Hindawi
Complexity
Volume 2022, Article ID 2463746, 22 pages
https://doi.org/10.1155/2022/2463746

WILEY | Hindawi

Research Article

Discriminative Extreme Learning Machine with Cross-Domain Mean Approximation for Unsupervised Domain Adaptation

Shaofei Zang,[1] **Xinghai Li**,[1] **Jianwei Ma**,[1] **Yongyi Yan**,[1] **Jinfeng Lv**,[1] **and Yuan Wei**[2]

[1]*College of Information Engineering, Henan University of Science and Technology, Luoyang 471000, China*
[2]*College of Vehicle and Traffic Engineering, Henan University of Science and Technology, Luoyang 471000, China*

Correspondence should be addressed to Jianwei Ma; majianwei_haust@163.com

Received 3 March 2022; Revised 10 August 2022; Accepted 26 August 2022; Published 3 October 2022

Academic Editor: Fanlin Meng

Extreme Learning Machine (ELM) is widely used in various fields because of its fast training and high accuracy. However, it does not primarily work well for Domain Adaptation (DA) in which there are many annotated data from auxiliary domain and few even no annotated data in target domain. In this paper, we propose a new variant of ELM called Discriminative Extreme Learning Machine with Cross-Domain Mean Approximation (DELM-CDMA) for unsupervised domain adaptation. It introduces Cross-Domain Mean Approximation (CDMA) into the hidden layer of ELM to reduce distribution discrepancy between domains for domain bias elimination, which is conducive to train a high accuracy ELM on annotated data from auxiliary domains for target tasks. Linear Discriminative Analysis (LDA) is also adopted to improve the discrimination of learned model and obtain higher accuracy. Moreover, we further provide a Discriminative Kernel Extreme Learning Machine with Cross-Domain Mean Approximation (DKELM-CDMA) as the kernelization extension of DELM-CDMA. Some experiments are performed to investigate the proposed approach, and the result shows that DELM-CDMA and DKELM-CDMA could effectively extend ELM suitable for domain adaptation and outperform ELM and many other domain adaptation approaches.

1. Introduction

It is a challenge for mankind to extract or mine valuable information from massive data generated from mobile devices, Internet, and industrial sensors in the current society. Classifiers based on machine learning play an important role in data and information processing systems. Extreme Learning Machine (ELM) [1] attracts attention due to its faster learning and higher accuracy compared with k Nearest Neighbor (kNN) [2], Back-Propagating (BP) [3], Naive Bayes (NB) [4], Support Vector Machine (SVM) [5], and Decision Tree (DT) [6], and has been widely promoted in many fields including image classification [7], traffic system [8], COVID-19 detection [9], fault diagnosis [10, 11], hyperspectral remote sensing images [12, 13], industrial sensors [14], facial expression recognition [15], and brain-computer interface (BCI) [16, 17] etc.

Due to its fast learning and strong generalization capability, Extreme Learning Machine attracted more attention. It randomly selects the input weights and the biases of hidden layer neurons without training data, and only obtains the optimal solution of output weight by minimizing the training error and the norm of output weights simultaneously [18]. Since the hidden layer parameters are randomly initialized and the output weights are solved by finding a least squares solution, the training time of the model is greatly reduced [1]. Recently, many variants of ELM have emerged for improving its performance and have been divided into three parts: supervised ELMs, semi-supervised, and unsupervised ones. Supervised ELMs need numerous labeled data to ensure its high performance, such as Kernel Extreme Learning Machine (KELM) [19], Weighted Extreme Learning Machine (WELM) [20], Twin Extreme Machines (TELM) [21], and Adaptive Regularized Extreme Learning Machine (A-RELM) [22]. Semi-supervised ELM usually requires unlabeled data together with labeled data to train models well, including Laplacian Twin Extreme Learning Machine (Lap-TELM) [23], Semi-Supervised Extreme Learning Machine (SS-ELM) [24], Robust Semi-Supervised Extreme Learning Machine (RSS-ELM) [25], and

Adaptive Safe Semi-Supervised Extreme Learning Machine (AdSafe-SSELM) [26]. In some other cases where no labeled data are available, some Unsupervised ELM (USELM) algorithms are proposed for clustering, dimension reduction, or data representation, such as Unsupervised Extreme Learning Machine (USELM) [24], Extreme Learning Machine as an Auto-Encoder (ELM-AE) [27], Enhanced Unsupervised Extreme Learning Machine (EUELM) [28], and Unsupervised Feature Selection based Extreme Learning Machine (UFSELM) [29]. Moreover, as deep learning has been successful in many fields, deep ELMs are also developed to extract more abstract and expressive features, such as Kernel-based Multi-Layer Extreme Learning Machine ELM (ML-KELM) [30], Hierarchical-ELM (H-ELM) [31], DS-ELM (a deep and stable extreme learning machine) [32], and Deep Residual Compensation Extreme Learning Machine (DRC-ELM) [33]. Although the above algorithms expand the application of ELM in various scenarios, they perform not well when training data are not related to the testing data.

In application, it is difficult to collect a large amount of data which keeps consistent with test data because of condition change, data noise, view variety, and so on. To address this issue, Domain Adaptation (DA) [34–36], as an important transfer learning technology, can remedy this shortcoming of ELM, in which abundant labeled data from other domains (source domain) is applied to help the current domain (target domain) and there is inconsistency in distribution between domains. Consequently, L. Zhang and D. Zhang [37] provided two ELMs with domain adaptation including Source Domain Adaptation Extreme Learning Machine (DAELM-S) and Target Domain Adaptation Extreme Learning Machine (DAELM-T). They can improve the generalization ability of ELM in multiple domains. Similar to DAELM, Zhao and Chen [38] also developed One Stage-Transfer-Learning ELM (OSTL-ELM) and Two-Stage-Transfer-Learning ELM (TSTL-ELM) to handle the problem of insufficient labeled data from target domains in aero engine fault diagnosis tasks. With the help of subspace alignment and the weight approximation, Zang et al. [39] put forward Transfer Extreme Learning Machine with Output Weight Alignment (TELM-OWA) to handle domain adaptation. TL-ELM (Transfer Learning-based ELM) was proposed by Li et al. [40], in which an output weight from two domains were forced to be close to each other for knowledge transfer. In the methods mentioned above, it is a requirement that there are few labeled data in target domains. However, annotating data are costly, laborious, and time-consuming. Consequently, some unsupervised models appear for unsupervised domain adaptation. Chen et al. [41] presented an transfer ELM, in which output weight alignment was applied to reduce domain bias and L2, 1-norm was imposed on output weight to enhance feature selection. For minimizing the distribution discrepancy between source and target domains, Li et al. [42], Chen et al. [43], and Zang et al. [44] utilized Maximum Mean Discrepancy (MMD) [45] to promote knowledge transfer in their respective models. Due to the insufficient target sample labels, the performance of unsupervised models is usually lower than that of supervised models, but it is hard to collect labeled target samples.

In this paper, we propose a novel ELM called Discriminative Extreme Learning Machine with Cross-Domain Mean Approximation (DELM-CDMA) for unsupervised domain adaptation. It introduces Cross-Domain Mean Approximation (CDMA) [36] into the objective function of ELM to jointly adapt the marginal and conditional distributions discrepancy between the output of hidden layers from source and target samples. CDMA could enhance the ability of transferring knowledge across domains. Moreover, to improve the discrimination, Linear Discriminant Analysis (LDA) [46] is also added into the objective function. It separates the samples with different categories and clusters the samples with the same category, which can improve the accuracy of ELM. Finally, we solve the designed objective function and obtain an optimal ELM model for domain adaptation. Moreover, we further present a Discriminative Kernel Extreme Learning Machine with Cross-Domain Mean Approximation (DKELM-CDMA) which not only kernelize DELM-CDMA suitable for nonlinear data but also eliminate the sensitivity of its accuracy to parameter initialization. DELM-CDMA is illustrated in Figure 1. Extensive experiments are carried out to verify the effectiveness and superiority of DELM-CDMA and DKELM-CDMA for unsupervised domain adaptation.

Contributions of this paper are as follows:

(1) Joint distribution adaptation based on CDMA measure is introduced into ELM to narrow distribution differences between domains. We apply LDA to boost the class separability of DELM-CDMA and thus improves its accuracy.

(2) As a kernel version of DELM-CDMA, it is proposed to enhance the robustness for initialization of the parameters of hidden layers and the adaptability to nonlinear data.

(3) We efficiently solve a least-squares problem for the objective function, and obtain an optimal ELM for unsupervised domain adaptation. Moreover, classification experiments on object recognition and text data sets are performed to investigate our approaches, and the results show that DELM-CDMA and DKELM-CDMA can efficiently achieve the capability of transferring knowledge across domains.

The rest of this paper is summarized as follows: We describe domain adaptation, CDMA and ELM very simply in Section 2. Then, we represent DELM-CDMA and DKELM-CDMA in Section 3. Next, the experiment is performed in Section 4 to verify the validity of DELM-CDMA. Finally, Section 5 is the conclusion of this paper.

2. Related Work

Since our method mainly extends ELM to handle domain adaptation, we will briefly introduce domain adaptation, CDMA and ELM in this section.

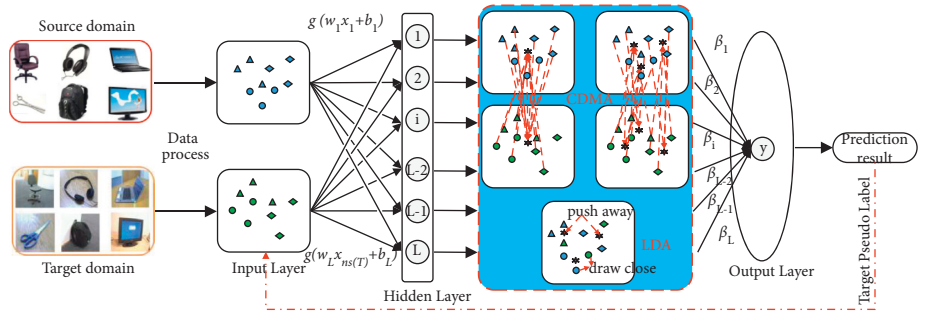


Figure 1: An illustration of DELM-CDMA. (1) Cross-domain mean approximation (CDMA) is adopted to reduce the domain shift. (2) Linear discriminant analysis (LDA) is added into object function to enhance the discrimination ability of DELM-CDMA. (3) The objective function is solved efficiently.

2.1. Domain Adaptation. Transfer learning [34, 36, 47, 48], as one of the important research branches of machine learning, helps target tasks to learn a high-quality model with the knowledge from source domains which have rich labeled data/samples but different distributions with target domains. Domain adaptation [36] is a hot topic in transfer learning and aims to adjust the distribution between domains in order to eliminate domain shift. In recent years, a large number of approaches have emerged for domain adaptation and have been roughly divided into five categories according to what is adapted, that is, instance adaptation, feature adaptation, parameter adaptation, deep network adaptation, and adversarial domain adaptation.

(1) *Instance Adaptation.* Most instance-adaptation methods attempt to find a strategy in which each instance is assigned a weight to balance the distribution difference between domains. Li et al. [49] proposed Prediction Reweighting for Domain Adaptation (PRDA) which first reweighed classifier training on source domain and then adopted manifold regularization to diffuse the labels from high confidence samples to low confidence ones. TrAdaBoost [50], as a classic domain adaptation method, weighs each sample in source and target domains by a dynamic mechanism to minimize the distribution discrepancy between domains. Moreover, bad samples and good samples from source domains are distinguished by weight, which promotes the knowledge transferring across domains.

(2) *Feature Adaptation.* This method usually finds shared feature subspace to reduce the domain distribution bias and transfer knowledge across domains. Many technologies such as K-L discrepancy [51], Bregman divergence [52], MMD [53], and Subspace alignment [54–56] are used to minimize distribution discrepancy and seek optimal common features between domains. Transfer Component Analysis (TCA) [57] and Joint Distribution Adaptation (JDA) [53] solve a projection matrix minimizing MMD measurement from source and target domains and extract shared features to obtain effective target learner training on source domain.

(3) *Parameter Adaptation.* Those approaches do not transfer shared knowledge in instance or feature levels but find optimal shared parameters of learner across domains. The DAELM-S(T), TELM-OWA, TL-ELM, and so on mentioned above belong to this method.

