Knowledge-Based Systems 309 (2025) 112879

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Prospective domain adaptation for longitudinal data

Sunghong Park [a], JeongHeun Yeon [b], Dong-gi Lee [c], Sang Joon Son [d], Hyun Goo Woo [a,e,f,*], Hyunjung Shin [b,g,**]

[a] Department of Physiology, Ajou University School of Medicine, Suwon, 16499, Republic of Korea
[b] Department of Artificial Intelligence, Ajou University, Suwon, 16499, Republic of Korea
[c] Department of Biostatistics, Epidemiology & Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, United States
[d] Department of Psychiatry, Ajou University School of Medicine, Suwon, 16499, Republic of Korea
[e] Department of Biomedical Science, Graduate School of Ajou University, Suwon, 16499, Republic of Korea
[f] Ajou Translational Omics Center, Research Institute for Innovative Medicine, Ajou University Medical Center, Suwon, 16499, Republic of Korea
[g] Department of Industrial Engineering, Ajou University, Suwon, 16499, Republic of Korea

ARTICLE INFO

Keywords:
Domain adaptation
Longitudinal data
Feature graph
Manifold adaptation
Distribution adaptation

ABSTRACT

Longitudinal data includes the information of samples in various timepoints. When applying machine learning algorithms to this data, the use of up-to-date information will yield more accurate results. In this case, if the labels are derived from the up-to-date information, out-of-date samples for which data has not yet been collected are excluded from the prediction. To alleviate this problem, domain adaptation can be a method for the prediction of out-of-date samples in that the method transforms those features similar with the up-to-date samples and bridges to the use of labels. Especially, domain adversarial training methods with a gradient reversal layer derive feature representation where samples in different domains appear to be one set so as to make the origin of them indistinguishable. However, since existing methods focus on different data with heterogeneous features, by considering that homogeneous features are continuously collected in longitudinal data, those need to be improved for the out-of-date and up-to-date samples so that their features are exactly matched. Motivated by this, we propose a method of domain adaptation, namely prospective domain adaptation, where the transformation of the out-of-date features purposes to match the properties of the up-to-date features. Therefore, in the proposed method, the out-of-date features are adapted to the manifold and distribution of the up-to-date features so that two feature sets are implicitly and explicitly matched. The experimental results demonstrated that the proposed method derives well-matched feature representation and outperforms comparative methods.

1. Introduction

Longitudinal data comprises information about samples at various time points [1]. Up-to-date information, reflecting the most recent status, would contribute more to our understanding of the characteristics of the sample than out-of-date information [2]. This is also the case in the application of machine learning (ML) algorithms to longitudinal data. Given that the features of the samples have been changing over time, an algorithm trained on them would yield more accurate predictions about the target task when the data contains the most recent information [3]. However, the application of ML to longitudinal data is hindered by the samples, including solely out-of-date information, given that the most recent data has yet to be gathered. If the samples are divided into two sets, one comprising older data and the other comprising newer data, the ML algorithms may encounter difficulties in training both sets simultaneously due to the temporal heterogeneity between the two sets.

To address this issue, it is essential to minimize the temporal heterogeneity between the out-of-date and up-to-date sets. This suggests that the strategy of transforming the older data to be similar to the newer data could be employed, thereby enabling ML algorithms to train the two sets together. The suitable methodology for implementing this strategy is domain adaptation [4], which entails the transformation of two different feature sets into a common feature space [5], with the objective of minimizing the discrepancy between them, thereby representing a single, integrated feature set [6]. Accordingly, the application of domain adaptation to longitudinal data enables representing the

* Correspondence author at: Department of Physiology, Ajou University School of Medicine, Worldcup-ro 164, Yeongtong-gu, Suwon, 16499, Republic of Korea.
** Co-correspondence author at: Department of Industrial Engineering, Ajou University, Worldcup-ro 206, Yeongtong-gu, Suwon, 16499, Republic of Korea.
E-mail addresses: hg@ajou.ac.kr (H.G. Woo), shin@ajou.ac.kr (H. Shin).

https://doi.org/10.1016/j.knosys.2024.112879
Received 20 September 2022; Received in revised form 28 November 2024; Accepted 12 December 2024
Available online 13 December 2024

feature of out-of-date samples that have not yet been collected similar to the features of up-to-date samples.

Domain adaptation minimizes the discrepancy between source and target domains, enabling a target domain deficient in information to benefit from a source domain sufficient in information within a common feature space [7,8]. In the case of longitudinal data, the up-to-date and out-of-date data correspond to the source and target data, respectively. Domain adaptation is typically implemented using a feature projection matrix or domain adversarial training [9]. Transfer Component Analysis (TCA) [6] is a representative method employing a feature projection matrix, which finds a set of projectors that minimize the maximum mean discrepancy (MMD) [10]. Domain-Adversarial Neural Network (DANN) [11] is a prevalent method utilizing domain adversarial training, which renders the origins of samples indistinguishable using a domain classifier trained by a gradient reversal layer (GRL) [12]. A number of subsequent methods have been developed that employ a feature projection matrix or domain adversarial training derived from TCA or DANN, respectively.

Despite these successes, the existing methods should be further improved in two respects. At first, the domain adaptation algorithm requires minimizing the information loss from the source domain data. Previous works have represented a new common feature space by transforming both source and target data. This approach carries a significant risk of compromising the information-rich source domain. Accordingly, domain adaptation needs to preserve the source data by only transforming the target data. Next, the domain adaptation algorithm requires considering a fundamental property of longitudinal data, the homogeneity of features between the out-of-date and up-to-date samples. Previous works have focused on representing an aligned common feature space of different data from heterogeneous domains. In contrast, samples in longitudinal data belong to the same domain and are homogeneous in their features, and thereby, if more recent information is collected on the out-of-date set, it will exhibit the same data property as the up-to-date set. Therefore, domain adaptation for longitudinal data needs to simultaneously aim for the minimization of temporal heterogeneity and the maximization of featural homogeneity between the out-of-date and up-to-date samples.

Motivated by the limitations above, we propose a novel domain adaptation method, *Prospective Domain Adaptation* (PDA), for longitudinal data. The proposed method is a one-way domain adaptation technique that transforms only out-of-date features to represent them similarly to up-to-date features, with the objective of matching the explicit and implicit properties of the transformed and up-to-date features. This is achieved through a combined process of feature projection matrix and domain adversarial training. As illustrated in Fig. 1(a), the proposed method entails transforming the temporal domain of the out-of-date features, followed by label prediction. PDA is constituted of three components: feature transformer, domain classifier, and label predictor. First, the feature transformer is a projection matrix applied to the out-of-date features with the purpose of minimizing the temporal domain discrepancy. As shown in Fig. 1(b), it also aims to maximize the homogeneity of the implicit and explicit features by matching the manifolds and distributions of the two feature sets. Second, the domain classifier reduces the overall discrepancy between the transformed and up-to-date features by employing the domain adversarial training. Third, the label predictor performs a target task on the out-of-date samples by training the up-to-date labels. As a result, the proposed method transforms the unlabeled, out-of-date samples to have similar implicit and explicit properties as the labeled, up-to-date samples, enabling predictions to be made about them.

