SEnsor Alignment for Multivariate Time-Series Unsupervised Domain Adaptation

Yucheng Wang[1,2], Yuecong Xu[2], Jianfei Yang[1], Zhenghua Chen[2,3]*, Min Wu[2], Xiaoli Li[1,2,3], Lihua Xie[1]

[1]Nanyang Technological University, Singapore
[2]Institute for Infocomm Research, A*STAR, Singapore
[3]Centre for Frontier AI Research, A*STAR, Singapore
{yucheng003, xuyu0014, yang0478, chen0832}@e.ntu.edu.sg, {wumin, xlli}@i2r.a-star.edu.sg, elhxie@ntu.edu.sg

Abstract

Unsupervised Domain Adaptation (UDA) methods can reduce label dependency by mitigating the feature discrepancy between labeled samples in a source domain and unlabeled samples in a similar yet shifted target domain. Though achieving good performance, these methods are inapplicable for Multivariate Time-Series (MTS) data. MTS data are collected from multiple sensors, each of which follows various distributions. However, most UDA methods solely focus on aligning global features but cannot consider the distinct distributions of each sensor. To cope with such concerns, a practical domain adaptation scenario is formulated as Multivariate Time-Series Unsupervised Domain Adaptation (MTS-UDA). In this paper, we propose SEnsor Alignment (SEA) for MTS-UDA to reduce the domain discrepancy at both the local and global sensor levels. At the local sensor level, we design the endo-feature alignment to align sensor features and their correlations across domains, whose information represents the features of each sensor and the interactions between sensors. Further, to reduce domain discrepancy at the global sensor level, we design the exo-feature alignment to enforce restrictions on the global sensor features. Meanwhile, MTS also incorporates the essential spatial-temporal dependencies information between sensors, which cannot be transferred by existing UDA methods. Therefore, we model the spatial-temporal information of MTS with a multi-branch self-attention mechanism for simple and effective transfer across domains. Empirical results demonstrate the state-of-the-art performance of our proposed SEA on two public MTS datasets for MTS-UDA. The code is available at https://github.com/Frank-Wang-oss/SEA

Introduction

Time-Series (TS) data have been widely studied thanks to their applications in various fields. Due to the powerful ability to model latent dependencies within data, Deep Learning (DL) methods have made notable advances in TS related problems (Zhao et al. 2019; Chen et al. 2021a; He et al. 2022). However, they require a vast amount of labeled TS data for training, which may not be available in real-world applications due to the high cost of data labeling.

To reduce the cost of labeling, Unsupervised Domain Adaptation (UDA) methods have been proposed to transfer the knowledge from a labeled source domain to an unlabeled target domain in the absence of labels of the target domain. To reduce discrepancy across domains, existing UDA methods learn domain-invariant features with two paradigms, which include metric-based methods (Tzeng et al. 2014) and adversarial-based methods (Ganin et al. 2016). These methods have achieved decent performance, showing the effectiveness of UDA in reducing label dependency. Subsequently, researchers applied UDA for TS data, aiming to reduce the discrepancy across domains by learning domain-invariant temporal features extracted by Recurrent Neural Network (RNN) (Purushotham et al. 2017), Long Short-Term Memory (LSTM) (Cai et al. 2021), and Convolutional Neural Network (CNN) (Liu and Xue 2021).

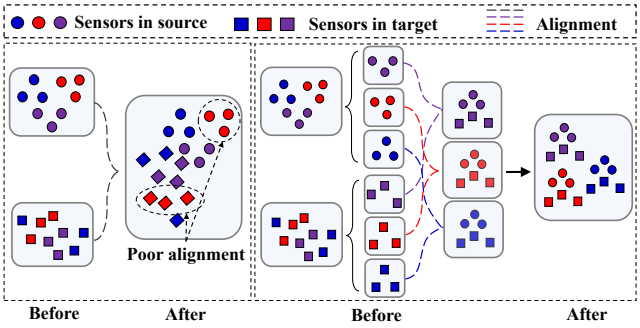

Figure 1: Comparisons between previous UDA methods and ours, where before and after represent before and after alignment respectively. Left: Only global sensor features are aligned, resulting in the poor alignment of the red sensor. Right: SEA aligns the sensor information between domains, so the feature discrepancy of each sensor can be reduced.

While substantial progress has been made, existing TS UDA methods generally focus on Univariate Time-Series (UTS) data by assuming that data originate from one single source (Cai et al. 2021). Yet, they are inapplicable for real-world applications where multiple sensors are deployed simultaneously, e.g. remaining useful life (RUL) prediction and human activity recognition (HAR). To cope with such applications, we thus formulate a more challenging yet practical scenario of Multivariate Time-Series Unsupervised Domain Adaptation (MTS-UDA). Different from UTS data,

*Corresponding Author

MTS data originate from multiple sensors where data from different sensors follow various data distributions. By treating all sensors as a whole, existing UDA methods can consider the global distributions of sensors (Ragab et al. 2022b). But they cannot take the sensor-level distributions into account, leading to the misalignment of each sensor. As presented in Fig. 1 (**Left**), previous UDA methods only align the global features between domains, resulting in the poor alignment of the features from the red sensor and limiting the transferability of the models. Further, due to its multiple-sources nature, MTS contains essential spatial-temporal dependencies information. Specifically, the spatial dependency refers to sensor correlations, representing the critical interactive relations between sensors. For example, the temperature sensor has strong correlations with the fan speed sensor in machine RUL prediction. Meanwhile, the temporal information represents the temporal dependency between time steps. Existing UDA methods are incapable of modelling and transferring both dependencies across domains. Due to these issues, MTS-UDA is more challenging than UDA for UTS that only aligns the data from one single source.

