Self-Supervised Autoregressive Domain Adaptation for Time Series Data

Mohamed Ragab, *Graduate Student Member, IEEE*, Emadeldeen Eldele, Zhenghua Chen, *Senior Member, IEEE*, Min Wu, *Senior Member, IEEE*, Chee-Keong Kwoh, and Xiaoli Li, *Senior Member, IEEE*

***Abstract*— Unsupervised domain adaptation (UDA) has successfully addressed the domain shift problem for visual applications. Yet, these approaches may have limited performance for time series data due to the following reasons. First, they mainly rely on the large-scale dataset (i.e., ImageNet) for source pretraining, which is not applicable for time series data. Second, they ignore the temporal dimension on the feature space of the source and target domains during the domain alignment step. Finally, most of the prior UDA methods can only align the global features without considering the fine-grained class distribution of the target domain. To address these limitations, we propose a SeLf-supervised AutoRegressive Domain Adaptation (SLARDA) framework. In particular, we first design a self-supervised (SL) learning module that uses forecasting as an auxiliary task to improve the transferability of source features. Second, we propose a novel autoregressive domain adaptation technique that incorporates temporal dependence of both source and target features during domain alignment. Finally, we develop an ensemble teacher model to align class-wise distribution in the target domain via a confident pseudo labeling approach. Extensive experiments have been conducted on three real-world time series applications with 30 cross-domain scenarios. The results demonstrate that our proposed SLARDA method significantly outperforms the state-of-the-art approaches for time series domain adaptation. Our source code is available at: https://github.com/mohamedr002/SLARDA.**

***Index Terms*— Autoregressive domain adaptation, ensemble teacher learning, self-supervised (SL) learning, time series data.**

I. Introduction

Time series classification (TSC) is a pivotal problem in many real-world applications including healthcare services and smart manufacturing [1], [2]. Several conventional approaches tried to learn the dynamics of the time series data for the classification task including dynamic time warping (DTW), hidden Markov models (HMMs), and artificial neural networks (ANNs) [3]. Yet, these approaches cannot cope with evolving complexity of real-world applications. Deep learning (DL) has shown notable success for time-series-based applications [1], [4], [5]. However, its success comes at the expense of laborious data annotation. Moreover, DL-based approaches always assume that the training data (i.e., source domain) and testing data (i.e., target domain) are drawn from the same distribution. This may not hold for real applications under dynamic environments, which is well-known as the domain shift problem.

The unsupervised domain adaptation (UDA) methods have achieved remarkable progress in mitigating the domain shift problem for visual applications [6], [7]. To avoid extensive data labeling, UDA is designed to leverage previously labeled datasets (i.e., source domain) and transfer knowledge to an unlabeled dataset of interest (i.e., target domain) in a transductive domain adaptation scenario [8]. One popular paradigm is to reduce the distribution discrepancy between the source and target domains via matching moments of distributions at different orders. For instance, the most prevailing method is based on the maximum mean discrepancy (MMD) as a distance, which is calculated via the weighted sum of the distribution moments [9]. Another paradigm for mitigating the distribution shift is inspired by generative adversarial networks (GANs). Particularly, it leverages adversarial learning between a feature extractor and a domain discriminator to find domain-invariant features [10], [11].

Nevertheless, applying UDA on time series data can be challenging for the following reasons. First, most of the existing approaches are specifically developed for visual data. Extending these approaches to time series could be suboptimal due to its temporal dynamics property. Second, most of the existing DA approaches rely on ImageNet pretraining as the initialization for the model, which is not applicable for time series data.

Manuscript received August 23, 2021; revised January 28, 2022; accepted June 8, 2022. This work was supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds under Grant A20H6b0151 and Career Development Award under Grant C210112046. The work of Mohamed Ragab and Emadeldeen Eldele was supported by the A*STAR SINGA Scholarship. *(Corresponding author: Zhenghua Chen.)*

Mohamed Ragab and Xiaoli Li are with the Institute for Infocomm Research, Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore 138632, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: mohamedr002@e.ntu.edu.sg; xlli@i2r.a-star.edu.sg).

Emadeldeen Eldele and Chee-Keong Kwoh are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: emad0002@ntu.edu.sg; asckkwoh@ntu.edu.sg)

Zhenghua Chen is with the Institute for Infocomm Research (I2R) and the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore (e-mail: chen0832@e.ntu.edu.sg).

Min Wu is with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore (e-mail: wumin@i2r.a-star.edu.sg).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3183252.

Digital Object Identifier 10.1109/TNNLS.2022.3183252

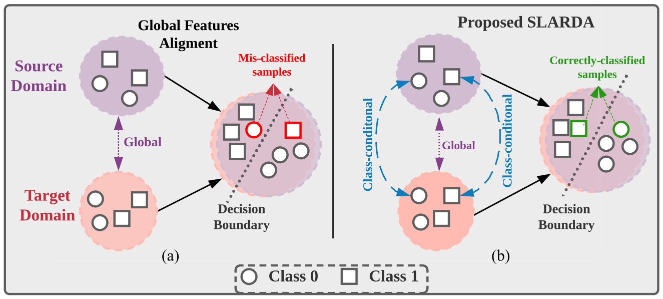

Fig. 1. Illustration of different domain alignment approaches. (a) Global distributions of the source and target domains are aligned, but the classes are misclassified between the source and the target. (b) In our proposed approach, both global feature alignment and class-conditional alignment are considered during the adaptation process to align the domains in feature and class levels.

Recently, few works have addressed domain adaptation for time series data by finding domain-invariant features [12], [13]. For instance, Purushotham *et al.* [12] used the variational recurrent networks to extract features and adversarial adaptation to align the source and target domains. Wilson *et al.* [13] leveraged information from multiple source domains to improve the performance on the unlabeled target domain. Both the approaches aim to find domain-invariant features by adversarially training the feature extractor to deceive the domain discriminator.

However, they ignore the temporal dimension when discriminating between the source and target features. As a result, the domain discriminator can be easily deceived without reaching a satisfactory alignment state. Furthermore, previous time series domain adaptation methods aim to only align the global distribution between domains, without considering the fine-grained class distributions within each domain as shown in Fig. 1.

To address all the aforementioned limitations, we propose a novel **Se**L**f-supervised **A**uto**R**egressive **D**omain **A**daptation (SLARDA) framework to boost the performance of time series UDA. First, unlike existing approaches that use self-supervised (SL) learning for unsupervised representation learning [14], [15], we design an SL pretraining approach to improve the transferability and generalization of the learned features in the source domain. With the lack of an ImageNet-like dataset for time series pretraining, we are the first to propose SL pretraining as a strong alternative for time series domain adaptation. Second, to incorporate temporal dependence of time series data during feature alignment, we propose a novel autoregressive domain adaptation approach. Particularly, an autoregressive domain discriminator is developed to consider the temporal dimension when classifying between the source and target features, which helps the feature extractor to learn better features.

Finally, to mitigate the class-conditional shift between the source and target domains, we propose a teacher-based approach with confident pseudo labels to guide the target model and correctly align the fine-grained source and target classes.

The main contributions of the proposed method can be summarized as follows.

