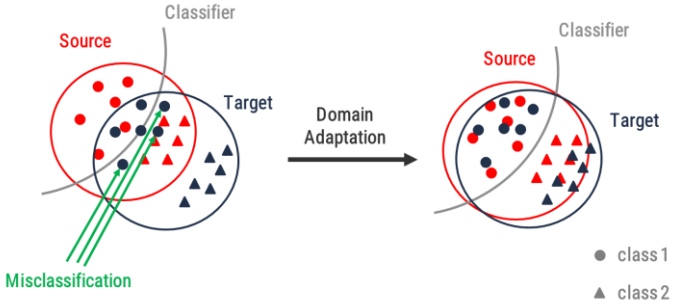


Figure 1: Illustration for domain shift and domain adaptation [11].

2.2 CORrelation ALignment (CORAL)

Correlation Alignment or CORAL is an efficient and simple unsupervised adaptive method commonly used to measure the distribution gap between the source and the target domains in pattern recognition because it aligns the distributions of both domains by exploring their second order statistic. The key calculation involved in CORAL is the covariance matrix in each domain. When incorporated into a neural network, it can be summarized as follows [13]:

$$L_{CORAL} = \frac{1}{4d^2}\left\|\mathrm{cov}(H_s) - \mathrm{cov}(H_t)\right\|_F^2 \tag{1}$$

Where H_s and H_t are of dimension d, $\mathrm{cov}(X)$ is the covariance matrix of X and $\|X\|_F$ is the Frobenius norm of X which is calculated by:

$$\|X\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|x_{i,j}\right|^2} \tag{2}$$

2.3 Wasserstein

Wasserstein distance, also referred to as Earth Mover's Distance, is a mathematical measure of the distance between two probability distributions [14]. It can also be interpreted as the minimum amount of effort required to transform one distribution into another, where effort is measured as the amount of distribution weight that must be multiplied by the distance it must be moved. This is a solid extension of the GAN model by shifting the focus from a discriminator that predicts the probability of an image being "real" to a critical model that assesses the "reality" of an image. The Wasserstein distance can be expressed as:

$$L_{Wasserstein} = \int_{-\infty}^{+\infty}\|U - V\|_F \tag{3}$$

Where U and V are the source and the target datasets respectively and $\|X\|_F$ is the Frobenius norm of X.

2.4 Maximum Mean Discrepancy

Although both CORAL and Wasserstein are non-parametric methods, it was empirically shown several times that Maximum Mean Discrepancy (MMD) outperforms both approaches and specially with the Gaussian kernel [6], [7], [8], [9]. MMD is a measure of the distribution deviation without explicitly calculating the intermediate density. The squared MMD distance between the source dataset X^s and the target dataset X^t is defined as [15]:

$$\begin{aligned} D_H^2\left(X^s, X^t\right) &= \left\|\frac{1}{n_s}\sum_{i=1}^{n_s}\varphi\left(X_i^s\right) - \frac{1}{n_t}\sum_{j=1}^{n_t}\varphi\left(X_i^t\right)\right\|_H^2 \\ &= \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}k\left(X_i^s, X_j^s\right) + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}k\left(X_i^t, X_j^t\right) - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}k\left(X_i^s, X_j^t\right) \end{aligned} \tag{4}$$

Where H is the Reproducing Kernel Hilbert Space (RKHS) and $k(a,b)$ is a Gaussian kernel function:

$$k(a,b) = e^{\frac{-|a-b|^2}{2\sigma^2}} \tag{5}$$

And where σ is the bandwidth parameter that influences the distribution of the studied signal.

3 Methodology

As seen in the formula (5) above, the Gaussian kernel requires a variance to be tuned with respect to both data distributions. One way to address is by applying standard Grid Search with respect to the objective metric such as the accuracy. However, this is not feasible since the target labels are not accessible in a

domain adaptation problem. Therefore, this variance should be deduced purely on each dataset distribution without any prior knowledge. In order to reduce the computational cost of the MMD method, all computations are performed in 2D. This transform enables the possibility to replicate the iteration summation as matrix multiplication, which significantly reduces the overall computation time. This strategy has been previously explored in other studies [16], which has shown its effectiveness in reducing computational costs for similar tasks.