To cope with the above limitations, we propose SEnsor Alignment (SEA) for MTS-UDA to reduce domain discrepancy at both the local and global sensor levels. At the local sensor level, we design endo-feature alignment to align sensor-level information between domains. We observe that the sensor-level information consists of both sensor features and sensor correlations, which represent the properties of each single sensor and their interactive information respectively. Therefore, we propose a contrastive sensor alignment to align the sensor features between domains and design the sensor correlation alignment to make the correlations between domains identical. Further, at the global sensor level, we design exo-feature alignment to reduce global feature discrepancy. To achieve this, we align the stacked sensor features by enforcing restrictions. Meanwhile, to transfer the spatial-temporal dependencies within MTS across domains in a simple and effective manner, we propose a multi-branch self-attention mechanism to model such information.

Our contributions can be summarized in three folds: first, we formulate a challenging scenario of Multivariate Time-Series Unsupervised Domain Adaptation (MTS-UDA) in accordance with the characteristics of MTS data. To our best knowledge, this is the first work to design the UDA method specifically for MTS data. Second, we analyze the problems underlying MTS-UDA and design SEA to address these problems uniformly. SEA reduces domain discrepancy at both local and global sensor levels and captures the spatial-temporal dependencies within MTS simultaneously for simple and effective transfer across domains. Lastly, we demonstrate the effectiveness of our proposed framework using different real-world MTS datasets, on which we achieve state-of-the-art performances.

Related Work

Unsupervised domain adaptation. UDA aims to reduce the cost of labeling by training networks to transfer knowledge in a labeled source domain to an unlabeled target domain. To achieve great performance on the target domain, existing UDA methods are trying to minimize the discrepancy across domains, which can be divided into two categories. The first category is adversarial based methods, which utilize domain discriminator networks to force the feature extractor to learn domain-invariant representations. The typical models are domain adversarial neural network (Ganin et al. 2016), conditional domain adversarial network (Long et al. 2018), adversarial discriminative domain adaptation (Tzeng et al. 2017) and so on. The another branch is metric-based methods, which enable networks to learn invariant features by enforcing metric restrictions. The typical models are deep domain confusion (Tzeng et al. 2014), correlation alignment via deep neural networks (Sun and Saenko 2016), domain conditioned adaptation network (Li et al. 2020) and so on. These UDA methods have shown their effectiveness in reducing label dependency for DL training.

Unsupervised domain adaptation for time-series data. Some UDA methods were recently proposed for TS data (Cai et al. 2021; Liu and Xue 2021), which reduce discrepancy by aligning temporal features across domains. VRADA (Purushotham et al. 2017) learned temporal features via a variational RNN and reduced the discrepancy by adversarial based methods. ADATIME (Ragab et al. 2022b) evaluated various CNN models to capture temporal dynamics. However, these works were designed specifically for the data from one source, i.e., the data from same distributions, which are inapplicable for the scenarios requiring multiple sensors. Generally, the data from different sensors follow various distributions because the sensors are deployed to measure various physical quantities, e.g., temperature and fan speed. Although existing works can be applied for MTS (Yang et al. 2021) by assuming that all sensors follow same distributions, i.e., treating all sensors as a whole, they still cannot consider the essential properties within MTS data.

Proposed SEA Framework

Problem Definition

For MTS-UDA, we are given a source domain with N_s labeled samples $\mathcal{D}_S = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and a target domain with N_t unlabeled samples $\mathcal{D}_T = \{x_i^t\}_{i=1}^{N_t}$. Each MTS sample x_i (either x_i^s or x_i^t) originates from N sensors with different distributions, i.e., $x_i = \{x_{im}\}_{m=1}^{N} \in \mathbb{R}^{N \times L}$, where L represents the time length. The goal of SEA is to train a network to transfer the knowledge in the source domain to the target domain to learn features $h_i \in \mathbb{R}^F$ from MTS data x_i. Further, we omit the index i in the following contents for clarity, so each sample is simplified as x^s and x^t, and the data from the m-th sensor are simplified as x_m^s and x_m^t.

Furthermore, to retain the temporal dependency in MTS data, we construct sequential graphs $\{\mathcal{G}_T\}_{T=1}^{\hat{L}}, \mathcal{G}_T = (Z_T, E_T)$ from the sample x, where $Z_T = \{z_{m,T}\}_{m=1}^{N}$ and $E_T = \{e_{mn,T}\}_{m,n=1}^{N}$ represent the sensor features and correlations, respectively, in the T-th graph. $\{\mathcal{G}_T^s\}_{T=1}^{\hat{L}}$ and $\{\mathcal{G}_T^t\}_{T=1}^{\hat{L}}$ represent the sequential graphs in the source and target domains respectively.