(4) *Deep Network Adaptation.* This method combines the traditional domain adaptation technology with the deep learning model. It not only uses the former to reduce the distribution differences between domains but also applies the deep network structure of the latter to extract the high-level respective and semantic features across domains, which enables more efficient knowledge transfer. Domain Adaptation Network (DAN) [58], Joint Adaptation Network (JAN) [59], and Residual Transfer Network (RTN) [60] all utilize MMD to reduce the distribution differences between different domains and the data set bias.

(5) *Adversarial Domain Adaptation.* Based on the generative adversarial network, this method produces target sample/data using domain-invariant feature generators, and subsequently applies the domain discriminator training on source samples to find the weakness of generated samples for generator adaptation, which could reduce the domain gap. As an unsupervised domain adaptation method based on deep feed-forward architecture, Domain Adversarial Neural Network (DANN) [61] introduces adversarial learning for domain adaptation. It simultaneously learns classifiers, feature extractors, and domain discriminators and obtains domain-invariant feature representation by minimizing classifier errors and maximizing the discriminant errors. Collaborative and Adversarial Network (CAN) [62] first uses collaborative learning to distinguish the domain-informative features of the sample belonging to source domain or target domain, and then utilizes adversarial learning to learn difficult-distinguish domain-uninformative features. Finally, domain-invariant features could be extracted.

In the methods mentioned in Section 2.1, the instance adaptation method has higher knowledge transferring efficiency, but it usually requires high-quality samples to ensure its excellent performance. Feature adaptation method has higher flexibility and wider application, but finding the shared feature with better generalization is a challenge. Due

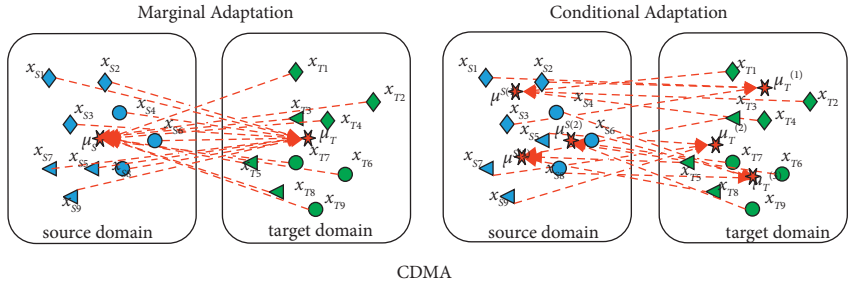


Figure 2: An illustration of CDMA.

to the lack of direct utilization of sample and feature information, the parameter adaptation method usually has lower efficiency of knowledge transfer compared with the two former ones, but its domain knowledge utilization is more diversified. Deep network adaptation and adversarial domain adaptation have good performance owing to higher-level feature extraction. They strictly belong to feature adaptation method because obtaining good features is their primary task. However, Deep network adaptation usually requires plenty of labeled samples, more consuming-time and memory, simultaneously finding efficient feature extractors and discriminators is also a challenge for adversarial domain adaptation. In this paper, DELM-CDMA belongs to parameter adaptation method, and DKELM-CDMA is shallow feature adaptation method.

2.2. Cross-Domain Mean Approximation (CDMA). To address the problem that MMD ignores sample individual difference, and thus insufficiently mining local information of data, Zang et al. [36] designed Cross-Domain Mean Approximation (CDMA), as shown in Figure 2. Given source domain $\{x_{Si}, y_{Si}\}_{i=1}^{n_S}$ with n_S sample x_{Si} and according to its label y_{Si}, and target domain $\{x_{Tj}, y_{Tj}\}_{j=1}^{n_T}$ including n_T sample x_{Tj} and according to its label y_{Tj}.The distribution discrepancy between source and target domains based on CDMA can be defined as follows:

$$L_{\mathrm{CDMA}}(D_S, D_T) = \sum_{i=1}^{n_S+n_T} \left\| x_i - \mu \right\|_2^2, \tag{1}$$

where $\mu = \begin{matrix} \mu_T, \text{if } x_i \in D_S \\ \mu_S, \text{if } x_i \in D_T \end{matrix}$, $\mu_{T(S)}$ is the mean vector of the target (source) domain samples. Moreover, to adapt the marginal and conditional distribution together, label information of sample is added into CDMA, and (1) is modified as given in the following equation:

$$L_{\mathrm{CDMA}}(D_S, D_T) = \sum_{i=1}^{n_S+n_T} \left\| x_i - \mu \right\|_2^2 + \sum_{c=1}^{C} \sum_{i=1}^{n_{Si}^{(c)}+n_{Ti}^{(c)}} \left\| x_i^{(c)} - \mu_i^{(c)} \right\|_2^2, \tag{2}$$

where $\mu^{(c)} = \begin{matrix} \mu_T^{(c)}, \text{if } x_i^{(c)} \in D_S^{(c)} \\ \mu_S^{(c)}, \text{if } x_i^{(c)} \in D_T^{(c)} \end{matrix}$, $\mu_{T(S)}^{(c)}$ is the mean vector of the target (source) domain $D_{T(S)}^{(c)}$ with c category.

ELM is a single-layer neural network with fast learning and high accuracy, without tuning input weight and bias. It efficiently obtains optimal output weight by solving a least squares problem. Given a data set $\{(x_i, y_i)\}_{i=1}^{N}$ including N samples x_i with label y_i. A classic ELM network is given:

CDMA is an efficient evaluation metric of the distribution discrepancy between domains, which is adopted in our DELM-CDMA and DKELM-CDMA to reduce the marginal and conditional distribution discrepancy. Different from Joint Distribution Adaptation based on Cross Domain Mean Approximation (JDA-CDMA) [36] which uses CDMA to construct the objective function and incorporates subspace learning for extracting shared feature between the source and target domains, DELM-CDMA applies CDMA into the hidden layer of ELM for domain bias elimination and enhancing the generalization of the parameters of learned ELM and DKELM-CDMA applies CDMA together with reconstruction error for shared feature extraction in KELM framework.

2.3. Extreme Learning Machine (ELM). ELM is a single-layer neural network with fast learning and high accuracy, without tuning input weight and bias. It efficiently obtains optimal output weight by solving a least squares problem. Given a data set $\{(x_i, y_i)\}_{i=1}^{N}$ including N samples x_i with label y_i. A classic ELM network is given by the following equation:

$$y_i^T = \sum_{j=1}^{L} \beta_j^T g\left(w_j \mathbf{x}_i + b_j\right), \tag{3}$$

where x_i is an input sample, w_j and b_j are the input layer weight and bias, respectively, $g(x)$ represents the nonlinear activation function, L represents the number of nodes in the hidden layer, and β_j represents the hidden layer output weight. We can obtain an optimal β^* by solving the objective function as given by the following equation:

$$\begin{aligned} &\min_{\beta} : \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\sum_{i=1}^{N}\left\|\varepsilon_i\right\|^2 \\ &\text{s.t.} g\left(\mathbf{x}_i\right)\boldsymbol{\beta} = \mathbf{y}_i^{\mathrm{T}} - \varepsilon_i^{\mathrm{T}}, i = 1, \ldots, N, \end{aligned} \tag{4}$$

where $\|\beta\|^2$ is applied to avoid model overfitting, ε_i is the error vector corresponding to the empirical risk of sample x_i with the tradeoff parameter λ. The constraint is removed and equation (4) changes into the following equation:

$$L_{ELM} = \min_{\boldsymbol{\beta}} : \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\|\mathbf{Y} - \mathbf{H}\boldsymbol{\beta}\|^2, \tag{5}$$

where $\mathbf{H} = [g(\mathbf{x}_i)^T, \ldots, g(\mathbf{x}_N)^T]^T$, $\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}$, and $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$.

The objective function is considered as a ridge regression or a regular least square problem. We set $\partial L_{ELM}/\partial\beta = 0$ and the optimal solution is got according to [18] as given by the following equation:

$$\boldsymbol{\beta}^* = \begin{cases} \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \dfrac{\mathbf{I}_L}{\lambda}\right)\mathbf{H}^{\mathrm{T}}\mathbf{Y}, & N > L, \\ \mathbf{H}^{\mathrm{T}}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \dfrac{\mathbf{I}_N}{\lambda}\right)\mathbf{Y}, & N \le L. \end{cases} \tag{6}$$

Then, the prediction result of test sample x_{Te} can be determined by the following equation:

$$\mathbf{y}_{Te} = \begin{cases} \operatorname{sign}\left(\mathbf{h}_{Te}^{\mathrm{T}}\boldsymbol{\beta}^*\right), & \text{for binary classification,} \\ \operatorname{argmax}\left(\mathbf{h}_{Te}^{\mathrm{T}}\boldsymbol{\beta}^*\right), & \text{for multi} - \text{classification,} \end{cases} \tag{7}$$

where $\mathbf{h}_{Te} = g(\mathbf{x}_{Te})$.

3. Proposed Methodology

In recent decades, ELM has been better than traditional classifiers of machine learning in speed and accuracy because of its ability in the generalization and global approximation. Thus, many variants of ELM appear for classification and regression in machine learning and pattern recognition to improve it in both theory and application. Nevertheless, ELM will be unsuccessful in the case of insufficient labeled samples [37]. In response to this issue, we develop Discriminative Extreme Learning Machine with Cross-Domain Mean Approximation (DELM-CDMA) for unsupervised domain adaptation, which use CDMA to narrow the marginal and conditional distributions between domains and adopt LDA to enhance category separability. Next, we will introduce it in detail in this Section 3.1.

3.1. Objective Function of DELM-CDMA. In unsupervised domain adaptation, given two different but related data sets: source domain $\mathbf{D}_S = \left\{\left(\mathbf{x}_{s(i)}, \mathbf{y}_{s(i)}\right)\right\}_{i=1}^{n_S}$ owning n_S sample x_{Si} with label y_{Si}, and target domain $\mathbf{D}_T = \left\{x_{Tj}\right\}_{j=1}^{n_T}$ including n_T unlabeled sample x_{Tj}. We hope DELM-CDMA training on $\mathbf{D}_S$ perform well on $\mathbf{D}_T$.

First, we transform $\mathbf{D}_S$ and $\mathbf{D}_T$ to $\mathbf{H}_S$ and $\mathbf{H}_T$ using activation function $g(x)$, and then construct the objective function of DELM-CDMA as given by the following equation:

$$L_{\mathrm{DELM-CDMA}} = L_{\mathrm{ELM}} + \alpha_1 L_{\mathrm{CDMA}} + \alpha_2 L_{\mathrm{LDA}}. \tag{8}$$

From equation (8), we can see that there are three parts in the objective function of DELM-CDMA. L_{ELM} represents the loss function of regularized ELM, L_{CDMA} denotes marginal or conditional CDMA measure between source and target domains, and L_{LDA} is LDA term. α_1 and α_2 are tradeoff parameters of L_{CDMA} and L_{LDA}, respectively.

3.1.1. Loss Function of Regularized ELM. In this paper, we carry out unsupervised domain adaptation task in which no labeled samples appear on target domain, so we can obtain a Regularized ELM on source domain as given by the following equation:

$$L_{ELM} = \min_{\boldsymbol{\beta}} : \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda_1}{2}\left\|\mathbf{H}_S\boldsymbol{\beta} - \mathbf{Y}_S\right\|^2. \tag{9}$$

In equation (9), the first term is a regularizer to avoid over-fitting, and the second term is classification error of samples from source domain. λ_1 is a parameter that balances the two terms.

In the iterative refinement process of labels, when the pseudo labels of the target samples $\hat{\mathbf{Y}}_{\mathbf{T}}$ are obtained, we could add the classification error from target domain, and equation (9) becomes equation (10).

$$\begin{aligned} L_{ELM} &= \min_{\boldsymbol{\beta}} : \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda_1}{2}\left(\left\|\mathbf{H}_S\boldsymbol{\beta} - \mathbf{Y}_S\right\|^2 + \left\|\mathbf{H}_T\boldsymbol{\beta} - \hat{\mathbf{Y}}_{\mathbf{T}}\right\|^2\right) \\ &= \min_{\boldsymbol{\beta}} : \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda_1}{2}\left(\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|^2\right), \end{aligned} \tag{10}$$

where $\mathbf{H} = \begin{pmatrix} \mathbf{H}_S \\ \mathbf{H}_T \end{pmatrix}$ and $\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_S \\ \hat{\mathbf{Y}}_{\mathbf{T}} \end{pmatrix}$.