3.1 Static representation

To estimate the variance of the 2D Gaussian kernel, a static representation is employed using the Pascal Triangle or binomial coefficients. The size of the Gaussian kernel is set to match the size of the data obtained from a specific layer of the model. In this paper, the fully connected layers are considered as they contain important information for classification. Thus, the width of the 2D Gaussian kernel is set to the square root of the 1D distribution size. Therefore, the Pascal Triangle of order $L = \sqrt{N}$, rounded to the nearest integer, is computed to determine the respective variance, such that N being the length of the chosen fully connected layer. This variance can be deduced from [17] and the Appendix as follows:

$$\sigma = \frac{2^{L-1}}{\sqrt{2\pi}\max(P_L)} \tag{6}$$

where P_L is the Pascal Triangle of order L.

3.2 Dynamic representation

While the static representation of the Gaussian kernel provides a fixed variance, it may not be optimal as the source and target distributions can significantly differ during model weight adjustments. To address this limitation, a dynamic representation is introduced to consider the difference between the source and target distributions. This dynamic representation acts as a weight for the static one, allowing for a more accurate estimation of the variance in the target domain. The product of the dynamic representation and the static representation yields the final variance estimation during each iteration, as shown in Eq. (7).

$$\sigma^2 = \sigma_{static} \times \sigma_d \tag{7}$$

Such that the variance σ of each pair of distributions depends on the dynamic one σ_d related to this pair. In addition, σ_{static} is the static representation since the same layer is considered upon iterations.

In a general manner, the goal is to minimize the discrepancy between two distributions A and B of sizes a and b respectively using a Gaussian kernel. This can be formulated as:

$$p\left(A_1, A_2, ..., A_a \mid B_1, B_2, ..., B_b, \sigma_d^2\right) = \prod_{i=1}^{a}\prod_{j=1}^{b} \mathrm{N}\left(A_i; B_j, \sigma_d^2\right) \tag{8}$$

where $\mathrm{N}(\cdot)$ denotes the probability density function of the Gaussian distribution defined as follows:

$$\mathrm{N}\left(A_i; B_j, \sigma_d^2\right) = \frac{1}{\sqrt{2\pi\sigma_d^2}} e^{\frac{-\left|A_i - B_j\right|^2}{2\sigma_d^2}} \tag{9}$$

Using the maximum Likelihood principle, one can estimate then σ_d^2 by maximizing the likelihood function, $L(\sigma_d^2)$, which is given by:

$$L(\sigma_d^2) = \prod_{i=1}^{a}\prod_{j=1}^{b} \frac{1}{\sqrt{2\pi\sigma_d^2}} e^{\frac{-\left|A_i - B_j\right|^2}{2\sigma_d^2}} \tag{10}$$

$$\begin{aligned}
\hat{\sigma}_d^2 &= argmax\left(L(\sigma_d^2)\right) \\
&= argmin\left(-\ln\left(L(\sigma_d^2)\right)\right) \\
&= argmin\left(-\frac{ab}{2}\ln\left(2\pi\sigma_d^2\right) - \frac{1}{2\sigma_d^2}\sum_{i=1}^{a}\sum_{j=1}^{b}\left|A_i - B_j\right|^2\right) \\
&= argmin\left(\ell\left(\sigma_d^2\right)\right)
\end{aligned}$$

To find a local minima of the negative log-likelihood function, the derivative of $\ell\left(\sigma_d^2\right)$ with respect to σ_d^2 is taken and set equal to zero:

$$\frac{\partial \ell}{\partial \sigma_d^2} = 0$$
$$\frac{\partial \ell}{\partial \sigma_d^2} = -\frac{ab}{2\sigma_d^2} + \frac{1}{2\sigma_d^4}\sum_{i=1}^{a}\sum_{j=1}^{b}\left|A_i - B_j\right|^2 = 0$$

$$\hat{\sigma}_d^2 = \frac{1}{ab}\sum_{i=1}^{a}\sum_{j=1}^{b}\left|A_i - B_j\right|^2 \tag{11}$$

Therefore, the final variance, being the product of static and dynamic representation, is estimated during each iteration with respect to each kernel as follows:

$$\begin{aligned}
\hat{\sigma}_{X_s,X_s}^2 &= \frac{2^{L-1}\sqrt{\frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\left|X_i^s - X_j^s\right|^2}}{\sqrt{2\pi}\max(TP_L)} \\
\hat{\sigma}_{X_t,X_t}^2 &= \frac{2^{L-1}\sqrt{\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\left|X_i^t - X_j^t\right|^2}}{\sqrt{2\pi}\max(TP_L)} \\
\hat{\sigma}_{X_s,X_t}^2 &= \frac{2^{L-1}\sqrt{\frac{1}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\left|X_i^s - X_j^t\right|^2}}{\sqrt{2\pi}\max(TP_L)}
\end{aligned} \tag{12}$$