The remainder of the paper is organized as follows. Section 2 introduces the related works on the proposed method with methodological comparisons. Section 3 presents a detailed description of the proposed method with mathematical implementations. Section 4 shows the experimental results on various datasets, including the ablation study. Section 5 concludes the paper with remarks on the proposed method.

2. Related works

In this section, we present an overview of the related works on the proposed method by categorizing the existing methods into two main approaches to domain adaptation: feature projection matrix-based approach and domain adversarial training-based approach, including the methodological comparison with the proposed method.

2.1. Feature projection matrix-based approach

This approach directly minimizes the discrepancy between the domains, as measured by the MMD, mapping the original data to a reproducing kernel Hilbert space, where the source and target distributions are assumed to be separable [6]. TCA is a representative method for implementing this approach, and thereby, it has been extended to

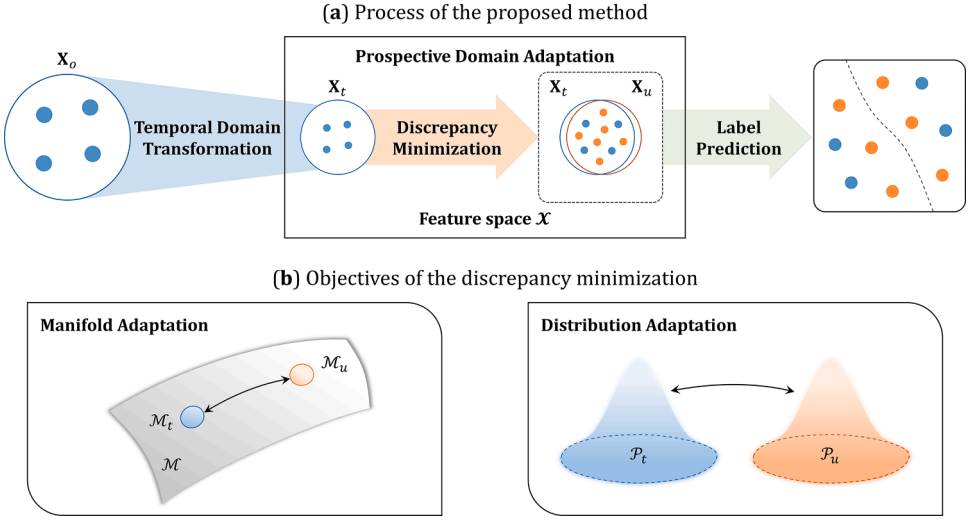

Fig. 1. Overview of the proposed method. The proposed method is a one-way domain adaptation technique that transforms only out-of-date features ($\mathbf{X}_o$) to represent them similarly to up-to-date features ($\mathbf{X}_u$). (a) PDA firstly transforms the temporal domain of the out-of-date feature, representing the transformed feature ($\mathbf{X}_t$). Subsequently, PDA minimizes the discrepancy between $\mathbf{X}_t$ and $\mathbf{X}_u$, followed by label prediction of $\mathbf{X}_t$. (b) For the discrepancy minimization, two objectives are implemented so that the manifolds and distributions of two feature sets are implicitly and explicitly matched.

numerous subsequent studies. A method for Joint Geometrical and Statistical Alignment (JGSA) was proposed in [13], which reduces the shift between domains both statistically and geometrically. As proposed in Manifold Embedded Distribution Alignment (MEDA) [14], the objective function for the projection matrix was improved to perform TCA-based manifold regularization by integrating unlabeled data with similarity constraints into the objective function for the feature projection matrix. Subsequently, the feature projection matrix was further advanced to reduce the domain discrepancy using label information by jointly matching the marginal and class-conditional distributions of both domains, such as Cross-Domain Structure Preserving Projection (CDSPP) [15] and Mutual Domain Adaptation (MDA) [16].

2.2. Domain adversarial training-based approach

This approach minimizes the domain discrepancy using a domain classifier, which is trained by a GRL, thereby rendering the origins of samples from different domains indistinguishable. DANN is a representative method for implementing this approach, where the domain discrepancy is measured by a binary cross-entropy (BCE) loss, and it has been extended to numerous subsequent studies. For example, a method for Adversarial Discriminative Domain Adaptation (ADDA) [17] was proposed, which minimizes the domain shift using a generative adversarial network (GAN)-based loss. This GAN-based framework was extended by Domain Invariant Feature Augmentation (DIFA) [18], which forces the learned feature extractor to be domain-invariant and trains it through data augmentation in the feature space. Subsequently, the more advanced techniques for the domain discriminator were proposed by Multi-Adversarial Domain Adaptation (MADA) [19] and Informative Discriminator-based Domain Adaptation (IDDA) [20], where the former focused on capturing multimode structures to enable fine-grained alignment of different data distributions based on multiple domain discriminators, while the latter presented an informed adversarial discriminator that guides the transformation of features to a more structure adapted space by utilizing all the information available including the class structure present in the source dataset.

2.3. Comparison of methodological characteristics

In the existing methods for domain adaptation based on feature projection matrix, the objective function for the projection matrix was designed to minimize the explicit domain discrepancy by default, including the MMD-based constraint for distribution matching, and it also aimed to minimize the implicit domain discrepancy via manifold matching with the graph Laplacian-based constraint. Although this approach is suitable for maximizing the featural homogeneity between the two domains, it introduces the problem that domain adaptation and label prediction are separately performed in the learning process, reducing the efficiency of the overall optimization. On the other hand, this problem is solved through the end-to-end learning process in the existing methods based on domain adversarial training, but the domain classifier only focuses on minimizing the distributional gap between the two domains. Furthermore, both approaches suffer from the problem that by transforming the source and target domains simultaneously, resulting in the loss of information in the information-rich source domain.

PDA, the method we propose in this study, is designed to alleviate the abovementioned problems of the previous approaches. Compared to existing methods, the key characteristics of the proposed method are summarized in Table 1. PDA combines the feature projection matrix and the domain adversarial training, matching the distributions and manifolds of two domains while simultaneously predicting labels for the target domain. In addition, PDA employs a one-way domain adaptation approach that only transforms the target domain while preserving the source domain, thus avoiding information loss in the source domain. Accordingly, the proposed method transforms the unlabeled, out-of-date samples to have similar implicit and explicit properties as the labeled, up-to-date samples, enabling predictions to be made about them.