3.1.2. Cross-Domain Mean Approximation Measure. In unsupervised domain adaptation, since source domain and target domain have distribution discrepancy, the prediction error of model training on $\mathbf{D}_S$ become increases on target domain. CDMA is an effective strategy for evaluation of inter-domain distribution differences. We apply it to measure the distribution discrepancy of output-layer data of ELM training on source and target domains, and compute it as given by the following equation:

$$L_{CDMA} = \min_{\boldsymbol{\beta}} : \left\{\frac{1}{2}\left(\left\|\mathbf{H}_T\boldsymbol{\beta} - \mathbf{H}_{\mathrm{S\_av}}\boldsymbol{\beta}\right\|^2 + \left\|\mathbf{H}_S\boldsymbol{\beta} - \mathbf{H}_{\mathrm{T\_av}}\boldsymbol{\beta}\right\|^2\right) + \frac{1}{2}\sum_{c=1}^{C}\left(\left\|\mathbf{H}_T^{(c)}\boldsymbol{\beta} - \mathbf{H}_{\mathrm{S\_av}}^{(c)}\boldsymbol{\beta}\right\|^2 + \left\|\mathbf{H}_S^{(c)}\boldsymbol{\beta} - \mathbf{H}_{\mathrm{T\_av}}^{(c)}\boldsymbol{\beta}\right\|^2\right)\right\}. \tag{11}$$

In equation (11), the first term measures the marginal distribution discrepancy, and the second term measures the conditional distribution discrepancy.

3.1.3. Linear Discriminant Analysis on Output Layer. To further improve the discrimination of ELM, we add Linear Discriminant Analysis (LDA) into it. LDA forces the samples with the same class closer and the ones with different class farther so as to enhance ELM category separability in unsupervised domain adaptation. We apply LDA on data from the output layer, in which $\mathbf{S}_b$ and $\mathbf{S}_w$, respectively, denote the between-class scatter and the within-class scatter and could been computed as given by the following equation:

$$\mathbf{S}_b = \sum_{c=1}^{C} n^{(c)} \left(\boldsymbol{\beta}\mathbf{H}^{(c)} - \boldsymbol{\beta}\mathbf{H}_{\text{av}}\right)^T \left(\boldsymbol{\beta}\mathbf{H}^{(c)} - \boldsymbol{\beta}\mathbf{H}_{\text{av}}\right), \quad (12)$$

$$\mathbf{S}_w = \sum_{c=1}^{C} \sum_{i=1}^{n^{(c)}} \left(\boldsymbol{\beta}\mathbf{H}^{(c)} - \boldsymbol{\beta}\mathbf{H}_{\text{av}}^{(c)}\right)^{\text{T}} \left(\boldsymbol{\beta}\mathbf{H}^{(c)} - \boldsymbol{\beta}\mathbf{H}_{\text{av}}^{(c)}\right). \quad (13)$$

In equations (12) and (13), $\mathbf{H}_{\text{av}} = \begin{pmatrix} \mathbf{H}_{\text{T\_av}} \\ \mathbf{H}_{\text{S\_av}} \end{pmatrix}$, $\mathbf{H}_{\text{av}}^{(c)} = \begin{pmatrix} \mathbf{H}_{\text{T\_av}}^{(c)} \\ \mathbf{H}_{\text{S\_av}}^{(c)} \end{pmatrix}$, $\mathbf{H}^{(c)}$ represents samples with c category in $\mathbf{H}_S$ and $\mathbf{H}_T$. $\mathbf{H}_{\text{S\_av}}$ and $\mathbf{H}_{\text{T\_av}}$ are mean vectors of $\mathbf{H}_S$ and $\mathbf{H}_T$, respectively. $H_{T(S)\_av}^{(c)}$ is the mean vector of the target (source) domain $H_{T(S)}^{(c)}$ with c category. Then, we construct the loss of LDA on the output layer as given by the following equation:

$$L_{LDA} = \min_{\boldsymbol{\beta}} : \frac{1}{2} \left(\mathbf{S}_w - \mathbf{S}_b\right). \quad (14)$$

3.1.4. Total Objective Function. To improve the ability of knowledge transferring and discrimination of ELM, DELM-CDMA joins CDMA and LDA to ELM. Therefore, we bring equations (9) or (10), (11) and (14) into (8), and then the objective function of DELM-CDMA is established:

$$\begin{aligned} L_{\text{DELM-CDMA}} &= L_{\text{ELM}} + \alpha_1 L_{\text{CDMA}} + \alpha_2 L_{\text{LDA}} \\ &= \min_{\boldsymbol{\beta}} : \begin{pmatrix} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda_1}{2}\|\mathbf{H}\boldsymbol{\beta} - Y\|^2 \\ + \frac{\alpha_1}{2}\left(\left\|\mathbf{H}\boldsymbol{\beta} - \mathbf{H}_{\text{av}}\boldsymbol{\beta}\right\|^2 + \sum_{c=1}^{C}\left(\left\|\mathbf{H}^{(c)}\boldsymbol{\beta} - \mathbf{H}_{\text{av}}^{(c)}\boldsymbol{\beta}\right\|^2\right)\right) \\ + \frac{\alpha_2}{2}\boldsymbol{\beta}^{\text{T}}\begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{H}^{(c)} - \mathbf{H}_{\text{av}}^{(c)}\right)^{\text{T}}\left(\mathbf{H}^{(c)} - \mathbf{H}_{\text{av}}^{(c)}\right) \\ - \sum_{c=1}^{C}\left(\mathbf{H}_{\text{av}}^{(c)} - \mathbf{H}_{\text{av}}\right)^{\text{T}}\left(\mathbf{H}_{\text{av}}^{(c)} - \mathbf{H}_{\text{av}}\right) \end{pmatrix}\boldsymbol{\beta} \end{pmatrix}. \end{aligned} \quad (15)$$

3.2. Model Learning. To solve optimal output weight, we let $\partial L_{DELM-CDMA}/\partial\beta = 0$ and yield

$$\begin{pmatrix} \boldsymbol{\beta} \\ -\lambda_1\mathbf{H}^{\text{T}}\left(\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\right) \\ + \frac{\alpha_1}{2}\begin{pmatrix} \left(\mathbf{H} - \mathbf{H}_{\text{av}}\right)^{\text{T}}\left(\mathbf{H} - \mathbf{H}_{\text{av}}\right) \\ + \sum_{c=1}^{C}\left(\mathbf{H}^{(c)} - \mathbf{H}_{\text{av}}^{(c)}\right)^{\text{T}}\left(\mathbf{H}^{(c)} - \mathbf{H}_{\text{av}}^{(c)}\right) \end{pmatrix} \\ + \alpha_2\begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{H}^{(c)} - \mathbf{H}_{\text{av}}^{(c)}\right)^{\text{T}}\left(\mathbf{H}^{(c)} - \mathbf{H}_{\text{av}}^{(c)}\right) \\ - \sum_{c=1}^{C}\left(\mathbf{H}_{\text{av}}^{(c)} - \mathbf{H}_{\text{av}}\right)^{\text{T}}\left(\mathbf{H}_{\text{av}}^{(c)} - \mathbf{H}_{\text{av}}\right) \end{pmatrix}\boldsymbol{\beta} \end{pmatrix} = 0 \quad (16)$$

According to [18], equation (16) has a closed form solution:

$$\boldsymbol{\beta}^* = \begin{cases} \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \alpha_1 \begin{pmatrix} (\mathbf{H}-\mathbf{H}_{\mathrm{av}})^{\mathrm{T}}(\mathbf{H}-\mathbf{H}_{\mathrm{av}}) \\ +\sum_{c=1}^{C}\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}}\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right) \end{pmatrix} + \alpha_2 \begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}}\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right) \\ -\sum_{c=1}^{C}\left(\mathbf{H}_{\mathrm{av}}^{(c)}-\mathbf{H}_{\mathrm{av}}\right)^{\mathrm{T}}\left(\mathbf{H}_{\mathrm{av}}^{(c)}-\mathbf{H}_{\mathrm{av}}\right) \end{pmatrix} + \frac{\mathbf{I}_{L\times L}}{\lambda_1}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Y}, & n_S + n_T > L, \\ \mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{H}^{\mathrm{T}} + \alpha_1 \begin{pmatrix} (\mathbf{H}-\mathbf{H}_{\mathrm{av}})(\mathbf{H}-\mathbf{H}_{\mathrm{av}})^{\mathrm{T}} \\ +\sum_{c=1}^{C}\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right)\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}} \end{pmatrix} + \alpha_2 \begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right)\left(\mathbf{H}^{(c)}-\mathbf{H}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}} \\ -\sum_{c=1}^{C}\left(\mathbf{H}_{\mathrm{av}}^{(c)}-\mathbf{H}_{\mathrm{av}}\right)\left(\mathbf{H}_{\mathrm{av}}^{(c)}-\mathbf{H}_{\mathrm{av}}\right)^{\mathrm{T}} \end{pmatrix} + \frac{\mathbf{I}_{(n_S+n_T)\times(n_S+n_T)}}{\lambda_1}\right)^{-1}\mathbf{Y}, & (n_S + n_T) \le L. \end{cases} \tag{17}$$

After optimal β^* is obtained, the test samples are classified by equation (7). A complete classification procedure of DELM-CDMA is summarized in Algorithm 1.

3.3. Kernelization. Although DELM-CDMA can improve the performance of ELM with fewer labeled samples or lack of conditions, it has two shortcomings: (1) The random initialization of hidden layer parameters will cause performance instability. (2) Inability to enhance the high-dimensional separability of nonlinear data. Therefore, we present a kernel version of DELM-CDMA named Discriminative Kernel Extreme Learning Machine with Cross-Domain Mean Approximation (DKELM-CDMA) to address above problems. DKELM-CDMA has two stages: feature extraction and classification.

3.3.1. Feature Extraction. At this stage, we let $\mathbf{K} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$, $K_{(\mathrm{ij})} = g(x_i)g(x_j)^{\mathrm{T}} = \Omega(x_i, x_j)$, $\Omega(x_i, x_j)$ is a kernel function such as sigmoid kernel, linear kernel, and radial basis kernel. The objective function of DKELM-CDMA is given by the following equation:

$$\begin{aligned} L_{DKELM-CDMA} &= L_{KELM} + \alpha_{k1}L_{CDMA} + \alpha_{k2}L_{LDA} \\ &= \min_{\boldsymbol{\beta}_k} : \begin{pmatrix} \frac{1}{2}\|\boldsymbol{\beta}_k\|^2 + \frac{\lambda_2}{2}\|\mathbf{K}\boldsymbol{\beta}_k - \mathbf{X}\|^2 \\ +\frac{\alpha_{k1}}{2}\left(\|\mathbf{K}\boldsymbol{\beta}_k - \mathbf{K}_{\mathrm{av}}\boldsymbol{\beta}_k\|^2 + \sum_{c=1}^{C}\left(\|\mathbf{K}^{(c)}\boldsymbol{\beta}_k - \mathbf{K}_{\mathrm{av}}^{(c)}\boldsymbol{\beta}_k\|^2\right)\right) \\ +\frac{\alpha_{k2}}{2}\boldsymbol{\beta}_k^{\mathrm{T}}\begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}}\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right) \\ -\sum_{c=1}^{C}\left(\mathbf{K}_{\mathrm{av}}^{(c)}-\mathbf{K}_{\mathrm{av}}\right)^{\mathrm{T}}\left(\mathbf{K}_{\mathrm{av}}^{(c)}-\mathbf{K}_{\mathrm{av}}\right) \end{pmatrix}\boldsymbol{\beta}_k \end{pmatrix}, \end{aligned} \tag{18}$$

where $\mathbf{X} = \left\{\mathbf{x}_{(i)}\right\}_{i=1}^{n_S+n_T}$ and $\mathbf{K}$ are its kernel maps of $\mathbf{K}_{\mathrm{av}} = \begin{pmatrix} \mathbf{K}_{\mathrm{T\_av}} \\ \mathbf{K}_{\mathrm{S\_av}} \end{pmatrix}$ and $\mathbf{K}_{\mathrm{av}}^{(c)} = \begin{pmatrix} \mathbf{K}_{\mathrm{T\_av}}^{(c)} \\ \mathbf{K}_{\mathrm{S\_av}}^{(c)} \end{pmatrix}$; $\mathbf{K}^{(c)}$ represents samples with c category in $\mathbf{K}_S$ and $\mathbf{K}_T$. $\mathbf{K}_S$ and $\mathbf{K}_T$ are the kernel maps of $\left\{\mathbf{x}_{S(i)}\right\}_{i=1}^{n_S}$ and $\left\{\mathbf{x}_{T(j)}\right\}_{j=1}^{n_T}$. $\mathbf{K}_{\mathrm{S\_av}}$ and $\mathbf{K}_{\mathrm{T\_av}}$ are mean vectors of $\mathbf{K}_S$ and $\mathbf{K}_T$, respectively. $K_{T(S)\_av}^{(c)}$ is the mean vector of the target (source) domain $K_{T(S)}^{(c)}$ with c category.