1) We develop an SL pretraining for the source domain via a contrastive predictive loss to improve representation learning and transferability of the learned features. To the best of our knowledge, we are the first to propose SL pretraining for time series domain adaptation.
2) To consider temporal dependence among the source and target features during domain alignment, we design an autoregressive domain discriminator for time series, which can boost the performance of feature learning and domain alignment.
3) We propose an ensemble teacher model confident pseudo labeling approach to generate reliable pseudo labels in the target domain for domain alignment, which can mitigate the class-conditional shift between the source and target domains.

II. Related Works

In this section, we will present the recent literature of general UDA and the existing techniques of time series domain adaptation.

A. Unsupervised Domain Adaptation

UDA, which is a subset of transfer learning, attempts to address the domain shift problem of labeled source and unlabeled target domains. The existing approaches can be classified into two major categories, namely, discrepancy-based methods and adversarial learning-based methods. Discrepancy-based approaches intend to align the two domains via minimizing statistical distances. For instance, some methods minimized MMD [16] to find invariant features between the two domains [17]–[19]. Chen *et al.* [20] presented a high-order MMD to match high-order moments between the source and target domains. Correlation alignment methods try to mitigate the domain shift by matching the second-order statistics between the source and target domains [21], [22]. In [23], central moment discrepancy (CMD) was proposed to align high-order central moments to obtain transferable features between the source and target domains.

Inspired by GANs, adversarial UDA methods optimize a feature extraction network to produce invariant features of the source and target domains such that a well-trained domain classification network cannot distinguish between them. For example, Ganin *et al.* [24] used a reverse gradient layer to adversarially train the domain discriminator and the feature extractor. While Tzeng *et al.* [25] proposed an adversarial discriminative domain adaptation (ADDA) approach via untying source and target networks and using GAN-based inverted labels' loss İn Wasserstein distance guided representation learning (WDGRL), a theoretically justified Wasserstein distance was used to tackle the stability issue of the GAN-based objective. Long *et al.* [27] proposed conditional adversarial domain adaptation (CDAN) via incorporating the task knowledge with features during the domain alignment step. The decision boundary iterative refinement training (DIRT) approach used virtual adversarial training and conditional entropy to align the source and target domains [28]. However, most of these approaches adopt conventional adversarial

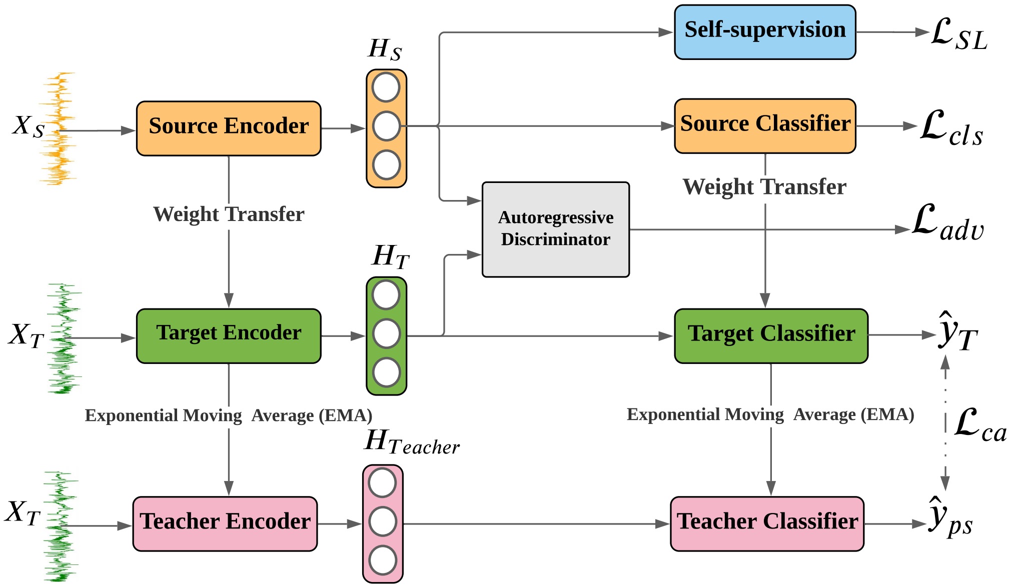

Fig. 2. Overall framework of the proposed SLARDA.

training on a vectorized feature space of the source and target domains, disregarding the temporal information during the domain alignment step. Differently, our approach leverages autoregressive domain discriminator to consider the temporal information during alignment, leading to a better discriminative adaptation between the source and target domains.

On the other hand, another related line of research has leveraged self-ensemble techniques to provide pseudo labels for the unlabeled target domain [29], [30]. Yet, these approaches cannot predict high-quality pseudo labels at the early stage of training due to lack of proper initializing. Differently, our approach is initialized by an SL pretrained model on the source domain, which can produce robust pseudo labels at both early and late stages of the adaptation step.

B. Domain Adaptation for Time Series Data

Few studies have investigated UDA for time series data. For instance, Purushotham *et al.* [12] used variational recurrent auto-encoder with adversarial training to mitigate the domain shift problem. Wilson *et al.* [31] proposed multi-source domain adaptation via a gradient reversal layer for human activity recognition (HAR) tasks. Most of these approaches directly adopted image-based UDA techniques for time series, which may be suboptimal as they ignored temporal dependence during domain alignment. Differently, our approach explicitly addresses temporal dependence during both feature learning and domain alignment steps by designing a novel SL pretraining and an innovative autoregressive domain discriminator, respectively. In addition to global feature alignment, our approach also adapts the fine-grained class distributions between the source and target domains, as shown in Fig. 1.

III. Methodology

A. Problem Formulation

In this work, we address the problem of UDA for time series data. Given a labeled source domain $\mathcal{D}_\mathrm{S} = \{X_S^i, \boldsymbol{y}_S^i\}_{i=1}^{n_S}$ with n_S samples, and an unlabeled target domain $\mathcal{D}_\mathrm{T} = \{X_T^j\}_{j=1}^{n_T}$, with n_T samples. The source and target domains are sampled from different distributions $P_S(X)$ and $P_T(X)$, respectively, where $P_S(X) \neq P_T(X)$. The samples of the source and target domains can be either univariate or multivariate time series. Formally, we have input source sample $X_S^i \in \mathbb{R}^{M\times K}$ with M channels and K time steps, and its corresponding label $\boldsymbol{y}_S^i \in \mathbb{R}^C$, where C is the number of classes. Our main goal is to design a predictive model that can accurately predict the label $\boldsymbol{y}_T^i$ of the unlabeled target sample $X_T^i \in \mathbb{R}^{M\times K}$.

B. Overview of SLARDA

Fig. 2 shows the proposed SLARDA framework, which is composed of three main components: 1) an SL pretraining module to improve the transferability of the learned source features; 2) an autoregressive discriminator model to explicitly consider temporal dependence among the source and target features during domain alignment; and 3) a class-conditional alignment module to address the class-conditional shift and adapt the fine-grained distribution of different categories for the unlabeled target domain. We will elaborate on each component in more detail in Sections III-C–III-E.