3. Prospective domain adaptation

3.1. Overview of the proposed method

The proposed method aims to transform the out-of-date features to have similar implicit and explicit properties as the up-to-date features, consisting of three components: feature transformer, domain classifier, and label predictor. At first, the feature transformer is a projection matrix applied to the out-of-date features, intending to minimize the temporal heterogeneity and maximize the featural homogeneity between two domains. In order to achieve the featural homogeneity, this component transforms the out-of-date feature set in a manner that aligns its topological and statistical properties with those of the up-to-date feature set. Specifically, the feature transformer adapts the manifolds and distributions between two feature sets. Manifolds represent topological properties, which are implicit information about the data based on the structural relationships between samples, while distributions represent statistical properties, which are explicit information about the data based on the feature values of the samples. Accordingly, the feature transformer, comprising the manifold and distribution adaptation, enables the representation of the out-of-date feature set to be both implicitly and explicitly similar to the up-to-date feature set. Next, the domain classifier reduces the overall discrepancy between the transformed and up-to-date features by employing domain adversarial training. At last, the label predictor performs a target task on the out-of-date samples by training the up-to-date labels. The following subsections provide a detailed description of the components of PDA, including the optimization process and the complexity analysis.

Table 1
Comparison of methodological characteristics for domain adaptation methods.

Method	Domain adaptation approach		Domain discrepancy matching		Source domain preserving
	Feature projection	Adversarial training	Distribution	Manifold	
TCA [6]	✓		✓		
JGSA [13]	✓		✓		
MEDA [14]	✓		✓	✓	
CDSPP [15]	✓		✓	✓	
MDA [16]	✓		✓	✓	
DANN [11]		✓	✓		
ADDA [17]		✓	✓		
DIFA [18]		✓	✓		
MADA [19]		✓	✓		
IDDA [20]		✓	✓		
PDA (Ours)	✓	✓	✓	✓	✓

3.2. Feature transformer

Let the feature matrices of out-of-date and up-to-date samples are $\mathbf{X}_o \in \mathbb{R}^{n_o \times d}$ and $\mathbf{X}_u \in \mathbb{R}^{n_u \times d}$, respectively, where $n_{o,u}$ are the number of samples for each feature matrix and d is the feature dimensionality. At first, the feature transformer linearly transforms $\mathbf{X}_o$ using the projection matrix, denoted as $\mathbf{P} \in \mathbb{R}^{d \times d}$, which is defined as a parameter reflecting the pattern of longitudinal changes in features. Then, the features of $\mathbf{X}_o\mathbf{P} \in \mathbb{R}^{n_o \times d}$ are smoothed by the graph $\mathbf{G}$, which is derived from the correlation matrix of $\mathbf{X}_u$, denoted as $\mathbf{W} \in \mathbb{R}^{d \times d}$, and thereby the transformed feature matrix of the out-of-date samples, denoted as $\mathbf{X}_t \in \mathbb{R}^{n_o \times d}$, is defined as follows:

$$\mathbf{X}_t = \mathbf{X}_o\mathbf{P}\mathbf{G} \tag{1}$$

where $\mathbf{G}$ is defined as $\mathbf{G} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{d \times d}$ with the diagonal matrix $\mathbf{D} \in \mathbb{R}^{d \times d}$ ($\mathbf{D}_{ii} = \sum_j |\mathbf{W}_{ij}|$). The objective of this feature smoothing is to achieve similarity between the correlation matrices of $\mathbf{X}_o\mathbf{P}$ and $\mathbf{X}_u$, thereby ensuring that those manifolds are consistently aligned. It is well known that the nonsingular correlation matrix is the symmetric positive definite and lies on a Riemannian manifold space $\mathscr{M}$ [21–23]. Accordingly, the manifold aligning can be implemented by matching the correlation matrices of each feature set. Subsequently, the distributions of $\mathbf{X}_t$ and $\mathbf{X}_u$ are matched by minimizing Kullback-Leibler Divergence (KLD) [24] as follows:

$$\mathscr{L}_k(\mathbf{P}) = \mathrm{KLD}(\mathscr{P}_t \| \mathscr{P}_u) = \mathscr{P}_t \left\{ \log\left(\frac{\mathscr{P}_t}{\mathscr{P}_u}\right) \right\}^{\mathrm{T}} \tag{2}$$

where $\mathscr{P}_t$ and $\mathscr{P}_u$ are defined as $\mathscr{P}_t = \mathrm{softmax}(\overline{\mathbf{X}}_t) \in \mathbb{R}^{1 \times d}$ and $\mathscr{P}_u = \mathrm{softmax}(\overline{\mathbf{X}}_u) \in \mathbb{R}^{1 \times d}$, respectively.

3.3. Domain classifier

The domain classifier f_d discriminates the domain labels indicating the origin of samples in the transformed data $\mathbf{X}_t$ and the up-to-date data $\mathbf{X}_u$. The domain label set is denoted as $\mathbf{Y}_d \in \mathbb{R}^{(n_o+n_u)\times 1}$, comprising binary elements, where the domain labels for samples in $\mathbf{X}_t$ and $\mathbf{X}_u$ are defined as +1 and –1, respectively. The predicted domain label set $\widehat{\mathbf{Y}}_d$ is derived by using the logistic function as follows:

$$\widehat{\mathbf{Y}}_d = f_d(\mathbf{X}) = \frac{1}{1 + e^{-\mathbf{X}\boldsymbol{\Theta}_d}} \in \mathbb{R}^{(n_o+n_u)\times 1} \tag{3}$$

where $\mathbf{X}$ is denoted as $\mathbf{X} = \begin{bmatrix} \mathbf{X}_t \\ \mathbf{X}_u \end{bmatrix} \in \mathbb{R}^{(n_o+n_u)\times d}$, and $\boldsymbol{\Theta}_d \in \mathbb{R}^{d \times 1}$ is the parameter of f_d. The parameter $\boldsymbol{\Theta}_d$ is optimized by minimizing the BCE loss $\mathscr{L}_d$ between $\widehat{\mathbf{Y}}_d$ and $\mathbf{Y}_d$ as below.

$$\mathscr{L}_d(\mathbf{P}, \boldsymbol{\Theta}_d) = -\left\{ \mathbf{Y}_d^{\mathrm{T}} \log \widehat{\mathbf{Y}}_d + (\mathbf{1}_{(n_o+n_u)\times 1} - \mathbf{Y}_d)^{\mathrm{T}} \log(\mathbf{1}_{(n_o+n_u)\times 1} - \widehat{\mathbf{Y}}_d) \right\} \tag{4}$$