The first term of equation (18) is $L_{\mathrm{KELM}} = \min_{\beta_k} : 1/2\|\beta_k\|^2 + \lambda_2/2\|\mathbf{K}\beta_k - \mathbf{X}\|^2$, and the second and third terms are CDMA measure and LDA, respectively. α_{k1} and α_{k2} are tradeoff parameters of L_{CDMA} and L_{LDA}, respectively.

The first part of L_{KELM} is the regularization parameter to prevent model overfitting, and the second part of L_{KELM} denotes the reconstruction error. λ_2 is a parameter that balances the two parts. We set $\partial L_{DKELM-CDMA}/\partial\beta_k = 0$, and get

Input: Data set D_S and D_T, tradeoff parameters α_1, α_2, and λ_1.
Output: Output layer weight β^*
Step 1: Randomly initialize input weights w and biases $\mathbf{b}$ of the hidden layer with L nodes, and set tradeoff parameters α_1, α_2, and λ_1.
Step 2: Transform $\mathbf{D}_S = \left\{(\mathbf{x}_{s(i)}, \mathbf{y}_{s(i)})\right\}_{i=1}^{n_S}$ and $\mathbf{D}_T = \left\{x_{Tj}\right\}_{j=1}^{n_T}$ to H_S and H_T using $g(\mathbf{w}x + \mathbf{b})$, and set $\mathbf{S}_b$ and $\mathbf{S}_w$ to zero.
Step 3: Calculate β^* according to equation (17).
Step 4: Use β^* to predict $\mathbf{D}_T$ and get its label $\hat{\mathbf{Y}}_{\mathbf{T}}$.
Step 5: Solve $\mathbf{S}_b$ and $\mathbf{S}_w$ by using D_S, D_T, and $\hat{\mathbf{Y}}_{\mathbf{T}}$ according to equations (12) and (13).
Step 6: Repeat steps 3–5 until $\hat{\mathbf{Y}}_{\mathbf{T}}$ unchanged.

Algorithm 1: DELM-CDMA

$$\begin{pmatrix} \boldsymbol{\beta}_k \\ -\lambda_2 \mathbf{K}\left(\mathbf{K}\boldsymbol{\beta}_k - \mathbf{X}\right)^{\mathrm{T}} \\ +\frac{\alpha_{k1}}{2}\begin{pmatrix} (\mathbf{K}-\mathbf{K}_{\mathrm{av}})(\mathbf{K}-\mathbf{K}_{\mathrm{av}})^{\mathrm{T}} \\ +\sum_{c=1}^{C}\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}} \end{pmatrix} \\ +\alpha_{k2}\begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}} \\ -\sum_{c=1}^{C}\left(\mathbf{K}_{\mathrm{av}}^{(c)}-\mathbf{K}_{\mathrm{av}}\right)\left(\mathbf{K}_{\mathrm{av}}^{(c)}-\mathbf{K}_{\mathrm{av}}\right)^{\mathrm{T}} \end{pmatrix}\boldsymbol{\beta}_k \end{pmatrix} = 0, \quad (19)$$

$$\boldsymbol{\beta}_k^* = \mathbf{K}^{\mathrm{T}}\left(\Omega + \alpha_{k1}\begin{pmatrix} (\mathbf{K}-\mathbf{K}_{\mathrm{av}})(\mathbf{K}-\mathbf{K}_{\mathrm{av}})^{\mathrm{T}} \\ +\sum_{c=1}^{C}\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}} \end{pmatrix} + \alpha_{k2}\begin{pmatrix} \sum_{c=1}^{C}\sum_{i=1}^{n^{(c)}}\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)\left(\mathbf{K}^{(c)}-\mathbf{K}_{\mathrm{av}}^{(c)}\right)^{\mathrm{T}} \\ -\sum_{c=1}^{C}\left(\mathbf{K}_{\mathrm{av}}^{(c)}-\mathbf{K}_{\mathrm{av}}\right)\left(\mathbf{K}_{\mathrm{av}}^{(c)}-\mathbf{K}_{\mathrm{av}}\right)^{\mathrm{T}} \end{pmatrix} + \frac{\mathbf{I}_{(n_S+n_T)\times(n_S+n_T)}}{\lambda_2}\right)\mathbf{X}. \quad (20)$$

After obtaining β_k^*, we could learn a new feature representation of D_S and D_T by $\mathbf{K}_{S1} = \mathbf{K}_S\beta_k^*$, $\mathbf{K}_{T1} = \mathbf{K}_T\beta_k^*$.

3.3.2. Classification. At this stage, we transform $\mathbf{K}_{S1}$ and $\mathbf{K}_{T1}$ to $\mathbf{K}_{S2}$ and $\mathbf{K}_{T2}$ by kernel mapping $k_{ij1} \longrightarrow^{\text{mapping}} \Omega(k_{ij1}) = k_{ij2}$. We obtain a standard representation of kernel ELM according to [19].

$$\mathbf{f}(x) = \Omega\left(k_{ij1}\right)\mathbf{K}_{S2}^{\mathrm{T}}\left(\Omega(\mathbf{K}_{S2}) + \frac{\mathbf{I}_{n_S}}{\lambda_3}\right)\mathbf{Y}_S, \quad (21)$$

where $\mathbf{f}(x)$ is output of a test sample x, λ_3 is the tradeoff parameter of empirical risk in kernel ELM. We summarize the DKELM-CDMA procedure in Algorithm 2.

4. Experiment and Analysis

In this section, we will perform experiments on object recognition and text data sets (described in Tables 1 and 2) for classification tasks under domain adaptation, and estimate DELM-CDMA and DKELM-CDMA. For fairness, all experiments were conducted on a PC with 8 GB memory and Windows 10 operating system. The algorithms are implemented in MATLAB 2017b. Each experiment runs 20 times, and the average is recorded. The accuracy rate [36, 39, 53] is adopted to evaluate each algorithm in experiments.

4.1. Data Set Description. USPS + MNIST [53]: USPS and MNIST (shown in Figure 3 and Table 1) are two image data sets with different but related distributions for handwritten number recognition. They share 10 categories from 0 to 9. USPS data set contains 9,298 images with 16×16 pixels, and MNIST data set collects 70,000 images with 28×28 pixels. In this section, 1800 images from USPS data set and 2000 images from MNIST data set are selected for domain adaptation, and then we construct two domain adaptation tasks: USPS $\longrightarrow$ MNIST, MNIST $\longrightarrow$ USPS. Moreover, all the images are uniformly transformed into grayscale images with 16×16 pixels, and each image is represented by a 256-dimensional vector encoding the grayscale values of all pixels.

MSRC + VOC2007 [63]: MSRC and VOC2007 (shown in Figure 4 and Table 1) are two common object recognition

Input: Data set D_S and D_T, tradeoff parameters α_{k1}, α_{k2} , and λ_2.
Output: Target output $\mathbf{Y_T}$.
Step 1: Map $\mathbf{D}_S = \left\{(\mathbf{x}_{s(i)}, \mathbf{y}_{s(i)})\right\}_{i=1}^{n_S}$ and $\mathbf{D}_T = \left\{x_{Tj}\right\}_{j=1}^{n_T}$ to K_S and K_T using $\Omega(x)$, and set $\mathbf{S}_b$ and $\mathbf{S}_w$ to zero.
Step 2: Calculate β_k^* according to equation (20).
Step 3: Set $\mathbf{K}_{S1} = \mathbf{K}_S\beta_k^*$, $\mathbf{K}_{T1} = \mathbf{K}_T\beta_k^*$, and map K_{S1} and K_{T2} to K_{S2} and K_{T2} using $\Omega(x)$.
Step 4: Predict $\mathbf{D}_T$ and get its label $\hat{\mathbf{Y}}_\mathbf{T}$ according to equation (21).
Step 5: Solve $\mathbf{S}_b$ and $\mathbf{S}_w$ by using $\{K_S, Y_S\}$, $\left\{K_T, \hat{\mathbf{Y}}_\mathbf{T}\right\}$ and according to equations (12) and (13).
Step 6: Repeat steps 2–5 until $\hat{\mathbf{Y}}_\mathbf{T}$ unchanged.

Algorithm 2: DKELM-CDMA.

Table 1: Description of image data sets.

Data set		Type of data	Number of samples	Dimension	Class	Contains subsets
USPS		Digit	1,800	256	10	USPS
MNIST		Digit	2,000	256	10	MNIST
MSRC		Object	1,269	240	18	MSRC
VOC2007		Object	1,530	240	20	VOC
Caltech-256		Object	1,123	800	10	Caltech
Office-10	AMAZON	Object	958	800	10	AMAZON
	Webcam	Object	295	800	10	Webcam
	DSLR	Object	157	800	10	DSLR
Office-31	Amazon	Object	2,817	2,048	31	Amazon
	Webcam	Object	795	2,048	31	Webcam
	Dslr	Object	498	2,048	31	Dslr

Table 2: Description of reuters-21578 data set.

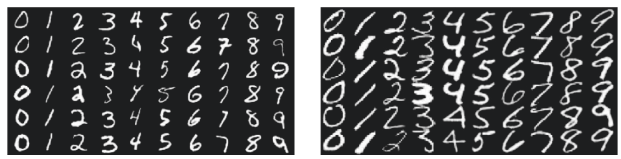
Data set		Type of data	Number of samples	Dimension	Class	Contains subsets
Reuters-21578:	Orgs	Text	1,237	4,771	Binary	Orgs
	People	Text	1,208	4,771	Binary	People
	Place	Text	1,016	4,771	Binary	Place

image data sets. MSRC has 4,323 images from 18 categories, and VOC2007 contains 5,011 images from 20 categories. Both are different but related. In MSRC and VOC2007 data sets, we, respectively, choose 1269 pictures and 1530 pictures from 6 shared classes including airplanes, birds, cows, family cars, sheep, and bicycles, and set two domain adaptation task: MSRC $\longrightarrow$ VOC and VOC $\longrightarrow$ MSRC. In this experiment, all the pictures are transformed into 0~256 gray pixels, and extract 240 dimensions. All images are rescaled to be 256 pixels in length, and extract 128-dimensional dense SIFT (DSIFT) features. A 240-dimensional codebook is created, where K-means clustering is used to obtain the codewords.

Office + Caltech [53]: Office + Caltech data set (shown in Figure 5 and Table 1) released by Gong [51] for visual cross-domain object recognition. Office data set includes three sub-data sets: Amazon (A), Webcam (W), and DSLR (D), in which 4,652 images with31 categories are collected. Caltech-256 (C) is also a benchmark data set for target recognition, in which 30,607 images from 256 categories are collected. In this experiment, we select 10 shared categories in four domains C (Caltech-256), A (Amazon), W (Webcam), and D (DSLR) with a total of 2,533 images with 800 SURF features, shown in Table 1. During the experiment, we can construct 12 groups of domain adaptation task, such as C $\longrightarrow$ A, C $\longrightarrow$ W, C $\longrightarrow$ D, . . ., and D $\longrightarrow$ W.

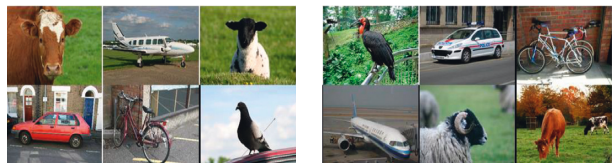
Office31 [64]: Office-31 (shown in Table 1) is a standard benchmark for domain adaptation, which contains 4,652 images with 31 categories and consists of three real-world object domains: amazon, webcam, and dslr. In this paper, we use 2,048-dim ResNet-50 [65] features to conduct 6 cross-domain tasks, that is, amazon vs dslr, amazon vs webcam, dslr vs amazon, dslr vs webcam, webcam vs amazon, webcam vs dslr.

Reuters-21578 [66]: Reuters-21578 text data set is commonly used for text categorization in domain adaptation, in which 21,577 news articles were collected and annotated by Reuters and divided into 5 categories: exchanges, orgs, people, places, and topics. In every text, a corpus becomes a digital data processing that takes out every word that appears as a feature. In this section, we adopt articles from 3 largest classes (shown in Table 2): orgs, people, and



(a) (b)

Figure 3: Image samples from (a) MNIST and (b) USPS.



