

Received September 8, 2019, accepted September 17, 2019, date of publication September 23, 2019, date of current version October 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943188

Domain Transfer Broad Learning System for Long-Term Drift Compensation in Electronic Nose Systems

BEI LIU, XIAOPING ZENG, FENGCHUN TIAN^{id}, SHUYA ZHANG, AND LEILEI ZHAO

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China
Chongqing Key Laboratory of Bio-perception and Intelligent Information Processing, Chongqing University, Chongqing 400044, China

Corresponding authors: Bei Liu (liubei@cqu.edu.cn), Xiaoping Zeng (zxp@cqu.edu.cn), and Fengchun Tian (fengchuntian@cqu.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2018CDXYTX0010, and in part by the National Science Foundation of China under Grant 61771080.

ABSTRACT The long-term sensor drift phenomenon seriously restricts the performance of Electronic Nose (E-nose) systems in their various applications. Due to frequent recalibrations, traditional drift compensation methods are costly and laborious, and their performance are limited due to the nonlinear dynamic properties of the drift. The latest proposed Broad Learning System (BLS) has been confirmed to be an efficient and effective learning technique for many machine learning problems. However, BLS with cross-domain learning capability has rarely been studied. In this paper, a novel unified framework called Domain Transfer Broad Learning System (DTBLS) is proposed based on BLS, to address the issue of drift via adaptive compensation. For the case where there is no labeled target sample, with simultaneous considerations of the empirical loss of source data, marginal distribution adaptation, conditional distribution adaptation and manifold regularization, the DTBLS framework learns a robust target classifier by using labeled source data and unlabeled target data to compensate the drift of sensor response adaptively. To the best of our knowledge, DTBLS is the first BLS-based transfer learning framework for the problem of dataset shift existing in E-nose systems. Like the basic BLS, high computation efficiency is achieved due to the existence of analytical solution. Parameter sensitivity analysis is also conducted to show that the optimal solution can be obtained in a wide range. Experiments on a public gas sensor drift dataset demonstrate that the proposed method outperforms the state-of-the-art methods well.

INDEX TERMS Broad learning system, domain transfer, drift compensation, electronic nose.

I. INTRODUCTION

Electronic Nose (E-nose), also called artificial olfactory system, has attracted wide attention from both academic and industrial societies, and the advanced fabrication of sensor material has accelerated this trend [1], [2]. Its latest application fields include (but not limited to) airport and train station checkpoints [3], food security [4], environmental monitoring [5] and clinical diagnosis [6]. A typical E-nose system consists of three parts, namely gas-sensor array, signal pre-processing unit and pattern recognition module. According to [7], ideally, when exposed to the same analyte under the same condition, the gas sensor array response would show identical or similar patterns and return to their baseline level immediately after the analyte is no longer present at the

sensor surface. However, the validity of this assumption is frequently challenged in real-world applications. For instance, poisoning or aging may change the response patterns of the sensors. This phenomenon is called sensor drift, which is formally defined as the gradual changes of transient response signals, caused by dynamic environment conditions or sensed characteristic modifications over time [8]. Actually, these slow changes can seriously affect the performance of pattern recognition. From a statistical point of view, since the sample distribution keeps changing over time, the performance of pattern recognition may dramatically degrade after some period. This long-term drift issue restricts the development of E-nose and has been challenging the research society for a long time.

A great deal of work have been done to mitigate the negative effects of drift. According to different perspectives, these proposed methods can be broadly grouped into

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Afzal^{id}.

three categories, i.e. component correction methods [9]–[14], adaptive methods [15], [16] and machine learning methods [17]–[27].

The component correction methods aim to calculate and remove drift components from the raw data before training a classifier. However, these methods are based on the hypothesis that drift components can be corrected additively which is not always the case and they may fail in handling time-dependent drift. The adaptive methods make an advance in tackling the problem of time-varying drift. Nonetheless, the hypothesis is still adopted and the capability of these methods are limited due to the nonlinear dynamic behavior of the drift phenomenon. In addition, these methods are costly and laborious owing to frequent recalibrations. Instead of removing drift a priori, the machine learning methods are designed to decrease the influence of drift by not modeling drift directly. They can be further divided into two classes: classifier-based approach and feature-based approach. For the classifier-based approach, the purpose is to adapt classifiers to drift directly; whereas the feature-based one is first to seek a common feature subspace between the original and the drifted data spaces in which the distributions of these two spaces are close to each other, then to use standard machine learning methods in this subspace. For the former, at first, Vergara *et al.* [7] introduced an ensemble method based on support vector machine, which used a weighted combination of classifiers trained at different points of time. However, a large amount of labeled data are needed in this method. Other methods based on ensemble learning, such as [17], [18], also have the disadvantage. In order to reduce the need for costly supervised samples and the influences of sensor drift, Semi-Supervised Learning (SSL) techniques that exploit simultaneously the benefits from supervised and unsupervised learning techniques were introduced into the Artificial Olfaction domain [8]. Two investigations indicate that the SSL approaches allow for significant reduction of the number of labeled samples needed to obtain a given performance goal and the effective reduction of drift-induced performance degradation. Inspired by semi-supervised domain adaptation, Liu *et al.* [19] proposed a drift compensation method using manifold regularization based on combination of weighted geodesic flow kernels (GFK). The experiments on the public gas sensor drift dataset [28] show that the proposed method outperforms the baseline methods well. The main limitation of this method is that the conditional distribution adaptation between source and target data domain (i.e. original and drifted data domain) is not taken into consideration. Other classifier-based approaches are given in [20]–[23]. Unfortunately, all of these methods assumed that some labeled target samples are available, which may not be satisfied in practice. For the feature-level approach, there are also plenty of investigations such as [24]–[27]. However, in this kind of method, common feature subspace and classifier are learned independently, probably resulting in weak optimality and robustness compared to the classifier-based method.

From the above discussions, we can see that there are very few classifier-level methods for drift compensation which focus on the case of no labeled target data. And the previous domain adaptation methods for this issue implicitly assume that drift only results in the variation of marginal distribution in feature space of raw data. Actually, conditional distribution of the data also keeps changing with drift. Thus, different from all the previous drift compensation methods, motivated by the idea of joint distribution adaptation strategy in [29], we propose a drift compensation method which simultaneously considers marginal distribution adaptation and conditional distribution adaptation between source and target domain data. Furthermore, inspired by manifold learning, the structure information of the target domain is also to be explored for constructing a more adaptive and accurate target classifier. And the prediction model is assumed to be constructed by the Broad Learning System (BLS) which is a newly proposed single layer feed-forward neural network for classification in [30]. A detailed introduction to BLS is presented in Section II-B. Thus, under the more challenging condition (i.e. there is no labeled target data), we propose a unified transfer framework based on BLS for drift compensation. We refer to the method as Domain Transfer Broad Learning System (DTBLS). The proposed DTBLS aims to learn an adaptive BLS-based classifier by simultaneously taking the following three aspects into account. First, obviously, in order to improve the target predictive function, the available labeled source data should be fully utilized. Hence, structural risk minimization for the source data is a basic consideration. Second, due to the inconsistency of data distributions between source domain data and target domain data, it is vital to reduce the distance between data distributions. By using the projected vision of maximum mean discrepancy [31], [32], marginal distribution and conditional distribution are adapted simultaneously in our framework. Third, to further exploiting the underlying structure information of the source and target data, the manifold learning is utilized. The contributions of this paper are summarized as follows.

- Based on BLS theory, a hybrid method, which incorporates distribution adaption and semi-supervised method, is proposed to deal with gas classification and sensor drift. Specially, the proposed DTBLS incorporates the standard classifier trained by regularized BLS, marginal distribution adaptation, conditional distribution adaptation, and manifold learning simultaneously. By considering these elements jointly, our framework can effectively deal with the drift. Comprehensive experiments on the public long-term gas sensor drift dataset [28] show that both the classification accuracy and time complexity of DTBLS are competitive.
- To the best of our knowledge, DTBLS is the first BLS-based transfer learning framework for the problem of dataset shift [33] in E-nose systems. This provides several new perspectives for exploring BLS theory and expands the application range of BLS.

Moreover, the proposed framework can be easily generalized to the case of available labeled target data.

- DTBLS can be formed into an unified BLS framework, in which two steps, namely feature generating and output weights training, are included. And the method holds the merits of BLS including the feature mapping with randomly generated input weights and bias, the analytically determined solution, and good generalization.

The rest of this paper is organized as follows. In Section II, we briefly review some related works. We then introduce the proposed DTBLS framework detailedly in Section III. Optimization algorithm and computational complexity are shown in Section IV and V. Experimental results and analysis are followed in Section VI. Finally, Section VII concludes this paper.