3.4. Label predictor

The label predictor f_l trains labels of samples in $\mathbf{X}_u$. By denoting the predicted label set as $\widehat{\mathbf{Y}}_l \in \mathbb{R}^{n_u \times c}$ where c is the number of classes, $\widehat{\mathbf{Y}}_l$ is derived by applying the softmax function as follows:

$$\widehat{\mathbf{Y}}_l = f_l(\mathbf{X}_u) = \mathrm{softmax}(\mathbf{X}_u\boldsymbol{\Theta}_l) \in \mathbb{R}^{n_u \times c} \tag{5}$$

where $\boldsymbol{\Theta}_l$ is the parameter of f_l. The parameter $\boldsymbol{\Theta}_l \in \mathbb{R}^{d \times c}$ is optimized by minimizing the cross-entropy loss $\mathscr{L}_l$ between the predicted label set $\widehat{\mathbf{Y}}_l$ and the true label set $\mathbf{Y}_l \in \mathbb{R}^{n_u \times c}$, as below.

$$\mathscr{L}_l(\boldsymbol{\Theta}_l) = -\mathrm{Tr}\left(\mathbf{Y}_l^{\mathrm{T}} \log \widehat{\mathbf{Y}}_l\right) \tag{6}$$

3.5. Optimization process

As illustrated in Fig. 2, the proposed method consists of three parameters: $\mathbf{P}$, $\boldsymbol{\Theta}_d$, and $\boldsymbol{\Theta}_l$, and the objective function for PDA is defined by combining Eq. (2), (4), and (6) as follows:

$$\underset{\mathbf{P},\boldsymbol{\Theta}_d,\boldsymbol{\Theta}_l}{\mathrm{argmin}}\ \gamma_k\mathscr{L}_k(\mathbf{P}) + \gamma_d\mathscr{L}_d(\mathbf{P},\boldsymbol{\Theta}_d) + \gamma_l\mathscr{L}_l(\boldsymbol{\Theta}_l) + \mathscr{L}_r(\mathbf{P},\boldsymbol{\Theta}_d,\boldsymbol{\Theta}_l) \tag{7}$$

where $\mathscr{L}_r$ denotes the L2-regularzation term for each parameter to penalize the complexity, and $\gamma_{k,d,l}$ are combining coefficients ($\gamma_* \geq 0$). The objective function in Eq. (7) is optimized by the gradient descent method [25,26]. First, the gradient with respect to the parameter $\boldsymbol{\Theta}_l$ of the label predictor is derived as follows.

$$\nabla\boldsymbol{\Theta}_l = \gamma_l\frac{\partial\mathscr{L}_l}{\partial\boldsymbol{\Theta}_l} + \frac{\partial\mathscr{L}_r}{\partial\boldsymbol{\Theta}_l} = \frac{\gamma_l}{n_u}\mathbf{X}_u^{\mathrm{T}}(\widehat{\mathbf{Y}}_l - \mathbf{Y}_l) + 2\boldsymbol{\Theta}_l \tag{8}$$

Second, the gradient with respect to the parameter $\boldsymbol{\Theta}_d$ of the domain classifier is derived as below.

$$\nabla\boldsymbol{\Theta}_d = \gamma_d\frac{\partial\mathscr{L}_d}{\partial\boldsymbol{\Theta}_d} + \frac{\partial\mathscr{L}_r}{\partial\boldsymbol{\Theta}_d} = \frac{\gamma_d}{n_o+n_u}\mathbf{X}^{\mathrm{T}}(\widehat{\mathbf{Y}}_d - \mathbf{Y}_d) + 2\boldsymbol{\Theta}_d \tag{9}$$

Third, the gradient with respect to the projection matrix $\mathbf{P}$ is derived as follows.

$$\nabla\mathbf{P} = \gamma_k\frac{\partial\mathscr{L}_k}{\partial\mathbf{P}} - \gamma_d\frac{\partial\mathscr{L}_d}{\partial\mathbf{P}} + \frac{\partial\mathscr{L}_r}{\partial\mathbf{P}} \tag{10}$$

By denoting $\overline{\mathbf{X}}_t$ as $\overline{\mathbf{X}}_t = \frac{1}{n_o}\mathbf{1}_{1\times n_o}\mathbf{X}_t$, the derivative of $\mathscr{L}_k$ *w.r.t.* $\mathbf{P}$ is presented as below.

$$\frac{\partial\mathscr{L}_k}{\partial\mathbf{P}} = \frac{1}{n_o}\mathbf{G}\mathbf{X}_o^{\mathrm{T}}\mathbf{1}_{1\times n_o}^{\mathrm{T}}\left\{\log\left(\frac{\mathscr{P}_t}{\mathscr{P}_u}\right) + \mathbf{1}_{1\times d}\right\}\left\{\mathrm{Diag}(\mathscr{P}_t) - \mathscr{P}_t^{\mathrm{T}}\mathscr{P}_t\right\} \tag{11}$$

Then, the derivative of $\mathscr{L}_d$ *w.r.t.* $\mathbf{P}$ is defined as follows.

$$\frac{\partial\mathscr{L}_d}{\partial\mathbf{X}} = \begin{bmatrix} \frac{\partial\mathscr{L}_d}{\partial\mathbf{X}_o} \\ \frac{\partial\mathscr{L}_d}{\partial\mathbf{X}_u} \end{bmatrix} = \frac{\gamma_d}{n_o+n_u}(\widehat{\mathbf{Y}}_d - \mathbf{Y}_d)\boldsymbol{\Theta}_d^{\mathrm{T}},\quad \frac{\partial\mathscr{L}_d}{\partial\mathbf{P}} = \left(\frac{\partial\mathscr{L}_d}{\partial\mathbf{X}_o}\right)^{\mathrm{T}}\mathbf{X}_o\mathbf{G}^{\mathrm{T}} \tag{12}$$

Finally, the gradient *w.r.t.* $\mathbf{P}$ is derived by combining Eq. (10), (11), and (12), as below.

$$\nabla\mathbf{P} = \frac{\gamma_k}{n_o}\mathbf{G}\mathbf{X}_o^{\mathrm{T}}\mathbf{1}_{1\times n_o}^{\mathrm{T}}\left\{\log\left(\frac{\mathscr{P}_t}{\mathscr{P}_u}\right) + \mathbf{1}_{1\times d}\right\}\left\{\mathrm{Diag}(\mathscr{P}_t) - \mathscr{P}_t^{\mathrm{T}}\mathscr{P}_t\right\} - \gamma_d\left(\frac{\partial\mathscr{L}_d}{\partial\mathbf{X}_o}\right)^{\mathrm{T}}\mathbf{X}_o\mathbf{G}^{\mathrm{T}} + 2\mathbf{P} \tag{13}$$

The overall procedure for PDA is summarized in Algorithm 1.

