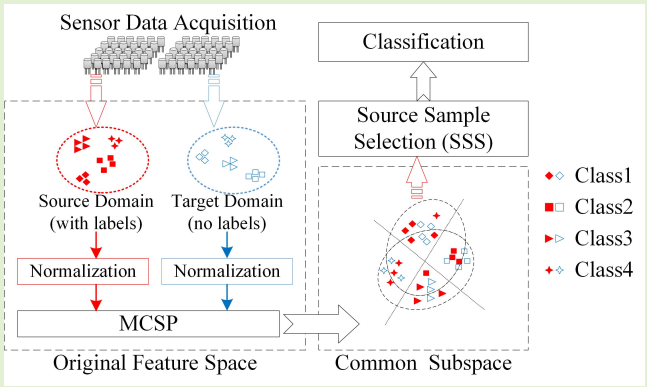
MCSP-SSS: A Domain Adaptive Framework for High-Accuracy Sensor Data Classification

Ran Liu, Xi Chen, Fengchun Tian, *Member, IEEE*, Junhui Qian, *Member, IEEE*, Feifei Wang, and Lin Yi

***Abstract*—Due to sensor drift, instrumental variation, or change of measurement object, the distributions of the datasets (domains) acquired by the sensors are often different. A domain adaptive framework called MCSP-SSS is proposed to reduce the distribution discrepancy across domains for high-accuracy sensor data classification. The framework consists of two key parts: Multi-Constraint Subspace Projection (MCSP) and Source Sample Selection (SSS). MCSP is a subspace-projection-based approach, which introduces four constraints to get an optimized projection matrix: Principal Component Analysis (PCA) is used to reduce the redundancy of features; Mean Distribution Discrepancy (MDD) is applied to minimize the difference between source and target data; Hilbert-Schmidt Independence Criterion (HSIC) is adopted to maximizing the dependence between features and labels; A so-called weighted-within-class scatter matrix is introduced to minimize the within-class variance to avoid samples with different labels to overlap in the subspace. SSS is designed to remove the outliers in projected source samples so as to reduce distribution discrepancy between the source and target samples projected by MCSP. Experimental results show that our framework can achieve the best accuracies in all classification tasks in comparison with other state-of-the-art approaches. The accuracy of MCSP-SSS is 3.57 percentage points (pps) higher on average than that of the second best approach for single-source domain adaptation, and 7.00 pps for multi-source domain adaptation. Source code is available at https://github.com/threedteam/tsc_subspace_projection.**



***Index Terms*— Domain adaption, subspace projection, sensor data classification, sensor drift, machine learning.**

I. Introduction

With the rapid development of sensor technology, sensors in different application fields have generated a large amount of sensor data [1]. Classifying these sensor data accurately is the key to harvest real benefits from sensor technology. At present, many machine learning algorithms have been used to classify the sensor data, such as *k*-Nearest Neighbor (*k*NN), Random Forest (RF), and Support Vector Machine (SVM) [2]–[4], etc. When applying these classifiers, the default hypothesis is that the training (source) and testing (target) data are extracted independently and identically from the same distribution [5]. However, the hypothesis is not necessarily true for the following reasons: (1) Sensor drift. It can be caused by sensor aging, poisoning, and environmental fluctuations [6]. It will cause the distributions of datasets acquired by sensors to be time-varying [7]. (2) Instrumental variation or change of measurement object [8], [9]. The distribution of datasets derived from different instruments or measurement objects are often different. As the difference in distribution across domain is not explicitly reduced, traditional classifiers may encounter difficulties in sensor data classification in these scenarios [10], [11]. Reducing the difference between cross-domain distributions is called domain adaption or domain correction [11].

Many approaches have been proposed to solve the problems above. For example, Ana *et al*. introduced a global calibration model for domain correction so as to reduce inherent sensor variability (i.e. instrumental variation) [12]. S. De Vito *et al*.

Manuscript received September 3, 2021; accepted October 5, 2021. Date of publication October 11, 2021; date of current version November 12, 2021. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2019CDYGZD004, in part by the Chongqing Foundation and Advanced Research Project under Grant cstc2019jcyj-msxmX0622, in part by the Sichuan Science and Technology Program under Grant 2019YFSY0026, and in part by the Open Fund Project of the Key Laboratory of Chongqing University Cancer Hospital. The associate editor coordinating the review of this article and approving it for publication was Prof. Pierluigi Salvo Rossi. *(Ran Liu and Xi Chen are co-first authors.) (Corresponding author: Lin Yi.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the IRB of the Mass Eye and Ear.

Ran Liu is with the College of Computer Science, Chongqing University, Chongqing 400044, China, and also with the Chongqing Key Laboratory of Bio-Perception & Intelligent Information Processing, Chongqing University, Chongqing 400044, China.

Xi Chen and Feifei Wang are with the College of Computer Science, Chongqing University, Chongqing 400044, China.

Fengchun Tian and Junhui Qian are with the Chongqing Key Laboratory of Bio-perception & Intelligent Information Processing, Chongqing University, Chongqing 400044, China.

Lin Yi is with the Chongqing University Cancer Hospital, Chongqing 400030, China (e-mail: linyi_cqu@163.com).

Digital Object Identifier 10.1109/JSEN.2021.3119320

proposed an adaptive learning strategy that using labeled update samples to compensate for drift [13]. These approaches usually rely on complex algorithms such as genetic algorithms or require repeating calibrations over a long period. Besides, the source labels are not fully utilized to reduce the distribution discrepancy across domains since they do not act directly on the domain distribution. Different from these approaches, the subspace-projection-based approaches [1], [7], [14]–[16] aim to directly reduce the distribution discrepancy between different domains (usually refers to source domain and target domain). The basic idea of the subspace projection is to find a projection matrix **P** to make the distribution of the source and target domains in the projected common subspace as similar as possible [7]. As the subspace-projection-based approach can simply but effectively represent the difference in distribution across domains, and can easily obtain the optimized projection matrix **P** by Eigen decomposition, it has been used for the classification of sensor data in the cases that the distributions of the source and target domains are different [7], [14], [16].

The key of subspace-projection-based approach is to find an optimized projection matrix **P** for domain adaption. Zhang *et al*. proposed an unsupervised subspace-projection-based approach DRCA (Domain Regularized Component Analysis) [7]. It optimizes the projection matrix **P** by minimizing the Mean Distribution Discrepancy (MDD) and maximizing the regularized variance of the source and target data. However, this approach does not take full use of the information of source labels, causing samples with different labels to overlap in the subspace [17]. To overcome this problem, Yi *et al*. proposed a method called D-DRCA (Discriminative Domain Regularization Component Analysis) [17]. The key idea of D-DRCA is to minimize the within-class variance of the projected source samples and maximize the between-class variance. This approach avoids overlapping samples with different labels in the subspace. Similar to D-DRCA, the MICF-IFLD (Maximum Independence of the Concentration Features-Iterative Fisher Linear Discriminant) has adopted the idea to reduce the difference within classes and increase the difference among classes, which can help to prevent inconsistent ratios of different types of samples among the domains [18]. Different from D-DRCA or MICF-IFLD, Liu *et al*. proposed a so-called MDMR (Maximizing label feature Dependency and Minimizing feature Redundancy) for projection matrix optimization [16]. Compared with D-DRCA or MICF-IFLD, MDMR takes the label-feature dependency into consideration when solving the projection matrix **P**. JDA (Joint Distribution Adaptation) method proposed by Jersson *et al*. simultaneously takes into account both marginal and conditional distributions among domains [1].

Although the methods above can do reduce the difference between cross-domain distributions, they still have disadvantages: 1) multiple constraints are not considered comprehensively when solving the projection matrix. 2) outliers (i.e. source samples far from the center of the target domain) will appear after projecting the source and target domains into a common subspace. These outliers may degrade the performance of the classifier if they are not removed.

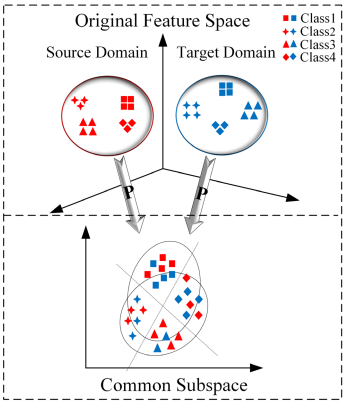


Fig. 1. Illustration of the basic idea of the proposed MCSP. In the original feature space, the distributions of the source and target domains are different. After projection, the distribution discrepancy is reduced and the overlapping samples are avoided as much as possible simultaneously.