3.3 Higher order

In some cases, the difference between the source and target distributions may be too large for the product of static and dynamic representations to effectively align the distributions. For instance, some studies have shown that using higher-order moments of the kernel can improve the alignment of distributions in some cases [18]. As the order of the Pascal Triangle increases, the corresponding Gaussian kernel becomes wider and considers more information. This means that higher-order kernels capture more information and have a greater chance of correctly estimating the variance within the range considered up to the chosen order P. Therefore, the choice of P affects the number of computations required during each iteration and, subsequently, the overall computational time. Eq. (13) presents the reformulated total representation considering higher-order moments:

$$\begin{aligned}
\hat{\sigma}_{X_s,X_s}^2 &= \left\{\sigma_{X_s,X_s}^2, 2^1\sigma_{X_s,X_s}^2, \ldots, 2^{P+1}\sigma_{X_s,X_s}^2\right\} \\
\hat{\sigma}_{X_t,X_t}^2 &= \left\{\sigma_{X_t,X_t}^2, 2^1\sigma_{X_t,X_t}^2, \ldots, 2^{P+1}\sigma_{X_t,X_t}^2\right\} \\
\hat{\sigma}_{X_s,X_t}^2 &= \left\{\sigma_{X_s,X_t}^2, 2^1\sigma_{X_s,X_t}^2, \ldots, 2^{P+1}\sigma_{X_s,X_t}^2\right\}
\end{aligned} \tag{13}$$

Such that $P \geq 0$.

Consequently, the final form of sigma will be a list of size $P+1$ such that $P+1$ MMD computations are needed, during each iteration, to fully adapt the target domain with respect to the source one. Based on experimental results, an order P of 5 is a reasonable choice as it strikes a balance between computational time and the probability of including the correct variance.

4 Tests and Discussions

4.1 Data description

Two datasets were utilized in this study: Case Western Reserve University (CWRU) bearings dataset and Jiangnan University (JNU) bearings dataset. The CWRU dataset served as the source dataset, while the JNU dataset was defined as the target one. The operating conditions varied in each dataset, with load variation in the CWRU dataset and speed variation in the JNU one. The sampling frequencies were approximately 12 kHz for the CWRU and 50 kHz for the JNU. To gather a large amount of training data, sliding windows of length 1568 (without overlap) were applied to the CWRU dataset, and windows of length 14884 were used for the JNU dataset. This windowing approach ensured that each window contained sufficient information to detect bearing faults. The signals were then normalized between 0 and 1 to avoid overfitting due to scaling factors. Both datasets underwent band pass filtering, with a passband between Fs/4 and 3Fs/8, where Fs represents the respective sampling frequency. The Hilbert transform was applied to obtain the envelope of each signal, which was used to compute the spectrum. Since the spectrum was even due to the real nature of the signal, only the right half of the spectrum (frequency range) was considered, resulting in sample sizes of 784 for the CWRU dataset and 7,442 for the JNU dataset. Table 1 provides a description of each subset of the datasets and their corresponding operating conditions. For model training, both datasets were resampled into a 2D format: the CWRU dataset as 28 x 28 images and the JNU dataset as 86 x 86 images. As the sizes of the two datasets were different, interpolation was performed to obtain a consistent input size for the model. Since the CWRU dataset served as the source dataset, the interpolation involved replacing every n samples in the JNU dataset (where n is the ratio of the JNU dataset size to the CWRU dataset size) with their mean. This down sampling of the JNU dataset ensured that major information was not excluded. The objective of the classification task was to achieve a 4-way classification, distinguishing between Normal, Inner Race Fault (IRF), Ball Fault (BF), and Outer Race Fault (ORF) under different working conditions. Several tasks were addressed to adapt a source domain to a target one as follows : source → target.