C. Self-Supervised Learning for Source Pretraining

Most of the existing UDA approaches initialize the target domain model by a supervised pretrained model on the labeled source domain. We argue that the learned representation from supervised objectives tends to be more specific toward a single domain and may have limited transferability to out-of-distribution domains. Inspired by van den Oord *et al.* [14], we propose a novel SL auxiliary task to improve the transferability of learned representations in the source domain. Specifically, given the encoded latent features, we pick a time step t and train the model to predict future time steps given the past ones, as shown in Fig. 3. Thus, the model will learn

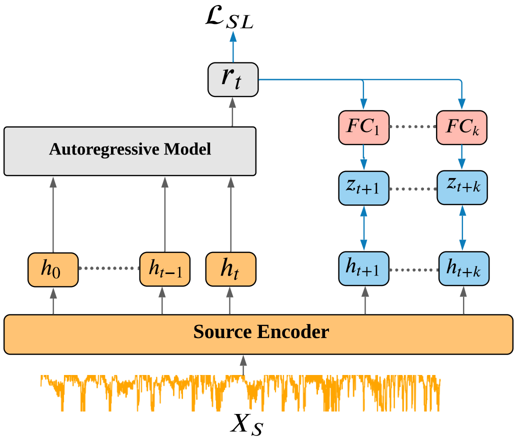

Fig. 3. SL learning in the source domain.

more general features that encompass the shared information among multiple time steps.

To map the input data into a latent space, we first design a 1-dimensional convolutional neural network (1-D-CNN) encoder model. Then, we leverage an autoregressive model to summarize the latent features into a context vector. Formally, given the output latent features from the encoder $H_{\leq t} = \{\boldsymbol{h_0}, \ldots, \boldsymbol{h_t}\}$, they are fed into an autoregressive model to obtain the context vector $\boldsymbol{r_t}$. Subsequently, we pass the context vector to a parameterized fully connected mapping layer FC_k to predict the future latent feature $z_{t+k} = FC_k(\boldsymbol{r_t})$.

To measure similarity between $\boldsymbol{h_{t+k}}$ and z_{t+k}, we leverage a dot product similarity measure between the predicted vector and the true latent future. The similarity matching function can be formulated as follows:

$$\phi_k(\boldsymbol{h_{t+k}}, z_{t+k}) = \exp\left(\boldsymbol{h}_{t+k}^{\top} z_{t+k}\right) \tag{1}$$

where ϕ_k is a log bilinear model. Here, we jointly optimize the encoder model, the autoregressive model, and the log bilinear model via the contrastive objective to maximize the similarity between the predicted future z_{t+k} and its corresponding true future latent feature $\boldsymbol{h_{t+k}}$. While the true latent feature changes during training, the predicted vector varies correspondingly to preserve their relationship and stabilize the training process.

This auxiliary task of predicting the future time steps via SL learning helps better model the temporal dependence of the input samples and produce more transferable features from the source domain. We formulate the problem as a binary classification problem between positive and negative samples. In our case, the future latent of the same sample is considered as a positive pair, while the future latent of all other samples in the mini-batch is considered as negative pairs. This can be formalized as follows:

$$\mathcal{L}_{\text{SL}} = -\mathop{\mathbb{E}}_{H_b}\left[\log \frac{\phi_k(\boldsymbol{h_{t+k}}, FC_k(\boldsymbol{r_t}))}{\sum_{\boldsymbol{h_j}\in H_b}\phi_k(\boldsymbol{h_j}, FC_k(\boldsymbol{r_t}))}\right] \tag{2}$$

where H_b represents a mini-batch of samples.

We design the aforementioned SL loss to optimize the source encoder E_S on the source domain data. Concurrently, we train the encoder model E_S to perform well on the main classification task via cross-entropy loss on the labeled source domain data, shown as follows:

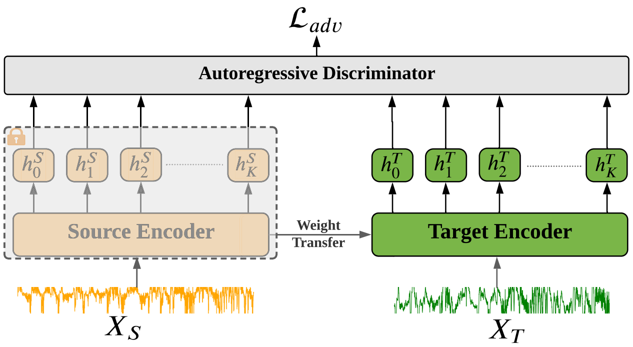

Fig. 4. Autoregressive discriminator.

$$\mathcal{L}_{\text{cls}} = -\mathbb{E}_{X_S\sim P_S}\left[\boldsymbol{y}_S^{\top}\log(C_S(E_S(X_S)))\right]. \tag{3}$$

Finally, we jointly train the source encoder E_S with the SL task along with the supervised objective to produce more transferable features as follows:

$$\min_{E_S} \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{SL}}. \tag{4}$$

D. Autoregressive Domain Adaptation

Adversarial domain adaptation has achieved remarkable performance for visual applications. However, the design of discriminator networks in the existing methods does not consider temporal dependence in the feature space of the time series data, resulting in a limited performance for domain alignment.

To address this critical issue, we propose an autoregressive domain discriminator to exhibit the temporal dynamic behavior of time series data during domain alignment, as shown in Fig. 4.

The autoregressive discriminator D_{AR} consists of two main components. First, an autoregressive network f_{AR} that encodes the temporal dependencies among both the source and target features into vector representations, shown as follows:

$$f_{AR}(\boldsymbol{h_0}, \ldots, \boldsymbol{h_K}) = p(\boldsymbol{h_K} \mid \boldsymbol{h_{<K}}) \tag{5}$$

where $p(\boldsymbol{h_K} \mid \boldsymbol{h_{<K}})$ is the conditional distribution among different time steps of the sequential features.

Second, a binary classification network f_D is applied on the summarized feature vectors to classify between the source and target features. Thus, the autoregressive discriminator can be represented as $D_{AR} = f_D(f_{AR}(\cdot))$. A detailed explanation of the autoregressive discriminator and its architecture are discussed in Section IV-B. To align the source and target domains, we first freeze the SL pretrained source model and transfer its weights to the target model. Then, we adversarially train the autoregressive domain discriminator against the target model to produce domain-invariant features. The autoregressive discriminator is optimized to discern between the source

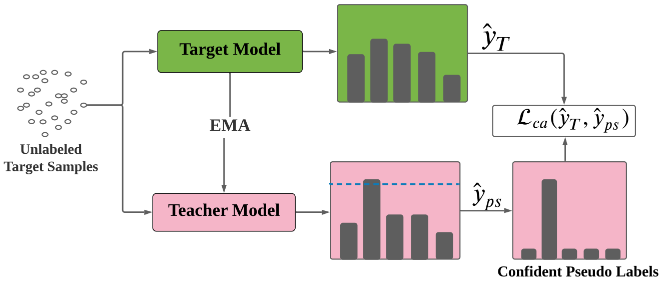

Fig. 5. Class-conditional alignment via teacher model.