II. PREVIOUS WORKS AND PRELIMINARIES

In this Section, we introduce the broad learning system and some background knowledge.

A. DOMAIN TRANSFER LEARNING

The problem of drift compensation to be studied herein can be categorized as the domain transfer learning defined in [34]. The definition of domain transfer learning can be described as follows. Given labeled source data $D_s = \left\{ \left(\mathbf{x}_s^{(i)}, \mathbf{y}_s^{(i)} \right) \right\}_{i=1}^{N_s}$ and unlabeled target data $D_t = \left\{ \mathbf{x}_t^{(j)} \right\}_{j=1}^{N_t}$, where $\mathbf{x}_s, \mathbf{x}_t \in \mathcal{X}$ represent the source and target data feature, respectively, $\mathbf{y}_s \in \mathcal{Y}$ is the corresponding source label. In addition, the real target data label \mathbf{y}_t is also assumed to belong to \mathcal{Y} . Denote all the source data features as $\mathbf{X}_s \in \mathbb{R}^{N_s \times d}$, all the target data features as $\mathbf{X}_t \in \mathbb{R}^{N_t \times d}$ and all the source data labels as $\mathbf{Y}_s \in \mathbb{R}^{N_s \times C}$, where d and C are the feature dimension and number of categories, respectively. The goal of domain transfer learning is to learn a target prediction function which can minimize the target prediction error under the condition that the marginal and conditional distributions of source and target domain are both different, namely $P_s(\mathbf{X}_s) \neq P_t(\mathbf{X}_t)$ and $Q_s(\mathbf{Y}_s | \mathbf{X}_s) \neq Q_t(\mathbf{Y}_t | \mathbf{X}_t)$.

B. BROAD LEARNING SYSTEM

The BLS is established in the form of a flat network, where the original inputs are mapped into several new feature representations to form the feature nodes and then these mapped features are expanded in wide sense to enhancement nodes by random mapping. Consider the input data set $\mathbf{X} \in \mathbb{R}^{N \times d}$ which contains N samples with dimension d , and its corresponding label matrix $\mathbf{Y} \in \mathbb{R}^{N \times C}$, where C is the number of categories. As shown in Fig. 1, the basic BLS first maps the original data \mathbf{X} into a set of mapped features to form feature nodes with randomly generated weights, then generates a set of enhancement nodes based on the feature nodes obtained from the previous step similarly. The combination of all the feature nodes and the enhancement nodes are fed into the input and \mathbf{Y} is fed into the output. Finally, a ridge regression

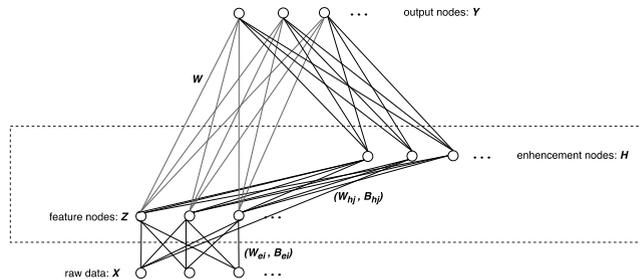


FIGURE 1. Illustration of BLS.

learning algorithm is designed to find the desired connection weights. Details are displayed below.

First, the i -th mapped feature \mathbf{Z}_i is constructed using the following mapping:

$$\mathbf{Z}_i = \phi_i(\mathbf{X}\mathbf{W}_{ei} + \mathbf{B}_{ei}) \quad (1)$$

where \mathbf{W}_{ei} and \mathbf{B}_{ei} are randomly generated weights and biases with proper dimensions respectively, ϕ_i is the activation function. Denote $\mathbf{Z}^i = [\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_i]$, which is the concatenation of all the first i groups of mapping features. The j -th group of enhancement nodes is generated as

$$\mathbf{H}_j = \xi_j(\mathbf{Z}^i\mathbf{W}_{hj} + \mathbf{B}_{hj}) \quad (2)$$

where \mathbf{W}_{hj} and \mathbf{B}_{hj} are the randomly generated weights and biases with the proper dimensions respectively, and ξ_j is the activation function. Similarly, denote $\mathbf{H}^j = [\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_j]$. In practice, the number of i and j can be selected according to the complexity of the modeling task.

Specially, in the first mapping, to obtain better features, BLS takes the advantages of sparse feature learning to fine-tune the initial \mathbf{W}_{ei} and \mathbf{B}_{ei} [35]. Assume that we have n groups of feature mappings and m groups of enhancement mappings with each feature mapping and enhancement mapping generating k nodes and q nodes, respectively. Then the obtained BLS feature can be represented as $\mathbf{A} = [\mathbf{Z}^n | \mathbf{H}^m] \in \mathbb{R}^{N \times (nk+mq)}$. In the following part, the concatenation of feature nodes and enhancement nodes is called BLS feature.

The l_2 norm regularized BLS can be expressed as the following convex optimization problem

$$\arg \min_{\mathbf{W}} \|\mathbf{A}\mathbf{W} - \mathbf{Y}\|^2 + \rho \|\mathbf{W}\|^2 \quad (3)$$

where the second term in (3) is the regularization term used to adjust generalization performance, ρ is the regularization coefficient, \mathbf{W} is the connecting weights for the broad structure.

Consequently, \mathbf{W} can be easily calculated as

$$\mathbf{W} = \left(\mathbf{I} + \rho\mathbf{A}^T\mathbf{A} \right)^{-1} \mathbf{A}^T\mathbf{Y} \quad (4)$$

where \mathbf{I} is the identical matrix with proper dimensions.

C. MAXIMUM MEAN DISCREPANCY

In [31], Gretton *et al.* introduced the Maximum Mean Discrepancy (MMD) for measuring the distance of two distributions in a Reproducing Kernel Hilbert Space (RKHS). Unlike many estimators such as Kullback–Leibler (K–L) divergence [36] and Bregman divergence [37], which are parametric or need an intermediate density estimation procedure, MMD is a nonparametric distance estimate designed by embedding distributions [38] that is both efficient to compute and relatively easy to incorporate into optimization problems whereas still allowing accurate distance measurement. Given two datasets $X = \{x_i\}_{i=1}^{n_1}$ and $Y = \{y_i\}_{i=1}^{n_2}$ drawn independently and identically distributed (i.i.d.) from distribution p and q defined on a domain \mathcal{X} , respectively. MMD is defined as $MMD[\mathcal{F}, p, q] := \sup_{f \in \mathcal{F}} (\mathbf{E}_{x \sim p}[f(x)] - \mathbf{E}_{y \sim q}[f(y)])$,

where \mathcal{F} is a given class of functions $f: \mathcal{X} \rightarrow \mathbb{R}$. It has been proved in [31] that when \mathcal{F} is a unit ball in an universal RKHS \mathcal{H} , $MMD[\mathcal{F}, p, q] = 0$ if and only if $p = q$. An empirical estimate of squared MMD between X and Y in an RKHS \mathcal{H} can be calculated as $MMD^2(X, Y) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(y_i) \right\|_{\mathcal{H}}^2$, where ϕ is the kernel-induced feature map, and $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm. It can be seen that the distance between two distributions is converted to the distance between the sample means of two distributions in an RKHS which is easier to be calculated.

Due to its obvious merits, the MMD measure has been widely adopted in domain adaptation problems. For instance, a classic approach named as transfer component analysis (TCA) [39] tried to learn a common subspace across domains in a RKHS using MMD. In the subspace, data distributions in different domains are close to each other and standard machine learning methods can be applied to train classifiers or regression models. Unfortunately, the original MMD can not be directly used in our framework. In [32], Quanz and Huan proposed the concept of projected MMD and utilized it as a regularizer to decrease the distance between source and target domains based on the the Support Vector Machine (SVM) paradigm. Take X and Y for example, the squared projected maximum mean discrepancy can be expressed as $MMD_p^2(X, Y) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \omega^T \phi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \omega^T \phi(y_i) \right\|_{\mathcal{H}}^2$, where ω is a vector in the Hilbert space defined by the corresponding kernel function. Furthermore, Long *et al.* [29] proposed a transfer learning method with joint distribution adaptation which means the marginal distribution and conditional distribution between domains can be simultaneously adapted. Inspired by the projected MMD and joint distribution adaptation, two regularizers denoting the marginal distribution and conditional distribution adaptation can be reasonably added to our DTBLS. The details are displayed in Section III-B and Section III-C, respectively. Therefore, the proposed framework not only inherits the learning capability and the computational efficiency of

BLS framework, but also owns the capability of transfer learning.