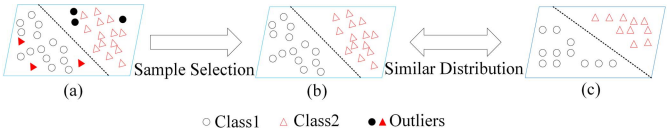


Fig. 2. Illustration of the basic idea of the proposed SSS. (a) Projected source samples. (b) Samples selected by SSS. (c) Projected target samples. The distribution discrepancy between (b) and (c) is reduced in contrast to that between (a) and (c).

A framework named MCSP-SSS is proposed in this paper. The framework consists of two key parts: Multi-Constraint Subspace Projection (MCSP) and Source Sample Selection (SSS). MCSP is proposed to get an optimized projection matrix **P**. This approach considers not only how to avoid the problem of overlapping samples but also the dependency between labels and features when solving the projection matrix. That is, multiple constraints are considered when solving the projection matrix. Fig.1 illustrates the basic idea of the proposed MCSP. SSS is proposed to select samples from the projected source domain with similar distribution of the projected target domain, and then use these samples to train a classifier. Fig.2 illustrates the basic idea of the proposed SSS. The main contributions of this paper are summarized in the following:

- A new projection matrix **P** optimization approach, MCSP, for domain adaption is proposed. This approach tries to avoid overlapping samples with different labels in the subspace as much as possible while maximizing the dependency between features and labels. Consequently, the approach improves the accuracy of the classifier.
- A so-called SSS is adopted to remove the outliers in projected source samples to reduce distribution discrepancy between the projected source and target samples as much as possible.

The remaining portions of this paper are organized as follows. In Sec. II, we describe the proposed domain adaption framework in detail. The experiments and results are presented in Sec. III. Sec. IV summarizes this paper.

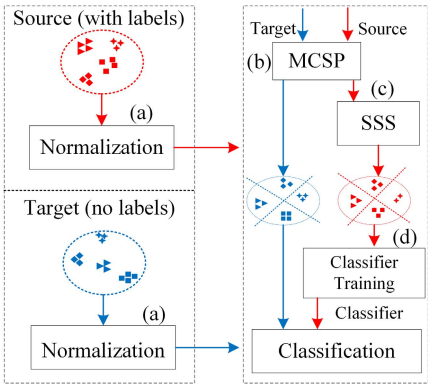


Fig. 3. Flowchart of the proposed framework. (a) L2 norm is used to normalize source and target data, respectively. (b) MCSP is used to get an optimized projection matrix. (c) SSS is used to select samples from the projected source domain hence remove the outliers. (d) Train a classifier using projected source samples and applied it for target data classification.

II. Proposed Framework

As mentioned above, we put forward a framework that can classify the target data whose distribution is different from the source data, where the source data has labels while the target data doesn't. As shown in Fig. 3, normalization is applied firstly to the source and target data respectively. Then, MCSP is used to find an optimized projection matrix $\mathbf{P}$ to project the source and target data into a low-dimensional common subspace so that the distributions of these two projected datasets are close to each other. Next, SSS is utilized to select samples from the projected source domain with similar distribution of the projected target domain to remove outliers. We use the samples selected by SSS for classifier training. Finally, classification is performed with the trained classifier. The details of each step are addressed in the following sections. For the convenience of description, we give some formal definitions below.

Let $\mathbf{X}_s = [\mathbf{x}_s^1, \mathbf{x}_s^2, \ldots, \mathbf{x}_s^{n_s}] \in \mathcal{R}^{d\times n_s}$ and $\mathbf{X}_t = [\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^{n_t}] \in \mathcal{R}^{d\times n_t}$ be the normalized samples from source and target domains respectively, where d is the dimension of the features; n_s and n_t represents the number of samples in the two domains. Let $\alpha, \beta, \gamma, \delta$ represent the trade-off parameters. $\mathbf{Y} = [\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{n_s}]^T \in \mathcal{R}^{n_s\times C}$ represents the source labels, where C is the number of classes. Notation $\mathbf{P}$ represents the projection matrix that maps the source data and the target data from the original space to a common subspace. $Tr(\cdot)$ represents the trace of a matrix, and $||\cdot||$ represents the matrix norm.

A. Normalization

Normalization can make the different samples "seen" by the classifier more similar to each other. In our framework, the L2 norm is applied to normalize both the source and target data. The main advantage of the L2 norm is that the Euclidean distance of a set of vectors and their cosine similarity can be equivalent after the data is normalized [19]. The normalization of a sample $\mathbf{V} = (v_1, v_2, \ldots, v_n)$ by L2 norm can be expressed mathematically by the following equation:

$$\mathbf{V}' = \left(\frac{v_1}{||\mathbf{V}||_2}, \frac{v_2}{||\mathbf{V}||_2}, \ldots, \frac{v_n}{||\mathbf{V}||_2}\right) \tag{1}$$

B. Multi-Constraint Subspace Projection

Multiple constraints are considered in our MCSP when solving the projection matrix. These constraints are the source and target variance maximization, MDD minimization, label-feature dependency maximization, and within-class variance minimization. Each of them will be addressed below.

1) Source and Target Variance Maximization: In MCSP, PCA (Principal Component Analysis) is adopted to maximize the variance of the data in source and target domains respectively so as to reduce redundant information in the source and target data [20]. The variance maximization can be achieved as follows:

$$\begin{aligned} &\arg\max_{\mathbf{P}} : Tr(\mathbf{P}^T\mathbf{S}_s\mathbf{P}) + \alpha Tr(\mathbf{P}^T\mathbf{S}_t\mathbf{P}) \\ &s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{2}$$

where

$$\mathbf{S}_s = \mathbf{X}_s\mathbf{H}_s\mathbf{X}_s^T, \mathbf{S}_t = \mathbf{X}_t\mathbf{H}_t\mathbf{X}_t^T \tag{3}$$

represents the scatter matrix of the source and target domain, respectively. $\mathbf{H}_s = \mathbf{I}_s - \frac{1}{n_s}\mathbf{1}_s$ and $\mathbf{H}_t = \mathbf{I}_t - \frac{1}{n_t}\mathbf{1}_t$ are centering matrices. $\mathbf{1}_s$ and $\mathbf{1}_t$ are the matrices with all ones. α is the trade-off parameter to avoid bias learning.

2) MDD Minimization: We try to reduce the difference between the source and target data by minimizing the MDD, so that there would be a similar distribution in the subspace after projection [7]. The expression is as follows:

$$\begin{aligned} \arg\min_{\mathbf{P}} :\ & ||\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{P}^T\mathbf{x}_s^i - \frac{1}{n_t}\sum_{j=1}^{n_t}\mathbf{P}^T\mathbf{x}_t^j||_2^2 \\ &= ||\mathbf{P}^T\boldsymbol{\mu}_s - \mathbf{P}^T\boldsymbol{\mu}_t||_2^2 \\ s.t.\ & \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{4}$$

where $\boldsymbol{\mu}_s = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{x}_s^i$ and $\boldsymbol{\mu}_t = \frac{1}{n_t}\sum_{j=1}^{n_t}\mathbf{x}_t^j$ represent the initial centers of the source domain and the target domain, respectively. Eq. (4) can be transformed into

$$\arg\min_{\mathbf{P}} : Tr(\mathbf{P}^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)^T\mathbf{P}) \tag{5}$$

3) Label-Feature Dependency Maximization: Commonly, the Hilbert-Schmidt Independence Criterion (HSIC) is adopted to describe the dependency between features and labels [21]. HSIC is a non-parametric dependency metric that takes into account all model dependencies between all variables. Using the linear kernel between feature and label space, we can get the empirical estimation of HSIC as follows:

$$HSIC = (l-1)^{-2}Tr(\mathbf{H}\mathbf{X}^T\mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{Y}^T) \tag{6}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the source data and label, respectively; $\mathbf{H}$ is the center matrix. l is the number of samples.