Datasets		Health Conditions	Number of Samples	Operating Conditions
CWRU	A	*N/IRF/BF/ORF*	4×150	0 HP (1797 rpm)
	B	*N/IRF/BF/ORF*	4×150	1 HP (1797 rpm)
	C	*N/IRF/BF/ORF*	4×150	2 HP (1797 rpm)
	D	*N/IRF/BF/ORF*	4×150	3 HP (1797 rpm)
JNU	E	*N/IRF/BF/ORF*	$N\times100, IRF\times33, BF\times33, ORF\times33$	600 rpm
	F	*N/IRF/BF/ORF*	$N\times100, IRF\times33, BF\times33, ORF\times33$	800 rpm
	G	*N/IRF/BF/ORF*	$N\times100, IRF\times33, BF\times33, ORF\times33$	1000 rpm

Table 1: Datasets Description.

4.2 Model and objective function

The proposed model's objective function consists of two parts:

- The classification error on the labelled source dataset $J\left(\theta(x_i^s), y_i^s\right)$.
- Multilayer adaptation with multi-core Maximum Mean Discrepancy (MMD) between source and target data d_K.

The model is optimized by minimizing Eq. (14), which combines the cross-entropy loss for classification and the MMD loss:

$$\min_{\theta}\left(\sum_{i=1}^{n} J\left(\theta(x_i^s), y_i^s\right) + d_K\right) \tag{14}$$

where θ represents the set of all convolutional neural network (CNN) parameters, n is the number of samples in the mini-batch, and $J\left(\theta(x_i^s), y_i^s\right) = -\left(y_i^s\right)^T \log\left(\theta(x_i^s)\right)$ is the training loss function of classification. The proposed model was optimized by the Adaptive moment (Adam) algorithm. The paper focuses on demonstrating the proposed method for automating domain adaptation rather than finding the optimal model for each specific dataset. The model topology is described in Table 2 and illustrated in Figure 2. The model includes convolutional neural network (CNN) layers, pooling layers, and fully connected layers.

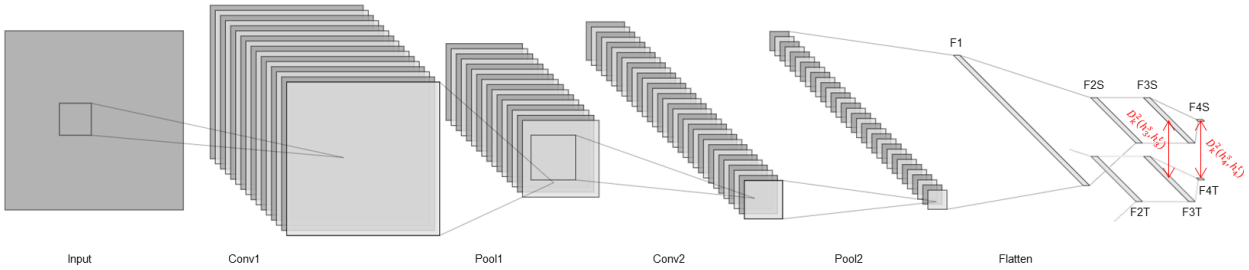


Figure 2: The proposed framework [11].

Layers	Parameters	Activation Functions	Output Size
Input	/	/	28×28
Conv1	Kernels: 5×5×16, bias: 16×1	ReLU	24×24×16
Pool1	Stride: 2	/	12×12×16
Conv2	Kernels: 7×7×32, bias: 32×1	ReLU	6×6×32
Pool2	Stride: 2	/	3×3×32
F1/Flatten	/	/	288×1
F2	Weights: 288×100, bias: 100×1	ReLU	100×1
F3	Weights: 100×100, bias: 100×1	ReLU	100×1
F4/Output	Weights: 100×4, bias: 4×1	Softmax	4×1

Table 2: Topology of the model.

The distribution of learned characteristics in the model's layers gradually transitions from a general approach to a more specific one. The transfer capacity decreases in the upper layers as the domain divergence increases. To address this, the hidden representations of the F_3 and F_4 layers are integrated into the Hilbert space of the Reproducing Kernel Hilbert Space (RKHS), where the mean embedding of different domain distributions can be explicitly aligned. Multiple Gaussian kernels with different bandwidths are used to match the lower-order and higher-order moments of the learned characteristics and reduce the discrepancy between domains [9]. A linear combination of Gaussian kernels is used, where h_3^s, h_3^t, h_4^s, and h_4^t are the outputs of layers F_3 and F_4 for the source and target domains, respectively. The multi-kernel-based MMD multilayer is calculated by:

$$d_K = D_K^2\left(h_3^s, h_3^t\right) + D_K^2\left(h_4^s, h_4^t\right) \tag{15}$$