D. MANIFOLD REGULARIZATION

To enhance the classification performance, Belkin *et al.* [40] proposed a family of semi-supervised algorithms which exploit the geometry of the marginal distribution. These algorithms were collectively referred to as manifold regularization algorithms. The manifold regularization is based on a specific assumption about the connection between the marginal and the conditional distribution. Consider labeled examples (x, y) generated according to a probability distribution P and unlabeled examples $x \in X$ drawn according to the marginal distribution \mathcal{P}_X of P . The assumption is that if two points $x_1, x_2 \in X$ are close in the intrinsic geometry of the underlying marginal distribution \mathcal{P}_X , then the conditional distributions $\mathcal{P}(y|x_1)$ and $\mathcal{P}(y|x_2)$ are similar. In other words, the conditional probability distribution $\mathcal{P}(y|x)$ varies smoothly along the geodesics in the intrinsic geometry of \mathcal{P}_X . By far, the manifold regularization has come into wide use in many semi-supervised and unsupervised learning methods. For instance, based on the BLS framework, a novel variant graph regularized broad learning system (GBLS) is proposed in [41]. By taking account of the locally invariant property of data, GBLS incorporates the manifold learning into the objective function of the standard BLS. Likewise, in our DTBLS, the manifold regularization is also incorporated as an additional regularization term to the optimization problem of original BLS. Different from the setting of supervised learning, in our DTBLS framework, we apply the manifold regularization to the scenario of cross-domain semi-supervised learning where labeled and unlabeled data are from different distributions. By exploiting the manifold properties of both labeled and unlabeled data, better discriminative hyperplanes are expected to be obtained.

III. DOMAIN TRANSFER BLS

In this Section, we introduce the DTBLS framework designed for the domain transfer learning problem in detail. First, a novel optimization problem based on BLS is established. Then the analytical solution to the above problem is given and the summary of the proposed framework is presented.

The DTBLS framework is constructed based on the following three considerations, namely

- Minimizing the structural risk functional on the labeled source domain data D_s ;
- Minimizing the distances between marginal distribution and conditional distribution of source and target domain simultaneously;
- Maximizing the manifold consistency underlying all the source and target data.

It can be formulated as

$$\arg \min_{\mathbf{W}} \sum_{i=1}^{N_s} \| \mathbf{e}_s^{(i)} \|^2 + \sigma \| \mathbf{W} \|^2 + \lambda \mathcal{M}(\mathbf{X}) + \gamma_1 dis^2(P_s, P_t) + \gamma_2 dis^2(Q_s, Q_t) \quad (5)$$

where \mathbf{W} is the output weights of the broad network, $\mathbf{e}_s^{(i)}$ is the training error of the i -th source sample, N_s is the number of source samples. We denote the marginal distribution distance as $dis(P_s, P_t)$ and the conditional distribution distance as $dis(Q_s, Q_t)$. The manifold regularization term is represented as $M(\mathbf{X})$, where $\mathbf{X} = [\mathbf{X}_s; \mathbf{X}_t]$ is all of the source and target data and each element is denoted as $\mathbf{x}^{(i)}$ ($i = 1, 2, \dots, N_s + N_t$). $\sigma, \gamma_1, \gamma_2, \lambda$ are the regularization parameters to control the model complexity, the distance between marginal distribution, the distance between conditional distribution and the manifold consistency, respectively. These factors are to be explained in the following sections.

A. STRUCTURAL RISK MINIMIZATION

The core of transfer learning is to exploit rich labeled source domain data to predict the target domain data which is few labeled or unlabeled. Thus, an important optimization goal is to minimize the the structural risk functional of the source domain data according to structural risk minimization principle [42]. The formulation is just the same as that of the basic BLS:

$$\min_{\mathbf{W} \in \mathbb{R}^{L \times C}} \|\mathbf{G}_s \mathbf{W} - \mathbf{Y}_s\|^2 + \sigma \|\mathbf{W}\|^2 \quad (6)$$

where σ is the regularization coefficient, \mathbf{W} is the output weights, \mathbf{G}_s is the BLS feature for all the source domain data. The generation process of \mathbf{G}_s is as follows. First, we randomly generate \mathbf{W}_{ei} and \mathbf{B}_{ei} , and fine-tune them with sparse learning. Then \mathbf{Z}_{si} and \mathbf{H}_{sj} is calculated by using

$$\mathbf{Z}_{si} = \phi_i(\mathbf{X}_s \mathbf{W}_{ei} + \mathbf{B}_{ei}) \quad (7)$$

and

$$\mathbf{H}_{sj} = \xi_j(\mathbf{Z}_{si} \mathbf{W}_{hj} + \mathbf{B}_{hj}) \quad (8)$$

respectively. With n groups of feature mappings and m groups of enhancement mappings, \mathbf{G}_s can be denoted as

$$\mathbf{G}_s = [\mathbf{Z}^{sn} \mid \mathbf{H}^{sm}]. \quad (9)$$

Let \mathbf{G} be the the BLS feature for \mathbf{X} which is calculated the same way as \mathbf{G}_s . Thus, Equation (6) can be rewritten as

$$\min_{\mathbf{W} \in \mathbb{R}^{L \times C}} \|\Lambda_e(\mathbf{G}\mathbf{W} - \mathbf{Y})\|^2 + \sigma \|\mathbf{W}\|^2 \quad (10)$$

where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_s \\ \mathbf{0} \end{bmatrix}_{(N_s+N_t) \times C}$$

and

$$\Lambda_e = \begin{bmatrix} \mathbf{I}_{N_s \times N_s} & \mathbf{0}_{N_s \times N_t} \\ \mathbf{0}_{N_t \times N_s} & \mathbf{0}_{N_t \times N_t} \end{bmatrix}.$$

B. MARGINAL DISTRIBUTION ADAPTATION

As mentioned in Section II-C, one method for estimating the distance between two distributions in an reproducing kernel Hilbert space is the maximum mean discrepancy (MMD) measure [31] which has been widely used in the context of transfer learning. Inspired by the projected maximum mean discrepancy distance measure [32], we introduce the marginal distribution adaptation into the BLS model by computing the distance between the predicted label means of the source and target data. The estimated marginal distribution distance between source and target domains is computed as follows.

Denote the BLS feature mapping as $g(\cdot)$, then the predicted label for $\mathbf{x}_s^{(i)}$ is that $f(\mathbf{x}_s^{(i)}) = g(\mathbf{x}_s^{(i)}) \mathbf{W}$. Thus, we have

$$\begin{aligned} dis^2(P_s(\mathbf{X}_s), P_t(\mathbf{X}_t)) &= \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} f(\mathbf{x}_s^{(i)}) - \frac{1}{N_t} \sum_{j=1}^{N_t} f(\mathbf{x}_t^{(j)}) \right\|^2 \\ &= \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} g(\mathbf{x}_s^{(i)}) \mathbf{W} - \frac{1}{N_t} \sum_{j=1}^{N_t} g(\mathbf{x}_t^{(j)}) \mathbf{W} \right\|^2 \\ &= \|\mathbf{1}^T \Lambda_0 \mathbf{G} \mathbf{W}\|^2 \end{aligned} \quad (11)$$

where

$$\mathbf{G} = [g^T(x^{(1)}), g^T(x^{(2)}), \dots, g^T(x^{(N_s+N_t)})]^T$$

and

$$\begin{aligned} \Lambda_0 &= \begin{bmatrix} \Lambda_s & \mathbf{0}_{N_s \times N_t} \\ \mathbf{0}_{N_t \times N_s} & \Lambda_t \end{bmatrix} \\ \Lambda_s &= \frac{1}{N_s} \mathbf{I}_{N_s \times N_s} \\ \Lambda_t &= \frac{1}{N_t} \mathbf{I}_{N_t \times N_t} \end{aligned}$$

Note that $\mathbf{1}$ in (11) denotes a $(N_s + N_t) \times 1$ vector whose elements are all one.