To facilitate the derivation, we use the least square method to construct the HSIC [22]:

$$\begin{aligned} &\min\ L(\mathbf{B},\mathbf{P}) = \min\left(||\mathbf{Y}^T\mathbf{H}\mathbf{X}^T - \mathbf{B}\mathbf{P}^T||_2^2\right) \\ &s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{7}$$

where $\mathbf{Y}^T\mathbf{H}\mathbf{X}^T$ and $\mathbf{B}$ represent the dependence between source features and labels, and coefficient matrix, respectively.

Let the partial derivation of $L(\mathbf{B}, \mathbf{P})$ with respect to $\mathbf{B}$ equal to 0:

$$\frac{\partial L(\mathbf{B}, \mathbf{P})}{\partial \mathbf{B}} = 2(\mathbf{B} - \mathbf{Y}^T\mathbf{H}\mathbf{X}^T\mathbf{P}) = 0 \Rightarrow \mathbf{B} = \mathbf{Y}^T\mathbf{H}\mathbf{X}^T\mathbf{P} \quad (8)$$

Putting (8) into (7) then we have

$$L(\mathbf{P}) = Tr(\mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}\mathbf{X}^T) - Tr(\mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}\mathbf{X}^T\mathbf{P}) \quad (9)$$

Minimizing (9) is equal to maximizing the last term in Eq. (9), since the first term is a constant term. Therefore, the finial optimization objective is

$$\begin{aligned} &\arg\max_{\mathbf{P}} : Tr(\mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}\mathbf{X}^T\mathbf{P}) \\ &s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \quad (10)$$

For the convenience of representation, we define a new matrix $\mathbf{S}_z = \mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}\mathbf{X}^T$, and use the one-hot encoding scheme for label encoding in the following sections.

4) Within-Class Variance Minimization: As the source labels are available, we can exploit the label information to construct the within-class distance, so that samples with same labels in the source domain are closer, and samples with different labels are as far away as possible to avoid overlapping after projection. This constraint can be expressed as:

$$\arg\min_{\mathbf{P}} : Tr(\mathbf{P}^T\mathbf{S}_w\mathbf{P}) \quad (11)$$

where $\mathbf{S}_w$ is the weighted-within-class scatter matrix of the source data which is used to reduce the impact of sample imbalance. $\mathbf{S}_w$ is defined by

$$\mathbf{S}_w = \sum_{c=1}^{C} \sum_{\mathbf{x}_i \in \mathbf{X}_s^{(c)}} w_c(\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T \quad (12)$$

where $w_c = 1 - \frac{n_c}{n_s}$ is the weight of samples labeled class c, $\mathbf{X}_s^{(c)}$ is the set of samples with label c and $|\mathbf{X}_s^{(c)}| = n_c$.

5) Optimization: In order to obtain an optimized projection matrix, constraints expressed by Eqs. (2), (5), (10), (11) are introduced into our proposed MCSP. As these constraints are introduced, our approach can make full use of the label information in the source domain, reduce redundant information in both source and target domains, and reduce the distribution discrepancy between the source and target data. Consequently, solving the optimized projection matrix is equivalent to the optimization problem below:

$$\begin{aligned} \arg\max_{\mathbf{P}} : L(\mathbf{P}) &= Tr(\mathbf{P}^T\mathbf{S}_s\mathbf{P}) + \alpha Tr(\mathbf{P}^T\mathbf{S}_t\mathbf{P}) \\ &\quad - \beta Tr(\mathbf{P}^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)\mathbf{P}) \\ &\quad + \gamma\, Tr(\mathbf{P}^T\mathbf{S}_z\mathbf{P}) - \delta Tr(\mathbf{P}^T\mathbf{S}_w\mathbf{P}) \\ s.t.\ \mathbf{P}^T\mathbf{P} &= \mathbf{I} \end{aligned} \quad (13)$$

where $\alpha, \beta, \gamma, \delta$ are trade-off parameters. We utilize the Lagrange Multiplier method to solve the optimization problem in (13):

$$\begin{aligned} L(\mathbf{P}, \rho) = Tr(&\mathbf{P}^T(\mathbf{S}_s + \alpha\mathbf{S}_t + \gamma\,\mathbf{S}_z - \delta\mathbf{S}_w \\ &- \beta(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t))\mathbf{P}) \\ &- \rho(Tr(\mathbf{P}^T\mathbf{P} - \mathbf{I})) \end{aligned} \quad (14)$$

where ρ denotes the Lagrange multiplier coefficient.

Let the partial derivation of $L(\mathbf{P}, \rho)$ with respect to $\mathbf{P}$ equal to 0, then we have

$$\begin{aligned} \rho\mathbf{P} = (&\mathbf{S}_s + \alpha\mathbf{S}_t - \beta(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t) \\ &+ \gamma\,\mathbf{S}_z - \delta\mathbf{S}_w)\mathbf{P} \end{aligned} \quad (15)$$

For the convenience of representation, let

$$\begin{aligned} \mathbf{M} = (&\mathbf{S}_s + \alpha\mathbf{S}_t - \beta(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t)^T(\boldsymbol{\mu}_s - \boldsymbol{\mu}_t) \\ &+ \gamma\,\mathbf{S}_z - \delta\mathbf{S}_w) \end{aligned} \quad (16)$$

thus (15) can be rewritten as:

$$\mathbf{M}\mathbf{P} = \rho\mathbf{P} \quad (17)$$

The projection matrix $\mathbf{P}$ can be obtained by Eigen decomposition of $\mathbf{M}$. $\mathbf{P}$ consists of the eigenvectors with respect to d' (a hyperparameter) of the largest eigenvalues of $\mathbf{M}$.

For easy implementation, the algorithm of MCSP is summarized in Algorithm 1.

Algorithm 1 Multi-Constraint Subspace Projection (MCSP)

Input:

Source data $\mathbf{X}_s \in \mathcal{R}^{d \times n_s}$
Target data $\mathbf{X}_t \in \mathcal{R}^{d \times n_t}$
Source labels $\mathbf{Y} \in \mathcal{R}^{n_s \times C}$
Trade-off parameters α, β, γ, and δ

Output:

Projection matrix $\mathbf{P}$

1: Compute the centers of source and target domains respectively: $\boldsymbol{\mu}_s, \boldsymbol{\mu}_t$;
2: Compute $\mathbf{S}_s, \mathbf{S}_t, \mathbf{S}_z, \mathbf{S}_w$ according to (3), (10), and (12);
3: Compute the matrix $\mathbf{M}$ according to (17);
4: Eigen decomposition for $\mathbf{M}$;
5: Compute and return the projection matrix $\mathbf{P}$.

C. Source Sample Selection

The SSS approach can select enough samples from the projected source domain so that the distributions of the selected samples and the projected target samples are as similar as possible. In this approach, MDD is used to measure the distance between the class after sample selection in the source domain and the same class in the target domain after projection. As shorter distance means greater similarity, our SSS approach can mathematically be expressed as an optimization problem:

$$\min \sum_{c=1}^{C} ||\mathbf{P}^T \frac{1}{n_{sc}} \sum_{i=1}^{n_{sc}} I \cdot \mathbf{x}_{sc}^i - \mathbf{P}^T\boldsymbol{\mu}_t||_2^2 \quad (18)$$

where n_{sc} represents the number of samples in class c in the source domain, $\mathbf{x}_{sc}^i$ is a sample belonging to class c, and $I \in \{0, 1\}$ indicates whether $\mathbf{x}_{sc}^i$ is selected.

Fig. 4 can help us better understand the "similarity" defined in the SSS approach. Note that SSS(i) (i = 1, 2, 3, 4, 5) represents the distance between a source sample from class c and the Center of the Target Domain (CTD, i.e. $\mathbf{P}^T\boldsymbol{\mu}_t$) after projection:

$$\mathrm{SSS}(i) = ||\mathbf{P}^T\mathbf{x}_{sc}^i - \mathbf{P}^T\boldsymbol{\mu}_t||_2^2 \quad (19)$$

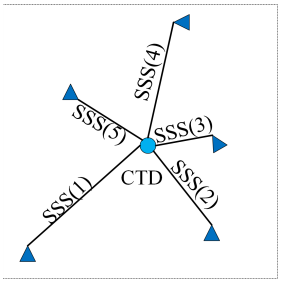


Fig. 4. Illustration of the similarity defined in the SSS approach. Similarity is inversely proportional to distance.