3.6. Computational complexity

We further analyze the computational complexity of Algorithm 1 with the $\mathscr{O}$ notation. First, in the forward propagation, the time complexity for the feature transformation, domain classification, and

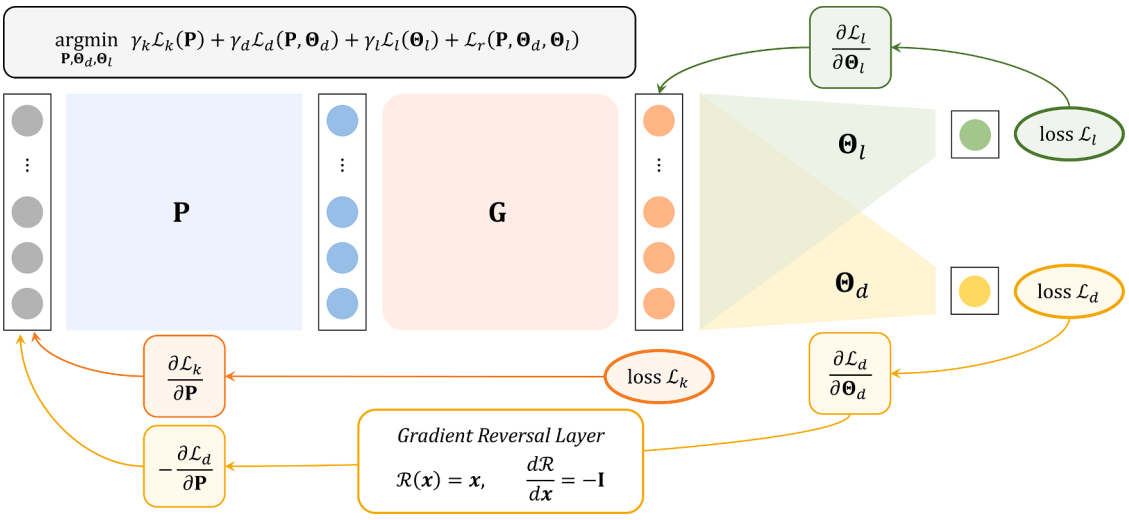

Fig. 2. Schematic description for training process of the proposed method.

Algorithm 1
Prospective Domain Adaptation.

Input:
Out-of-date feature $\{\mathbf{X}_o \in \mathbb{R}^{n_o \times d}\}$
Up-to-date feature and label $\{\mathbf{X}_u \in \mathbb{R}^{n_u \times d}, \mathbf{Y}_u \in \mathbb{R}^{n_u \times c}\}$
Output: Transformed feature $\mathbf{X}_t$ and Predicted label $\widehat{\mathbf{Y}}_o$ for $\mathbf{X}_o$
Initialize parameters: $\mathbf{P}$, Θ_d, and Θ_l
While (stopping criterion is not satisfied)
Forward propagation
Feature transformation: $\mathbf{X}_t$ by (1)
Domain classification: $\widehat{\mathbf{Y}}_d$ by (3)
Label prediction: $\widehat{\mathbf{Y}}_l$ by (5)
Loss functions
Kullback-Leibler divergence $\mathscr{L}_k(\mathbf{P})$ by (2)
Binary cross-entropy loss $\mathscr{L}_d(\mathbf{P},\Theta_d)$ by (4)
Cross-entropy loss $\mathscr{L}_l(\Theta_l)$ by (6)
Backward propagation
Label predictor $\Theta_l := \Theta_l - \eta\nabla\Theta_l$ by (8)
Domain classifier $\Theta_d := \Theta_d - \eta\nabla\Theta_d$ by (9)
Projection matrix $\mathbf{P} := \mathbf{P} - \eta\nabla\mathbf{P}$ by (13)
End while
Return Transformed feature $\mathbf{X}_t$ and Predicted label $\widehat{\mathbf{Y}}_o$ for $\mathbf{X}_o$

label prediction is $\mathscr{O}\left(n_o d^2\right)$, $\mathscr{O}((n_o + n_u)d)$, and $\mathscr{O}(n_u dc)$, respectively, and the space complexity for the feature transformation, domain classification, and label prediction is $\mathscr{O}\left(n_o d + d^2\right)$, $\mathscr{O}((n_o + n_u)d)$, and $\mathscr{O}(n_u d + dc + n_u c)$. If $n_u \gg d \gg c$, the space complexity for the label prediction is reduced to a simplified form, expressed as $\mathscr{O}(n_u d)$. Then, the overall time and space complexity for the forward propagation is $\mathscr{O}\left(n_o d^2 + n_o d + n_u d\right)$ and $\mathscr{O}\left(n_o d + n_u d + d^2\right)$, respectively. Next, in the backward propagation, the time complexity for updating the projection matrix, domain classifier, and label predictor is $\mathscr{O}\left(n_o d^2 + d^3\right)$, $\mathscr{O}((n_o + n_u)d)$, and $\mathscr{O}(n_u dc)$, respectively. The space complexity for updating the projection matrix, domain classifier, and label predictor is $\mathscr{O}\left(n_o d + d^2\right)$, $\mathscr{O}((n_o + n_u)d)$, and $\mathscr{O}(n_u d + n_u c + dc)$, respectively. If $n_u \gg d \gg c$, the space complexity for updating the label predictor is simplified to $\mathscr{O}(n_u d)$. Therefore, the overall time and space complexity for the backward propagation is $\mathscr{O}\left(n_o d^2 + d^3 + n_u d\right)$ and $\mathscr{O}\left(n_o d + n_u d + d^2\right)$, respectively.

4. Experimental results

4.1. Data description

The proposed method was applied to a total of eight benchmark datasets: one electrocardiogram (ECG) dataset (ECG200), three sensor datasets (Strain, Trace, and Wafer), and four synthetic datasets (BME, CBF, Pattern, and UMD). **BME** [27] is a synthetic dataset with three classes: one class is characterized by small positive bells occurring in the initial period (Begin), one without bells (Middle), and one with positive bells occurring in the last period (End). **CBF** [28] is a synthetic dataset

Table 2
Summary of benchmark datasets.