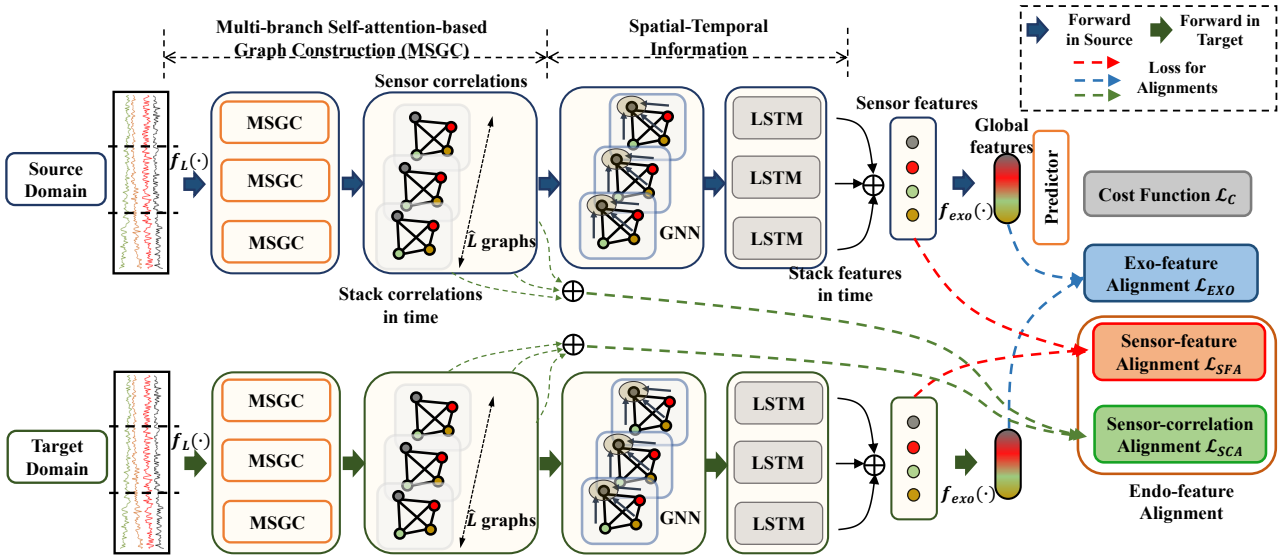

Figure 2: The overall structure of SEA. (1) To transfer spatial-temporal dependencies across domains, source and target domains share the same feature extractor, including MSGC, GNN, and LSTM. The samples are cropped as mini-pieces, which are constructed as sequential graphs. With the combination of GNN and LSTM, the decent sensor information is learned, including sensor features and their correlations. (2) Except for supervised learning on the source domain, with the sensor information, endo-feature alignment and exo-feature alignment are designed to reduce domain discrepancy at the local and global sensor levels. Endo-feature alignment consists of sensor feature alignment and sensor correlation alignment, which align sensor features and their correlations between domains. Exo-feature alignment aligns the global features mapped by sensor features.

Overall Structure

The overall structure of SEA is shown in Fig. 2. As the forward processes in the source and target domains share the same feature extractor, including Multi-branch Self-attention based Graph Construction (MSGC), Graph Neural Network (GNN), and LSTM, the superscripts s and t are omitted in Section , , and . Firstly, we model the spatial-temporal dependencies within MTS data for simple transfer across domains. Given the sample x, we crop it as mini-pieces, which are constructed as sequential graphs by MSGC. Then, with the sequential graphs, GNN and LSTM are combined together to capture the spatial-temporal information for learning sensor information, i.e., sensor features and correlations. Secondly, we utilize the sensor information to reduce the domain discrepancy at the local and global sensor levels. At the local sensor level, we design endo-feature alignment by leveraging sensor features and their correlations to design sensor feature alignment (SFA) and sensor correlation alignment (SCA) respectively. At the global sensor level, we stack sensor features and design exo-feature alignment to align the global features by enforcing restrictions. We introduce each module in the following sections.

Multi-Branch Self-Attention-Based Graph Construction

Sampling. To retain temporal information between time steps, we divide the original sample $x \in \mathbb{R}^{N \times L}$ into sequential features $\{Z_T\}_{T=1}^{\hat{L}} \in \mathbb{R}^{N \times \hat{L} \times d}$, where $\hat{L}$ is the time length representing the time part, and d is the feature dimension representing the feature part. In this way, we obtain the sequential features, the T-th of which is $Z_T \in \mathbb{R}^{N \times d}$. For each time step, the feature of the m-th sensor is $z_{m,T} \in \mathbb{R}^d$. Furthermore, a nonlinear function $f_L(\cdot)$ is added to improve the nonlinear expressiveness as shown in Fig. 2.

Graph construction. We adopt the sequential features to construct sequential graphs. To fully represent the complex correlations between sensors, we propose MSGC, aiming to construct multiple graphs using different initialized weights.

$$q^i_{m,T} = z_{m,T} W^i_Q, k^i_{n,T} = z_{n,T} W^i_K,$$
$$e^i_{mn,T} = \frac{\frac{q^i_{m,T}(k^i_{n,T})^T}{\sqrt{d}}}{\sum_{j=1}^{N} \frac{q^i_{m,T}(k^i_{j,T})^T}{\sqrt{d}}}. \quad (1)$$

In each branch, we adopt query and key to learn the weights between sensors. Eqn. (1) describes the graph construction between m and n of the i-th branch, where d is the feature dimension, and we treat the m-th sensor $z_{m,T}$ as the query sensor and the n-th sensor $z_{n,T}$ as the key sensor. In this way, we can learn the edge $e^i_{mn,T}$ for the i-th branch. Then, we adopt softmax to make the edge within $[0, 1]$.

To make constructed graph robust, we averagely combine the obtained graphs in all branches, i.e., $e_{mn,T} = \sum_{i=1}^{n_b} \frac{e^i_{mn,T}}{n_b}$, where n_b is the number of branches. We thus obtain the sensor correlations $E_T = \{e_{mn,T}\}_{m,n=1}^{N} \in \mathbb{R}^{N \times N}$ in the T-th graph and the sequential correlations $\{E_T\}_{T=1}^{\hat{L}} \in \mathbb{R}^{\hat{L} \times N \times N}$.

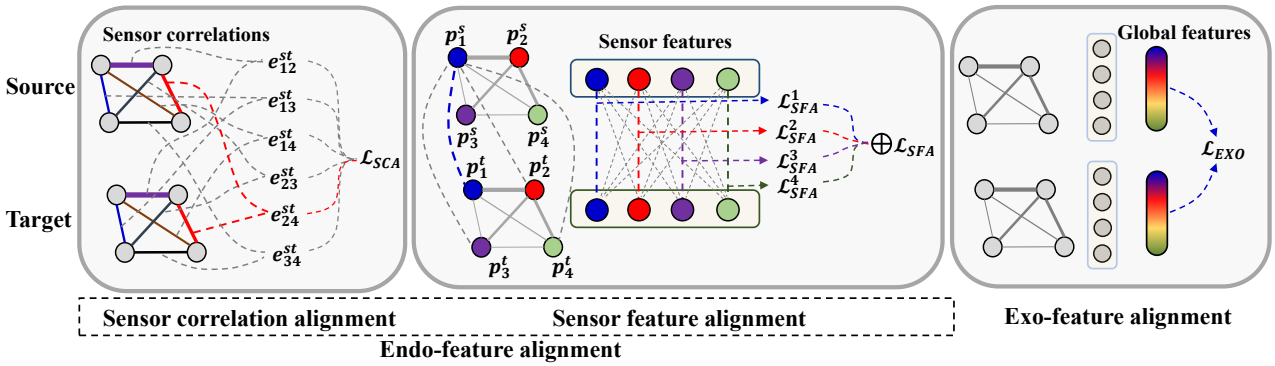

Figure 3: Left: Sensor correlation alignment is to make the correlations between domains identical. Middle: Sensor feature alignment is to make the data from the same sensor between domains similar and the data from the different sensors between domains different. Right: Exo-feature alignment is to reduce the discrepancy between the global features in different domains.