and target features, which can be formalized as

$$\begin{aligned}\min_{D_{AR}} \mathcal{L}_{\mathrm{D}} = &-\mathbb{E}_{X_S \sim P_S}\left[\log D_{AR}(H_S)\right] \\ &- \mathbb{E}_{X_T \sim P_T}\left[\log(1 - D_{AR}(H_T))\right]\end{aligned} \tag{6}$$

where $H_S = E_S(X_S)$ and $H_T = E_T(X_T)$ are the temporal output features from the source and target encoders, respectively, and D_{AR} represents the autoregressive discriminator network. Concurrently, we train the target encoder to confuse the discriminator by mapping the target features to be similar to the source ones. The target encoder loss can be formalized as

$$\min_{E_T} \mathcal{L}_{\mathrm{adv}} = \mathbb{E}_{X_T \sim P_T}\left[\log(1 - D_{AR}(H_T))\right]. \tag{7}$$

E. Class-Conditional Alignment via Teacher Model

Autoregressive domain adaptation can successfully align the marginal distribution of the source and target temporal features. However, it can still misalign the different classes among the source and target domains due to the class-conditional shift. To overcome this issue, we develop a teacher-based confident pseudo labeling approach to adapt the fine-grained distribution of different categories among the source and target domains.

1) Teacher Model: Inspired by the mean teacher for semi-supervised learning [29], we design an ensemble teacher model f_ψ to produce robust pseudo labels for the unlabeled target domain, as shown in Fig. 5. We obtain the weights of the teacher model $\mathcal{W}_\psi$ by applying the exponential moving average (EMA) over the target model parameters $\mathcal{W}_{\theta_{\mathrm{T}}}$ across successive training steps. The momentum updates of the teacher model parameters can be represented as follows:

$$\mathcal{W}_\psi = \alpha \mathcal{W}_\psi + (1 - \alpha)\mathcal{W}_{\theta_{\mathrm{T}}} \tag{8}$$

where α is a momentum parameter that controls the speed of the weight updates of the teacher model. Given the teacher model f_ψ, we obtain the output predictions as follows:

$$\boldsymbol{p}_\psi = f_\psi(X_T) \tag{9}$$

$$\hat{\boldsymbol{y}}_\psi = \mathrm{softmax}\left(\boldsymbol{p}_\psi\right) \tag{10}$$

where $\boldsymbol{p}_\psi$ are the output predictions of the teacher model, and $\hat{\boldsymbol{y}}_\psi$ are the corresponding probabilities.

2) Confident Pseudo Labels: To further refine the predicted labels of the teacher model, we only preserve the confident labels that are above a predefined confidence threshold ζ. This can be formalized as follows:

$$\hat{y}_{\mathrm{ps}} = \hat{\boldsymbol{y}}_\psi\left[\max\left(\boldsymbol{p}_\psi\right) > \zeta\right] \tag{11}$$

where $\hat{\boldsymbol{y}}_{\mathrm{ps}}$ are the retained confident pseudo labels. To align the class-conditional distribution, we leverage the obtained confident pseudo labels to train the target model by a cross-entropy loss

$$\mathcal{L}_{\mathrm{ca}} = -\mathbb{E}_{X_T \sim P_T}\left[\sum_{k=1}^{K} \mathbb{1}_{[y_{ps}=k]} \log\left(\hat{\boldsymbol{y}}_T^k\right)\right] \tag{12}$$

where $\mathcal{L}_{\mathrm{ca}}$ is the class-conditional alignment loss, and $\hat{\boldsymbol{y}}_T = C_T(E_T(X_T))$ are the predicted labels by the target classifier C_T.

Algorithm 1: Autoregressive Domain Adaptation

```
**Input:** Source domain: $\mathcal{D}_S = \{X_S^i, y_S^i\}_{i=1}^{n_S}$
Target domain:$\mathcal{D}_T = \{X_T^i\}_{i=1}^{n_T}$
**Output:** Trained target encoder $E_T$
$E_S \leftarrow$ Pretrained source encoder
$E_T \leftarrow$ Initialize with $E_s$ parameters
$f_\psi \leftarrow$ Teacher model
$D_{AR} \leftarrow$ Autoregressive Domain Discriminator
**for** *number of iterations* **do**
    1) Sample mini-batch of $m$ source samples $X_S \sim P_S$
    2) Sample mini-batch of $m$ target samples $X_T \sim P_T$
    3) Extract source features: $H_S = E_S(X_S)$
    4) Extract target features: $H_T = E_T(X_T)$
    5) Feed $H_S$ and $H_T$ to $D_{AR}$
    6) Assign labels of ones to $H_S$ and zeros to $H_T$
    7) Compute discriminator loss $\mathcal{L}_D$ by Eq. 6
    8) Update $D_{AR}$ by $\mathcal{L}_D$
    9) Invert the labels of $H_T$
    10) Compute $\mathcal{L}_{adv}$ with the inverted labels by Eq. 7
    11) Pass $X_T$ to the teacher model $f_\psi$
    12) Obtain the confident pseudo labels by Eq. 11
    13) Compute the class-conditional loss $\mathcal{L}_{CA}$ by Eq. 12
    14) Update $E_T$ using both $\mathcal{L}_{adv}$ and $\mathcal{L}_{CA}$ via Eq. 13
**end**
```

F. Overall Objective Function

In our approach, we jointly optimize the target encoder E_T to minimize both the autoregressive domain adaptation loss and class-conditional alignment loss in an end-to-end learning manner. Our overall objective can be formalized as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{overall}} &= \mathcal{L}_{\mathrm{adv}} + \lambda \mathcal{L}_{\mathrm{ca}} \\ &= \min_{E_T} \mathbb{E}_{X_T \sim P_T}\big[\log(1 - D_{AR}(E_T(X_T))) \\ &\qquad - \lambda \hat{\boldsymbol{y}}_{ps}^{\mathsf{T}} \log(C_T(E_T(X_T)))\big]\end{aligned} \tag{13}$$

where λ is the weight of the class-conditional loss. Algorithm 1 shows the detailed procedures of our autoregressive adaptation approach.

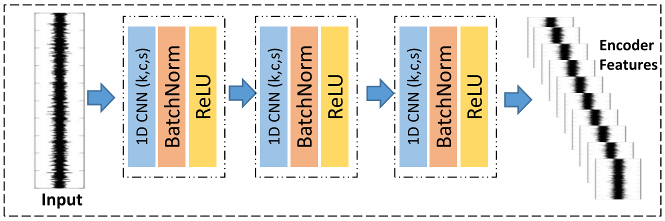

Fig. 6. Architecture of feature extraction network.

G. Testing on the Target Domain

In the testing phase, we only use the pretrained target encoder E_T and target classifier C_T while ablating both the transformer model and the autoregressive network, ensuring consistency of the backbone network when evaluating against other UDA algorithms. Given the test data from the target domain, the encoder model E_T will extract the target adapted features. Subsequently, the target classifier C_T will predict the corresponding class predictions

$$\hat{\mathbf{p}}_{\text{test}} = \mathbb{E}_{X_{\text{test}} \sim P_{\text{test}}}[\sigma(C_T(E_T(X_{\text{test}})))] \tag{14}$$

$$\hat{y}_{\text{test}} = \operatorname{argmax}(\hat{\mathbf{p}}_{\text{test}}) \tag{15}$$

where $\sigma(a)_i = (e^{a_i} / \sum_{j=1}^{k} e^{a_j})$ represents the softmax function, $\hat{\mathbf{p}}_{\text{test}}$ is the output probability vector, and $\hat{y}_{\text{test}}$ is the predicted label.