Dataset	Type	# Samples	# Features	# Classes
BME	Synthetic	180	128	3
CBF	Synthetic	930	128	3
ECG200	ECG	200	96	2
Pattern	Synthetic	5,000	128	4
Strain	Sensor	1,272	84	2
Trace	Sensor	200	274	4
UMD	Synthetic	180	150	3
Wafer	Sensor	7,164	152	2

including 930 signals of 128 lengths, which are classified into three shapes: cylinder, bell, and funnel. **ECG200** [29] is an ECG dataset, where each series represents electrical activity recorded during one heartbeat, which is classified as normal or myocardial infarction. **Pattern** [30] is a synthetic dataset containing 5,000 signals of 128 lengths, which are classified into four shapes: 'down-down', 'up-down', 'down-up', and 'up-up'. **Strain** [31] is a sensor dataset that involves distinguishing between two sensor types: humidity and temperature sensors. **Trace** [27] is a sensor dataset designed to simulate instrumentation failures in a nuclear power plant, with four different types of failures. **UMD** [27] is a synthetic dataset with three classes: one class is characterized by a small up bell arising at the initial or final period (Up), one does not have any bell (Middle), and one has a small down bell arising at the initial or final period (Down). **Wafer** [29] is a sensor dataset of semiconductor microelectronics fabrication, where each data contains measurements recorded by one sensor while processing one wafer with one tool, and the two classes are normal and abnormal. Table 2 provides a summary of the dataset, and more information about the dataset is available in the Time Series Classification repository (https://www.timeseriesclassification.com).

4.2. Experimental settings

For the experiment, samples of each dataset were divided in half and set as out-of-date and up-to-date sets. Also, we divided the length of time in half, the first half and the second half were set as the features of each set. The hyperparameters, γ_l, γ_d, and γ_k, in (4) were varied in the range of $\{0.01,\ 0.1,\ 1,\ 10,\ 100\}$ and determined to be the value that yielded the best results. The experimental results of the proposed methods were compared with a total of 10 existing methods: five feature projection matrix-based methods TCA [6], JGSA [13], MEDA [14], CDSPP [15], and MDA [16] and five domain adversarial training-based methods DANN [11], ADDA [17], DIFA [18], MADA [19], and IDDA [20]. The model architectures of comparison methods were constructed with reference to the best performance reported in each paper. The domain adversarial training-based methods, including the proposed method, were trained by using the ADAM optimizer [32] with a learning rate of 0.005. The entire experiment was repeated 100 times for each setting and the performance was measured by two metrics: the proxy A-distance (PAD) [33] for the domain adaptation and the area under receiving operating characteristic curve (AUC) for the label prediction.

4.3. Performance comparison

4.3.1. Results for domain adaptation

The results for comparing the domain adaptation performance of the proposed method with existing methods are shown in Fig. 3. At first, the baseline PAD results for the eight datasets used in the experiment were 0.8224 on average. The domain adaptation by the proposed method showed an average PAD performance of 0.2932, which is a 64.3% improvement over the baseline result and an average 28.8% better performance than the 10 comparison methods. Among the comparison methods, the feature projection matrix-based methods yielded an average PAD performance of 0.3965, which is 10.3% better than the average PAD performance of 0.4418 for the domain adversarial training-based methods. Furthermore, including the proposed method, the domain adaptation methods that match both the distribution and manifold of data from out-of-date and up-to-date samples showed better PAD results than the other methods. The detailed results for the performance comparison of domain adaptation are delineated in Table 3.

4.3.2. Results for label prediction

The comparison of label prediction performance between the proposed and existing methods is illustrated in Fig. 4. The baseline AUC results averaged 0.6495 across the eight datasets utilized in the experiment. The proposed method demonstrated an average AUC performance of 0.8454, reflecting a 30.2% improvement over the baseline and an enhancement of 7.2% compared to the ten other evaluated methods. Among the comparison methods, the domain adversarial training-based methods achieved an average AUC performance of 0.8164, surpassing the feature projection matrix-based methods, which had an average performance of 0.7635, by 6.9%. The comparative results showing the difference in AUC performance between the two approaches suggest that domain adaptation and label prediction performed end-to-end learning produce better results than when performed individually.

4.4. Ablation study

We further conducted the ablation experiment for the proposed method. For this experiment, we configured two ablated models $\Phi_{w/K}$ and $\Phi_{w/D}$ by ablating $\mathscr{L}_d$ and $\mathscr{L}_k$ in the objective function, respectively, which are terms for domain adaptation. In other words, $\Phi_{w/K}$ performs domain adaptation using the KLD without the domain classifier, while

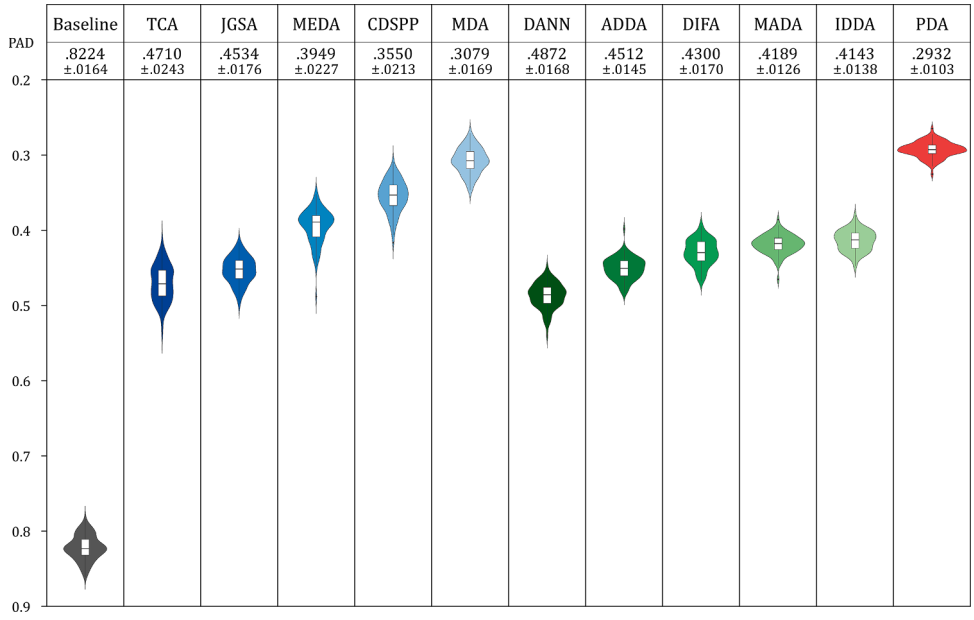

Fig. 3. Performance comparison of domain adaptation.

Table 3
Comparison results for performance of domain adaptation.