Further,

$$dis^2(P_s(\mathbf{X}_s), P_t(\mathbf{X}_t)) = \text{tr}(\mathbf{W}^T \mathbf{G}^T \mathbf{M}_0 \mathbf{G} \mathbf{W}) \quad (12)$$

where

$$\mathbf{M}_0 = \Lambda_0^T \mathbf{1} \mathbf{1}^T \Lambda_0. \quad (13)$$

To reduce the complexity of constructing \mathbf{M}_0 , further calculation results in the following expression:

$$(\mathbf{M}_0)_{ij} = \begin{cases} \frac{1}{N_s^2}, & \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D_s \\ \frac{1}{N_t^2}, & \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D_t \\ -\frac{1}{N_s N_t}, & \mathbf{x}^{(i)} \in D_s, \mathbf{x}^{(j)} \in D_t \\ & \text{or } \mathbf{x}^{(j)} \in D_s, \mathbf{x}^{(i)} \in D_t \\ 0, & \text{others} \end{cases} \quad (14)$$

C. CONDITIONAL DISTRIBUTION ADAPTATION

In order to implement conditional distributions adaptation [34], the pseudo labels of the target data are required. To this end, one can train a model with source domain data by utilizing supervised machine learning methods or transfer learning methods. Then, the conditional probability distributions distance is estimated approximately as follows.

$$\begin{aligned} &dis^2(Q_s(\mathbf{Y}_s | \mathbf{X}_s), Q_t(\mathbf{Y}_t | \mathbf{X}_t)) \\ &= \sum_{c=1}^C \left\| \frac{1}{N_s^{(c)}} \sum_{\mathbf{x}_s^{(i)} \in D_s^{(c)}} f(\mathbf{x}_s^{(i)}) - \frac{1}{N_t^{(c)}} \sum_{\mathbf{x}_t^{(j)} \in D_t^{(c)}} f(\mathbf{x}_t^{(j)}) \right\|^2 \\ &= \sum_{c=1}^C \left\| \frac{1}{N_s^{(c)}} \sum_{\mathbf{x}_s^{(i)} \in D_s^{(c)}} g(\mathbf{x}_s^{(i)}) \mathbf{W} - \frac{1}{N_t^{(c)}} \sum_{\mathbf{x}_t^{(j)} \in D_t^{(c)}} g(\mathbf{x}_t^{(j)}) \mathbf{W} \right\|^2 \end{aligned} \quad (15)$$

where $c = 1, 2, \dots, C$ is the category numbers, $D_s^{(c)} = \{\mathbf{x}_s^{(i)} | y(\mathbf{x}_s^{(i)}) = c\}$ is the set of source samples belonging to class c , $y(\mathbf{x}_s^{(i)})$ is the true labels of $\mathbf{x}_s^{(i)}$ and $N_s^{(c)} = |D_s^{(c)}|$. Correspondingly, $D_t^{(c)} = \{\mathbf{x}_t^{(j)} | \hat{y}(\mathbf{x}_t^{(j)}) = c\}$ is the set of target samples belonging to class c , $\hat{y}(\mathbf{x}_t^{(j)})$ is the pseudo labels of $\mathbf{x}_t^{(j)}$, and $N_t^{(c)} = |D_t^{(c)}|$.

Similar to equation (11), we can rewrite the above formula as the matrix form

$$dis^2(Q_s(\mathbf{Y}_s | \mathbf{X}_s), Q_t(\mathbf{Y}_t | \mathbf{X}_t)) = \sum_{c=1}^C \|\mathbf{1}^T \Lambda_c \mathbf{G} \mathbf{W}\|^2 \quad (16)$$

where $\Lambda_c = \begin{bmatrix} \Lambda_s^c & \mathbf{0}_{N_s \times N_t} \\ \mathbf{0}_{N_t \times N_s} & \Lambda_t^c \end{bmatrix}$, whereas Λ_s^c and Λ_t^c are both diagonal matrices whose elements can be calculated as

$$(\Lambda_s^c)_{ii} = \begin{cases} -\frac{1}{N_s^{(c)}}, & \text{if } \mathbf{x}_s^{(i)} \in D_s^{(c)} \\ 0, & \text{otherwise} \end{cases}$$

and

$$(\Lambda_t^c)_{jj} = \begin{cases} -\frac{1}{N_t^{(c)}}, & \text{if } \mathbf{x}_t^{(j)} \in D_t^{(c)} \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore,

$$dis^2(Q_s(\mathbf{Y}_s | \mathbf{X}_s), Q_t(\mathbf{Y}_t | \mathbf{X}_t)) = \text{tr}(\mathbf{W}^T \mathbf{G}^T \mathbf{M}_C \mathbf{G} \mathbf{W}) \quad (17)$$

where

$$\begin{aligned} \mathbf{M}_C &= \sum_{c=1}^C \mathbf{M}_c \\ &= \Lambda_C^T \mathbf{1} \mathbf{1}^T \Lambda_C. \end{aligned} \quad (18)$$

Likewise, to reduce the complexity of constructing \mathbf{M}_c , further calculation is as follows.

$$(\mathbf{M}_c)_{ij} = \begin{cases} \frac{1}{N_s^{(c)2}}, & \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D_s^{(c)} \\ \frac{1}{N_t^{(c)2}}, & \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D_t^{(c)} \\ -\frac{1}{N_s^{(c)} N_t^{(c)}}, & \mathbf{x}^{(i)} \in D_s^{(c)}, \mathbf{x}^{(j)} \in D_t^{(c)} \\ & \text{or } \mathbf{x}^{(j)} \in D_s^{(c)}, \mathbf{x}^{(i)} \in D_t^{(c)} \\ 0, & \text{others} \end{cases} \quad (19)$$

D. MANIFOLD REGULARIZATION

According to the manifold assumption, similar samples have similar labels. With this regularization, the resulting classifier is enforced to give smooth predictions and not to cut through the high density regions. Specially, first, a graph can be defined on the data set, where the nodes correspond to the labeled or unlabeled samples, and the weights of edges reflect the similarity between the samples; then, the label smoothness can be enforced over the graph as a regularization term.

Based on geodesic smoothness, the manifold regularization is computed as

$$\begin{aligned} M(\mathbf{X}) &= \frac{1}{2} \sum_{i,j=1}^{N_s+N_t} \omega_{ij} \|f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(j)})\|^2 \\ &= \frac{1}{2} \sum_{i,j=1}^{N_s+N_t} \omega_{ij} \|g(\mathbf{x}^{(i)}) \mathbf{W} - g(\mathbf{x}^{(j)}) \mathbf{W}\|^2 \\ &= \text{tr}(\mathbf{W}^T \mathbf{G}^T (\mathbf{D} - \Omega) \mathbf{G} \mathbf{W}) \\ &= \text{tr}(\mathbf{W}^T \mathbf{G}^T \Delta \mathbf{G} \mathbf{W}) \end{aligned} \quad (20)$$

where Δ is the graph Laplacian given by

$$\Delta = \mathbf{D} - \Omega. \quad (21)$$

Here $\Omega \in \mathbb{R}^{(N_s+N_t) \times (N_s+N_t)}$ is the graph adjacency matrix composed of ω_{ij} and \mathbf{D} is a diagonal matrix with each item as

$$D_{ii} = \sum_{j=1}^{N_s+N_t} \omega_{ij} \quad (22)$$

where ω_{ij} is given in (23), as shown at the bottom of the next page. Note we use $\mathcal{N}_u(\mathbf{x}^{(i)})$ to denote the set of u -nearest neighbors of point $\mathbf{x}^{(i)}$. In fact, when we use the graph Laplacian for simplicity, the normalized Laplacian

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} \Delta \mathbf{D}^{-\frac{1}{2}} \quad (24)$$

can be used interchangeably. Using \mathbf{L} instead of Δ provides certain theoretical guarantees [43] and seems to perform as well or better in many practical tasks. Thus, we use \mathbf{L} in all our empirical studies. The final form of manifold regularization is

$$M(\mathbf{X}) = \text{tr}(\mathbf{W}^T \mathbf{G}^T \mathbf{L} \mathbf{G} \mathbf{W}). \quad (25)$$

IV. OPTIMIZATION ALGORITHM

By incorporating (10), (12), (18) and (25), the DTBLS framework can be formulated as the following optimization objective

$$\min_{\mathbf{W} \in \mathbb{R}^{L \times C}} \|\Lambda_e(\mathbf{G}\mathbf{W} - \mathbf{Y})\|^2 + \sigma \|\mathbf{W}\|^2 + \text{tr}(\mathbf{W}^T \mathbf{G}^T (\gamma_1 \mathbf{M}_0 + \gamma_2 \mathbf{M}_C + \lambda \mathbf{L}) \mathbf{G} \mathbf{W}) \quad (26)$$

By setting the gradient of (26) w.r.t \mathbf{W} to be zero, we obtain its solution. Specifically, if the dimension of BLS feature L is less than the number of samples $(N_s + N_t)$, the solution is (27), as shown at the bottom of this page; otherwise, we get (28), as shown at the bottom of this page.