Capturing Spatial-Temporal Information

Prior works have proved the capability of GNN to capture spatial dependency (Deng and Hooi 2021; Jia et al. 2020). Therefore, we introduce GNN to capture sensor correlations in the sequential graphs. We assume that the sensor correlations in each of these graphs are independent, so we adopt GNN to process each graph, i.e., $\mathcal{G}_T = (Z_T, E_T)$, with shared GNN. As shown in Eqn. (2), Message Passing Neural Network (Gilmer et al. 2017), a variant of GNN, is adopted to process each graph.

$$\begin{aligned} h_{m,T} &= \sum_{j=1}^{N} e_{mj,T} z_{j,T}, \\ z_{m,T} &= ReLU(h_{m,T} W_G). \end{aligned} \tag{2}$$

Further, it is noted that the data from a specific sensor show temporal dependency between sequential graphs. Therefore, given the features $z_m \in \mathbb{R}^{\hat{L} \times d}$ for a sensor m in sequential graphs, we learn the temporal dependency for the sensor m with LSTM. With GNN and LSTM, we update sequential features as $\{Z_T\}_{T=1}^{\hat{L}} \in \mathbb{R}^{\hat{L} \times N \times d}$ by capturing the spatial-temporal dependency within MTS data, which can be then transferred across domains by the alignment at local and global sensor levels in the following parts.

Alignment for UDA

To reduce the discrepancy between domains for MTS-UDA, we design endo-feature alignment and exo-feature alignment at the local and global sensor levels respectively.

Endo-feature alignment. Existing UDA methods reduce feature discrepancy between domains by considering only global features, which is applicable for image and UTS data as they come from a single source. However, it is noted that MTS data originate from multiple sensors and the data from different sensors follow various distributions. When applying existing UDA methods for MTS data, they only consider the global distribution by treating all sensors as a whole but ignore the distribution of each sensor. This will result in the misalignment of each sensor, restricting the performance of the learned model transferring from the source domain to the target domain. Therefore, endo-feature alignment is proposed to align the sensor-level information so that the misalignment for each sensor between domains can be avoided.

To make better endo-feature alignment, it is necessary to note that the sensor-level information consists of sensor features and sensor correlations. Sensor features represent the properties of each sensor, and sensor correlations represent the essential interaction information between sensors. For example, a temperature sensor and a fan speed sensor are deployed together to detect the status of a machine. The data from the temperature sensor between source and target domains should follow a similar distribution. Meanwhile, the correlation of them between domains should be also similar, as the fan speed usually increases when the temperature rises. Based on these considerations, we believe it is necessary to align the sensor features and sensor correlations simultaneously to make endo-feature alignment better.

With learned sequential graphs $\{\mathcal{G}_T\}_{T=1}^{\hat{L}}$ consisting of sequential features $\{Z_T\}_{T=1}^{\hat{L}}$ and sequential correlations $\{E_T\}_{T=1}^{\hat{L}}$, we can get the sensor information by stacking the sequential features and correlations through time as $P = \sum_{T=1}^{\hat{L}} \frac{Z_T}{\hat{L}}$ and $E = \sum_{T=1}^{\hat{L}} \frac{E_T}{\hat{L}}$, where $P = \{p_m\}_{m=1}^{N} \in \mathbb{R}^{N \times d}$ and $E = \{e_{mn}\}_{m,n=1}^{N} \in \mathbb{R}^{N \times N}$ are the sensor features and correlations. We thus obtain P^s and E^s in the source domain and P^t and E^t in the target domain.

We first align the sensor correlations between domains. To reduce domain discrepancy, we expect the interactions of sensors to have similar trends between domains. To achieve this, sensor correlation alignment should make the sensor correlations between domains identical. Therefore, we propose to align each edge of the graphs between domains, as shown in Fig. 3 **Left**. The process can be denoted as Eqn. (3), where $|\cdot|$ is absolute value. We compute the expectation of the discrepancy of sensor correlations between domains.

$$\begin{aligned} e_{mn}^{st} &= e_{mn}^{s} - e_{mn}^{t}, \\ \min \mathcal{L}_{SCA} &= \mathbb{E}(|e_{mn}^{st}|). \end{aligned} \tag{3}$$

We then align the sensor features between domains. We

assume that the features from the same sensor should have similar distributions. Meanwhile, compared with the features from the same sensor, the features from the different sensors between domains should have different properties. For example, as shown in Fig. 3 **Middle**, the features p_1^s should be more similar with p_1^t than other sensors in the target domain, i.e., p_2^t, p_3^t, and p_4^t. To achieve this goal, we propose a sensor contrastive mechanism as shown in Eqn. (4) to align the sensor features between domains.

$$\begin{aligned} \min \mathcal{L}_{SFA} &= -\mathbb{E}(log \frac{e^{\varphi(p_m^s, p_m^t)}}{\sum_{p_j^t \in P^t} e^{\varphi(p_m^s, p_j^t)}}), \\ \varphi(p_m^s, p_m^t) &= p_m^s (p_m^t)^T. \end{aligned} \tag{4}$$

With the proposed sensor feature alignment and sensor correlation alignment, we can get the endo-feature alignment as Eqn. (5), where λ_{SCA} and λ_{SCA} are the hyperparameters to tune the effect of SFA and SCA respectively.

$$\min \mathcal{L}_{Endo} = \lambda_{SCA} \mathcal{L}_{SCA} + \lambda_{SFA} \mathcal{L}_{SFA}. \tag{5}$$