(a) (b)

Figure 4: Image samples from (a) MSRC and (b) VOC2007.

(a) (b) (c) (d)

Figure 5: Image samples from Amazon, Caltech, DSLR, and Webcam. (a) DSLR. (b) Amazon. (c) Webcam. (d) Caltech-256.

place, and then construct 6 domain adaptation tasks: orgs vs people, people vs orgs, orgs vs place, place vs orgs, people vs place, and place vs people.

4.2. Compared Algorithm and Setting. To investigate our method, we choose several classification algorithms for comparison.

4.2.1. Non-Adaptation Domain Classifiers. 1NN: k Nearest Neighbor classifier; we choose one sample as target nearest neighbor.

SVM: Support vector machine with linear kernel; we set SVM penalty parameter belong to $\{0.1, 0.5, 1, 5, 10, 50, 100\}$.

ELM: Standard extreme learning machine.

SSELM: Semi-Supervised ELM with graph regularization [24].

4.2.2. Shallow Adaptation Domain Classifiers. TCA1, TCA2: Classifier combining TCA [57] with 1NN (TCA1) and classifier combining TCA with SVM (TCA2).

JDA1, JDA2: Classifier combining JDA [53] with 1NN (JDA1) and classifier combining JDA with SVM (JDA2). We set parameters in TCA and JDA according to [53].

DAELM_S, DAELM_T [37]: Supervised ELMs for domain adaptation.

KMM [67]: Kernel Mean Matching with resampling weights. Its result in this section is cited from [42].

LMPROJ [68]: A large margin hyperplane classifier based on SVM. We cite its result from [42].

ARRLS [66]: A general transfer learning framework referred to adaptation regularization based on transfer learning using squared loss. We set its parameters according to [66].

TELM-OWA [39]: Supervised ELM using output weight alignment for domain adaptation. We set its parameters referred [39].

CDELM-M, CDELM-C: An unsupervised ELM using MMD and manifold regularization for domain adaptation. We cited its result in [42] for comparison.

JUC-SDELM [69]: An unsupervised ELM using MMD and the scalable factor for domain adaptation. We cited its result in [69] for comparison.

4.2.3. Deep Domain Adaptation Classifiers. DAN [58]: A deep domain adaptation network with multi-kernel MMD for domain adaptation. We cited its result in [59] for comparison.

JAN [59]: A deep domain adaptation network for domain adaptation by aligning the joint distributions of multiple domain-specific layers across domains. We cited its result in [59] for comparison.

DANN [61]: An adversarial domain adaptation network with domain-invariant feature extractor. We cited its result in [62] for comparison.

CAN [62]: A adversarial domain adaptation network adopting collaborative learning and adversarial learning for domain adaptation. We cited its result in [62] for comparison.

ELM-CDMA, DELM-CDMA, and DKELM-CDMA are our methods. ELM-CDMA is a special case of DELM-CDMA when the LDA term is 0.

It is noteworthy that we set the penalty parameter $\lambda_{(1)} \in [10^{-3}, 10^{-1}]$ in ELM, SSELM, DAELM_S, DAELM_T, TELM-OWA, ELM-CDMA, and DELM-CDMA. In ELM-CDMA and DELM-CDMA, we set $\alpha_1 \in [10^{-2}, 10^{2}]$, $\alpha_2 \in [10^{-4}, 10^{-2}]$, and $L = 2000$ on Office + Caltech and offfice31 data sets, $L = 3000$ on USPS + MNIST and MSRC + VOC2007 data sets, $L = 3000$ on Reuters-21578 data set. We evaluate DAELM_S, DAELM_T, and TELM-OWA in unsupervised domain adaptation task by selecting 0.5% labeled target samples on USPS + MNIST, MSRC + VOC2007, and Reuters-21578 data sets and 1% labeled target samples on Office + Caltech data set to train model. For DKELM-CDMA, we set balance coefficient $\alpha_{k1} \in [10^{-1}, 10^{1}]$ on USPS + MNIST, Reuters-21578, Office + Caltech and offfice31 data sets, $\alpha_{k1} = 10^{-4}$ on MSRC + VOC2007 data set, and set $\alpha_{k2} \in [10^{-2}, 10^{1}]$, penalty coefficient $\lambda_2 \in [10^{-1}, 10^{3}]$ and kernel parameter $\sigma_1 \in [10^{-1}, 10^{3}]$ on 5 data sets at feature extraction stage. We let $\lambda_3 \in [10^{1}, 10^{3}]$ and kernel parameter $\sigma_2 \in [10^{1}, 10^{4}]$ on 5 data sets at classification stage.

4.3. Results and Analysis. We present the accuracy of all algorithms in Tables 3–5 and Figures 6–10 and find that

(1) DKELM-CDMA achieves best average result in Tables 3 and 4 with help of CDMA and LDA. Especially in Table 3, the average result is 56.7%, much higher than other methods, which indicate that CDMA could efficiently reduce the distribution discrepancy of output-layer data from source and target domains and significantly promote the knowledge transfer capability of ELM in unsupervised domain adaptation. DKELM-CDMA could further mine shared features across domains, so it achieves better accuracy than DELM-CDMA.

(2) DELM-CDMA and ELM-CDMA perform better than other methods on most tasks in total average, which shows again that CDMA is efficient for unsupervised domain adaptation. It extends ELM in wide application scope and scene. DELM-CDMA outperform ELM-CDMA on most tasks with the help of LDA, showing that LDA also improves discrimination of ELM for domain adaptation.

(3) Compared with DKELM-CDMA, DELM-CDMA, and ELM-CDMA, TELM-OWA and DAELM_T get bad results, because they are supervised classifiers requiring a few labeled samples on target domain. Moreover, the traditional classifiers, such as 1NN, SVM, and ELM, are unsuccessful because of inability in knowledge transfer. ELM gains slightly better performance than 1NN and SVM due to its good generation. SSELM has better performance than ELM by mining manifold structure information of data.

(4) Although ARRLS, CDELM-M(C), JUC-SDELM, KMM, and LMPROJ also can gain better results with the help of statistical adaptation mechanism like MMD, (D)ELM-CDMA are more successful duo to CDMA is more efficient than MMD. TCA1 (2) and JDA1 (2) have higher accuracy than 1NN and SVM, depending on cross-domain shared feature extraction.

(5) In Table 5 and Figure 10, we run 1NN, SVM, ELM, SSELM, DAELM_S, DAELM_T, ARRLS, TELM-OWA, ELM-CDMA, DELM-CDMA, and DKELM-CDMA on the Office-31 data set with ResNet-50 features comparing with deep adaptation domain methods DAN, DANN, JAN, and CAN. As a machine learning classification model with shallow network structure, DELM-CDMA, although not extracting shared high-level features, performs well on more complex data sets combined with deep feature extraction networks. It clearly demonstrates that DELM-CDMA equipped with deep generic features could further reduce the cross-domain discrepancy and achieve the best adaptation performance, which demonstrates the potential of our method. However, DKELM-CDMA does not work well on data office31. A possible explanation is that its feature extraction process hurts the quality of features already extracted by ResNet-50.

We carry out experiment on MNIST⟶USPS data set to compare the running-time in 1NN,SVM, ELM,SSELM,TCA1(2), JDA1(2), DAELM_S, DAELM_T, ARRLS, DELM-CDMA, and DKELM-CDMA, and the results are shown in Table 6. It can be seen: (1) ELM has least running-time because of fast learning speed, and the algorithm based on ELM also spend less time cost than ones based on 1NN and SVM, such as TCA1(2) and JDA1(2). (2) DELM-CDMA spends more running-time than DAELM_S and DAELM_T because of the additional calculation generated by CDMA and LDA, but its time cost is lower than ARRLS in which MMD matrix and graph regulation term are required to construct. (3) TELM-OWA and DKELM-CDMA have similar time consumption more time than ELM, SSELM,

Table 3: Classification accuracies (%) of all algorithms on office + caltech and USPS + MNIST.

Methods	Data set														
	C ⟶ A (1)	C ⟶ W (2)	C ⟶ D (3)	A ⟶ C (4)	A ⟶ W (5)	A ⟶ D (6)	W ⟶ C (7)	W ⟶ A (8)	W ⟶ D (9)	D ⟶ C (10)	D ⟶ A (11)	D ⟶ W (12)	USPS ⟶ MNIST	MNIST ⟶ USPS	Average
1NN	23.7	25.8	25.5	26.0	29.8	25.5	19.9	23.0	59.2	26.3	28.5	63.4	44.7	65.9	34.8
SVM	52.4	40.7	42.0	44.1	41.7	40.8	29.9	34.1	87.9	31.4	32.4	73.6	35.1	54.7	45.8
ELM	50.6	42.7	42.0	43.0	37.6	42.7	32.2	36.9	80.3	30.1	31.5	71.9	30.7	51.4	44.5
SSELM	52.9	45.4	38.9	41.8	35.9	38.9	34.3	37.1	79.0	30.4	32.1	80.0	29.1	43.3	44.2
TCA1	55.9	48.8	51.6	44.2	40.3	36.9	32.3	38.4	81.5	36.2	39.4	82.4	35.6	53.3	48.3
TCA2	55.0	48.5	49.7	44.4	41.0	43.3	32.6	38.9	79.6	34.6	38.3	83.9	34.7	52.8	48.4
JDA1	53.3	46.8	52.9	42.7	36.3	40.8	33.7	47.6	86.9	36.5	43.3	82.7	34.8	54.9	49.5
JDA2	54.8	51.9	51.2	41.2	39.7	44.6	35.1	45.5	80.3	33.4	40.9	83.1	33.9	53.1	49.2
DAELM_S	50.3	45.4	37.6	42.9	38.3	36.9	34.2	36.4	80.9	32.3	34.4	72.9	44.4	47.1	45.3
DAELM_T	27.9	31.9	22.9	25.3	19.3	36.3	41.9	46.5	53.5	48.1	51.2	54.2	31.1	60.3	39.3
KMM	48.4	45.8	53.6	42.3	42.5	42.7	31.6	32.0	72.0	31.7	32.2	42.9	33.0	42.0	42.3
LMPROJ	52.5	46.9	40.8	43.6	38.6	38.2	30.6	36.4	83.5	30.7	34.0	81.7	46.3	57.7	47.2
ARRLS	54.9	50.5	44.6	42.5	39.0	38.9	34.7	39.9	78.3	32.2	37.2	82.7	34.6	52.5	47.3
TELM-OWA	31.7	32.2	33.1	38.9	38.6	43.3	33.5	38.0	77.1	32.5	36.2	78.6	59.6	59.4	45.2
CDELM-M	52.1	51.1	45.9	42.3	42.9	45.9	30.3	39.8	81.2	30.3	35.7	81.8	46.5	62.5	49.2
CDELM-C	56.3	51.0	43.8	43.2	40.8	41.8	34.4	39.6	83.1	34.4	39.5	84.3	45.9	42.6	48.6
JUC-SDELM	49.6	39.8	44.5	42.5	39.0	44.1	29.7	35.6	81.5	27.7	36.4	74.6	39.8	62.7	46.2
ELM-CDMA	53.2	47.1	45.2	43.6	44.1	47.1	33.1	36.6	84.1	33.1	36.5	74.6	45.3	82.6	50.5
DELM-CDMA	55.3	48.5	45.9	42.2	44.8	45.2	33.9	37.3	84.1	34.0	38.1	81.0	47.5	84.4	51.6
DKELM-CDMA	57.9	56.6	59.9	47.4	45.8	46.5	38.7	41.4	93.6	38.0	42.8	89.5	55.6	79.8	56.7

Table 4: Classification accuracies (%) of all algorithms on MSRC + VOC2007 and reuters-21578.