4.3 Results

Table 3 presents the different transfer methods used in the transfer learning process. The experiments employed an 80% training and 20% validation split for the source dataset, while the target dataset was used for testing. The methods included CNN (no transfer), CORAL, Wasserstein, conventional MMD, and the proposed method. For conventional MMD, optimal variances were determined through Grid Search, resulting in values of 10, 35, and 14 for the CWRU, JNU, and CWRU → JNU subsets, respectively. In the proposed method, an order of P = 5 was chosen to adapt the variance automatically during transfer learning.
To evaluate the performance and stability of the methods, 1000 iterations were conducted, and each method was executed around 50 times. The results, presented in Table 4, consistently demonstrate the superiority of the proposed method over other non-parametric approaches, with a slight improvement compared to

conventional MMD. The reported results in Table 4 provide the average accuracy in percentage, along with the variance that accounts for the observed variation resulting from the random seed used in each iteration. For visualizing the transfer performance, t-distributed stochastic neighbour embedding (t-SNE) was utilized to reduce dimensionality nonlinearly. Figure 5b showcases how the proposed method effectively regroups both domains with respect to each class, in contrast to Figure 5a that represents the results using CNN. This visualization demonstrates the capability of the proposed method to enable the classifier to operate effectively on both distributions. Refer to Table 5 for the legend explaining the distribution visualization.

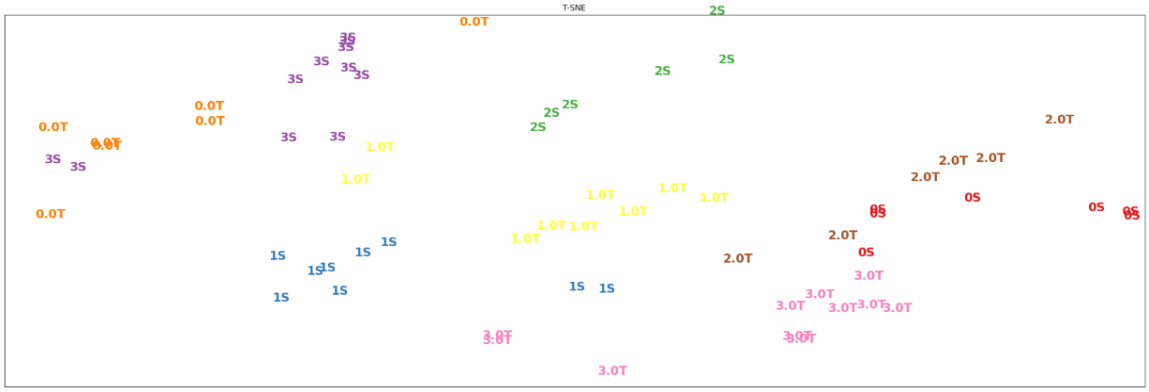
Method	Transfer Approach
CNN	No Transfer
CORAL	Non-parametric Transfer
Wasserstein	Non-parametric Transfer
Conventional MMD	Transfer + Variance Tuning
Proposed Method	Non-parametric Transfer

Table 3: Various Transfer Methods.

Task \ Method	CNN	CORAL	Wasserstein	MMD	Proposed Method
A → C	82.17 ± 2.97%	85.24 ± 1.58%	88.81 ± 1.14%	97.98 ± 0.98%	**98.33 ± 0.74%**
B → D	64.92 ± 1.26%	79.34 ± 1.35%	86.93 ± 1.02%	96.11 ± 0.81%	**97.75 ± 1.61%**
C → B	97.33 ± 3.75%	97.40 ± 2.07%	97.36 ± 0.66%	97.32 ± 0.45%	**98.45 ± 0.39%**
D → A	58.83 ± 2.10%	86.47 ± 1.52%	84.72 ± 0.85%	95.88 ± 0.72%	**97.30 ± 1.06%**
E → F	92.48 ± 1.48%	92.71 ± 1.39%	93.02 ± 1.37%	94.73 ± 0.99%	**95.46 ± 1.33%**
F → G	80.93 ± 3.39%	82.11 ± 2.78%	85.19 ± 2.64%	**88.53 ± 0.88%**	86.42 ± 2.21%
G → E	82.41 ± 2.68%	84.78 ± 1.51%	86.62 ± 1.09%	91.44 ± 0.53%	**92.96 ± 0.48%**
CWRU → JNU	54.62 ± 1.05%	78.74 ± 1.12%	79.91 ± 0.73%	87.55 ± 0.55%	**88.93 ± 0.67%**

Table 4: Performance Metrics For Transfer Tasks.