Exo-feature alignment. To reduce the domain discrepancy at the global level, we enforce restrictions on the global features mapped by sensor features between domains. As shown in Fig. 3 **Right**, we stack the sensor features and map the stacked features to the global features, which are then aligned by Deep Coral (Sun and Saenko 2016). The exo-feature alignment can be shown as Eqn. (6):

$$\begin{aligned} h^s = f_{exo}(p_1^s, ..., p_N^s) &= \sigma(cat(p_1^s, ..., p_N^s) W_{exo}) \in \mathbb{R}^F, \\ h^t = f_{exo}(p_1^t, ..., p_N^t) &= \sigma(cat(p_1^t, ..., p_N^t) W_{exo}) \in \mathbb{R}^F, \\ C^s &= \frac{(h^s)^T h^s - \frac{1}{B^s}(1^T h^s)(1^T h^s)}{B^s - 1}, \\ C^t &= \frac{(h^t)^T h^t - \frac{1}{B^t}(1^T h^t)(1^T h^t)}{B^t - 1}, \\ \min \mathcal{L}_{EXO} &= \frac{1}{4F^2} ||C^s - C^t||_F^2, \end{aligned} \tag{6}$$

where $cat(\cdot)$ represents concatenation. $||\cdot||_F^2$ is the squared matrix Frobenius norm. B^s and B^t are the number of samples in the source and target domains. Finally, the overall loss function (7) is minimized to train SEA:

$$\begin{aligned} \min \mathcal{L} &= \mathcal{L}_C + \lambda_{EXO} \mathcal{L}_{EXO} + \mathcal{L}_{Endo}, \\ \mathcal{L}_{Endo} &= \lambda_{SCA} \mathcal{L}_{SCA} + \lambda_{SFA} \mathcal{L}_{SFA}, \end{aligned} \tag{7}$$

where λ_{EXO} represents the hyperparameter to tune the effect of exo-feature alignment. $\mathcal{L}_C$ is the cost function (i.e., Mean Square Error loss or Cross Entropy loss, depending on the specific tasks) computed by the source data and labels.

Experimental Results

To evaluate the effectiveness of SEA, we test our model on two public datasets, C-MAPSS for remaining useful life prediction (Saxena et al. 2008) and Opportunity HAR for human activity recognition (Roggen et al. 2010).

Datasets and Setup

C-MAPSS describes the degradation process of aircraft engines. This dataset includes four sub-datasets collected in different working conditions and fault modes, where each sub-dataset represents one domain. The MTS data in C-MAPSS originate from 14 sensors deployed in different locations to measure various physical quantities. Following the data preparation in the previous work (Ragab et al. 2021), we process the datasets, where data annotations represent the remaining useful life circle of engines.

Opportunity HAR describes the human activities collected from four subjects, each of which represents one domain. The data annotations include two levels: 1). locomotion representing the low-level tasks including four categories, sitting, standing, walking, and lying down; 2). gestures representing the high-level tasks including 17 various actions. Following the experimental setting in the previous work (Ragab et al. 2022a), we adopt the low-level tasks. Meanwhile, only the sensors attached to the body are used (i.e., 113 sensors in total). As some values in the data are missing, we adopt the linear interpolation approach to fill in the missing positions. To construct the training dataset, we adopt a sliding window with a size of 128 and an overlapping of 50% as (Ragab et al. 2022a) did.

The experiments include three parts, the comparisons with state-of-the-art, the ablation study, and the sensitivity analysis. All experiments run ten times and the average results are shown to prevent the effect of random initialization. Besides, we set batch size as 50, optimizer as Adam, learning rate as 0.001, and training epoch as 10 for training our model. Furthermore, we built and trained our model based on Pytorch 1.9 and NVIDIA GeForce RTX 3080Ti GPU.

We adopt different indicators to evaluate our method on two datasets. For the experiments on C-MAPSS, we adopt Root MSE (RMSE) and Score (Ragab et al. 2021; Chen et al. 2021b; Xu et al. 2022), as it is a prediction task, i.e., predicting the RUL of an engine. The lower the two indicators are, the better the model is. For the experiments on Opportunity HAR, we adopt accuracy (Acc.), as it is a classification task. The higher the indicator is, the better the model is.

Comparisons with State-of-the-Art

We compare SEA with SOTA UDA methods, including DDC (Tzeng et al. 2014), Coral (Sun, Feng, and Saenko 2017), DANN (Ganin et al. 2017), BNM (Cui et al. 2020), LMMD (Zhu et al. 2021), CtsADA (Ragab et al. 2021), SLARDA (Ragab et al. 2022a), CLUDA (Vayyat et al. 2022), and SDAT (Rangwani et al. 2022). For fair comparisons, we adopt the same feature extractor (i.e., GNN-LSTM) for these methods, each of which runs ten times to show average results. Notably, as LMMD and SLARDA were designed specifically for multi-class classification UDA, they only have results on Opportunity HAR. Further, the results of source only (Source) are also compared.

Table 1 shows the results in 12 cross-domain scenarios for RUL prediction. From the results, we observe that SEA achieves the best performance in 10 cross-domain scenarios and the second best in 2 cross-domain scenarios for both