Data set	Methods														
	1NN	SVM	ELM	SSELM	TCA1	TCA2	JDA1	JDA2	DAELM_S	DAELM_T	ARRLS	TELM-OWA	ELM-CDMA	DELM-CDMA	DKELM-CDMA
MSRC ⟶ VOC	36.0	35.1	38.6	37.5	27.9	34.9	27.1	34.8	38.3	33.6	34.6	41.4	38.0	38.2	39.0
VOC ⟶ MSRC	45.2	54.7	58.0	63.2	47.9	54.2	48.3	54.6	58.4	30.6	50.4	66.2	62.7	63.5	64.5
Orgs vs people (1)	72.9	75.3	77.2	80.3	76.5	77.5	80.1	81.8	76.6	48.7	80.6	63.7	82.5	83.4	82.2
People vs orgs (2)	72.0	77.1	79.3	84.7	77.3	79.4	85.1	85.3	78.4	47.7	84.6	64.3	86.1	86.3	85.4
Orgs vs place (3)	67.5	70.2	63.6	69.6	72.4	71.8	75.7	76.2	68.5	43.8	72.2	68.1	78.3	79.1	78.2
Place vs orgs (4)	61.1	63.8	64.6	71.2	68.2	65.8	76.4	76.1	68.8	42.5	74.1	74.7	79.5	80.6	83.3
People vs place (5)	52.7	60.6	57.0	66.0	61.2	60.6	66.9	65.8	59.1	42.5	65.2	59.0	68.9	69.3	67.2
Place vs people (6)	53.4	57.9	58.7	61.7	56.1	51.1	58.1	51.2	58.8	39.9	58.6	60.5	66.5	65.1	69.0
Total average	57.6	61.8	62.1	66.8	60.9	61.9	64.7	65.7	63.4	41.2	65.0	62.2	70.3	70.7	71.1

Table 5: Accuracy (%) comparison of different algorithms on office-31 (resnet-50).

Data set	Methods														
	1NN	SVM	ELM	SSELM	DAN	DANN	JAN	CAN	DAELM_S	DAELM_T	ARRLS	TELM-OWA	ELM-CDMA	DELM-CDMA	DKELM-CDMA
Amazon vs dslr (1)	79.1	80.7	82.1	83.3	78.6	79.7	84.7	85.5	83.9	38.2	85.7	77.7	86.7	86.7	85.3
Amazon vs webcam (2)	75.9	75.4	77.5	77.6	80.5	82.0	85.4	81.5	77.7	32.1	83.4	76.0	80.6	84.4	78.5
Dslr vs amazon (3)	60.2	62.8	64.0	66.1	63.6	68.2	68.6	65.9	64.6	29.4	71.7	65.2	68.7	69.5	67.3
Dslr vs webcam (4)	96.0	96.2	94.2	95.5	97.1	96.9	97.4	98.2	94.0	12.8	97.1	94.2	95.4	97.7	96.1
Webcam vs amazon (5)	59.9	62.3	65.6	66.2	62.8	67.4	70.0	63.4	66.0	83.2	70.4	66.1	68.3	68.4	67.3
Webcam vs dslr (6)	99.4	99.6	98.6	98.0	99.6	99.1	99.8	99.7	99.0	81.3	99.0	98.4	99.0	99.4	99.4
Total average	78.4	79.5	80.3	81.1	80.4	82.2	84.3	82.4	80.9	46.2	84.6	79.6	83.1	84.4	82.3

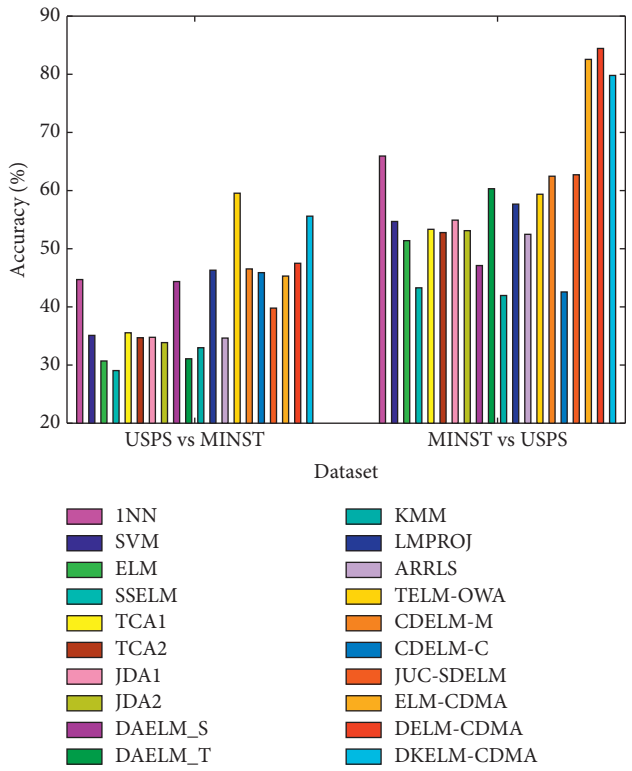


Figure 6: Classification accuracy of different algorithms on USPS + MNIST data set.

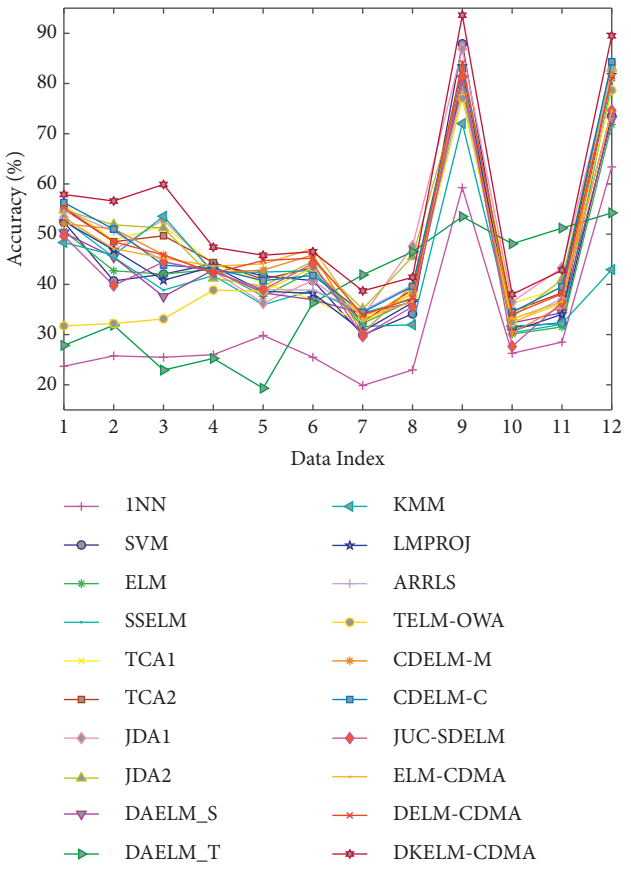


Figure 7: Classification accuracy of different algorithms on office + caltech data set.

DAELM_S, and DAELM_T, because they all need to solve the output weights twice during training. (4) SSELM needs more time than ELM because of manifold regulation. (5) JDA2 has most running time because it need to construct the marginal and conditional MMD matrix, and SVM has a complex solving process.

Through the above analysis, it can be seen that compared with other methods, DKELM-CDMA and DELM-CDMA do not require more time consumption while achieving higher accuracy.

4.4. Parameter Sensitivity Analysis. To evaluate the sensitivity of DELM-CDMA to parameters α_1, α_2, λ_1 and number of hidden layer nodes(L) and its convergence, we run experiments on org vs people, MSRC vs VOC, MNIST vs USPS, A vs D and amazon vs dslr data sets, and the results are shown in Figures 11(a)–11(e). We can see that: (1) the accuracy increases first and then little decreases with α_1, α_2, λ_1 growing, and DELM-CDMA achieve optimal results when $\alpha_1 \in [10^{-2}, 10^{0}]$, $\alpha_2 \in [10^{-3}, 10^{-1}]$, and $\lambda_1 \in [10^{-3}, 10^{-2}]$, respectively. Because α_1, α_2, and λ_1 adjust CDMA, LDA, and output weight regularization terms, results in Figures 11(a)–11(c) show that CDMA, LDA, and output weight regularization term, adjusted the appropriate range, can improve the knowledge transfer ability and discrimination of ELM and avoid model from overfitting. (2) From Figure 11(d), we can see that the accuracy of DELM-CDMA increases first and then slightly decreases with L growing on most data sets. Wider hidden layer as soon as possible can improve the approximation ability of the function, but it will weaken the ELM performance of domain adaptation by damaging cross-domain metric performance of CDMA. (3) As shown in Figure 11(e), by observing the variation of accuracy with the increase of iterations, we find that finally converge after several iterations, which shows that DELM-CDMA has good robustness.

Similar to DELM-CDMA, we carry out some experiments on org vs people, MSRC vs VOC, MNIST vs USPS, A vs D data set, and amazon vs dslr data sets to observe influence of parameter α_{k1}, α_{k2}, and λ_2 on the accuracy of DKELM-CDMA, its convergence, and CDMA distance. We record results in Figure 12 and can see that: (1) Generally, the sensitivity of parameter α_{k1}, α_{k2}, and λ_2 are different for various data sets, but DKELM-CDMA mainly achieves good performance when $\alpha_{k1} \in [10^{-2}, 10^{0}]$, $\alpha_{k2} \in [10^{-3}, 10^{-1}]$, and $\lambda_2 \in [10^{-3}, 10^{0}]$, respectively, shown in Figures 12(a)–12(c). (2) From Figure 12(d), we can see that the accuracy is increasing iteratively, and finally converges after several iterations, so the robustness of DKELM-CDMA is strong. (3) We investigate CDMA distance with iteration growing, and the result is shown in Figure 12(f). We can see that: with the increase of iteration number, the CDMA distance becomes small, and the accuracy becomes high, which indicate that DKELM-CDMA could extract effective shared feature representation across domains relying on CDMA to reduce

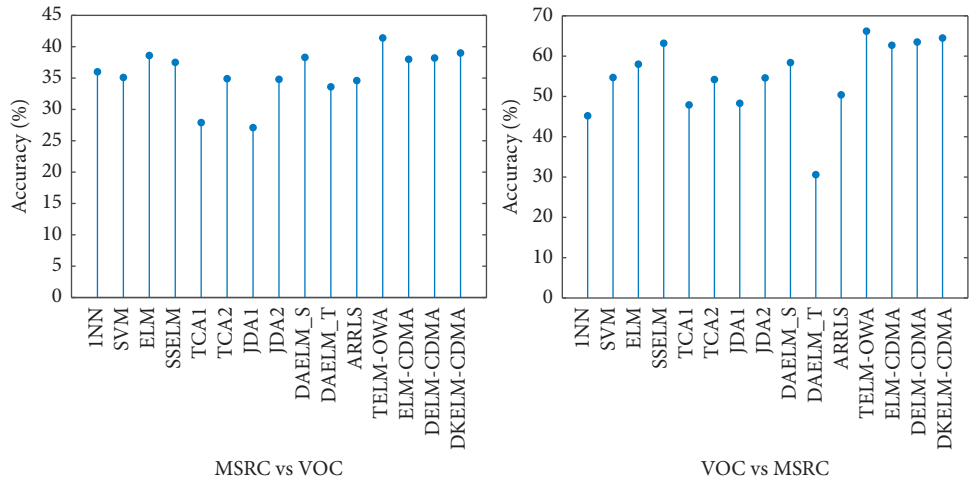


Figure 8: Classification accuracy of different algorithms on MSRC + VOC data set.

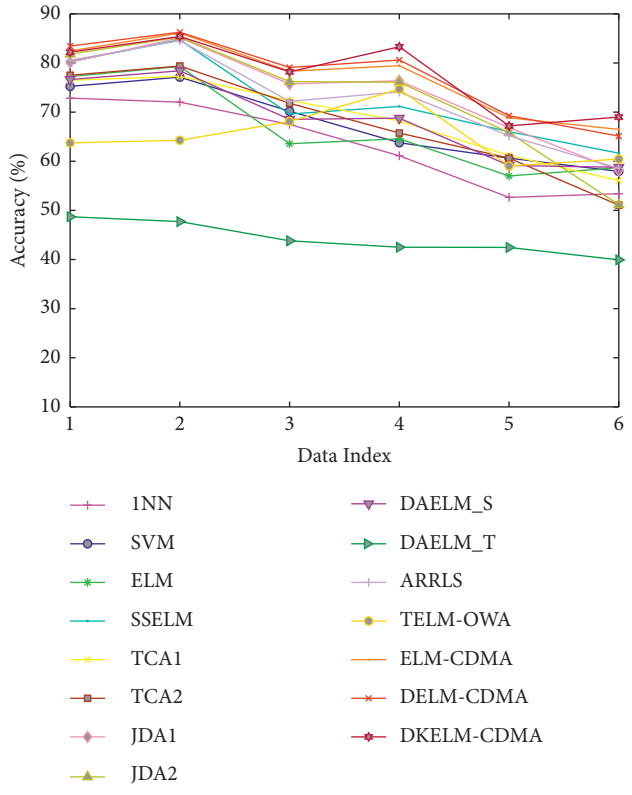


Figure 9: Classification accuracy of different algorithms on reuters-21578 data set.