For a new test sample \mathbf{x} , the predicted label is computed as $f(\mathbf{x}) = g(\mathbf{x}) \mathbf{W}$.

According to the above statement, we summary our framework in Algorithm 1.

Algorithm 1 DTBLS

Input: Data \mathbf{X} , \mathbf{Y}_s ; parameters σ , γ_1 , γ_2 , λ .

Output: \mathbf{W} .

begin

(1) **for** $i = 1, \dots, n$ **do**

Random \mathbf{W}_{ei} , \mathbf{B}_{ei} ;

Calculate $\mathbf{Z}_i = \phi_i(\mathbf{X}\mathbf{W}_{ei} + \mathbf{B}_{ei})$;

Fine-tune \mathbf{Z}_i using sparse learning;

end

(2) set the feature mapping group $\mathbf{Z}^n = [\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n]$.

(3) **for** $j = 1, \dots, m$ **do**

Random \mathbf{W}_{hj} , \mathbf{B}_{hj} ;

Calculate $\mathbf{H}_j = \xi_j(\mathbf{X}\mathbf{W}_{hj} + \mathbf{B}_{hj})$;

end

(4) Set the enhancement nodes group

$\mathbf{H}^m = [\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_m]$.

(5) Set $\mathbf{G} = [\mathbf{Z}^n, \mathbf{H}^m]$.

(6) Construct \mathbf{M}_0 by (13), \mathbf{M}_C by (18), and \mathbf{L} by (21)–(24).

(7) Normalize the graph Laplacian Δ , namely $\mathbf{L} \leftarrow \mathbf{D}^{-\frac{1}{2}} \Delta \mathbf{D}^{-\frac{1}{2}}$.

(8) Calculate \mathbf{W} by (27) or (28).

V. COMPUTATIONAL COMPLEXITY

Let s denote the average number of non-zero elements per example. It is natural that $s \leq d$. In addition, generally, $u \leq \min(d, N_s + N_t)$. The computational complexity of the framework consists of three parts.

(1) For a $m \times m$ matrix, the computational complexity of inverse operation is $O(m^3)$. Thus, when it comes to the first case, namely (27), $O((N_s + N_t)^2)$ is needed. While for the second case, namely (28), the computational complexity is $O((N_s + N_t)^3)$.

(2) For constructing the BLS feature matrix \mathbf{G} , the DTBLS needs $O((\max(d, N_s + N_t))^2)$ at most.

(3) For constructing the graph Laplacian matrix \mathbf{L} , the DTBLS needs $O(s(N_s + N_t)^2)$.

(4) For constructing the \mathbf{M}_0 , the computational complexity is $O((N_s + N_t)^2)$.

(5) For constructing the \mathbf{M}_C , the computational complexity is $O(C(N_s + N_t)^2)$.

In summary, the computational complexity of Algorithm 1 is

$O((N_s + N_t)^3 + (s + C)(N_s + N_t)^2 + (\max(d, N_s + N_t))^2)$ or $O((s + C)(N_s + N_t)^2 + (\max(d, N_s + N_t))^2)$.

VI. EXPERIMENTS

In this section, we will examine our DTBLS framework on the E-nose Gas Sensor Drift Dataset in UCI Machine Learning Repository [28].

A. EXPERIMENTAL DATA AND SETTINGS

The long-term sensor drift dataset consists of 13,910 measurements collected over a period of 36 months. There recordings were sampled by an E-Nose system with a 16 metal-oxide semiconductor gas sensor array exposed to six gases including Ammonia, Acetaldehyde, Acetone, Ethylene, Ethanol and Toluene at different concentration levels. Ignoring the concentration information, each sample contains a class label and a 128-dimensional feature vector. The feature vector is formulated by extracting eight features from each sensor. Interested readers can refer to [7] for specific technical details on how to determine the 8 features for each sensor. In order to make the number of measurements as uniformly distributed as possible, the measurements from 36 months were combined to form 10 batches. Details on the period of collection and number of measurements in each batch is given in Table 1.

For the visualization of the time-varying sensor drift across batches, we apply Principal Component Analysis (PCA) to the data set, and project the data into a 2-D subspace based on the first two PCs. As shown in Fig. 2, the projected 2-D subspace for each batch is displayed and in particular

$$\omega_{ij} = \begin{cases} \cos(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), & \text{if } \mathbf{x}^{(i)} \in \mathcal{N}_u(\mathbf{x}^{(j)}) \vee \mathbf{x}^{(j)} \in \mathcal{N}_u(\mathbf{x}^{(i)}) \\ 0, & \text{else} \end{cases} \quad (23)$$

$$\mathbf{W} = (\sigma \mathbf{I} + \gamma_1 \mathbf{G}^T \mathbf{M}_0 \mathbf{G} + \gamma_2 \mathbf{G}^T \mathbf{M}_C \mathbf{G} + \lambda \mathbf{G}^T \mathbf{L} \mathbf{G} + \mathbf{G}^T \Lambda_e \mathbf{G})^{-1} (\mathbf{G}^T \Lambda_e \mathbf{Y}) \quad (27)$$

$$\mathbf{W} = \mathbf{G}^T (\sigma \mathbf{I} + \gamma_1 \mathbf{M}_0 \mathbf{G} \mathbf{G}^T + \gamma_2 \mathbf{M}_C \mathbf{G} \mathbf{G}^T + \lambda \mathbf{L} \mathbf{G} \mathbf{G}^T + \Lambda_e \mathbf{G} \mathbf{G}^T)^{-1} (\Lambda_e \mathbf{Y}) \quad (28)$$

TABLE 1. Long-term drift data set details.

Batch ID	Month	Ammonia	Acetaldehyde	Acetone	Ethylene	Ethanol	Toluene	Total
Batch 1	1,2	83	30	70	98	90	74	445
Batch 2	3-10	100	109	532	334	164	5	1244
Batch 3	11,12,13	216	240	275	490	365	0	1586
Batch 4	14,15	12	30	12	43	64	0	161
Batch 5	16	20	46	63	40	28	0	197
Batch 6	17,18,19,20	110	29	606	574	514	467	2300
Batch 7	21	360	744	630	662	649	568	3613
Batch 8	22,23	40	33	143	30	30	18	294
Batch 9	24,30	100	75	78	55	61	101	470
Batch 10	36	600	600	600	600	600	600	3600

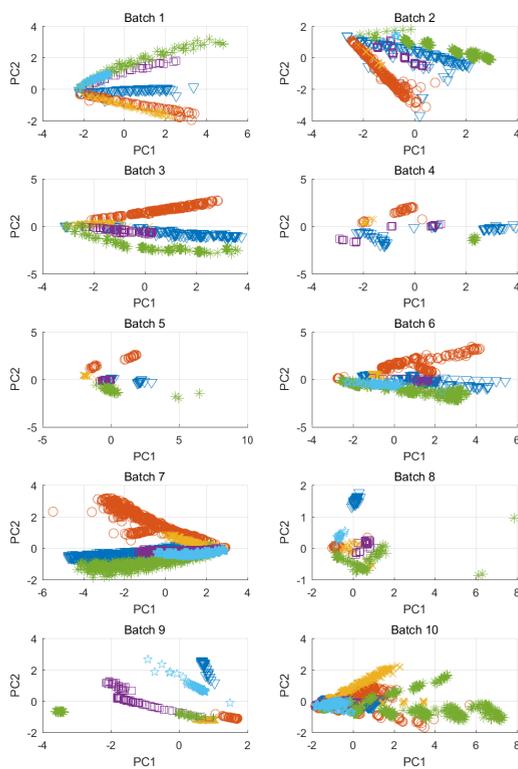


FIGURE 2. Scatter plots of samples in batches 1-10 in the gas sensor array drift dataset. Samples are projected to a 2D subspace using PCA. Here ∇ , \circ , \times , \square , $*$ and \star indicate Acetone, Acetaldehyde, Ethanol, Ethylene, Ammonia and Toluene respectively.

different categories are showed in different colors. Ignoring the class information, it can be seen that the outline of the scatter plots change a lot across different batches most of the time. Besides, the scatter diagrams of any kind of gas also vary greatly between different batches. In other words, both the marginal distribution and conditional distribution of the data set change significantly across different batches. The pattern of sensor drift is so complicated that modeling and compensating drift directly is difficult. Thus, compensating drift from the perspective of transfer learning is a more reasonable and promising direction.

To generate several classification tasks, according to [19], two experimental settings as follows are given to verify our algorithm:

- Setting 1: fix Batch 1 (source domain) as the training set, and then test on other batches (target domain).
- Setting 2: dynamically changing the training set (source domain) with the batch $T - 1$, and then test on batch T ($T = 2, \dots, 10$).