Models	1→2	1→3	1→4	2→1	2→3	2→4	3→1	3→2	3→4	4→1	4→2	4→3	Avg.
Source	36.06	42.43	38.91	32.22	41.01	31.29	46.69	45.84	25.82	35.49	32.56	21.68	35.83
DDC	43.25	39.48	42.99	40.07	39.46	43.01	40.93	43.43	43.72	41.58	43.38	39.61	41.74
Coral	16.73	33.86	26.21	14.17	32.76	30.07	29.22	25.80	23.29	30.70	23.27	16.54	25.22
DANN	17.69	36.73	33.76	17.83	39.28	31.39	28.35	36.62	22.16	30.83	22.68	30.63	29.00
BNM	48.18	41.61	46.45	56.66	48.40	62.02	49.39	48.89	42.98	40.76	43.71	41.89	47.58
CtsADA	17.95	27.74	27.92	19.47	51.32	38.99	36.01	24.43	19.24	31.05	25.25	23.16	28.54
CLUDA	25.47	37.48	32.59	22.58	33.44	33.14	25.36	23.42	26.60	25.05	25.73	22.23	27.76
SDAT	17.97	32.77	**25.34**	16.78	33.91	**26.48**	26.24	27.67	22.55	22.27	20.72	27.65	25.03
SEA	**15.83**	**20.03**	25.67	**13.00**	**25.40**	27.43	**21.75**	**19.36**	**18.49**	**18.51**	**16.16**	**15.59**	**19.77**
Source	12349	7868	27449	5440	10410	7803	96166	163951	10349	17411	28953	1356	32459
DDC	23323	6164	34891	11581	6578	35122	22236	59400	80774	29592	58029	15239	31911
Coral	1239	4473	5004	378	3497	6509	10637	25612	17668	6166	5159	497	7237
DANN	1355	65652	12133	897	18660	29828	4891	14923	9393	11359	3790	13846	15561
BNM	24265	44650	19017	526203	56769	29689	46262	29188	39559	11618	20566	65807	76133
CtsADA	1424	1682	5756	2619	22657	18315	41127	19285	**1953**	9844	11377	1425	11455
CLUDA	2704	4617	6181	919	3224	6384	1846	3243	3300	1963	6715	1036	3511
SDAT	1106	8352	**3962**	689	19477	6519	5736	19280	11620	1401	2294	6803	7270
SEA	**1023**	**783**	4807	**298**	**1412**	**5416**	**1720**	**2995**	2181	**863**	**1606**	**451**	**1963**

Table 1: The Comparisons with State-of-the-art Models on C-MAPSS (Up: RMSE; Down: Score)

Models	1→2	1→3	1→4	2→1	2→3	2→4	3→1	3→2	3→4	4→1	4→2	4→3	Avg.
Source	47.50	64.50	60.83	73.00	71.00	65.67	64.83	56.67	72.17	66.67	65.50	74.67	65.25
DDC	40.50	58.50	55.50	43.35	58.30	55.50	45.00	40.50	55.50	45.00	40.50	58.50	49.72
Coral	78.55	82.71	78.17	81.58	79.93	79.07	78.50	75.40	84.88	78.56	80.38	87.88	80.47
DANN	69.62	85.12	79.00	85.62	81.62	74.88	76.50	75.75	86.12	84.25	74.12	86.62	79.94
BNM	44.00	61.00	52.00	45.00	61.00	52.00	45.00	44.00	52.00	45.50	44.00	61.00	50.54
LMMD	56.25	61.17	59.08	45.33	58.50	55.50	49.42	46.83	57.33	59.00	44.75	61.75	54.58
CtsADA	80.00	73.67	76.08	69.75	78.17	79.08	67.75	77.33	64.75	74.58	52.17	75.00	72.36
SLARDA	80.38	80.50	79.50	82.75	78.88	79.25	75.88	78.62	70.12	80.62	84.00	77.25	78.98
CLUDA	77.60	79.45	76.00	73.20	79.25	76.15	61.75	61.95	67.70	61.90	61.05	75.80	70.98
SDAT	84.92	72.42	77.00	**87.25**	83.00	76.17	**87.75**	**84.25**	87.25	74.58	66.75	91.50	81.07
SEA	**86.35**	**86.15**	**86.00**	86.35	**84.35**	**81.75**	87.25	83.05	**88.70**	**84.90**	**84.25**	**92.25**	**85.42**

Table 2: The Comparisons with State-of-the-art Models on Opportunity HAR (%)

RMSE and Score. In the scenarios where we are the best, we achieve significant improvements, e.g., 27.7% and 22.4% better than the second best method in $1 \rightarrow 3$ and $2 \rightarrow 3$ for RMSE respectively. In the scenarios where we are the second best, the gaps between the best methods and ours are quite narrow, e.g., only 1.3% weaker than SDAT in $1 \rightarrow 4$ for RMSE. Further, in terms of the average improvements, we observe that SEA is 21.0% and 44.1% better than the second best methods for RMSE and Score respectively.

Similar improvements can be observed in the experiments for HAR in Table 2, which records the comparisons in 12 cross-domain scenarios. We observe that SEA achieves the best performance in nine cross-domain scenarios and the second best in three cross-domain scenarios. In the scenarios where we achieve the best performance, we obtain significant improvements. For example, we are 6.5% better than the second best in $1 \rightarrow 4$. Further, in the scenarios where we are the second best, the gaps between the best methods and ours are still very narrow, e.g., only 0.9%, 0.5%, and 1.2%. Besides, in terms of the average improvements, we observe that SEA is 4.35% better than the second best method.

The results of the two datasets show that our SEA is effective for MTS-UDA. By reducing the domain discrepancy at the local and global sensor levels, our method can achieve state-of-the-art performance.

Ablation Study

To evaluate whether each alignment in SEA is effective, we conduct ablation study in 24 cross-domain scenarios of two datasets. We first evaluate the variant $\mathcal{L}_C$ w/o endo, describing the model with exo-feature alignment only. The second variant is $\mathcal{L}_C$ w/o $\mathcal{L}_{SCA}$, which introduces the endo-feature alignment, while sensor correlation alignment is excluded. The third variant is $\mathcal{L}_C$ w/o $\mathcal{L}_{SFA}$, which introduces the endo-feature alignment while sensor feature alignment is excluded. The final one is our complete SEA.