Method	Datasets								Overall average	*P*-value*
	BME	CBF	ECG200	Pattern	Strain	Trace	UMD	Wafer		
Baseline	.8338 ±.0530	.8943 ±.0178	.7199 ±.0063	.8000 ±.0303	.9048 ±.0158	.7850 ±.0152	.8375 ±.0203	.8039 ±.0050	.8224 ±.0164	2.49×10^{-257}
TCA	.4117 ±.0335	.4077 ±.0379	.4367 ±.0272	.2992 ±.0307	.2558 ±.0494	.5465 ±.0547	.7176 ±.0318	.6928 ±.0152	.4710 ±.0243	2.57×10^{-138}
JGSA	.4160 ±.0515	.4113 ±.0517	.4321 ±.0456	.3275 ±.0442	.3001 ±.0540	.4986 ±.0557	.6345 ±.0474	.6070 ±.0408	.4534 ±.0176	5.01×10^{-151}
MEDA	.3328 ±.0572	.3336 ±.0667	.3698 ±.0622	.2474 ±.0710	.2016 ±.0729	.4640 ±.0694	.6188 ±.0610	.5916 ±.0561	.3949 ±.0227	2.24×10^{-98}
CDSPP	.3039 ±.0604	.2935 ±.0573	.3227 ±.0626	.2029 ±.0556	.1645 ±.0734	.4165 ±.0733	.5776 ±.0635	.5586 ±.0608	.3550 ±.0213	3.79×10^{-66}
MDA	.2611 ±.0488	.2604 ±.0506	.2844 ±.0502	.1840 ±.0459	.1464 ±.0521	.3667 ±.0557	.4954 ±.0569	.4649 ±.0433	.3079 ±.0169	3.58×10^{-12}
DANN	.4442 ±.0495	.4468 ±.0507	.4595 ±.0440	.3614 ±.0420	.3388 ±.0561	.5367 ±.0520	.6675 ±.0479	.6430 ±.0509	.4872 ±.0168	3.29×10^{-170}
ADDA	.4163 ±.0377	.4214 ±.0464	.4339 ±.0458	.3467 ±.0362	.3175 ±.0438	.4933 ±.0477	.5990 ±.0432	.5816 ±.0365	.4512 ±.0145	2.12×10^{-161}
DIFA	.3931 ±.0451	.3854 ±.0427	.4078 ±.0433	.3196 ±.0467	.2858 ±.0520	.4805 ±.0549	.5899 ±.0469	.5774 ±.0452	.4300 ±.0170	3.94×10^{-140}
MADA	.3705 ±.0358	.3701 ±.0415	.3962 ±.0347	.2978 ±.0378	.2617 ±.0418	.4749 ±.0397	.5994 ±.0387	.5802 ±.0315	.4189 ±.0126	1.16×10^{-149}
IDDA	.3707 ±.0218	.3689 ±.0255	.3895 ±.0207	.2892 ±.0213	.2581 ±.0323	.4717 ±.0330	.5923 ±.0199	.5739 ±.0151	.4143 ±.0138	3.70×10^{-142}
PDA	.2585 ±.0254	.2536 ±.0183	. 2743 ±.0323	.1749 ±.0278	.1353 ±.0179	.3530 ±.0138	.4737 ±.0279	.4223 ±.0172	.2932 ±.0103	–

* *P*-values were calculated by comparing the overall average performance between the proposed and comparison methods.

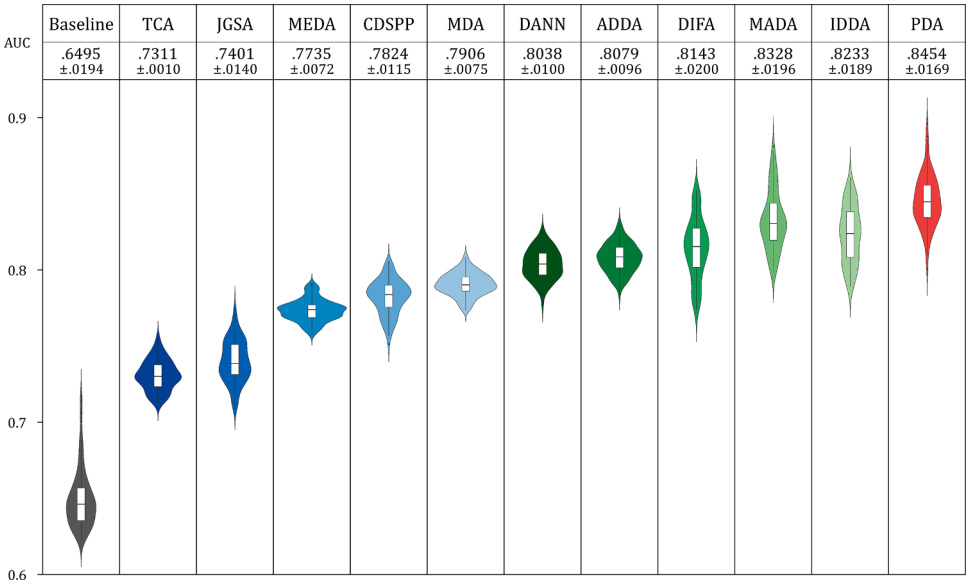

Fig. 4. Performance comparison of label prediction.

$\Phi_{w/D}$ includes the domain classifier for domain adaptation without the KLD, and the proposed method is denoted by Φ_{K+D} as it encompasses both components. The experimental settings for the two ablated models were applied in the same way as the proposed method, and the results are shown in Fig. 5. The proposed method Φ_{K+D} performed the best in both domain adaptation and label prediction, followed by $\Phi_{w/K}$. The performance of PAD and AUC of $\Phi_{w/K}$ were 0.3568 and 0.8053 respectively, which was 15.7% and 8.6% better than the performance of PAD of 0.4230 and AUC of 0.7416 of $\Phi_{w/D}$. It can thus be demonstrated that the optimization of the feature projection matrix through the use of KLD constituted a significant contribution to the superior performance of the proposed method in comparison to other methods.