Indeed, we compare the proposed DTBLS framework with the following 11 kinds of algorithms that have been applied in sensor drift compensation:

- Component Correction-Principal Component Analysis (CC-PCA) [13]
- Support Vector Machine-radial basis function (SVM-rbf)
- Support Vector Machine-geodesic flow kernel (SVM-gfk)
- Support Vector Machine-combination geodesic flow kernel (SVM-comgfk) [19]
- Manifold Regularization-radial basis function (MI-rbf) [19]
- Manifold Regularization-combination geodesic flow kernel (MI-comgfk) [19]
- Extreme Learning Machine-radial basis function (ELM-rbf) [44]
- source Domain Adaptation Extreme Learning Machine (DAELM-S) [23]
- target Domain Adaptation Extreme Learning Machine (DAELM-T) [23]
- multi-feature kernel semi-supervised joint learning (MFKS) [20]
- Broad Learning System (BLS) [30]

CC-PCA is a representative signal correction based method. According to [7], the SVM-rbf trained in Setting 2 is believed to be a natural and strong baseline for any drift-correcting machine learning algorithm to compare against and performing better than or as well as this setting is a positive result. MI-comgfk is the algorithm proposed in [19] and SVM-gfk, SVM-comgfk, MI-rbf are its comparison algorithms. The regularized ELM with RBF function in

TABLE 2. Comparisons of recognition accuracy(%) under the experimental setting 1.

Method	Batch2	Batch3	Batch4	Batch5	Batch6	Batch7	Batch8	Batch9	Batch10	Average
CC-PCA	67.00	48.50	41.00	35.50	55.00	31.00	56.50	46.50	30.50	45.72
SVM-rbf	74.36	61.03	50.93	18.27	28.26	28.81	20.07	34.26	34.48	38.94
SVM-gfk	72.75	70.08	60.75	75.08	73.82	54.53	55.44	69.62	41.78	63.76
SVM-comgfk	74.47	70.15	59.78	75.09	73.99	54.59	55.88	70.23	41.85	64.00
MI-rbf	42.25	73.69	75.53	66.75	77.51	54.43	33.50	23.57	34.92	53.57
MI-comgfk	80.25	74.99	78.79	67.41	77.82	71.68	49.96	50.79	53.79	67.28
ELM-rbf	70.63	66.44	66.83	63.45	69.73	51.23	49.76	49.83	33.50	57.93
DAELM-S(5)	72.66	75.72	61.30	86.29	53.45	59.40	31.16	66.85	44.39	61.25
DAELM-T(10)	55.80	70.70	59.94	86.55	52.94	55.54	73.23	75.32	42.22	63.58
MFKS(10)	80.79	80.64	86.75	79.14	80.69	36.19	68.30	63.04	37.10	68.07
BLS	44.62	89.29	64.72	77.01	73.42	42.70	66.53	62.62	43.97	62.76
DTBLS	78.67	96.36	74.60	85.23	83.20	81.53	58.67	56.19	63.10	75.28

hidden layer (ELM-rbf) from [44] has been a popular algorithm in recent years. Two algorithms called source DAELM (DAELM-S) and target DAELM (DAELM-T) which utilize a few labeled target samples has demonstrated the effectiveness in drift compensation. MFKS is a multi-task learning algorithm. BLS is a basic classifier compared with the proposed DTBLS.

The classification results of CC-PCA in Setting 1 and the results of ELM-rbf are cited from [23]. For SVM-rbf, SVM-gfk, SVM-comgfk, MI-rbf, and MI-comgfk, the results are extracted from [19]. The results of DAELM-S and DAELM-T are obtained via running the program exposed by the author. Results of MFKS in [20] are compared directly with the proposed framework in Setting 1.

For BLS and DTBLS, the former is trained on the labeled source data, and tested on the unlabeled target data, whereas the latter is trained on all data in a transductive way to directly induce domain-adaptive classifiers [34]. Following [34], under the experimental setup that there is no labeled data in target domain, it is impossible to automatically tune the optimal parameters for the target classifier using cross validation. Therefore, the two methods on the dataset are evaluated by empirically searching the parameter space for the optimal parameter settings, and the best results of each method are reported. For BLS, the input data are scaled into $[-1, 1]$, and the associated random parameters \mathbf{W}_{ei} , \mathbf{B}_{ei} , \mathbf{W}_{hj} , \mathbf{B}_{hj} are drawn from the standard uniform distribution on the interval $[-1, 1]$. For the generation of enhancement nodes, a sigmoid function is chosen as the activation function. In Setting 1, the structure is initialized with 10×100 feature nodes and 3000 enhancement nodes. The model regularization parameter is set to be 0.01. In Setting 2, the structure is initialized with 20×10 feature nodes and 100 enhancement nodes. The model regularization parameter is set to be 100. For DTBLS, by default, the structure is initialized with 10×100 feature nodes and 2000 enhancement nodes. The input features are scaled to lie in $[-1, 1]$ and the associated random parameters \mathbf{W}_{ei} , \mathbf{B}_{ei} , \mathbf{W}_{hj} , \mathbf{B}_{hj} are drawn from the standard uniform

distribution on the interval $[-1, 1]$. For the generation of enhancement nodes, a sigmoid function is chosen as the activation function. The base classifier generating pseudo labels of target data is trained by a simple neural network. There are four tunable parameters involved in the DTBLS. In the comparative study, we fix $\sigma = 0.001$, $\gamma_1 = 10$, $\gamma_2 = 0.1$ and $\lambda = 0.1$ for setting 1, and $\sigma = 0.01$, $\gamma_1 = 100$, $\gamma_2 = 10$ and $\lambda = 0.1$ for setting 2. The accuracy of the test data (unlabeled target data) is used as the assessment criteria. Besides, due to the existence of random weights in the BLS and DTBLS, we run 10 times for each set of parameters.

B. EXPERIMENTAL RESULTS

The performance of the proposed framework is compared with some other methods in Table 2 and Table 3 under Setting 1 and Setting 2, respectively. The digits after DAELM-S and DAELM-T are the number of labeled samples needed in target domain, respectively. Furthermore, for visualization of these results, we plot the corresponding line charts in Fig. 3 and Fig. 4.

From Table 2 and Fig. 3, we can reach the following observations:

(1) Except SVM-rbf, other machine learning algorithms perform much better than the classical CC-PCA. It indicates that machine learning methods are more suitable for coping with the problem of drift than traditional calibration methods.

(2) MI-comgfk achieves better results than SVM-comgfk. This demonstrates that manifold regularization can enhance the model performance significantly.

(3) ML-comgfk obtains an average accuracy of 67.28% and outperforms almost all baseline methods. It demonstrates that manifold regularization and combined GFK are effective in addressing domain adaptation issues.

(4) The proposed DTBLS achieves the highest average accuracy and the highest accuracies on Batch 3, 6, 7, 10. Compared with BLS, the DTBLS behaves much better except Batch 8 and the average accuracy of DTBLS is 12.52% higher than that of BLS. It can be drawn that distribution adaptation

TABLE 3. Comparisons of recognition accuracy(%) under the experimental setting 2.

Method	1→2	2→3	3→4	4→5	5→6	6→7	7→8	8→9	9→10	Average
SVM-rbf	74.36	87.83	90.06	56.35	42.52	83.53	91.84	62.98	22.64	68.01
SVM-gfk	72.75	74.02	77.83	63.91	70.31	77.59	78.57	86.23	15.76	68.56
SVM-comgfk	74.47	73.75	78.51	64.26	69.97	77.69	82.69	85.53	17.76	69.40
MI-rbf	42.25	58.51	75.78	29.10	53.22	69.17	55.10	37.94	12.44	48.17
MI-comgfk	80.25	98.55	84.89	89.85	75.53	91.17	61.22	95.53	39.56	79.62
ELM-rbf	70.63	40.44	64.16	64.37	72.70	80.75	88.20	67.00	22.00	63.36
DAELM-S(5)	72.66	69.99	72.61	79.54	52.93	87.18	91.36	56.66	29.05	68.00
DAELM-T(10)	55.80	71.83	56.58	86.35	52.79	55.41	71.46	75.32	42.21	63.08
BLS	73.87	82.07	73.04	44.16	74.43	76.88	87.76	74.53	39.88	69.62
DTBLS	88.75	97.65	79.88	67.01	75.34	90.44	95.10	68.09	54.47	79.64

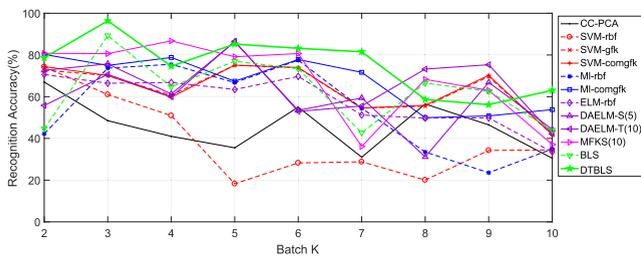


FIGURE 3. Performance of the classifiers under Setting 1.