Table 3 and Table 4 show the ablation study on C-MAPSS and Opportunity HAR respectively. From the results, we observe that the models with additional SFA or SCA achieve better performance than the model with exo-feature align-

Variants	1→2	1→3	1→4	2→1	2→3	2→4	3→1	3→2	3→4	4→1	4→2	4→3	Avg.
$\mathcal{L}_C$ w/o endo	16.73	33.86	26.21	14.17	32.76	30.07	29.22	25.80	23.29	30.70	23.27	16.54	25.22
$\mathcal{L}_C$ w/o $\mathcal{L}_{SCA}$	15.92	22.89	26.08	13.11	26.49	28.14	24.11	21.54	18.72	28.20	22.05	16.36	21.97
$\mathcal{L}_C$ w/o $\mathcal{L}_{SFA}$	16.34	23.64	26.74	13.88	28.12	28.4	29.17	23.12	19.41	26.89	21.14	16.38	22.77
SEA	**15.83**	**20.03**	**25.67**	**13.00**	**25.40**	**27.43**	**21.75**	**19.36**	**18.49**	**18.51**	**16.16**	**15.59**	**19.77**
$\mathcal{L}_C$ w/o endo	1239	4472	5004	377	3496	6509	10636	25612	17668	6166	5159	496	7237
$\mathcal{L}_C$ w/o $\mathcal{L}_{SCA}$	1124	1408	7426	**296**	2018	8057	3739	5184	**1590**	6235	4343	708	3511
$\mathcal{L}_C$ w/o $\mathcal{L}_{SFA}$	1067	1157	**4473**	375	2230	5741	8696	10132	3431	5578	4190	539	3968
SEA	**1022**	**783**	4807	297	**1411**	**5416**	**1719**	**2995**	2181	**862**	**1606**	**451**	**1963**

Table 3: The Results of Ablation Study on C-MAPSS (Up: RMSE; Down: Score)

Variants	1→2	1→3	1→4	2→1	2→3	2→4	3→1	3→2	3→4	4→1	4→2	4→3	Avg.
$\mathcal{L}_C$ w/o endo	78.55	82.71	78.17	81.58	79.93	79.07	78.50	75.40	84.88	78.56	80.38	87.88	80.47
$\mathcal{L}_C$ w/o $\mathcal{L}_{SCA}$	82.94	83.62	80.38	84.25	83.45	80.55	78.80	75.38	86.20	79.05	81.90	87.50	82.00
$\mathcal{L}_C$ w/o $\mathcal{L}_{SFA}$	84.45	84.90	82.20	84.00	81.95	81.00	81.50	75.81	86.95	81.80	81.60	87.35	82.79
SEA	**86.35**	**86.15**	**86.00**	**86.35**	**84.35**	**81.75**	**83.25**	**80.70**	**88.70**	**84.90**	**84.25**	**92.25**	**84.89**

Table 4: The Results of Ablation Study on Opportunity HAR (%)

ment only. For example, regarding the average improvements of RMSE on C-MAPSS, the models with additional $\mathcal{L}_{SFA}$ and $\mathcal{L}_{SCA}$ are 12.8% and 9.7% better than the model only with $\mathcal{L}_{EXO}$. However, the single perspective of sensor information is insufficient. From the results, we find that it is difficult for the model with a single perspective of endo-feature alignment (i.e., SFA or SCA) to achieve the best performance in each case. For example, the model with $\mathcal{L}_{SFA}$ achieves better performance than that with $\mathcal{L}_{SCA}$ in $1 \rightarrow 2$ for RUL prediction, while the model with $\mathcal{L}_{SCA}$ achieves better performance than that with $\mathcal{L}_{SFA}$ in $1 \rightarrow 2$ for HAR. Compared with these variants, SEA achieves better performance in each cross-domain scenario and obtains significant improvements. For example, regarding the average RMSE on C-MAPSS, SEA is 21.6%, 10.0%, and 13.1% better than the model only with $\mathcal{L}_{EXO}$, the model with additional $\mathcal{L}_{SFA}$, and the model with additional $\mathcal{L}_{SCA}$, respectively. These results further show that our endo-feature alignment is effective, and it is necessary to consider both sensor feature and correlation alignment for endo-feature alignment.

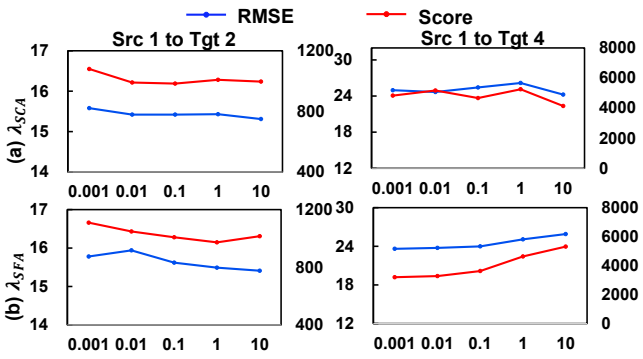

Figure 4: The sensitivity analysis for different λ_{SCA} and λ_{SFA} on C-MAPSS.

Sensitivity Analysis

We further analyze λ_{SCA} and λ_{SFA} to see how the hyper-parameters will affect MTS-UDA. We have conducted experiments with various λ_{SCA} and λ_{SFA} varying 0.001 from 10 with the interval of $10\times$. Fig. 4 shows the analysis on C-MAPSS. From the results, we observe that SEA is stable in most cross-domain scenarios. Further, we also observe that SEA tends to achieve poor performance when λ_{SCA} is too small or λ_{SFA} is too large. For example, in $1 \rightarrow 2$ of Fig. 4 (a), SEA shows the worst performance when λ_{SCA} is 0.001, while it is stable in other cases. Meanwhile, in $1 \rightarrow 4$ of Fig. 4 (b), SEA shows the worse performance than other cases when λ_{SFA} is 10. Therefore, to achieve stable and decent performance, we recommend that the values of λ_{SCA} and λ_{SFA} should be set as 0.1 or 1.

Conclusion and Future Work

In this paper, we formulate the unsupervised domain adaptation for multivariate time-series data (MTS-UDA). We analyze the problems underlying this task and propose SEnsor Alignment (SEA) to address these issues. To reduce the domain discrepancy at both the local and global sensor levels, we design endo-feature alignment and exo-feature alignment. At the local sensor level, we align the sensor features and correlations between domains to prevent the misalignment of each sensor. Furthermore, we enforce restrictions on the global sensor features to reduce domain discrepancy at the global sensor level. Meanwhile, we model the spatial-temporal dependencies within MTS data for simple transfer across domains. Our extensive experiments demonstrate the effectiveness of SEA for MTS-UDA.