IV. Experiments

A. Datasets

We evaluate our SLARDA on three real-world time series applications including HAR, sleep stage classification (SSC), and machine fault diagnosis (MFD). Table I shows the summarized details about each dataset. To calculate the total number of samples for each dataset, we summed all the training and testing parts for all the domains. We will elaborate further about each dataset in Sections IV-A1–IV-A3.

1) HAR Dataset: The Opportunity[1] is a benchmark dataset for HAR [32].

In our experiments, following the existing baselines in the data challenge [33], we only selected 113 sensors. The data annotations comprised of two main levels: 1) locomotion represents low-level tasks such as sitting, standing, walking, and lying down and 2) gestures: high-level tasks which comprised 17 different actions. We only adopted low-level annotations, and hence, we have four main classes (i.e., sitting, standing, walking, and lying down). The missing values in the data have been filled via the linear interpolation approach. Four users have been involved in the experiments, where the data from each user represent one domain. We aim to apply domain adaptation across different users. To construct the training samples for each user, we adopted sliding window approach with a window size of 128 and overlapping of 50%, as in [33].

2) SSC Dataset: SSC includes classifying electroencephalogram (EEG) signals into five stages: wake (W), nonrapid eye movement (N1, N2, and N3), and rapid eye movement (REM). In our experiments, we evaluate our domain adaptation method with cross-dataset scenarios. Therefore, we use three real-world datasets, namely, Sleep-EDF,[2] SHHS-1, and SHHS-2,[3] with sampling rates of 100, 125, and 250 Hz, respectively. The different sampling rates incur significant domain shifts among datasets. Notably, we down-sampled the data from SHHS-1 and SHHS-2 such that their sequence lengths become the same as Sleep-EDF (i.e., 3000 time steps).

TABLE I
Dataset Statistics

	HAR Dataset	SSC Dataset	MFD Dataset
Domain names	(A,B,C,D)	(EDF, SH1, SH2)	(H,I,J,K)
# Training Samples	8224	81740	32736
# Testing Samples	1426	30390	10912
# Channels	113	1	1
# Classes	4	5	3
Sequence Length	128	3000,3750,7500	5120

TABLE II
Parameter Setting for the CNN Encoder and the Autoregressive Feature Extractor

Parameters	HAR Dataset	SSC Dataset	MFD Dataset
Encoder model:			
# of Layers	3	3	5
# of Channels (c)	16	32	8
Kernel size (k)	8	25	32
# stride (s)	2	3	2
Transformer (Adaptation):			
FC Layer	64	512	128
Input Channels	16	64	8
# of Layers	8	8	4
# Num of Heads	2	4	4
GRU (Pretraining):			
Hidden Dimension	16	64	64
Input Dimension	16	128	8
# of Layers	1	1	1

3) MFD Dataset: The MFD[4] dataset contains sensor readings of bearing machine under four different operating conditions, with each having three different classes, i.e., healthy, inner bearing damage, and outer bearing damage. Each operating condition refers to different operating parameters, including rotational speed, load torque, and radial force [34]. In our experiments, each operating condition is considered as one domain. Eventually, we can perform 12 cross-condition scenarios for domain adaptation. To construct the data samples for each domain, we adopted a sliding window to segment the data into small segments. We set the window size of 5120 and shifting size of 4096, as in [35].

B. Model Architectures

Our algorithm has two main models, namely, the feature extractor model and the autoregressive discriminator model. We provide further details about the architecture of each model in Sections IV-B1 and IV-B2.

[1] https://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition

[2] physionet.org/content/sleep-edf/1.0.0/

[3] https://sleepdata.org/datasets/shhs

[4] https://mb.uni-paderborn.de/en/kat/main-research/datacenter/bearing-datacenter/data-sets-and-download

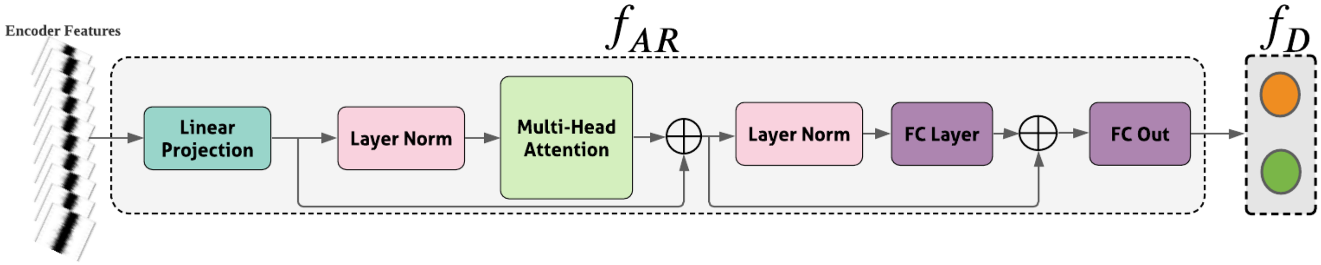

Fig. 7. Architecture of autoregressive discriminator.

1) Feature Extractor: We adopt the 1-D-CNN architecture to extract features for the three datasets, as shown in Fig. 6. Due to the large variation among different applications, different kernel sizes and different number layers are selected for each dataset. Table II shows the detailed encoder parameters for each dataset. We adopted the commonly used architecture in the literature for each application. Particularly, for the MFD dataset, we used a five-layer 1-D-CNN with a kernel size of 32, as in [35]. While for both SSC and HAR, we used a three-layer 1-D-CNN with a kernel size of 25 and 8, respectively, as in [36].

2) Autoregressive Discriminator: We use the transformer model [38] to model the temporal dependence among time steps for both the source and target domains. The transformer model uses SL, which has an advantage over other sequential model such as recurrent neural networks in terms of efficiency and speed [39]. The model architecture is shown in Fig. 7. First, a linear projection layer is used to map from the input dimension to the hidden dimension of the transformer model. Then, layer normalization is applied to the input features. After that, a multihead SL is used to normalized features. Table II shows the detailed parameters for the autoregressive discriminator. As each dataset has different characteristics, we adopt different parameters for each dataset.

3) Autoregressive Network (Pretraining): In our pretraining step, we leverage gated recurrent network (GRU) to summarize the latent features into a context vector. Particularly, we used a single-layer GRU network for all the datasets, while input and hidden dimensions vary according to each dataset. Table II illustrates the detailed architectures of the GRU network on each dataset.

C. Implementation Details

In our experiments, we use labeled data from the source domain and unlabeled data from the target domain, following the standard protocol of UDA [27], [28]. All the experiments have been conducted using PyTorch 1.7 on NVIDIA GeForce RTX 2080 Ti GPU. We use a batch size of 512 for MFD and 128 for HAR and SSC. We adopt Adam optimizer with a learning rate of 1e−3 for SSC and 1e−4 for HAR and MFD, and a weight decay of 3e−4, as in [35], [36], and [39]. For the teacher model, the conditional alignment weight λ is set to 0.005, the momentum of updating the teacher model α is set to 0.996, and the confidence threshold ζ for pseudo labels is set to 0.9. For all the datasets, we randomly split the data into 60% for training, 20% for validation, and 20% for testing. We report the mean value of five consecutive runs with different random seeds.