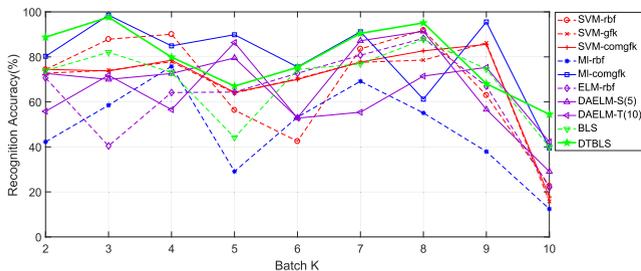


FIGURE 4. Performance of the classifiers under Setting 2.

and manifold regularization are quite effective in improving the recognition rates.

From Table 3 and Fig. 4, we have the following observation:

(1) Compared with the case of Setting 1, nearly all algorithms work better under the case of Setting 2. This may be explained by the general trend that the longer the interval between the two batches is, the more serious the drift will be.

(2) Except Batch 7→8, MI-comgfk outperforms SVM-comgfk. It implies that the manifold regularization has greatly improved the model performance.

(3) MI-comgfk obtains an average accuracy of 79.62% which is only second to the DTBLS and outperforms all baseline methods. It indicates that manifold regularization and combined GFK are highly effective in dealing with drift.

(4) The proposed DTBLS achieves the highest average accuracy and the highest accuracy in Batch 1→2, 7→8 and

9→10. In Batch 2→3, 5→6, 6→7, the recognition accuracies are next only to the highest values in all of methods respectively. Compared with BLS, the DTBLS performs much better except the Batch 8→9 and the average accuracy of DTBLS is 10.02% higher than that of BLS. It can also be deduced that distribution adaptation and manifold regularization play an important role in improving the recognition rates.

C. PARAMETER SENSITIVITY

In this section, parameter sensitivity analysis is conducted to show that the proposed DTBLS framework can achieve the optimal results over a wide range of parameter values. Limited to the space, the discussions will be carried out only on the tasks “Batch 1 → 2”, “Batch 2 → 3”, “Batch 5 → 6” and “Batch 7 → 8”.

1) NEAREST NEIGHBORS PARAMETER p

As shown in Fig. 5(a), we plot the classification accuracy on the above tasks w.r.t. different values of p . It can be seen that the curves are fairly flat in the interval [2, 100], which indicates a wide range for optimal parameter values. However, theoretically, a small p may lead to a limited capture of similar information between samples, whereas a large p may connect two samples that are not similar at all. Thus, in practice, it is usually to set p to 10.

2) MARGINAL DISTRIBUTION ADAPTATION REGULARIZATION γ_1

As shown in Fig. 5(b), we plot the classification accuracy w.r.t. different values of γ_1 . It can be seen that the interval [50, 400] is flat, which also shows a wide range for optimal parameter values. Theoretically, the larger γ_1 is, the more effective marginal distribution adaptation will be. However, the actual situation is that too large γ_1 may lead to degradation of model performance. That is probably because the weight of this term is too heavy that other terms are weakened.

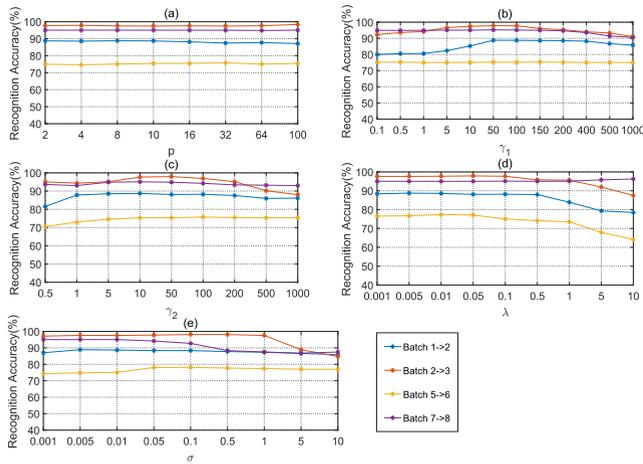


FIGURE 5. Parameter sensitivity study for DTBLS on selected tasks: (a) Nearest neighbors p . (b) Marginal distribution adaptation regularization γ_1 . (c) Conditional distribution adaptation regularization γ_2 . (d) Manifold regularization λ . (e) Model regularization σ .

3) CONDITIONAL DISTRIBUTION ADAPTATION REGULARIZATION γ_2

Similarly, we plot the classification accuracy w.r.t. different values of γ_2 in Fig. 5(c). It seems that the interval [5, 200] is a good choice. And we observe that for some tasks, when the value of γ_2 increases to a certain extent, the classification performance starts to decline. This may be due to the high inaccuracy of pseudo labels.

4) MANIFOLD REGULARIZATION λ

We plot the classification accuracy w.r.t. different values of λ in Fig. 5(d). The interval [0.001, 0.5] is a reasonable range. In theory, when the value of λ is too large, only manifold consistency is preserved while labeled information is discarded and degenerates into an unsupervised situation. The curves in the graph is consistent with the theory.

5) MODEL REGULARIZATION σ

We run the DTBLS with varying σ and plot the classification accuracy w.r.t. different values of σ in Fig. 5(e). From the line chart, we can see that the variation is steady in the the interval [0.001, 0.1]. And as the value increases, the performance has a sharp decline for most tasks. It is in accordance with the theory that a large model regularization parameter can lead to a oversimplified model which can not use the label information effectively.

D. TIME COMPLEXITY

We empirically check the time complexity of DTBLS and almost all the other algorithms by running them in Setting 1 and 2, and show the corresponding results in Table 4 and Table 5, respectively. The environment is an Intel Core i7-7700 CPU with 16 GB memory. It can be seen that except

¹CC-PCA is a data correction method which has to work with a classifier in practical use.

TABLE 4. Time complexity of DTBLS and the baseline methods in setting 1.

Method	Running Time (s)	Method	Running Time (s)
SVM-rbf	0.85	ELM-rbf	0.19
SVM-gfk	4.41	DAELM-S	0.77
SVM-comgfk	4.99	DAELM-T	17.74
MI-rbf	99.19	BLS	0.63
MI-comgfk	709.57	DTBLS	18.17
MFKS	>1000	CC-PCA ¹	-

TABLE 5. Time complexity of DTBLS and the baseline methods in setting 2.

Method	Running Time (s)	Method	Running Time (s)
SVM-rbf	1.7	ELM-rbf	0.21
SVM-gfk	22.94	DAELM-S	1.29
SVM-comgfk	24.44	DAELM-T	20.12
MI-rbf	241.98	BLS	0.85
MI-comgfk	756.22	DTBLS	26.18
MFKS	>1000	CC-PCA ¹	-

its superiority in classification accuracy, DTBLS also achieve comparable time complexity as the baseline methods.

VII. CONCLUSION

In this paper, a novel unified framework called DTBLS has been proposed for drift compensation. By taking structural risk minimization, marginal distribution adaptation, conditional distribution adaptation and manifold regularization into account simultaneously, the DTBLS not only maintains the merits of BLS including the feature mapping with randomly generated input weights and bias, the analytically determined solution, and good generalization, but also possesses the capability of transferring. Experiments on the public Sensor Drift Dataset in UCI Machine Learning Repository have shown the effectiveness of the proposed framework and superiority to other typical methods in classification. Moreover, there is no need for auxiliary samples in our approach.

In addition, since the drift phenomenon could be considered as the variation of data distribution in the feature space, the proposed DTBLS can be applied to other similar problems. For example, in computer vision, object classifiers optimized on one benchmark dataset often exhibit significant degradation in recognition accuracy when evaluated on another one. Similarly, in text analysis, one might want to train a document classifier on one corpus and apply to another one. As the two corpora have mismatched distributions of words and their usages, the trained classifier would not perform well. Obviously, the proposed framework can be easily applied to the above situations.

ACKNOWLEDGMENT

The authors would like to thank Alexander Vergara, San Diego and Ramon Huerta for the public gas sensor drift data they provide.