(a)



(b)

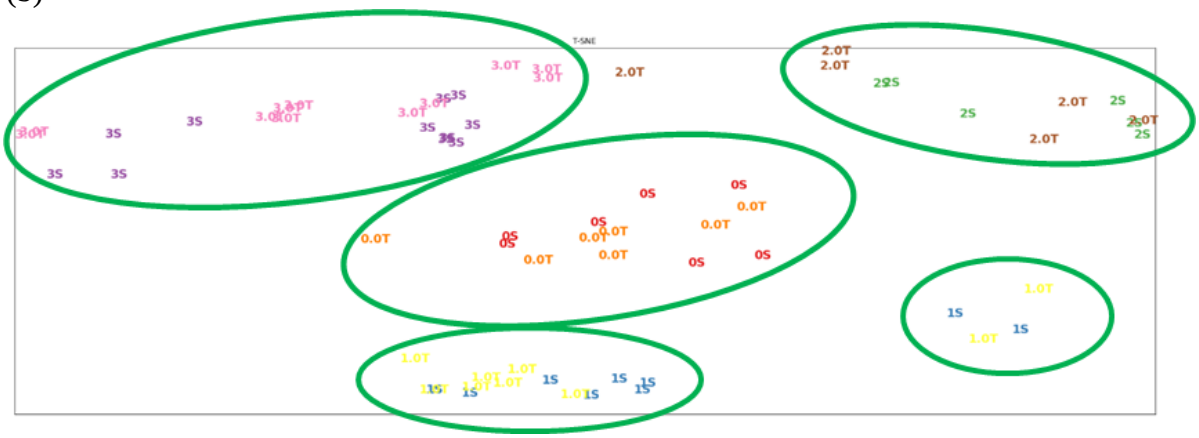


Figure 3: The two-dimensional visualization of learned features exposed by t-SNE embedding of transfer task CWRU → JNU using (a) CNN (b) Proposed method.

Label	Element
0	Normal/Healthy Case
1	Inner Race Fault
2	Ball Fault
3	Outer Race Fault
S	Source Domain
T	Target Domain

Table 5: Distribution Visualization Legend.

5 Conclusion

In this study, a novel approach for automating knowledge transfer in fault classification of rotating machinery was proposed. The method utilized a convolutional neural network (CNN) with multi-kernel-based maximum mean discrepancy (MMD) multilayer adaptation. The results demonstrated that the proposed method outperformed other existing transfer learning techniques and achieved high accuracy in the diagnostic tasks. By leveraging the knowledge learned from the source dataset, the proposed method effectively transferred that knowledge to the target dataset, improving the fault diagnosis performance. The self-adaptive nature of the method, utilizing multi-kernel-based MMD, allowed for better alignment of the distributions between the source and target domains. The findings of this study highlight the potential of automated knowledge transfer in enhancing fault diagnosis in rotating machinery and reducing the need for extensive labelled data in the target dataset.

Acknowledgments

F. Karkafi gratefully acknowledges the European Commission for its support of the Marie Sklodowska Curie program through the ETN MOIRA project (GA 955681).

Appendix

The Pascal's Triangle is a mathematical construct used in various fields, including probability theory and combinatory. A Pascal's Triangle of order L (P_L) is represented as follows:

1					$\times\frac{1}{2^0}$
1	1				$\times\frac{1}{2^1}$
1	2	1			$\times\frac{1}{2^2}$
1	3	3	1		$\times\frac{1}{2^3}$
1	⋯	⋯	⋯	1	$\times\frac{1}{2^{L-1}}$

The 2D Gaussian kernel can be reconstructed upon the L^{th} row of interest after normalizing it as seen in the matrix above where the sum of the L^{th} row is equal to 2^{L-1}. Taking an example of $L=5$ results in the final row as $\frac{1}{16}\times\begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix}$. Therefore, the Gaussian kernel $G(a,b)$ of width 5 can be constructed as follows:

$$G(a,b)=\frac{1}{16}\times\begin{bmatrix}0&0&1&0&0\\0&0&4&0&0\\0&0&6&0&0\\0&0&4&0&0\\0&0&1&0&0\end{bmatrix}\times\begin{bmatrix}1&4&6&4&1\\1&4&6&4&1\\1&4&6&4&1\\1&4&6&4&1\\1&4&6&4&1\end{bmatrix}\times\frac{1}{16}=\begin{bmatrix}1&4&6&4&1\\4&16&24&16&4\\16&24&36&24&16\\4&16&24&16&4\\1&4&6&4&1\end{bmatrix}\times\frac{1}{256}$$