D. Results

1) Baselines: To evaluate the performance of the proposed SLARDA, we have compared against some strong baselines. As most of the state-of-the-art approaches are implemented for image-related datasets, we reimplement nine state-of-the-art methods to fit our time series datasets. In addition, to promote fair evaluation, we adopt our backbone architecture which works well on time series for all the baseline methods. In particular, we compare our SLARDA with the following state-of-the-art methods: deep adaptation networks (**DAN**) [17], **WDGRL** [26], **Deep CORAL** [22], minimum discrepancy domain adaptation (**MDDA**) [37], **HoMM** [20], domain adversarial neural networks (**DANN**) [24], CDAN [27], and virtual adversarial domain adaptation (**VADA**) [28]. It is worth noting that some baselines failed to outperform Source Only on some datasets as they are not specifically designed for time series data. Hence, we only reported the methods that outperform Source Only for each dataset. In Tables III–V, the best performance is **bolded**, while the second best is underlined.

2) Results on the HAR Dataset: We first evaluate our proposed SLARDA on the HAR dataset which contains data from four subjects, namely, A, B, C, and D. Table III shows the evaluation results on 12 cross-domain scenarios. Our proposed approach achieves the best performance on six cross-domain scenarios and the second best on five cross-domain scenarios. Besides, the proposed SLARDA significantly outperforms the benchmark methods in the overall performance with a 2.62% improvement over the second best method, i.e., DIRT. It is worth noting that the adaptation sometimes may deteriorate the performance when the domain gap is small as in the B → A scenario.

3) Results on the SSC Dataset: The SSC dataset contains three domains, namely, EDF, SH1, and SH2, with sampling rates of 100, 125, and 250 Hz, respectively. Table IV shows the results on six cross-domain scenarios. Overall, our SLARDA approach performs best on five out of six cross-domains scenarios with 5% average improvement over the state-of-the-art method. Notably, our approach performs best when mapping from higher resolution to lower resolution datasets (i.e., SH2 → SH1, SH2 → EDF, and SH1 → EDF). The

TABLE III

RESULTS ON HAR DATASET AMONG 12 CROSS-DOMAIN SCENARIO (ACCURACY %)

Method	A→B	A→C	A→D	B→A	B→C	B→D	C→A	C→B	C→D	D→A	D→B	D→C	Average	P-Value
Source Only	66.55	**71.46**	60.40	**83.78**	72.02	27.68	56.04	30.97	53.43	47.09	64.79	58.58	57.73	1.2E-06
DANN [24]	75.92	59.67	63.01	81.04	65.41	49.10	70.46	72.44	57.72	68.95	61.80	62.70	65.68	1.1E-05
DAN [17]	75.09	61.72	**66.64**	81.11	66.66	47.12	75.59	71.94	58.92	69.59	66.28	67.27	67.33	8.7E-04
WDGRL [26]	76.79	60.09	64.66	81.50	62.88	52.14	64.56	60.46	59.80	68.75	64.22	64.71	65.05	7.3E-04
MDDA [37]	72.88	61.23	55.38	76.12	61.40	50.38	54.10	60.33	56.37	70.63	53.63	63.64	61.34	7.6E-06
HoMM [20]	73.99	58.74	60.94	76.76	61.59	47.34	71.36	68.38	57.63	65.21	64.82	58.07	63.74	3.3E-06
CDAN [27]	77.53	60.60	53.89	77.59	63.23	44.60	53.76	50.45	59.04	70.61	**71.06**	61.96	62.03	9.2E-05
DIRT [28]	70.03	65.14	60.17	74.04	65.88	56.62	**78.92**	69.49	58.95	**71.97**	73.55	**76.87**	68.47	3.0E-02
SLARDA	**79.66**	63.09	65.87	83.53	**76.25**	**60.35**	78.18	**77.42**	**59.87**	71.58	66.85	70.42	**71.09**	-

TABLE IV

EXPERIMENTAL RESULTS ON SSC DATASET AMONG SIX CROSS-DOMAIN SCENARIO (ACCURACY %)

Method	EDF→SH1	EDF→SH2	SH1→EDF	SH1→SH2	SH2→EDF	SH2→SH1	Average	P-Value
Source Only	49.12	55.98	67.50	52.27	58.33	76.83	60.00	4.2E-05
DAN [17]	59.98	57.98	70.68	60.35	65.69	77.78	65.41	6.7E-04
Deep Coral [22]	61.43	58.86	71.05	60.85	67.33	77.51	66.17	8.3E-04
DANN [24]	57.91	59.01	72.30	57.31	66.57	76.06	64.86	6.5E-04
CDAN [27]	62.76	63.62	72.94	**67.72**	73.39	77.71	69.69	3.1E-03
DIRT [28]	59.92	57.80	75.92	63.66	68.91	73.82	66.67	2.2E-03
SLARDA	**68.19**	**64.71**	**82.73**	67.01	**82.36**	**81.91**	**74.49**	-

TABLE V

EXPERIMENTAL RESULTS ON FAULT DIAGNOSIS DATASET AMONG 12 CROSS-DOMAIN SCENARIO (ACCURACY %)

Method	H→I	H→J	H→K	I→H	I→J	I→K	J→H	J→I	J→K	K→H	K→I	K→J	Average	P-Value
Source Only	25.70	36.18	25.81	36.62	71.74	**99.89**	32.26	90.91	93.81	38.09	98.90	78.23	60.68	1.6E-07
Deep Coral [22]	38.05	47.07	45.37	41.30	66.98	92.63	36.92	82.31	81.60	42.80	96.29	69.48	61.73	9.8E-07
DAN [17]	50.86	53.57	56.30	38.86	65.16	98.82	26.13	91.09	87.97	45.31	98.27	69.71	65.17	1.1E-04
WDGRL [26]	40.67	51.70	52.02	51.37	72.56	94.89	52.73	67.73	76.74	51.28	97.98	65.79	64.62	2.6E-07
MDDA [37]	38.15	48.65	49.14	35.35	72.28	97.79	23.56	85.53	81.61	39.60	**99.42**	70.86	61.83	1.9E-04
HoMM [20]	46.78	45.47	51.28	41.15	75.19	98.43	34.17	84.97	83.35	44.82	98.99	75.43	65.00	4.9E-07
CDAN [27]	52.95	61.38	53.55	31.64	74.25	99.66	55.20	91.98	93.14	42.08	98.71	72.90	68.95	1.8E-04
DIRT [28]	47.21	54.13	51.46	45.71	**85.91**	98.26	31.06	**99.28**	**99.14**	45.64	99.23	**84.66**	70.14	2.3E-04
SLARDA	**84.38**	**75.70**	**96.04**	**86.60**	79.47	99.68	**75.59**	90.10	92.94	**91.17**	97.40	80.69	**87.48**	-

reason is that our SLARDA, in contrast to the baseline approaches, better exploits the rich temporal information in the feature space to improve the alignment between domains. For example, in scenarios SH2 → SH1 and SH2 → EDF, our approach significantly outperforms the second best method with improvements of nearly 9% and 4%, respectively. On the other hand, adapting from domains with lower sampling rates to the ones with higher sampling rates can be quite challenging due to the extrapolation effect. Yet, our SLARDA can still perform best in EDF → SH1 and EDF → SH2 and second best in SH1 → SH2.