REFERENCES

- [1] S.-T. Han, H. Peng, Q. Sun, S. Venkatesh, K.-S. Chung, S. C. Lau, Y. Zhou, and R. Val, "An overview of the development of flexible sensors," *Adv. Mater.*, vol. 29, no. 33, 2017, Art. no. 1700375.
- [2] Z. Ye, M. Zhang, Z. Guo, L. Miao, S.-T. Han, Z. Wang, X. Zhang, H. Zhang, and Z. Peng, "Recent advances in black phosphorus-based photonics, electronics, sensors and energy devices," *Mater. Horizons*, vol. 4, no. 6, pp. 997–1019, 2017.
- [3] S. Giannoukos, A. Agapiou, and S. Taylor, "Advances in chemical sensing technologies for VOCs in breath for security/threat assessment, illicit drug detection, and human trafficking activity," *J. Breath Res.*, vol. 12, no. 2, 2018, Art. no. 027106.
- [4] J. Chilo, J. Pelegri-Sebastian, M. Cupane, and T. Sogorb, "E-nose application to food industry production," *IEEE Instrum. Meas. Mag.*, vol. 19, no. 1, pp. 27–33, Jan. 2016.
- [5] S. Deshmukh, R. Bandyopadhyay, N. Bhattacharyya, R. A. Pandey, and A. Jana, "Application of electronic nose for industrial odors and gaseous emissions measurement and monitoring—an overview," *Talanta*, vol. 144, pp. 329–340, Nov. 2015.
- [6] J. E. Fitzgerald, E. T. Bui, N. M. Simon, and H. Fenniri, "Artificial nose technology: Status and prospects in diagnostics," *Trends Biotechnol.*, vol. 35, no. 1, pp. 33–42, 2017.
- [7] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sens. Actuators B, Chem.*, vols. 166–167, pp. 320–329, May 2012.
- [8] S. De Vito, G. Fattoruso, M. Pardo, F. Tortorella, and G. Di Francia, "Semi-supervised learning techniques in artificial olfaction: A novel approach to classification problems and drift counteraction," *IEEE Sensors J.*, vol. 12, no. 11, pp. 3215–3224, Nov. 2012.
- [9] C. Distanto, M. Leo, P. Siciliano, and K. C. Persaud, "On the study of feature extraction methods for an electronic nose," *Sens. Actuators B, Chem.*, vol. 87, no. 2, pp. 274–288, Dec. 2002.
- [10] L. Carmel, S. Levy, D. Lancet, and D. Harel, "A feature extraction method for chemical sensors in electronic noses," *Sens. Actuators B, Chem.*, vol. 93, nos. 1–3, pp. 67–76, 2003.
- [11] C. Distanto, P. Siciliano, and K. C. Persaud, "Dynamic cluster recognition with multiple self-organising maps," *Pattern Anal. Appl.*, vol. 5, no. 3, pp. 306–315, Aug. 2002.
- [12] S. Wold, H. Antti, F. Lindgren, and J. Öhman, "Orthogonal signal correction of near-infrared spectra," *Chemometrics Intell. Lab. Syst.*, vol. 44, nos. 1–2, pp. 175–185, 1998.
- [13] T. Artursson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, and M. Holmberg, "Drift correction for gas sensors using multivariate methods," *J. Chemometrics*, vol. 14, nos. 5–6, pp. 711–723, 2000.
- [14] H. Liu, R. Chu, J. Ran, and J. Xia, "Long-term drift compensation algorithms based on the kernel-orthogonal signal correction in electronic nose systems," in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Aug. 2015, pp. 1583–1587.
- [15] S. Di Carlo, M. Falasconi, E. Sánchez, A. Scionti, G. Squillero, and A. Tonda, "Increasing pattern recognition accuracy for chemical sensing by evolutionary based drift compensation," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1594–1603, 2011.
- [16] M. Zuppa, C. Distanto, P. Siciliano, and K. C. Persaud, "Drift counteraction with multiple self-organising maps for an electronic nose," *Sens. Actuators B, Chem.*, vol. 98, nos. 2–3, pp. 305–317, Mar. 2004.
- [17] M. Verma, S. Asmita, and K. K. Shukla, "A regularized ensemble of classifiers for sensor drift compensation," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1310–1318, Nov. 2015.
- [18] T. Wiezbicki and E. P. Ribeiro, "Sensor drift compensation using weighted neural networks," in *Proc. IEEE Conf. Evolving Adapt. Intell. Syst. (EAIS)*, May 2016, pp. 92–97.
- [19] Q. Liu, X. Li, M. Ye, S. S. Ge, and X. Du, "Drift compensation for electronic nose by semi-supervised domain adaptation," *IEEE Sensors J.*, vol. 14, no. 3, pp. 657–665, Mar. 2014.
- [20] L. Zhang, D. Zhang, X. Yin, and Y. Liu, "A novel semi-supervised learning approach in artificial olfaction for E-nose application," *IEEE Sensors J.*, vol. 16, no. 12, pp. 4919–4931, Jun. 2016.
- [21] K. Yan and D. Zhang, "Calibration transfer and drift compensation of e-noses via coupled task learning," *Sens. Actuators B, Chem.*, vol. 225, pp. 288–297, Mar. 2016.
- [22] K. Yan, D. Zhang, and Y. Xu, "Correcting instrumental variation and time-varying drift using parallel and serial multitask learning," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2306–2316, Jun. 2017.
- [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] 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.
- [25] 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.
- [26] K. Yan, L. Kou, and D. Zhang, "Learning domain-invariant subspace using domain features and independence maximization," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 288–299, Jan. 2018.
- [27] L. Zhang and D. Zhang, "Efficient solutions for discreteness, drift, and disturbance (3D) in electronic olfaction," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 2, pp. 242–254, Feb. 2016.
- [28] *UCI Machine Learning Repository*. Accessed: Aug. 4, 2019. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [29] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.
- [30] C. L. P. Chen and Z. L. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- [31] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 513–520.
- [32] B. Quanz and J. Huan, "Large margin transductive transfer learning," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1327–1336.
- [33] J. Q. Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*, 2009.
- [34] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.
- [35] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3263–3277, Dec. 2015.
- [36] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3208–3215.
- [37] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.
- [38] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A Hilbert space embedding for distributions," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Springer, 2007, pp. 13–31.
- [39] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
- [40] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 2399–2434, 2006.
- [41] J. Jin, Z. Liu, and C. L. P. Chen, "Discriminative graph regularized broad learning system for image recognition," *Sci. China Inf. Sci.*, vol. 61, no. 11, 2018, Art. no. 112209.
- [42] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2013.
- [43] U. Von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *Ann. Statist.*, vol. 36, no. 2, pp. 555–586, 2008.
- [44] D. A. P. Daniel, K. Thangavel, R. Manavalan, and R. S. C. Boss, "ELM-based ensemble classifier for gas sensor array drift dataset," in *Computational Intelligence, Cyber Security and Computational Models*. Springer, 2014, pp. 89–96.



BEI LIU received the M.Sc. degree in signal and information processing from Southwest University, Chongqing, China, in 2013. She is currently pursuing the Ph.D. degree with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing. Her research interests include electronic nose technology and machine learning algorithm.



FENGCHUN TIAN received the B.Sc., M.Sc., and Ph.D. degrees in radio engineering, biomedical instruments and engineering, and theoretical electric engineering from Chongqing University, Chongqing, China, in 1984, 1986, and 1996, respectively. Since 1984, he has been a Teacher with Chongqing University, where he has been a Professor, since 2001. Since 2007, he has also been an Adjunct Professor with the University of Guelph, Guelph, ON, Canada. He is currently the Director of the Key Laboratory of Chongqing for Bio-perception and Intelligent Information Processing. His research interests include artificial olfaction (electronic nose), image processing, biomedical, and modern signal processing technology.



SHUYA ZHANG received the bachelor's degree in electronic information engineering from Hunan University, China, in 2018. She is currently pursuing the M.Sc. degree from Chongqing University. Her research interests include electronic nose and machine learning algorithm.



XIAOPING ZENG received the B.E., M.S., and Ph.D. degrees in electrical engineering from Chongqing University, Chongqing, China, in 1982, 1987, and 1996, respectively. He was a Visiting Scholar with Ohio State University, Columbus, OH, USA, from 2002 to 2003. He is currently a Professor with the College of Communication Engineering, Chongqing University. His current research interests include mobile aeronautical communications, the Internet of Vehicles, and wireless local area networks.



LEILEI ZHAO received the B.S. degree from the Academy of Mathematics and Statistics, Zhengzhou University, Zhengzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing. His research interests include electronic nose and machine learning.

...