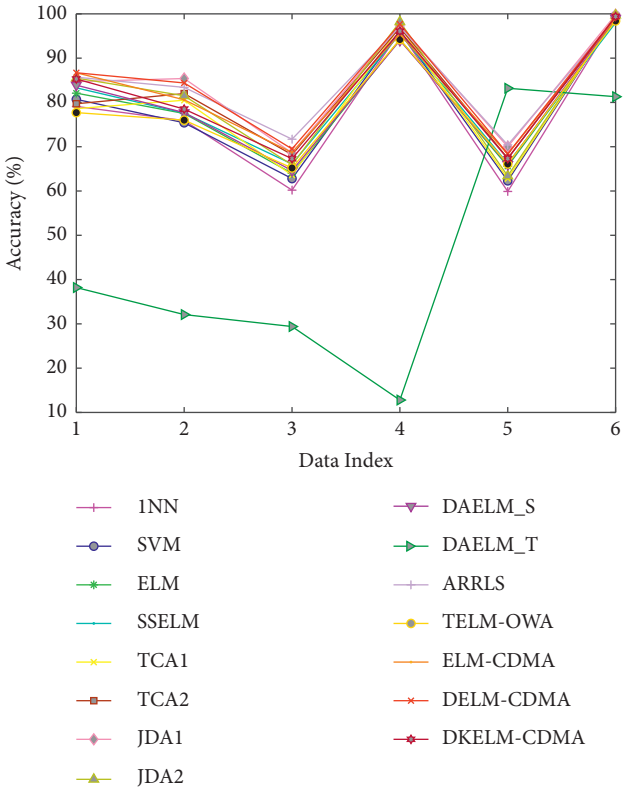


Figure 10: Classification accuracy of different algorithms on office-31 data set.

Table 6: Comparison of running time on MNIST ⟶ USPS.

Algorithm	1NN	SVM	ELM	SSELM	TCA1	TCA2	JDA1	JDA2	DAELM_S	DAELM_T	ARRLS	TELM-OWA	DELM-CDMA	DKELM-CDMA
Time(s)	0.6	8.8	0.4	3.5	4.7	5.5	48.3	53.6	0.8	0.6	2.1	3.7	1.9	3.7

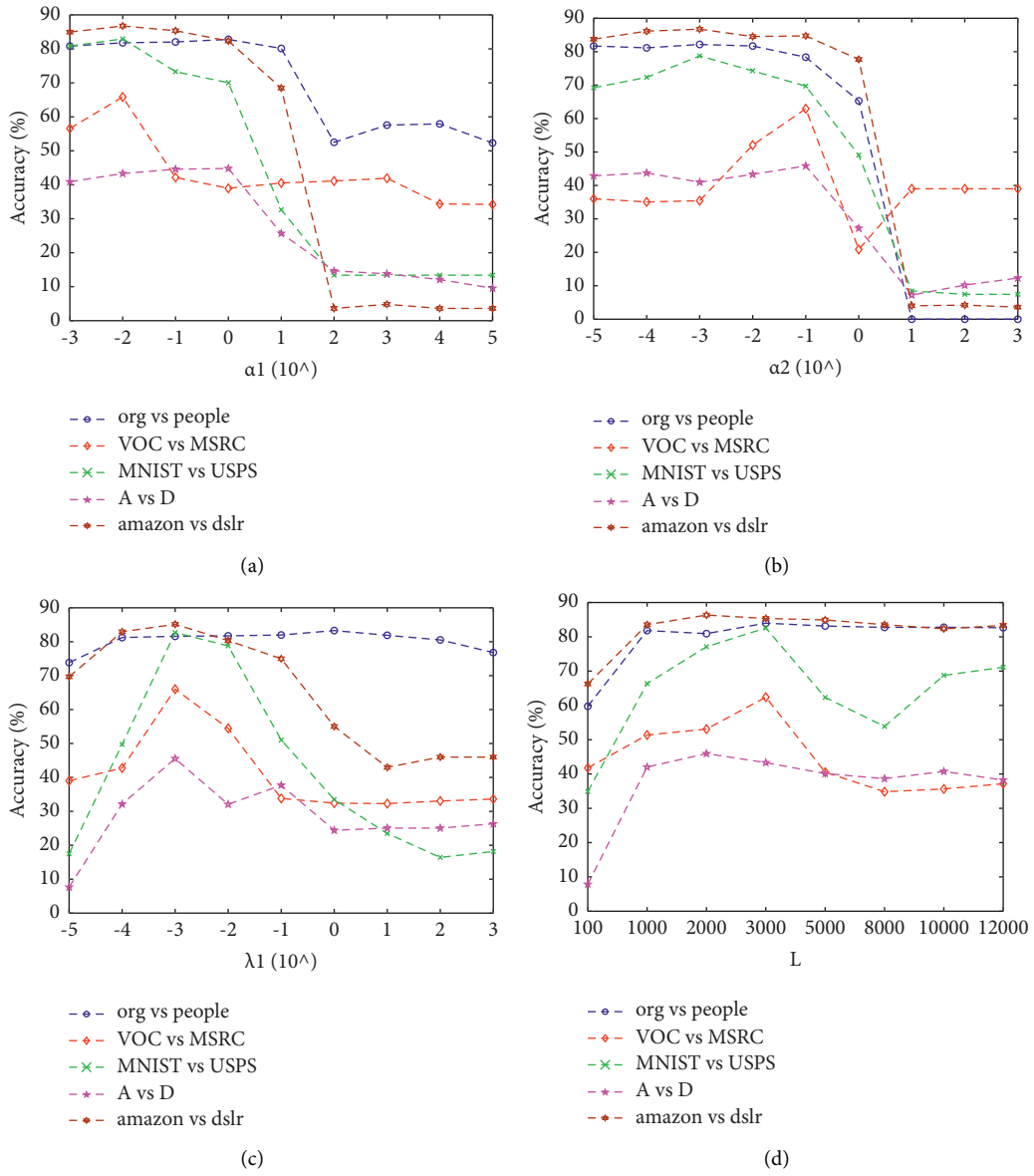
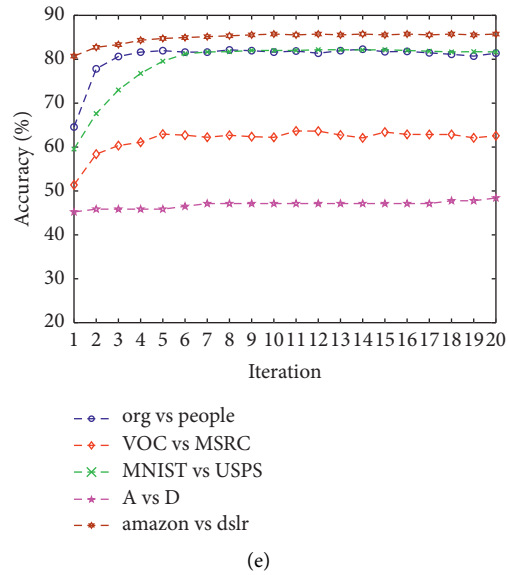
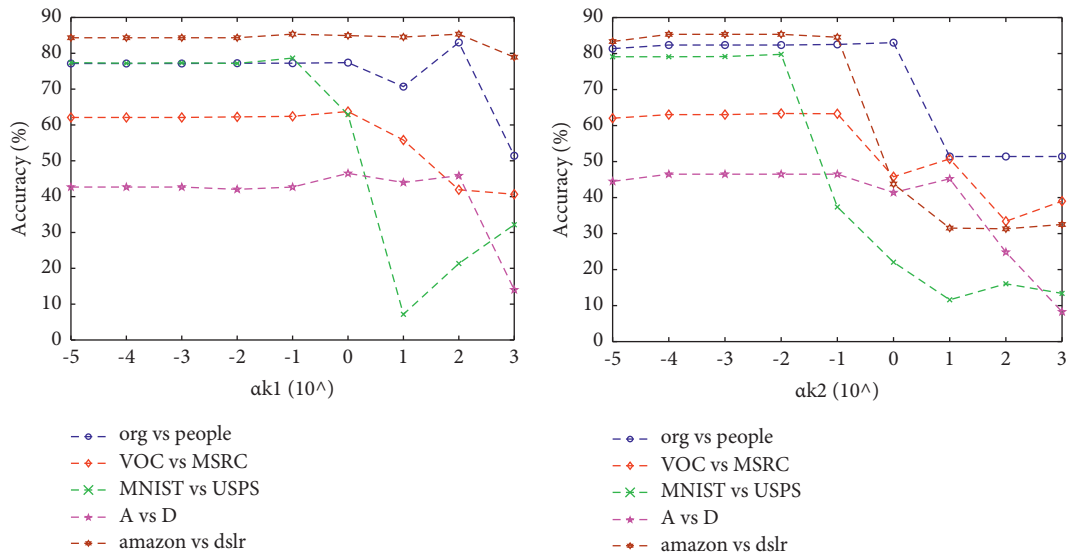


Figure 11: Continued.



(e)

Figure 11: Effect of iteration, number of hidden layer nodes, and parameters α_1, α_2, and λ_1 on the accuracy of DELM-CDMA.



(a) (b)

Figure 12: Continued.

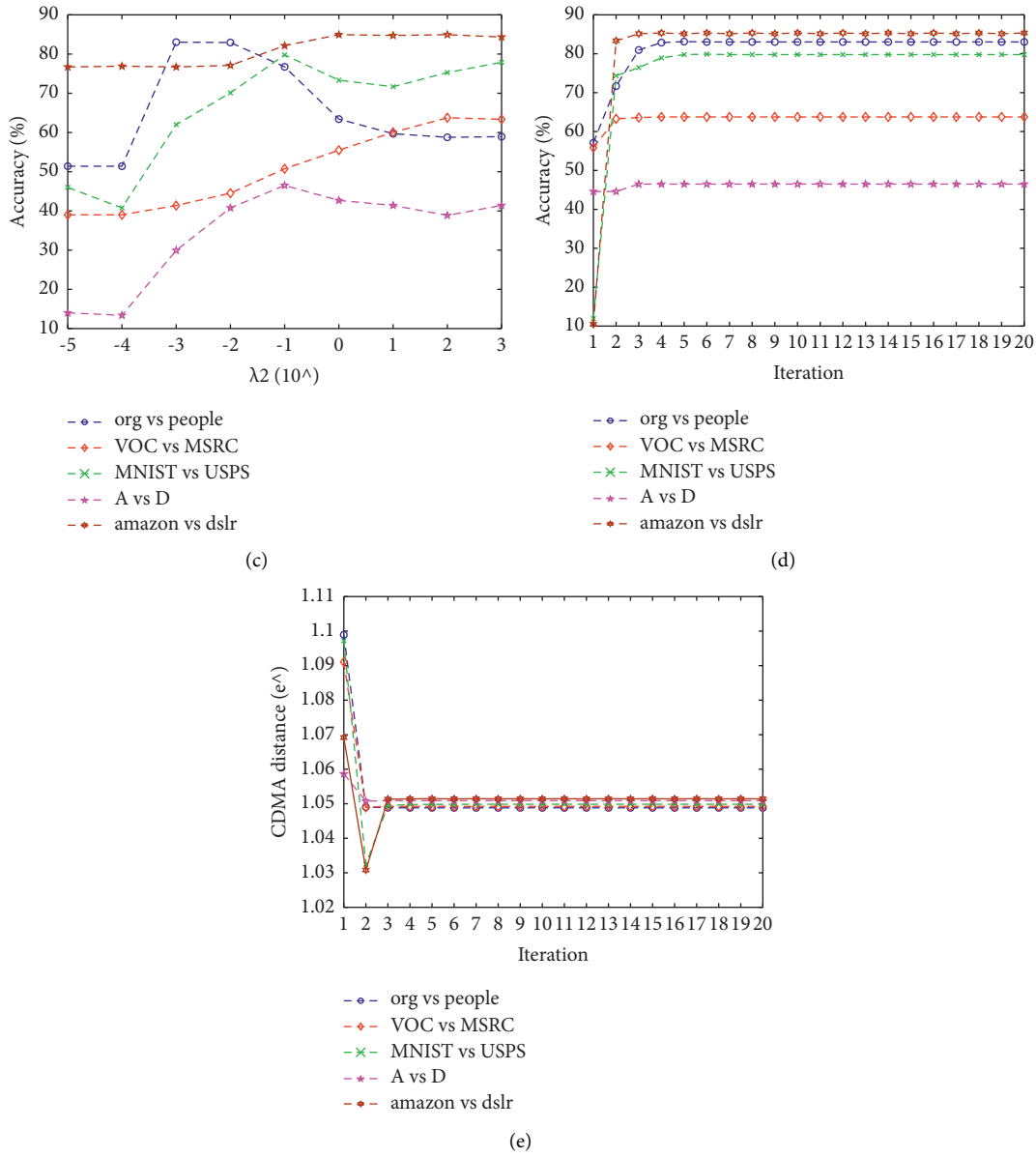


Figure 12: CDMA distance and Effect of iteration and parameter α_{k1}, α_{k2}, and λ_2 on the accuracy of DKELM-CDMA.

differences in marginal and conditional distributions between domains, and facilitate the cross-domain transfer of knowledge.