Suppose we need to select k samples from the projected source domain. Then, which samples should be selected is a critical issue. Usually, making the sample size of different classes balanced is beneficial to classification task. In order to balance the number of samples in different classes as much as possible, we first sort the samples in each class according to distance and then select the samples with the shortest distance from each class with (as much as possible) the same number. The number of samples selected from all classes equals k. The algorithm of SSS is summarized in Algorithm 2.

Algorithm 2 Source Sample Selection (SSS)

Input:
Projected source samples $\mathbf{P}^T\mathbf{X}_s$
Projected target samples $\mathbf{P}^T\mathbf{X}_t$
Number of samples selected k
Output:
Samples selected from the projected source domain $\mathbf{X}'_s$

Calculate the CTD
For $c := 1$ **to** C **do**
  For each sample $\mathbf{P}^T\mathbf{x}_s^i$ in class c calculate SSS(i) using (19);
  Sort the samples in class c in ascending order according to SSS(i);
  If $n_{sc} < k/C$ **then**
    All the samples from class c are selected into $\mathbf{X}'_s$;
  Else
    k/C (or close to k/C) samples with the shortest distance are selected into $\mathbf{X}'_s$.
End
Adjust properly the sample sizes of different classes so that they are as balanced as possible, and ensure a total of k samples are selected into $\mathbf{X}'_s$;
Return $\mathbf{X}'_s$.

D. Summary of the Framework

The complete procedure of MCSP-SSS is summarized in Algorithm 3. One important aspect of the framework is that SSS should be carried out after MCSP. This is because the outliers in the original source domain may not be outliers in the common subspace after being projected. If SSS is performed first, these samples will be removed by mistake.

We can learn from the algorithms of the framework that our framework can be applied to the cases where there is only one sample in the target data. In these cases, the only sample is just the center of the target domain. This means that our framework does not require a batch of target data to work properly; it can be applied to a single measurement. In other words, it does not need to know which classes are in the target data.

Algorithm 3 MCSP-SSS

Input:
Source data $\mathbf{X}_s$
Target data $\mathbf{X}_t$
Source labels $\mathbf{Y}$
Number of samples selected k
Output:
Predicted target labels $\mathbf{Y}_t$

Normalize the source and target data;
Adapt Algorithm 1 to calculate the projection matrix $\mathbf{P}$;
Adapt Algorithm 2 to obtain the source samples selected $\mathbf{X}'_s$;
Classification: train the classifier on $\mathbf{X}'_s$ and then apply it to $\mathbf{X}_t$ for classification;
Return the classification results $\mathbf{Y}_t$.

Note that our framework is easy to extend. Firstly, the L2 norm can be replaced by other normalization method according to the characteristics of the dataset. Secondly, the proposed framework can also be extended to semi-supervised learning, where the target domain contains a small number of labels. In this situation, the SSS approach calculates the CTD of each class only using the labeled samples belonging to this class in the target domain. Consequently, (18) is changed to

$$\min \sum_{c=1}^{C} ||\mathbf{P}^T \frac{1}{n_{sc}} \sum_{i=1}^{n_{sc}} I \cdot \mathbf{x}_{sc}^i - \mathbf{P}^T \frac{1}{n_{tc}} \sum_{j=1}^{n_{tc}} \mathbf{x}_{tc}^j||_2^2 \qquad (20)$$

where n_{tc} represents the number of samples in class c in the target domain, $\mathbf{x}_{tc}^j$ is a sample of class c. Accordingly, (19) is changed to

$$\text{SSS}(i) = ||\mathbf{P}^T\mathbf{x}_{sc}^i - \mathbf{P}^T \frac{1}{n_{tc}} \sum_{j=1}^{n_{tc}} \mathbf{x}_{tc}^j||_2^2 \qquad (21)$$

Note that if no labeled sample for a class is available in the target domain, then all unlabeled samples in the target domain are used to calculate the CTD for this class. In addition, we can choose classification methods such as DC-AELM [11], DAELM [23], TBLS [24], and TrLightGBM [25] for the semi-supervised learning situation.

We give a brief analysis of the computational complexity of the proposed approach. For MCSP, the time complexity of $\mathbf{S}_s$, $\mathbf{S}_t$, $\mathbf{S}_w$, $\mathbf{S}_z$, and $\mathbf{M}$ are $O(d^2n_s)$, $O(d^2n_t)$, $O(d^2n_s)$, $O(d^2n_s^2)$, and $O(d^3)$, respectively. For SSS, the time complexity is $O(d^2n_s)$. Thus the time complexity of the proposed approach is $O(d^2n_s + d^2n_t + d^2n_s^2 + d^3)$.

III. Experiments and Discussions

We used the long-term sensor drift dataset [26], [27], short-term sensor drift dataset, instrumental variation dataset [28],

TABLE I
DETAILS OF THE LONG TERM SENSOR DRIFT DATASET FROM UCI

Batch ID	Month	Acetone	Acetaldehyde	Ethanol	Ethylene	Ammonia	Toluene	Total
Batch 1	1,2	90	98	83	30	70	74	455
Batch 2	3∼10	164	334	100	109	532	5	1244
Batch 3	11,12,13	365	490	216	240	275	0	1586
Batch 4	14,15	64	43	12	30	12	0	161
Batch 5	16	28	40	20	46	63	0	197
Batch 6	17∼20	514	574	110	29	606	467	2300
Batch 7	21	649	662	360	744	630	586	3613
Batch 8	22,23	30	30	40	33	143	18	294
Batch 9	24,30	61	55	100	75	78	101	470
Batch 10	36	600	600	600	600	600	600	3600

and Visually Induced Motion Sickness (VIMS) dataset [29] to verify the effectiveness of our proposed framework for various classification tasks. The VIMS dataset is actually a dataset in which the measurement object changes. We also carried out experiments to show that our framework support both Single-Source Domain Adaptation (SSDA) and Multi-Source Domain Adaptation (MSDA) [30]. For the convenience of comparison, we took the 1NN (kNN with $k = 1$) as a "reference classifier" for all experiments. Note that the 1NN was not only used as the reference classifier for the classification step in all the approaches for comparison but also as an approach for comparison (i.e. a benchmark).

A. Single-Source Domain Adaptation

1) Long-Term Sensor Drift Dataset: To verify the effectiveness of the proposed framework, we adopted a long-term sensor drift dataset from UCI archive for our experiments. It consists of 13910 samples gathered by Vergara *et al*. [26]. They took 36 months totally, from January 2008 to February 2011, to collect this dataset. Sixteen sensors were used for the collection. Eight features were extracted from each sensor. Hence a total of 128 features were extracted. The dataset were collected from 6 types of gases at different concentrations, including Acetone, Acetaldehyde, Ethanol, Ethylene, Ammonia, and Toluene. It was divided into 10 batches according to the collection time. TABLE I provides a summary of the number of samples for each batch. For more details of this dataset, please refer to [26] and [27].

Our comparative experiments used the Setting 1 below and adopted the classification accuracy as the evaluation metric [7].

Setting 1: Batch 1 is used as the source data (training data) for training, and batch K ($K = 2, 3, \ldots, 10$) is used in turn as the target data (testing data) for testing.

Ten approaches were used for comparison. Note that 1NN, PCA, Linear Discriminant Analysis (LDA) are commonly used benchmark approaches. JDA Linear, JDA Primal, JDA RBF, JDA SAM are the joint distribution adaptation approaches using Linear, Primal, Radial Basis Function (RBF), and Spectral Angle Mapper (SAM) kernel, respectively [1]. Other approaches for comparison include DRCA [7], D-DRCA [17], and MDMR [16].

TABLE X lists the classification accuracies of various approaches for Setting 1. The results demonstrate that our MCSP-SSS achieves the best accuracy. Note that the average accuracy (Avg *Acc*) of MCSP is 76.13%, which is 6.07 percentage points (pps) higher than that of D-DRCA; and the accuracy of MSCP-SSS is 78.10%, which is higher than that of MCSP. That's to say, SSS has a positive effect on accuracy.