However, dirty data caused by sampling rate jitter and timestamp misalignment across sensors may affect the performance. Therefore, we will focus on introducing self-supervised module into our method so that our method will be able to learn robust representations even facing dirty data.

Acknowledgements

We thank anonymous reviewers for their constructive comments to this work. This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151) and Career Development Award (Grant No. C210112046).

References

Cai, R.; Chen, J.; Li, Z.; Chen, W.; Zhang, K.; Ye, J.; Li, Z.; Yang, X.; and Zhang, Z. 2021. Time Series Domain Adaptation via Sparse Associative Structure Alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8): 6859–6867.

Chen, K.; Zhang, D.; Yao, L.; Guo, B.; Yu, Z.; and Liu, Y. 2021a. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4): 1–40.

Chen, Z.; Wu, M.; Zhao, R.; Guretno, F.; Yan, R.; and Li, X. 2021b. Machine Remaining Useful Life Prediction via an Attention-Based Deep Learning Approach. *IEEE Transactions on Industrial Electronics*, 68(3): 2521–2531.

Cui, S.; Wang, S.; Zhuo, J.; Li, L.; Huang, Q.; and Tian, Q. 2020. Towards Discriminability and Diversity: Batch Nuclear-Norm Maximization Under Label Insufficient Situations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, A.; and Hooi, B. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4027–4035.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.*, 17(1): 2096–2030.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2017. *Domain-Adversarial Training of Neural Networks*, 189–209. Cham: Springer International Publishing. ISBN 978-3-319-58347-1.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. *arXiv preprint arXiv:1704.01212*.

He, H.; Zhang, Q.; Bai, S.; Yi, K.; and Niu, Z. 2022. CATN: Cross Attentive Tree-Aware Network for Multivariate Time Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4): 4030–4038.

Jia, Z.; Lin, Y.; Wang, J.; Zhou, R.; Ning, X.; He, Y.; and Zhao, Y. 2020. GraphSleepNet: Adaptive Spatial-Temporal Graph Convolutional Networks for Sleep Stage Classification. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 1324–1330. International Joint Conferences on Artificial Intelligence Organization. Main track.

Li, S.; Liu, C.; Lin, Q.; Xie, B.; Ding, Z.; Huang, G.; and Tang, J. 2020. Domain Conditioned Adaptation Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07): 11386–11393.

Liu, Q.; and Xue, H. 2021. Adversarial Spectral Kernel Matching for Unsupervised Time Series Domain Adaptation. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2744–2750. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional Adversarial Domain Adaptation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 1647–1657. Red Hook, NY, USA: Curran Associates Inc.

Purushotham, S.; Carvalho, W.; Nilanon, T.; and Liu, Y. 2017. Variational Recurrent Adversarial Deep Domain Adaptation. In *ICLR*.

Ragab, M.; Chen, Z.; Wu, M.; Foo, C. S.; Kwoh, C. K.; Yan, R.; and Li, X. 2021. Contrastive Adversarial Domain Adaptation for Machine Remaining Useful Life Prediction. *IEEE Transactions on Industrial Informatics*, 17(8): 5239–5249.

Ragab, M.; Eldele, E.; Chen, Z.; Wu, M.; Kwoh, C.-K.; and Li, X. 2022a. Self-Supervised Autoregressive Domain Adaptation for Time Series Data. *IEEE Transactions on Neural Networks and Learning Systems*, 1–11.

Ragab, M.; Eldele, E.; Tan, W. L.; Foo, C.-S.; Chen, Z.; Wu, M.; Kwoh, C.-K.; and Li, X. 2022b. ADATIME: A Benchmarking Suite for Domain Adaptation on Time Series Data. *arXiv preprint arXiv:2203.08321*.

Rangwani, H.; Aithal, S. K.; Mishra, M.; Jain, A.; and Babu, R. V. 2022. A Closer Look at Smoothness in Domain Adversarial Training. In *Proceedings of the 39th International Conference on Machine Learning*.

Roggen, D.; Calatroni, A.; Rossi, M.; Holleczek, T.; Förster, K.; Tröster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Ferscha, A.; Doppler, J.; Holzmann, C.; Kurz, M.; Holl, G.; Chavarriaga, R.; Sagha, H.; Bayati, H.; Creatura, M.; and Millàn, J. d. R. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, 233–240.

Saxena, A.; Goebel, K.; Simon, D.; and Eklund, N. 2008. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, 1–9.

Sun, B.; Feng, J.; and Saenko, K. 2017. *Correlation Alignment for Unsupervised Domain Adaptation*, 153–171. Cham: Springer International Publishing. ISBN 978-3-319-58347-1.

Sun, B.; and Saenko, K. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In Hua, G.; and Jégou, H., eds., *Computer Vision – ECCV 2016 Workshops*, 443–450. Cham: Springer International Publishing. ISBN 978-3-319-49409-8.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial Discriminative Domain Adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2962–2971.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv preprint arXiv:1412.3474*.

Vayyat, M.; Kasi, J.; Bhattacharya, A.; Ahmed, S.; and Tallamraju, R. 2022. CLUDA: Contrastive Learning in Unsupervised Domain Adaptation for Semantic Segmentation. *arXiv preprint arXiv:2208.14227*.

Xu, Q.; Chen, Z.; Wu, K.; Wang, C.; Wu, M.; and Li, X. 2022. KDnet-RUL: A Knowledge Distillation Framework to Compress Deep Neural Networks for Machine Remaining Useful Life Prediction. *IEEE Transactions on Industrial Electronics*, 69(2): 2022–2032.

Yang, T.; Yu, X.; Ma, N.; Zhao, Y.; and Li, H. 2021. A novel Domain Adaptive Deep Recurrent Network for multivariate time series prediction. *Engineering Applications of Artificial Intelligence*, 106: 104498.

Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; and Gao, R. X. 2019. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115: 213–237.

Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; and He, Q. 2021. Deep Subdomain Adaptation Network for Image Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4): 1713–1722.