Since $G(a,b)$ represents a Gaussian kernel, then it can be denoted as: $G(a,b)=\frac{1}{2\pi\sigma^2}e^{\frac{-(a^2+b^2)}{2\sigma^2}}$

Considering a special case where $G(a,b)$ is maximal, its value corresponds to the maximum value of

$e^{\frac{-(a^2+b^2)}{2\sigma^2}}$. Consequently, G is maximal when $a=b=0$. From the example of order 5, $G_{\max}={}^{36}\!/_{256}$ which is none other than ${}^{6^2}\!/_{16^2}$ respectively equal to ${}^{\max(P_L)^2}\!/_{(2^{L-1})^2}$.

As a conclusion, generally this means that $G(0,0)=\frac{1}{2\pi\sigma^2}=\frac{\max(P_L)^2}{(2^{N-1})^2}$. Then $\sigma=\frac{2^{L-1}}{\sqrt{2\pi}\max(P_L)}$

References

[1] R. B. Randall and J. Antoni, "*Rolling element bearing diagnostics—A tutorial*", 2011, Mech. Syst. Signal Process., vol. 25, no. 2, pp. 485–520.

[2] S. Braun, "*Mechanical signature analysis: theory and applications*", 1986.

[3] S. J. Pan, Q. Yang et al., "*A survey on transfer learning*", 2010, IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359.

[4] S. Shao, S. McAleer, R. Yan, and P. Baldi, "*Highly accurate machine fault diagnosis using deep transfer learning*", 2018, IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2446–2455.

[5] Z. Chen, K. Gryllias, and W. Li, "*Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network*", 2019, IEEE Transactions on Industrial Informatics.

[6] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "*Deep model based domain adaptation for fault diagnosis*", 2017, IEEE Transactions on Industrial Electronics, vol. 64, no. 3, pp. 2296–2305.

[7] L. Wen, L. Gao, and X. Li, "*A new deep transfer learning based on sparse auto-encoder for fault diagnosis*", 2019 January, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 1, pp. 136–144.

[8] 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*", 2018, IEEE Transactions on Industrial Electronics, pp. 1–1.

[9] B. Yang, Y. Lei, F. Jia, and S. Xing, "*An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings*", 2019, Mechanical Systems and Signal Processing, vol. 122, pp. 692–706.

[10] C. Che, H. Wang, X. Ni and Q. Fu, "*Domain adaptive deep belief network for rolling bearing fault diagnosis*", 2020 Computers & Industrial Engineering 143 (2020) 106427.

[11] W. Qian, S. Li and X. Jiang, "*Deep transfer network for rotating machine fault analysis*", 2019, Pattern Recognition 96 (2019) 106993.

[12] J. Zhu, N. Chen, Member, IEEE, and C. Shen, Member, IEEE, "*A New Deep Transfer Learning Method for Bearing Fault Diagnosis under Different Working Conditions*", 2019, DOI 10.1109/JSEN.2019.2936932, IEEE Sensors Journal.

[13] B. Sun and K. Saenko, "*Deep CORAL: Correlation Alignment for Deep Domain Adaptation*", 2016 July 6, arXiv:1607.01719v1.

[14] Z. Goldfeld, K. Kato, S. Nietert and G. Rioux, "*Limit Distribution Theory for p-Wasserstein Distances*", 2022 March 1, http://arxiv.org/pdf/2203.00159v1.

[15] B. Yang, Y. Lei, F. Jia and S. Xing, "*An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings*", 2018, https://doi.org/10.1016/j.ymssp.2018.12.051.

[16] J. Zhao, D. Meng, "*Fast MMD: Ensemble of Circular Discrepancy for Efficient Two-Sample Test*", 2015 June 18, arXiv:1405.2664v2.

[17] Farina, A., Giompapa, S., Graziano, A. et al. "*Tartaglia-Pascal's triangle: a historical perspective with applications*", 2013, SIViP 7, 173–188 (2013). https://doi.org/10.1007/s11760-011-0228-6.

[18] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "*Learning transferable features with deep adaptation networks*", 2015, arXiv preprint arXiv:1502.02791.