Please note that Setting 1 covers only a subset of the possible permutations/combinations of different batches in a dataset. To avoid the dependence of results from a particular case, we conducted similar experiments for all possible permutations/combinations. Detailed experimental results can be found from the hyperlink presented in the abstract. A similar situation exists in the subsequent experimental settings.

TABLE II
DETAILS OF THE SHORT-TERM SENSOR DRIFT DATASET COLLECTED

Batch ID	N-propanol	Isopropanol	Acetone	Total
Bach 1	30	30	30	90
Bach 2	15	15	15	45

TABLE III
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON SHORT-TERM SENSOR DRIFT DATASET WITH SETTING 2

Batch ID	1NN	PCA	LDA	DRCA	D-DRCA	MDMR	MCSP	MCSP-SSS
1→2	0.7778	0.8000	0.3333	0.4921	0.8518	0.9259	**0.9556**	**0.9556**
2→1	0.6667	0.7667	0.3333	0.4815	0.8518	0.9259	**0.9815**	**0.9815**

2) Short-Term Sensor Drift Dataset: We used a short-term sensor drift dataset gathered by our electronic nose [31] to verify the proposed framework. We took 10 weeks totally to collect this dataset. Thirty-two sensors were used for data collection. The dataset was collected from gases that contain N-propanol, Isopropanol, and Acetone. Sampling only occurs in the first and tenth weeks. The data collected in the first week were saved as Bach 1, and the tenth week Bach 2. A total of 135 samples were collected and then 32 features were extracted from these samples. The details of this dataset are summarized in TABLE II.

Setting 2 was used for the experiments on the short-term sensor drift dataset. The results are presented in TABLE III.

Setting 2: One batch is used for training and the other batch is used for testing.

From TABLE III we can see that the accuracy of MCSP-SSS is 95.56% for Batch 1→2, and 98.15% for Batch 2→1, both of which are the best in contrast to that of other approaches. This indicates that our MCSP-SSS can reduce the distribution discrepancy of the short-term sensor drift dataset effectively.

3) Instrumental Variation Dataset: We used a public dataset to verify the effectiveness of MCSP-SSS for the dataset with instrumental variation [28]. The dataset was collected from 4 gases: Ethanol (GEa), Methane (GMe), Ethylene (GEy), and Carbon Monoxide (GCO), then it was divided into 5 batches according to the instruments from which the data came. TABLE IV provides a summary of the number of samples in each batch. Please refer to [28] for more details. We used Setting 3 for experiments on instrumental variation dataset.

TABLE IV
DETAILS OF THE INSTRUMENTAL VARIATION DATASET

Batch ID	GEa	GMe	GEy	GCO	Total
Batch 1	40	40	40	40	160
Batch 2	40	40	40	40	160
Batch 3	40	40	40	40	160
Batch 4	20	20	20	20	80
Batch 5	20	20	20	20	80

TABLE V
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON INSTRUMENTAL VARIATION DATASET WITH SETTING 3

Approach	5→1	5→2	5→3	5→4	Avg *Acc*
1NN	0.7375	0.7938	0.8563	0.9626	0.8375
PCA	0.7688	0.7938	0.8563	0.9626	0.8458
LDA	0.1875	0.1313	0.1000	0.5100	0.2322
DRCA	0.7750	0.8188	0.8813	0.9375	0.8536
D-DRCA	0.8688	0.9250	0.9125	0.9875	0.9235
MDMR	0.8875	0.9125	0.9000	0.9750	0.9188
MCSP	0.8500	0.8875	0.8937	0.9875	0.9048
MCSP-SSS	0.8688	0.9250	0.9438	0.9875	**0.9313**

TABLE VI
DETAILS OF THE VIMS DATASET

Subject ID	VIMS	No-VIMS	Total
S1	686	545	1231
S2	97	315	412
S3	455	706	1161
S4	291	346	637
S5	367	494	861
S6	379	500	879
S7	529	850	1379
S8	131	502	633

Setting 3: Take one batch (e.g. Batch 5) as the source data for training, and each batch rest as the target data for testing in turn.

The experimental results are presented in TABLE V. It can be seen that the average accuracy of MCSP-SSS achieves 93.13%, which indicates that MCSP-SSS can solve the problem of instrumental variation effectively.

4) VIMS Dataset: We utilized the VIMS dataset to verify the effectiveness of MCSP-SSS for the dataset with change of measurement object. The VIMS dataset was collected from 8 subjects (S1∼S8). All subjects voluntarily signed the informed content approved by the Institutional Review Board (IRB) before data collection. The VIMS data collection was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the IRB of the Massachusetts Eye and Ear (16-015H, 1/5/2016). The VIMS dataset used in this study comprises of 7193 samples, which can be divided into 8 batches, one batch for one subject. The number of samples per subject is summarized in TABLE VI. There are 20-dimensional features for each subject. For more details about this dataset please refer to [29].

TABLE VII
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON VIMS DATASET WITH SETTING 4

Approach	1→2	1→3	1→4	1→5	1→6	1→7	1→8	Avg *Acc*
1NN	0.5690	0.5206	0.4749	0.8550	0.6716	0.4565	0.4795	0.5753
PCA	0.5763	0.5835	0.5123	0.8596	0.6909	0.5094	0.5457	0.6110
LDA	0.2349	0.3916	0.4561	0.4257	0.4307	0.3833	0.2066	0.3613
DRCA	0.7831	0.3742	0.3867	0.8406	0.5909	0.3841	0.7598	0.5885
D-DRCA	0.9036	0.6452	0.7695	0.7942	0.8579	0.7590	0.7559	0.7849
DMDR	0.5542	0.5806	0.6015	0.8551	0.7897	0.4801	0.5512	0.6303
MCSP	0.8193	0.6888	0.7344	0.8986	0.8267	0.5471	0.7871	0.7546
MCSP-SSS	0.8253	0.6731	0.8203	0.9043	0.8409	0.8442	0.7918	**0.8141**

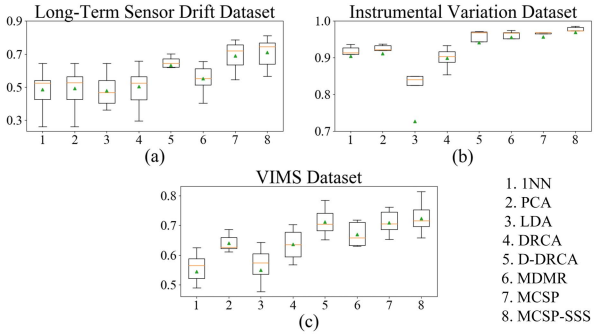


Fig. 5. Box plots plotted based on the experimental results on different datasets: (a) Long-Term Sensor dataset; (b) Instrumental Variation dataset; (c) VIMS dataset. Orange line: Median; Green triangle: Mean.

Similarly, classification accuracy was used as the evaluation metric, and Setting 4 below was used in the experiments.

Setting 4: Take the data from Subject 1 as the source data for training, and the data from Subject K ($K = 2, 3, \ldots, 8$) as the target data for testing in turn.

The experimental results are presented in TABLE VII. We can see that the average accuracy of MCSP-SSS is 2.92 pps higher than the second highest approach (D-DRCA), which demonstrates that it can greatly reduce the distribution discrepancy between the source domain and the target domain.

To give an overview of the performances of different approaches on various data sets, we drawn the box plots based on our experimental results with all the possible permutations/combinations of batches were considered. These results were reported in terms of median, percentile, and mean for each classifier in Fig.5. As can be seen from Fig.5, our MCSP-SSS outperforms all other approaches in terms of both median and mean.