5. Conclusion

In this paper, we proposed *prospective domain adaptation* for longitudinal data. The most pronounced feature of our method is to perform domain adaptation, which enables prediction even if there is no label by transforming the out-of-date data implicitly and explicitly similarly to the up-to-date data. This feature-transforming process includes adaptation to the manifolds and distributions of data. The transformed features are matched more precisely, so it is difficult to distinguish the origin domain of the samples by the domain discriminator and the gradient reversal layer. Then, the label predictor trains the labels in the up-to-date set and predicts the labels in the out-of-date set. The preceding experiments on the benchmark datasets validated the proposed method, and the results indicated that the proposed method performed better

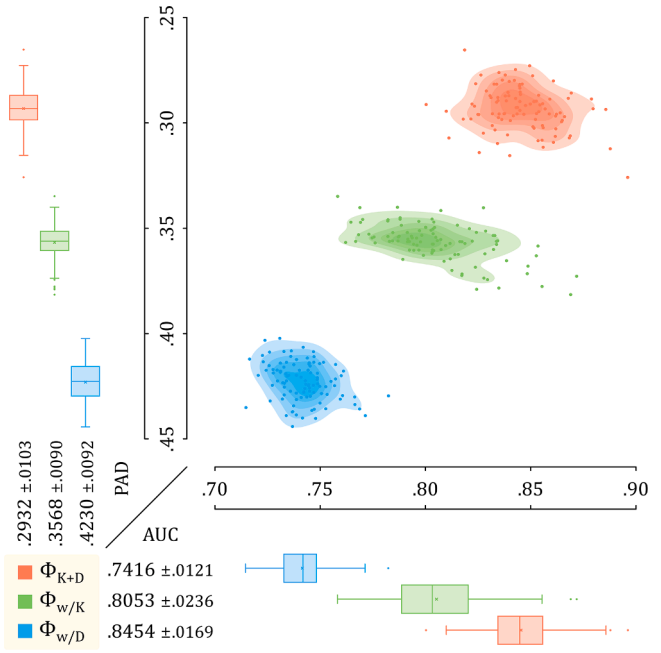

Fig. 5. Results for ablation study.

than the comparison method in both domain adaptation and label prediction.

CRediT authorship contribution statement

Sunghong Park: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **JeongHeun Yeon:** Formal analysis, Investigation, Methodology, Validation, Writing – review & editing. **Dong-gi Lee:** Conceptualization, Data curation, Validation. **Sang Joon Son:** Formal analysis, Resources, Writing – original draft. **Hyun Goo Woo:** Funding acquisition, Supervision, Writing – review & editing. **Hyunjung Shin:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE), Republic of Korea (2022R1A6A3A01086784) and Ajou University research fund. This work was also supported by the NRF grants funded by the Ministry of Science and ICT (MSIT), Republic of Korea (2019R1A5A2026045, 2021R1A2C2003474, and RS-2022-001653), the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the MSIT (RS-2023-00255968), the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI) funded by Ministry of Health and Welfare (MOHW), Republic of Korea (RS-2021-KH113821), and a grant of 'Korea Government Grant Program for Education and Research in Medical AI' through the KHIDI funded by the MOE and MOHW.

Data availability

The data that has been used is confidential.

References

[1] K.-Y. Liang, S.L. Zeger, Longitudinal data analysis using generalized linear models, Biometrika 73 (1986) 13–22.
[2] X. Yin, W. Tan, Semi-supervised truth discovery, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 217–226.
[3] H. Khabbaz, M.H. Karimi-Jafari, A.A. Saboury, B. BabaAli, Prediction of antimicrobial peptides toxicity based on their physico-chemical properties using machine learning techniques, BMC Bioinformatics 22 (2021) 1–11.
[4] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transac. Knowl. Data Eng 22 (2009) 1345–1359.
[5] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn 79 (2010) 151–175.
[6] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Transact. Neur. Netw 22 (2010) 199–210.
[7] S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J.E. Gonzalez, A. L. Sangiovanni-Vincentelli, S.A. Seshia, A review of single-source deep unsupervised visual domain adaptation, IEEE Transact. Neur. Netw. Learn. Syst 33 (2020) 473–493.
[8] S. Park, S.J. Son, K. Park, Y. Nam, H. Shin, L.S. of Ageing, A.s.D.N. Initiative, In-house data adaptation to public data: multisite MRI harmonization to predict Alzheimer's disease conversion, Exp. Syst. Applic 238 (2024) 122253.
[9] P. Oza, V.A. Sindagi, V.V. Sharmini, V.M. Patel, Unsupervised domain adaptation of object detectors: a survey, IEEE Transac. Pattern Anal. Mach. Intel (2023).
[10] A. Smola, A. Gretton, L. Song, B. Schölkopf, A Hilbert space embedding for distributions, in: International conference on algorithmic learning theory, Springer, 2007, pp. 13–31.
[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res 17 (2016) 1–35.
[12] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189.
[13] J. Zhang, W. Li, P. 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.
[14] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, P.S. Yu, Visual domain adaptation with manifold embedded distribution alignment, in: Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 402–410.
[15] Q. Wang, T.P. Breckon, Cross-domain structure preserving projection for heterogeneous domain adaptation, Pattern Recog 123 (2022) 108362.
[16] S. Park, M.J. Kim, K. Park, H. Shin, Mutual domain adaptation, Pattern Recog 145 (2024) 109919.
[17] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176.
[18] R. Volpi, P. Morerio, S. Savarese, V. Murino, Adversarial feature augmentation for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5495–5504.
[19] Z. Pei, Z. Cao, M. Long, J. Wang, Multi-adversarial domain adaptation, in: Proceedings of the AAAI conference on artificial intelligence, 2018.
[20] V.K. Kurmi, V.P. Namboodiri, Looking back at labels: a class based domain adaptation technique, in: 2019 international joint conference on neural networks (IJCNN), IEEE, 2019, pp. 1–8.
[21] M. Moakher, P.G. Batchelor, Symmetric positive-definite matrices: from geometry to applications and visualization. Visualization and Processing of Tensor Fields, Springer, 2006, pp. 285–298.
[22] Z. Lin, Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition, SIAM J. Matrix Anal. Applic 40 (2019) 1353–1370.
[23] M. Moakher, M. Zéraï, The Riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data, J. Math. Imag. Vis 40 (2011) 171–187.
[24] S. Kullback, R.A. Leibler, On information and sufficiency, Annals Math. Stat 22 (1951) 79–86.
[25] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747, (2016).
[26] M. Andrychowicz, M. Denil, S. Gomez, M.W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, N. De Freitas, Learning to learn by gradient descent by gradient descent, Adv. Neur. Infor. Process. Sys 29 (2016).
[27] H.A. Dau, A. Bagnall, K. Kamgar, C.-C.M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, The UCR time series archive, IEEE/CAA J. Automatica Sinica 6 (2019) 1293–1305.
[28] N. Saito, Local Feature Extraction and Its Applications Using a Library of Bases, Yale University, 1994.
[29] R.T. Olszewski, Generalized Feature Extraction For Structural Pattern Recognition in Time-Series Data, Carnegie Mellon University, 2001.
[30] P. Geurts, Contributions to decision tree induction: bias/variance tradeoff and time series classification, (2002).

[31] J. Sun, S. Papadimitriou, C. Faloutsos, Online latent variable detection in sensor networks, in: 21st International Conference on Data Engineering (ICDE'05), IEEE, 2005, pp. 1126–1127.

[32] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).

[33] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, Adv. Neur. Infor. Process. Sys 19 (2006).