5. Conclusion

In this paper, we propose a novel ELM model called Discriminative Extreme Learning Machine with Cross-Domain Mean Approximation for Unsupervised Domain Adaptation. Based on traditional ELM, it introduces Cross-Domain Mean Approximation to jointly minimize the marginal and conditional distribution from the data of output layers between source and target domains which enhance the ability of ELM to transfer knowledge. We also add Linear Discriminant Analysis to further improve the discrimination which improves the accuracy of ELM. In addition, in order to overcome the sensitivity of DELM-CDMA to the initialization of hidden layer parameters, we further propose DKELM-CDMA as the kernel version of DELM-CDMA, which further facilitates the fitting for nonlinear data. Finally, we run DELM-CDMA and DKELM-CDMA on many experiments for unsupervised domain adaptation, and the results show that the proposed approach can effectively enhance the efficiency of cross-domain knowledge transfer in ELM. Although DELM-CDMA and DKELM-CDMA work well in domain adaptation, they cannot fully utilize high-level semantic features representation because of their own shallow network structure. Therefore, we design a DKELM-CDMA with deep learning for feature extraction in future.

Data Availability

The data used to support the findings of this study are found in https://github.com/jindongwang/transferlearning/blob/master/data/dataset.md.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62073124 and 51805149, in part by the Key Scientific Research Projects of Universities in Henan Province under Grant 22A120005, the National Aviation Fund Projects under Grant 201701420002, and Henan Province Key Scientific and Technological Projects under Grant 222102210095.

References

[1] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[2] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, vol. 1, no. 12, pp. 1559–1615, 2019.

[3] H. Azami, M. R. Mosavi, and S. Sanei, "Classification of GPS satellites using improved back propagation training algorithms," *Wireless Personal Communications*, vol. 71, no. 2, pp. 789–803, 2013.

[4] A. Wood, V. Shpilrain, K. Najarian, and D. Kahrobaei, "Private naive Bayes classification of personal biomedical data: application in cancer data analysis," *Computers in Biology and Medicine*, vol. 105, pp. 144–150, 2019.

[5] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[6] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[7] G. S. Nandini, A. S. Kumar, and K. Chidananda, "Dropout technique for image classification based on extreme learning machine," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 111–116, 2021.

[8] S. C. Nayak, S. K. Nayak, and A. Dash, "Extreme learning with artificial electric field algorithm (EL-AEFA) for estimation of short-term vehicular traffic system," in *Proceedings of the 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pp. 1–4, Dehradun, India, June 2021.

[9] M. Turkoglu, "COVID-19 detection system using chest CT images and multiple kernels-extreme learning machine based on deep neural network," *IRBM*, vol. 42, no. 4, pp. 207–214, 2021.

[10] Y. P. Zhao and Y. B. Chen, "Extreme learning machine based transfer learning for aero engine fault diagnosis," *Aerospace Science and Technology*, vol. 121, Article ID 107311, 2022.

[11] S. Pang, X. Yang, X. Zhang, and Y. Sun, "Fault diagnosis of rotating machinery components with deep ELM ensemble induced by real-valued output-based diversity metric," *Mechanical Systems and Signal Processing*, vol. 159, Article ID 107821, 2021.

[12] Y. Cai, Z. Zhang, Q. Yan, D. Zhang, and M. J. Banu, "Densely connected convolutional extreme learning machine for hyperspectral image classification," *Neurocomputing*, vol. 434, pp. 21–32, 2021.

[13] X. Liu, Q. Hu, Y. Cai, and Z. Cai, "Extreme learning machine-based ensemble transfer learning for hyperspectral image classification," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3892–3902, 2020.

[14] H. Tian, M. Shuai, K. Li, and X. Peng, "An incremental learning ensemble strategy for industrial process soft sensors," *Complexity*, vol. 2019, 12 pages, Article ID 5353296, 2019.

[15] S. Saurav, R. Saini, and S. Singh, "Facial expression recognition using dynamic local ternary patterns with kernel extreme learning machine classifier," *IEEE Access*, vol. 9, pp. 120844–120868, 2021.

[16] M. H. Annaby, M. H. Said, A. M. Eldeib, and M. Rushdi, "EEG-based motor imagery classification using digraph Fourier transforms and extreme learning machines," *Biomedical Signal Processing and Control*, vol. 69, Article ID 102831, 2021.

[17] Y. Zhang, Y. Wang, G. Zhou et al., "Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces," *Expert Systems with Applications*, vol. 96, pp. 302–310, 2018.

[18] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics: A Publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 42, no. 2, pp. 513–529, 2012.

[19] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

[20] W. Zong, G. B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[21] Y. Wan, S. Song, G. Huang, and S. Li, "Twin extreme learning machines for pattern classification," *Neurocomputing*, vol. 260, pp. 235–244, 2017.

[22] Y. Zhang, Q. Wu, and J. Hu, "An adaptive learning algorithm for regularized extreme learning machine," *IEEE Access*, vol. 9, pp. 20736–20745, 2021.

[23] S. Li, S. Song, and Y. Wan, "Laplacian twin extreme learning machine for semi-supervised classification," *Neurocomputing*, vol. 321, pp. 17–27, 2018.

[24] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.

[25] H. Pei, K. Wang, Q. Lin, and P. Zhong, "Robust semi-supervised extreme learning machine," *Knowledge-Based Systems*, vol. 159, pp. 203–220, 2018.

[26] J. Ma and C. Yuan, "Adaptive safe semi-supervised extreme machine learning," *IEEE Access*, vol. 7, pp. 76176–76184, 2019.

[27] L. Kasun, H. Zhou, and G. B. Huang, "Representational learning with ELMs for big data," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 31–34, 2013.

[28] L. Shao, R. Kang, W. Yi, and H. Zhang, "An enhanced unsupervised extreme learning machine based method for the

nonlinear fault detection," *IEEE Access*, vol. 9, pp. 48884–48898, 2021.

[29] J. Chen, Y. Zeng, Y. Li, and G. B. Huang, "Unsupervised feature selection based extreme learning machine for clustering," *Neurocomputing*, vol. 386, pp. 198–207, 2020.

[30] C. M. Wong, C. M. Vong, P. K. Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 757–762, 2018.

[31] J. Tang, C. Deng, and G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.

[32] L. Cao, W. Huang, and F. Sun, "A deep and stable extreme learning approach for classification and regression," in *Proceedings of the ELM-2014*, pp. 141–150, Marina Bay Sands, Singapore, December 2015.

[33] Y. Chen, X. Xie, T. Zhang, J. Bai, and M. Hou, "A deep residual compensation extreme learning machine and applications," *Journal of Forecasting*, vol. 39, no. 6, pp. 986–999, 2020.

[34] S. J. Pan and Q. Yang, "'A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct, 2010.

[35] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

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

[37] L. Zhang and D. Zhang, "Domain adaptation extreme learning machines for drift compensation in E-nose systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1790–1801, 2015.

[38] Y. P. Zhao and Y. B. Chen, "Extreme learning machine based transfer learning for aero engine fault diagnosis," *Aerospace Science and Technology*, vol. 121, Article ID 107311, 2022.

[39] S. Zang, Y. Cheng, X. Wang, and Y. Yan, "Transfer Extreme learning machine with output weight alignment," *Computational Intelligence and Neuroscience*, pp. 2021–14, 2021.

[40] X. Li, W. Mao, and W. Jiang, "Extreme learning machine based transfer learning for data classification," *Neurocomputing*, vol. 174, pp. 203–210, 2016.

[41] C. Chen, B. Jiang, and X. Jin, "Parameter transfer extreme learning machine based on projective model," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2018)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, July 2018.

[42] S. Li, S. Song, G. Huang, and C. Wu, "Cross-domain extreme learning machines for domain adaptation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1194–1207, 2019.

[43] Y. Chen, S. Song, S. Li, L. Yang, and C. Wu, "Domain space transfer extreme learning machine for domain adaptation," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1909–1922, 2019.

[44] S. Zang, X. Li, J. Ma, Y. Yan, J. Gao, and Y. Wei, "TSTELM: two-stage transfer extreme learning machine for unsupervised domain adaptation," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–18, 2022.

[45] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, 2006.

[46] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: a detailed tutorial," *AI communications*, vol. 30, no. 2, pp. 169–190, 2017.

[47] A. Farahani, B. Pourshojae, and K. Rasheed, "A concise review of transfer learning," in *Proceedings of the IEEE International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 344–351, Las Vegas, NV, USA, December 2020.

[48] Z. Wan, R. Yang, M. Huang, N. Zeng, and X. Liu, "A review on transfer learning in EEG signal analysis," *Neurocomputing*, vol. 421, pp. 1–14, 2021.

[49] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1682–1695, 2017.

[50] L. Yan, R. Zhu, Y. Liu, and N. Mo, "TrAdaBoost based on improved particle swarm optimization for cross-domain scene classification with limited samples," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 9, pp. 3235–3251, 2018.

[51] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the 2012 IEEE International Conference on Computer Vision (CVPR 2012)*, pp. 2066–2073, Rhode Island Convention Center United States, June 2012.

[52] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2010.

[53] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the International Conference on Computer Vision (ICCV 2013)*, pp. 2200–2207, Sydney Australia, December 2013.

[54] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the 2013 IEEE International Conference on Computer Vision (CVPR 2013)*, pp. 2960–2967, Sydney Australia, December 2013.

[55] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation," in *Proceedings of the British Machine Vision Conference (BMVC 2015)*, pp. 24.1–24.10, Swansea UK, September 2015.

[56] A. Raj, V. P. Namboodiri, and T. Tuytelaars, "Subspace alignment based domain adaptation for RCNN detector," in *Proceedings of the British Machine Vision Conference (BMVC 2015)*, pp. 10–24, Swansea UK, September 2015.

[57] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[58] M. Long, Y. Cao, and J. Wang, "'Learning transferable features with deep adaptation networks," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, pp. 97–105, Lille France, July 2015.

[59] M. Long, H. Zhu, and J. Wang, "'Deep transfer learning with joint adaptation networks," in *Proceedings of the International Conference on International Conference on Machine Learning(ICML)*, pp. 2208–2217, Sydney Australia, August 2017.

[60] M. Long, H. Zhu, and J. Wang, "'Unsupervised domain adaptation with residual transfer networks," in *Proceedings of the advances in Neural Information Processing Systems (NerIPS)*, pp. 136–144, Barcelona Spain, December 2016.

[61] Y. Ganin and V. Lempitsky, ""Unsupervised domain adaptation by backpropagation," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, pp. 1180–1189, Lille France, June 2015.

[62] W. Zhang, W. Ouyang, and W. Li, "'Collaborative and adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 3801–3809, Salt Lake, UT, USA, June 2018.

[63] M. Long, J. Wang, and G. Ding, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 1410–1417, Columbus, OH, USA, June 2014.

[64] K. Saenko, B. Kulis, and M. Fritz, "Adapting visual category models to new domains," in *Proceedings of the IEEE European Conference on Computer Vision(ECCV)*, pp. 213–226, Heraklion Greece, September 2010.

[65] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 770–778, Las Vegas USA, July 2016.

[66] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: a general framework for transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[67] J. Huang, A. Gretton, and K. Borgwardt, "Correcting sample selection bias by unlabeled data," in *Proceedings of the Advances in Neural Information Processing Systems(NIPS)*, pp. 601–608, Cambridge, MA, USA, December 2006.

[68] B. Quanz and J. Huan, "Large margin transductive transfer learning," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management(CIKM09)*, pp. 1327–1336, New York, NY, USA, November 2009.

[69] B. Zhang, Y. Liu, H. Yuan, L. Sun, and Z. Ma, "A joint unsupervised cross-domain model via scalable discriminative extreme learning machine," *Cognitive Computation*, vol. 10, no. 4, pp. 577–590, 2018.