B. Multi-Source Domain Adaptation

The experiments above have demonstrated that MCSP-SSS plays a positive role in single-source domain adaptation. However, a more practical scenario of classification of sensor data is that there are multiple source domains, which refers to a problem of multi-source domain adaptation. As the distributions of multiple source domains are often different, it is challenging to reduce the distribution discrepancy between these source domains and the target domains simultaneously. In this

TABLE VIII
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON INSTRUMENTAL VARIATION DATASET WITH SETTING 6

Approach	5+4→1	5+4→2	5+4→3	Avg *Acc*
1NN	0.8938	0.9500	0.9000	0.9146
PCA	0.9313	0.9563	0.9000	0.9292
LDA	0.9688	0.9875	0.9813	0.9792
DRCA	0.8063	0.8813	0.9125	0.8667
D-DRCA	0.9750	0.9813	0.9875	0.9813
MDMR	0.9438	0.9688	0.9688	0.9605
MCSP	0.9750	0.9875	0.9875	0.9833
MCSP-SSS	0.9813	0.9938	0.9938	**0.9896**

TABLE IX
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON VIMS DATASET WITH SETTING 7

Approach	1+2→3	1+2→4	1+2→5	1+2→6	1+2→7	1+2→8	Avg *Acc*
1NN	0.5129	0.5690	0.7784	0.7057	0.4630	0.4211	0.5750
PCA	0.5534	0.6787	0.8584	0.7295	0.4862	0.4953	0.6335
LDA	0.6351	0.6411	0.7935	0.7852	0.5029	0.5994	0.6596
DRCA	0.5720	0.4609	0.6725	0.6733	0.4293	0.5551	0.5619
D-DRCA	0.6000	0.7539	0.8232	0.8295	0.6612	0.5316	0.6999
MDMR	0.6710	0.6797	0.8551	0.6932	0.4475	0.6732	0.6700
MCSP	0.6538	0.8359	0.8650	0.7841	0.6486	0.6693	0.7380
MCSP-SSS	0.6796	0.8359	0.8696	0.8722	0.6721	0.6929	**0.7704**

section, we designed experiments to show that MCSP-SSS also supports multi-source domain adaptation effectively. Experiments on the long-Term Sensor Drift Dataset, Instrumental Variation Dataset and VIMS Dataset were taken as examples for discussion. Similar conclusions can be achieved for other datasets.

1) Long-Term Sensor Drift Dataset: We used the same long-term sensor drift dataset used for Setting 1 to verify the effectiveness of the proposed framework for MSDA, and Setting 5 was applied to the experiments.

Setting 5: Batch 1 and Batch 2 are combined together to form a new source domain for training, and batch K ($K = 3$, 4, …, 10) is used as the target domain for testing in turn.

TABLE XI lists the classification accuracies of various approaches for Setting 5. As can be seen from TABLE XI, our approach achieves the best accuracy: The average accuracy of it is 79.81%, which is 4.85 pps higher than the second highest approach (MSCP). The results indicate that our MCSP-SSS can improve the classification accuracy of the classifier effectively on long-term sensor drift dataset for MSDA.

2) Instrumental Variation Dataset: The same instrumental variation dataset used for Setting 3 was used to verify the effectiveness of the proposed framework for MSDA. Setting 6 below was adopted for the verification.

Setting 6: Take two batches (e.g. Batch 5 and Batch 4) as the source data for training, and each batch rest as the target data for testing in turn.

It can be seen from TABLE VIII that the average accuracy of MCSP-SSS is 98.96%, which outperforms all other approaches. This shows that MCSP-SSS is effective for MSDA when instrumental variation occurs.

3) VIMS Dataset: Similarly, we conducted experiments on VIMS dataset to verify the effectiveness of the proposed framework for MSDA. Setting 7 below was adopted for the verification.

Setting 7: Take the data from Subject 1 and Subject 2 together as the source data for training, and the data from Subject K ($K = 3$, 4, …, 8) as the target data for testing in turn.

It can be seen from Table IX that the average accuracy of MCSP-SSS is 77.04%, which outperforms all other approaches. This shows that MCSP-SSS is effective for MSDA when the measurement object changes.

C. Parameter Sensitivity

Similar to many other domain adaption approaches, there are also hyperparameters in our proposed framework that should be set in advance. These hyperparameters include the trade-off parameters in MCSP and the parameter k (number of samples selected) in SSS. They need to be fine-tuned during training to get an optimized model. As the distributions of training and testing data are different, we used a technique that combined grid search algorithm [16] and A-distance metric [15] for fine-tuning. The basic idea of this technique is to find a set of optimized hyperparameters for the framework with grid search so that the A-distance between the source data and the target data achieves the smallest.

In this section, parameter sensitivity analysis for these hyperparameters is conducted to show that the proposed framework can achieve optimal results over a wide range of parameter values.

1) Trade-Off Parameters in MCSP: There are 4 trade-off parameters, α, β, γ, and δ, in our proposed MCSP. Referring to the trade-off parameters settings in Refs. [7], [16], and [17], we adopted the values in the following set to tune the trade-off parameters: $\{10^k, k = -4, -3, -2, -1, 0, 1, 2, 3, 4\}$. Due to space limitations, we list only four representative experimental results on the long-term sensor drift dataset here: Batch 1→5, in which the accuracy was the best; Batch 1→4 and Batch 1→9, in which the accuracy was average; Batch 1→10, in which the accuracy was the worst. Note when tuning one parameter, we kept other parameters fixed.

Fig. 6 shows the results of the four experiments when each trade-off parameter varied independently. Each subgraph in Fig. 6 describes the influence of the above hyperparameters on classification accuracy for the four experiments. The values along x-axis of each subgraph show the logarithms of the parameter values, and the y-axis the classification accuracies. From Fig. 6 we can learn that:

- For each trade-off parameter, there is a large interval in which the curve is flat. This indicates that our framework can achieve optimal results over a wide range of parameter values.
- With the increase of the parameter value, MCSP-SSS becomes sensitive to the change of the trade-off parameter. Therefore, it's better for us to choose small values for the trade-off parameters at the beginning of the

TABLE X
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON LONG-TERM SENSOR DRIFT DATASET WITH SETTING 1

Approach	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6	Batch 7	Batch 8	Batch 9	Batch 10	Avg *Acc*
1NN	0.7789	0.6778	0.6087	0.6650	0.6652	0.4174	0.2211	0.4404	0.3301	0.5338
PCA	0.8295	0.8058	0.7143	0.7107	0.7365	0.4888	0.2211	0.4660	0.4108	0.5971
LDA	0.3416	0.4376	0.4161	0.5127	0.3196	0.3687	0.6361	0.3638	0.2036	0.4000
JDA Linear	0.7838	0.6097	0.6708	0.4924	0.7500	0.7667	0.2313	0.5404	0.3583	0.5782
JDA Primal	0.7942	0.7945	0.6273	0.7056	0.7048	0.6275	0.1667	0.6787	0.3583	0.6236
JDA RBF	0.7854	0.7926	0.7019	0.5635	0.8309	0.7312	0.4932	0.5553	0.3400	0.6438
JDA SAM	0.7918	0.8026	0.7267	0.5533	0.8320	0.7628	0.5170	0.5533	0.4025	0.6594
DRCA	0.7717	0.7598	0.6646	0.7107	0.7122	0.4520	0.6259	0.5936	0.4272	0.6353
D-DRCA	0.7998	0.7705	0.7516	0.8883	0.8352	0.5267	0.7380	0.6000	0.3953	0.7006
MDMR	0.8296	0.8058	0.7578	0.7411	0.7404	0.5217	0.5476	0.5127	0.4200	0.6530
MCSP	0.8473	0.8000	0.7846	0.9493	0.8413	0.5546	0.8136	0.7447	0.5167	0.7613
MCSP-SSS	0.8594	0.8268	0.7846	0.9747	0.8891	0.5589	0.8305	0.7500	0.5188	**0.7810**

TABLE XI
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT APPROACHES ON LONG-TERM SENSOR DRIFT DATASET WITH SETTING 5

Approach	1+2→3	1+2→4	1+2→ 5	1+2→ 6	1+2→7	1+2→8	1+2→9	1+2→10	Avg *Acc*
1NN	0.8045	0.4907	0.6599	0.5683	0.5115	0.3095	0.3766	0.4567	0.5222
PCA	0.8045	0.4907	0.6751	0.5683	0.5256	0.4183	0.3787	0.4603	0.5402
LDA	0.4515	0.2236	0.4772	0.4713	0.2845	0.5986	0.1383	0.1256	0.3463
DRCA	0.8707	0.5777	0.5888	0.7913	0.4805	0.5748	0.3936	0.4400	0.5918
D-DRCA	0.8953	0.6770	0.7411	0.7313	0.6438	0.8027	0.6978	0.4300	0.7023
MDMR	0.8493	0.6149	0.7208	0.6965	0.5782	0.4728	0.5787	0.4633	0.6218
MCSP	0.9339	0.5692	0.8607	0.8740	0.7427	0.7203	0.7394	0.5569	0.7496
MCSP-SSS	0.9370	0.7538	0.8860	0.9022	0.7524	0.7966	0.7553	0.6001	**0.7981**