4) Results on the MFD Dataset: The MFD dataset has four different working conditions, denoted as H, I, J, and K. Table V shows the results on the 12 cross-condition scenarios. Similarly, our proposed approach outperforms baselines in 6 out of 12 cross-domain scenarios with an average improvement of 17.34% over the second best method, i.e., VADA. Clearly, SLARDA outperforms the benchmark methods on the challenging transfer tasks with large domain shifts, e.g., H → I, H → J, and H → K.

5) Statistical Significance: We performed a comparative analysis on the statistical significance of our SLARDA approach against all the other baselines. Specifically, we leveraged Wilcoxon signed-rank test to measure the P-value of our SLARDA against other baseline methods [38]. Tables III–V show the P-value of our SLARDA against other baselines in the HAR, SSC, and MFD datasets, respectively. Clearly, for all the baseline methods, our SLARDA achieves P-value <0.05 and is significantly better than other approaches on all the datasets with 95% confidence level.

E. Ablation Study and Sensitivity Analysis

1) Ablation Study: To show the contribution of each component in our proposed method, we conduct an ablation study on the MFD dataset. The model variants are defined as follows.

1) ***SLARDA (w/o SL):*** We replace SL pretraining with conventional supervised pretraining.
2) ***SLARDA (w/o AR):*** We replace the autoregressive domain discriminator with a conventional fully

TABLE VI
TOTAL TRAINING TIME OF EACH APPROACH ON FAULT DIAGNOSIS DATASET (s)

Method	DAN	Deep Coral	HoMM	MMDA	DANN	CDAN	WDGRL	DIRT	SLARDA
Computational Time	1,125	1,411	1,793	1,427	1,467	1,862	2,736	2,929	1,765

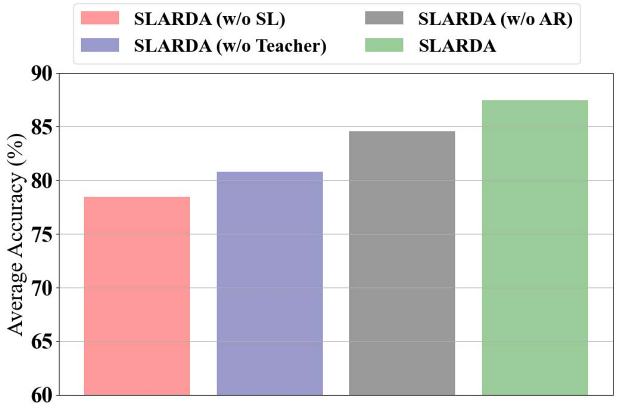

Fig. 8. Ablation Study on the MFD dataset.

connected discriminator network trained with standard GAN loss.

3) ***SLARDA (w/o Teacher)****:* We remove the conditional alignment component from the SLARDA model.

4) ***SLARDA (full)****:* We include all the model's components.

Fig. 8 shows the average results of different variants for the 12 cross-domain scenarios. It can be seen that removing SL pretraining can be detrimental to the performance with more than 8% degradation. This is because removing SL can reduce the feature's transferability between domains, which can also affect the efficacy of our remaining modules (i.e., AR and teacher). Similarly, removing the class-conditional alignment (i.e., Teacher) also has a significant impact on the model performance. Finally, adding the autoregressive component by addressing the temporal features can improve the overall performance by about 3%. To sum up, this ablation clearly shows the effectiveness of each component in our SLARDA model.

2) Sensitivity Analysis of the Class Conditional Loss: There are some key parameters in the proposed approach, which may have a significant impact on model performance. One of the key parameters is λ in (12), which indicates the contribution of the class-conditional loss. Here, we investigate the impact of this key parameter on model performance. We conduct experiments on the MFD dataset and report the average performance of 12 cross-domain scenarios. We vary the weight parameter λ from 0.0001 to 1. Fig. 9 shows the results of our proposed SLARDA with different values of λ. Clearly, gradually increasing λ improves the performance of our SLARDA. Yet, over-weighting the class-conditional loss deteriorates the performance as the predicted pseudo labels can still be noisy. In a nutshell, our SLARDA approach performs best with λ values between 0.001 and 0.005.

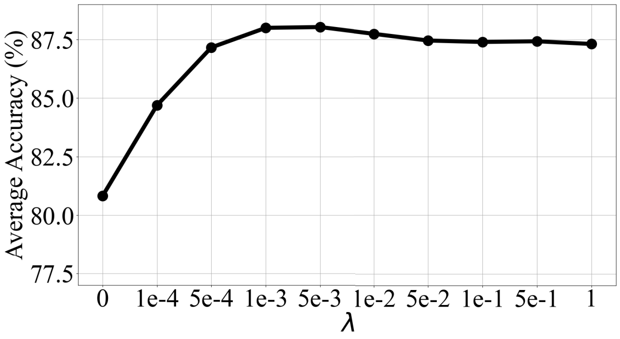

Fig. 9. Sensitivity analysis of class-conditional loss in (12).

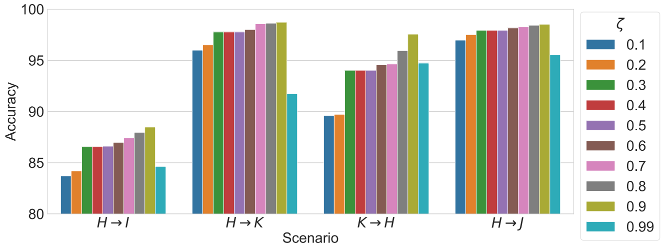

Fig. 10. Sensitivity analysis of confidence threshold parameter ζ.

3) Sensitivity Analysis of the Confidence Threshold: We conducted a sensitivity analysis experiment to measure the sensitivity of our approach to the confidence threshold parameter. Fig. 10 shows the evaluation performance on four randomly selected cross-domain scenarios for the MFD dataset. We varied the confidence threshold from 0.1 to 0.99 and reported the corresponding performance. Clearly, lower values of the confidence threshold can degrade the generalization performance across domains as noisy pseudo labels can be used to train the target model. In comparison, higher confidence thresholds consistently yield better performance across the four experimented cross-domain scenarios. However, a very large confidence threshold, e.g., 0.99, can deteriorate the performance on cross-domain scenarios, as we may not be able to find sufficient amount of pseudo labels that satisfy this large threshold.

F. Computational Complexity

To evaluate the time complexity of our proposed approach against other baseline methods, we calculated the total running time over all the cross-domain scenarios on the Fault Diagnosis dataset, as shown in Table VI. Generally, discrepancy-based approaches (i.e., DAN, Deep Coral, HoMM, and MMDA) have lower computational complexity, when compared with adversarial-based methods. Among all the adversarial-based methods, our SLARDA approach has the second lowest computational cost with a total computational time of 1765 s.