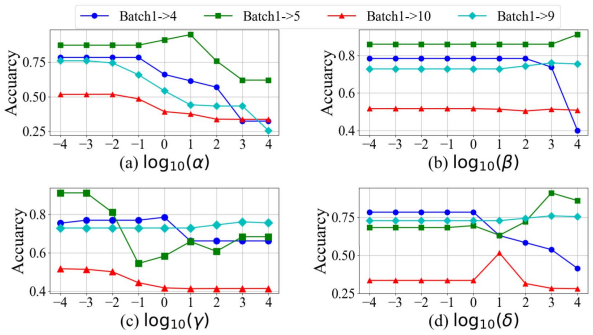


Fig. 6. Trade-off parameter sensitivity analysis for framework MCSP-SSS. (a) Parameter α. (b) Parameter β. (c) Parameter γ. (d) Parameter δ.

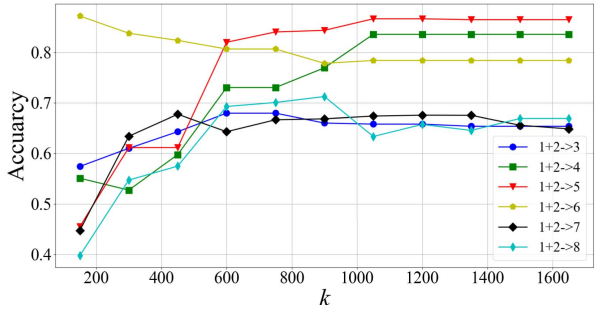


Fig. 7. Parameter sensitivity analysis of parameter k on VIMS dataset for framework MCSP-SSS.

experiment, and then fine-tune these parameters by grid search.

- When parameter β changes, the accuracies of the four experiments vary slightly within a certain range. This indicates that our proposed framework is not sensitive to the changes in parameter β.
- Fig. 6(c) indicates intuitively that our framework is more sensitive to parameter γ than to other parameters. We can fine-tune this parameter by grid search.

2) Parameter in SSS: There is only one hyperparameter, the number of samples selected k, in SSS. The following experiments show the influence of this hyperparameter on classification accuracy. Due to space limitations, we present only the experimental results on VIMS dataset with Setting 6, which are shown in Fig. 7. The values along the x-axis show the values of parameter k, and the y-axis the classification accuracies.

It can be seen from Fig. 7 that with the increase of k the accuracy of MCSP-SSS increases gradually, then it reaches the peak, and finally it becomes stable. Each curve in the figure has a large flat interval with high accuracy (e.g. [1000, 1600]), which indicates that our framework can achieve optimal results over a wide range of values of k. We can also learn from this figure that more samples selected (i.e., larger k) does not always lead to better accuracy.

In our experiment, the hyperparameters are set according to the parameter sensitivity analysis results above. A description of our hyperparameter settings can be found from the hyperlink presented in the abstract.

D. Discussions

MCSP is a key module in our framework. Visualizing the distributions before and after processing by MCSP can give us a view into how MCSP projects the samples and why it can achieve a better classification performance. In order

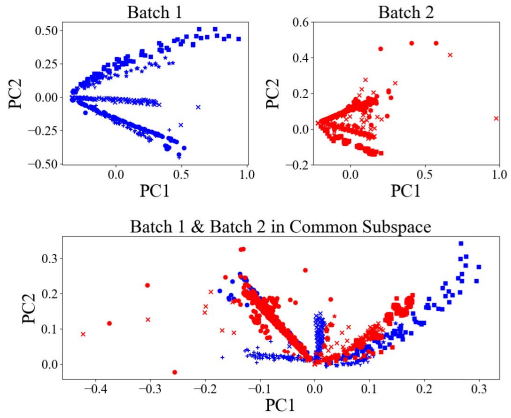


Fig. 8. Distribution discrepancy of long-term sensor drift dataset before and after processing by MCSP, taking experimental results of Batch 1→2 as an example.

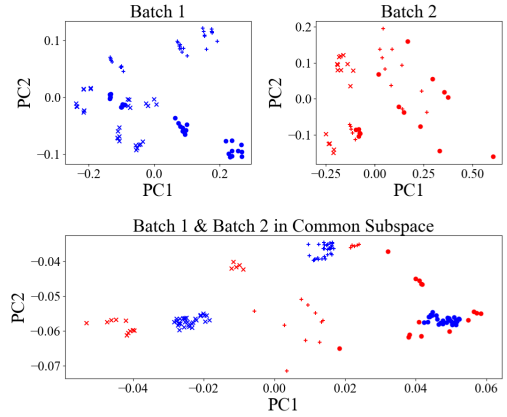


Fig. 9. Distribution discrepancy of short-term sensor drift dataset before and after processing by MCSP, experimental results of Batch 1→2 are shown.

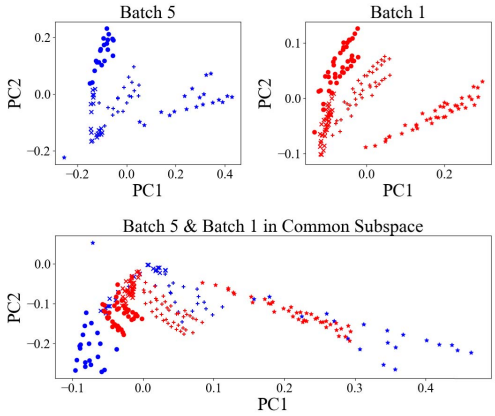


Fig. 10. Distribution discrepancy of instrumental variation dataset before and after processing by MCSP, taking experimental results of Batch 5→1 as an example.

to show intuitively the distribution discrepancy between the source and the target domain before and after processing by MCSP, PCA is applied to the samples to get the first two principal components (PC1 & PC2) so we can plot the PCA scatter points in a 2-D subspace based on these PCs. Fig. 8,

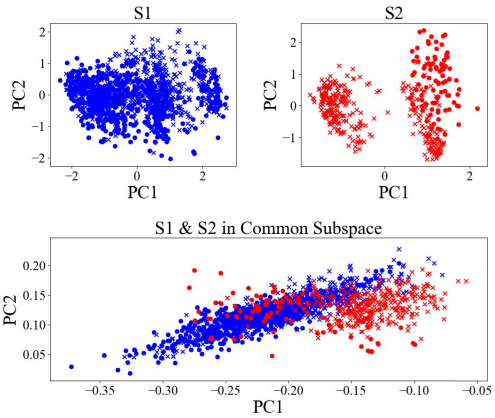


Fig. 11. Distribution discrepancy of VIMS dataset before and after processing by MCSP, taking experimental results of S1→S2 as an example.

Fig. 9, Fig. 10, and Fig. 11 show the distribution discrepancy in the 2-D subspaces before and after processing by MCSP. The subgraphs in the top row of these figures are PCA plots of the samples in the original feature space, and the subgraph in the bottom row is a PCA plot of the samples processed by MCSP in the common subspace.

Compared with the original distributions of the datasets shown in the subgraphs in the top row of these figures, the subgraphs in the bottom row show clearly that the distribution discrepancy in common subspace has been reduced. Hence our MCSP is effective for domain adaption if there are distribution divergences caused by long-term/short-term sensor drift, instrumental variation, or changes of measurement object in the senor data.

SSS also contributes to the improvement of performance. In the previous experiments, we have compared the accuracies of MCSP with and without SSS. These experiments indicate that SSS has a positive effect on accuracy. In contrast to the second-highest approach, the accuracy of MCSP-SSS is increased by 3.57 pps on average for SSDA, and 7.00 pps for MSDA.

IV. Conclusion

This paper proposes a framework called MCSP-SSS to solve the domain adaption problem for sensor data. In this framework, a subspace-projection-based approach, called MCSP, is used to minimize the differences of distribution between source and target domains. Several important constraints are introduced in order to get an optimized projection: PCA is used to reduce the redundancy of features, MDD is applied to minimize the difference between source and target data, and HSIC is adopted to maximize the dependency between features and labels. In addition, we propose a weighted-within-class scatter matrix to minimize the within-class variance so as to avoid samples with different labels to overlap in the subspace. Besides, we propose a so-called SSS approach to further reduce the distribution discrepancy between the source and the target domain by removing the outliers in projected source samples.

Our experimental results show that the proposed MCSP-SSS achieves the best classification accuracy in both single- and multi-source domain adaptation tasks on the datasets that contain distribution divergences caused by long-term/short-term sensor drift, instrumental variation, or changes of measurement object. Besides, the evaluation results of parameter sensitivity indicate that the proposed framework can achieve optimal results over a wide range of parameter values. These results demonstrate that MCSP-SSS is effective for domain adaption for the sensor data with distribution divergences caused by different factors.

ACKNOWLEDGMENT

The authors thank all the scientists, engineers, and volunteers from Schepens and Chongqing University who contributed to data collection.

REFERENCES

[1] J. X. Leon-Medina, W. A. Pineda-Muñoz, and D. A. T. Burgos, "Joint distribution adaptation for drift correction in electronic nose type sensor arrays," *IEEE Access*, vol. 8, pp. 134413–134421, 2020.

[2] Y.-H. Yu, P.-C. Lai, L.-W. Ko, C.-H. Chuang, B.-C. Kuo, and C.-T. Lin, "An EEG-based classification system of Passenger's motion sickness level by using feature extraction/selection technologies," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, New York, NY, USA, Jul. 2010, pp. 1–6.

[3] K. Van Laerhoven, "Combining the self-organizing map and k-means clustering for on-line classification of sensor data," in *Proc. Int. Conf. Artif. Neural Netw.* Berlin, Germany: Springer, Aug. 2001, pp. 464–469.

[4] G. Wei, Z. Jie, Z. Yu, Y. Feng, and S. Xue, "An effective gas sensor array optimization method based on random forest," in *Proc. IEEE SENSORS*, New York, NY, USA, Oct. 2018, pp. 1–4.

[5] B. Schölkopf, J. Platt, and T. Hofmann, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2007, pp. 601–608.

[6] F. Hossein-Babaei and V. Ghafarinia, "Compensation for the drift-like terms caused by environmental fluctuations in the responses of chemoresistive gas sensors," *Sens. Actuators B, Chem.*, vol. 143, pp. 641–648, Jan. 2010.

[7] L. Zhang, Y. Liu, Z. He, J. Liu, P. Deng, and X. Zhou, "Anti-drift in E-nose: A subspace projection approach with drift reduction," *Sens. Actuators B, Chem.*, vol. 253, pp. 407–417, Dec. 2017.

[8] L. Xie, Z. Deng, P. Xu, K.-S. Choi, and S. Wang, "Generalized hidden-mapping transductive transfer learning for recognition of epileptic electroencephalogram signals," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2200–2214, Jun. 2019.

[9] K. Yan and D. Zhang, "Correcting instrumental variation and time-varying drift: A transfer learning approach with autoencoders," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2012–2022, Sep. 2016.

[10] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, Feb. 2016, pp. 2058–2065.

[11] Z. Liang, F. Tian, C. Zhang, H. Sun, A. Song, and T. Liu, "Improving the robustness of prediction model by transfer learning for interference suppression of electronic nose," *IEEE Sensors J.*, vol. 18, no. 3, pp. 1111–1121, Nov. 2018.

[12] S. Marco *et al.*, "Multi-unit calibration rejects inherent device variability of chemical sensor arrays," *Sens. Actuators B, Chem.*, vol. 265, pp. 142–154, Jul. 2018.

[13] S. De Vito, G. Di Francia, E. Esposito, S. Ferlito, F. Formisano, and E. Massera, "Adaptive machine learning strategies for network calibration of IoT smart air quality monitoring devices," *Pattern Recognit. Lett.*, vol. 136, pp. 264–271, Aug. 2020.

[14] S. Zang, Y. Cheng, X. Wang, Q. Yu, and G.-S. Xie, "Cross domain mean approximation for unsupervised domain adaptation," *IEEE Access*, vol. 8, pp. 139052–139069, 2020.

[15] B. Schölkopf, J. Platt, and T. Hofmann, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, Dec. 2007, pp. 137–144.

[16] T. Liu, Y. Chen, D. Li, T. Yang, J. Cao, and M. Wu, "Drift compensation for an electronic nose by adaptive subspace learning," *IEEE Sensors J.*, vol. 20, no. 1, pp. 337–347, Jan. 2020.

[17] Z. Yi and C. Li, "Anti-drift in electronic nose via dimensionality reduction: A discriminative subspace projection approach," *IEEE Access*, vol. 7, pp. 170087–170095, 2019.

[18] H. Liu, Q. Li, Z. Li, and Y. Gu, "A suppression method of concentration background noise by transductive transfer learning for a metal oxide semiconductor-based electronic nose," *Sensors*, vol. 20, no. 7, p. 1913, Mar. 2020.

[19] M. Terashima, F. Shiratani, T. Hashimoto, and K. Yamamoto, "A normalization method of input data that conserves the norm information for competitive learning neural network using inner product," *Opt. Rev.*, vol. 3, no. 6, pp. A414–A417, Nov. 1996.

[20] T. Steinherz, N. Intrator, and E. Rivlin, "Skew detection via principal components analysis," in *Proc. 5th Int. Conf. Document Anal. Recognit. (ICDAR)*, New York, NY, USA, Sep. 1999, pp. 153–156.

[21] L. Song, A. Gretton, K. Borgwardt, and A. J. Smola, "Colored maximum variance unfolding," in *Proc. Adv. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, Dec. 2008, pp. 1385–1392.

[22] J. Xu, J. Liu, J. Yin, and C. Sun, "A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously," *Knowl.-Based Syst.*, vol. 98, pp. 172–184, Apr. 2016.

[23] L. Zhang and D. Zhang, "Domain adaptation extreme learning machines for drift compensation in E-nose systems," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1790–1801, Jul. 2015.

[24] L. Yang, S. Song, and C. L. P. Chen, "Transductive transfer learning based on broad learning system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, New York, NY, USA, Oct. 2018, pp. 912–917.

[25] X. Wang, Y. Gu, and H. Liu, "A transfer learning method for the protection of geographical indication in China using an electronic nose for the identification of Xihu Longjing tea," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8065–8077, Mar. 2021.

[26] A. Vergara *et al.*, "Chemical gas sensor drift compensation using classifier ensembles," *Sens. Actuators B, Chem.*, vol. 166, pp. 320–329, May 2012.

[27] I. Rodriguez-Lujan, J. Fonollosa, A. Vergara, M. Homer, and R. Huerta, "On the calibration of sensor arrays for pattern recognition using the minimal number of experiments," *Chemometrics Intell. Lab. Syst.*, vol. 130, pp. 123–134, Jan. 2014.

[28] J. Fonollosa, L. Fernandez, A. Gutierrez-Galvez, R. Huerta, and S. Marco, "Calibration transfer and drift counteraction in chemical sensor arrays using direct standardization," *Sens. Actuators B, Chem.*, vol. 236, pp. 1044–1053, Nov. 2016.

[29] R. Liu, M. Xu, Y. Zhang, E. Peli, and A. D. Hwang, "A pilot study on electroencephalogram-based evaluation of visually induced motion sickness," *J. Imag. Sci. Technol.*, vol. 64, no. 2, pp. 20501-1–20501-10, Mar. 2020.

[30] C. Chen, W. Xie, Y. Wen, Y. Huang, and X. Ding, "Multiple-source domain adaptation with generative adversarial nets," *Knowl.-Based Syst.*, vol. 199, Jul. 2020, Art. no. 105962.

[31] B. Liu, H. Yu, X. Zeng, D. Zhang, and R. Liu, "Lung cancer detection via breath by electronic nose enhanced with a sparse group feature selection approach," *Sens. Actuators B, Chem.*, vol. 339, no. 12, Jul. 2021, Art. no. 129896.