V. Conclusion

In this article, we proposed a time series domain adaptation method, which explicitly considers temporal dynamics of data during both feature learning and domain alignment. In particular, we showed that the proposed SL pretraining of the source domain model can produce more transferable features than supervised pretraining. Hence, we suggest adopting SL pretraining for time series domain adaptation methods. Second, we proved that addressing the temporal dependence during domain alignment can significantly boost performance. Finally, we demonstrated that providing confident pseudo labels can successfully address the class-conditional shift of time series data. The efficacy of the proposed method has been verified using three real-world time series datasets. We believe that our approach can promote the direction of time series domain adaptation. Our approach can still be limited as it assumes the availability of rich-labeled source domain data, which may be laborious. Hence, in our future works, we aim to design SL learning [40] to learn representations with few labeled data and a large amount of unlabeled in the source domain.

References

[1] A. Gharehbaghi and M. Lindén, "A deep machine learning method for classifying cyclic time series of biological signals using time-growing neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4102–4115, Sep. 2018.

[2] A. Osmani, M. Hamidi, and S. Bouhouche, "Monitoring of a dynamic system based on autoencoders," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1836–1843.

[3] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.

[4] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[5] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1407–1418, May 2018.

[6] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2387–2397, Jul. 2020.

[7] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[9] W. Wang *et al.*, "Rethinking maximum mean discrepancy for visual domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 9, 2021, doi: 10.1109/TNNLS.2021.3093468.

[10] Q. Kang, S. Yao, M. Zhou, K. Zhang, and A. Abusorrah, "Effective visual domain adaptation via generative adversarial distribution matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 3919–3929, Sep. 2021.

[11] L. Li, M.-W. Mak, and J.-T. Chien, "Contrastive adversarial domain adaptation networks for speaker recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 5, pp. 2236–2245, May 2022.

[12] S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu, "Variational recurrent adversarial deep domain adaptation," in *Proc. ICLR*, 2016.

[13] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020.

[14] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[15] T. Han, W. Xie, and A. Zisserman, "Video representation learning by dense predictive coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1483–1492.

[16] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[17] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[19] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[20] C. Chen *et al.*, "HoMM: Higher-order moment matching for unsupervised domain adaptation," in *Proc. AAAI*, 2020, vol. 34, no. 4, pp. 3422–3429.

[21] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 2058–2065.

[22] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 443–450.

[23] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *Proc. ICLR*, 2017.

[24] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.

[25] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.

[26] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 4058–4065.

[27] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. NIPS*, vol. 31, 2018, pp. 1640–1650.

[28] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[29] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. NIPS*, vol. 30, 2017, pp. 1195–1204.

[30] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," 2017, *arXiv:1706.05208*.

[31] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1768–1778.

[32] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Networked Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.

[33] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 1533–1540.

[34] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification," in *Proc. Eur. Conf. Prognostics Health Manage. Soc.*, 2016, pp. 5–8.

[35] M. Ragab *et al.*, "Adversarial multiple-target domain adaptation for fault classification," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[36] E. Eldele *et al.*, "Time-series representation learning via temporal and contextual contrasting," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 2352–2359.

[37] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "On minimum discrepancy estimation for deep domain adaptation," in *Domain Adaptation for Visual Understanding*. Cham, Switzerland: Springer, 2020, pp. 81–94.

[38] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[39] E. Eldele *et al.*, "An attention-based deep learning approach for sleep stage classification with single-channel EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 809–818, 2021.

[40] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "4S-DT: Self-supervised super sample decomposition for transfer learning with application to COVID-19 detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 2798–2808, Jul. 2021.

Mohamed Ragab (Graduate Student Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees from the Department of Electrical Engineering, Aswan University, Tingar, Egypt, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore.

He is also a Scientist with the Machine Intellection (MI) Department, Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include deep learning, transfer learning, and intelligent fault diagnosis and prognosis.

Mr. Ragab received the Finalist Academic Paper Award at the IEEE International Conference on Prognostics and Health Management (ICPHM) 2020.

Emadeldeen Eldele received the B.Sc. and M.Sc. degrees in computer engineering from the Faculty of Engineering, Tanta University, Tanta, Egypt, in 2012 and 2018, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore.

He is also with the Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include deep learning, self-supervised learning, transfer learning, and sensory data analytics.

Zhenghua Chen (Senior Member, IEEE) received the B.Eng. degree in mechatronics engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2011, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2017.

He is currently a Scientist and Lab Head at the Institute for Infocomm Research and an Early Career Investigator at the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A∗STAR), Singapore. His research interests include smart sensing, data analytics, machine learning, and transfer learning and related applications.

Dr. Chen received several competitive awards, such as First Place Winner for CVPR 2021 UG2+ Challenge, A*STAR Career Development Award, First Runner-Up Award for Grand Challenge at IEEE International Conference on Visual Communications and Image Processing (VCIP) 2020, and Finalist Academic Paper Award at IEEE International Conference on Prognostics and Health Management (ICPHM) 2020. He is currently the Vice Chair for the IEEE Sensors Council Singapore Chapter. He serves as an Associate Editor for Elsevier *Neurocomputing* and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Min Wu (Senior Member, IEEE) received the B.S. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2006, and the Ph.D. degree in computer science from Nanyang Technological University (NTU), Singapore, in 2011.

He is currently a Senior Scientist at the Data Analytics Department, Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include machine learning, data mining, and bioinformatics.

Dr. Wu received the Best Paper Awards at the International Conference on Bioinformatics (InCoB) 2016 and the International Conference on Database Systems for Advanced Applications (DASFAA) 2015. He also won the IJCAI Competition on repeated buyers prediction in 2015.

Chee-Keong Kwoh received the bachelor's degree (Hons.) in electrical engineering and the master's degree in industrial system engineering from the National University of Singapore, Singapore, in 1987 and 1991, respectively, and the Ph.D. degree from the Imperial College of Science, Technology, and Medicine, University of London, London, U.K., in 1995.

He has been with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore, since 1993, where he is currently the Deputy Executive Director of PaCE. His research interests include data mining, soft computing, and graph-based inference; application areas including bioinformatics and engineering. He has done significant research work in his research areas and has authored many quality international conferences and journal articles.

Dr. Kwoh is a member of the Association for Medical and Bio-Informatics and Imperial College Alumni Association of Singapore. He has provided many services to professional bodies in Singapore and was conferred the Public Service Medal by the president of Singapore in 2008.

Xiaoli Li (Senior Member, IEEE) is currently a Principal Scientist at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. He also holds an adjunct full professor position at Nanyang Technological University, Singapore. He has authored more than 270 high-quality articles. His research interests include AI, machine learning, data mining, and bioinformatics.

Dr. Li received eight best paper/benchmark competition awards. He has been serving as area chairs/senior PC members in leading AI and data-mining-related conferences, including IJCAI, AAAI, KDD, and ICDM. He is currently serving as an Editor-in-Chief of *World Scientific Annual Review of Artificial Intelligence* and an Associate Editor of the IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE and *Machine Learning with Applications* (Elsevier).