Which Time Series Domain Shifts can Neural Networks Adapt to?

Henri Hoyez*‡, Bruno Mirbach †, Jason Rambach†, Cedric Schockaert*, Didier Stricker†‡
*Paul Wurth S.A
{henri.hoyez, cedric.schockaert}@sms-group.com
†German Research Center for Artificial Intelligence (DFKI)
{bruno.mirbach, jason.rambach, didier.stricker}@dfki.de
‡RPTU Kaiserslautern

***Abstract*—Machine Learning on time series has great potential in solving real-life problems, from medical monitoring to machine predictive maintenance. In practice, however, the deployment domain can differ from the training conditions. These domain shifts give rise to the domain adaptation field which extends to time series data. However, despite the fact that broad domain shift definitions were already proposed, there is no clear definition adapted specifically to time series in the current literature. Moreover, even though evaluation methods were proposed, evaluations on real data do not allow a full control over the benchmarks. In this paper, we first propose a novel definition of domain shift based on a State Space Model (SSM). Then, we introduce a new dataset based on these SSMs to provide a benchmark using controlled domain shifts. Lastly, we perform an evaluation of the State of the Art on these generated domain adaptation problems, and hence systematically evaluate the domain shift effects on the adaptation performances.**

***Index Terms*—Domain Adaptation, Domain Shift, Machine Learning, Multivariate Time Series, Time Series**

I. Introduction

In research, a theoretical model has often to adapt to the experiment results. This analogy can also be found in the field of machine learning, where models trained on a particular dataset may be confronted with different configurations during deployment. This domain shift has been extensively studied in the scientific literature and has given rise to the field of domain adaptation originally in computer vision with various methods [1].

Domain adaptation aims to help a model to bridge the gap between a source and a target domain. Preliminary methods for domain adaptation try to minimize a statistical distance, such as Maximum Mean Discrepancy (MMD), in order to bring source and target domains closer together [2]–[4]. Moreover, Adversarial-based methods were also introduced, which train a feature extractor network to extract a domain invariant representation with the constraint of a domain discriminator [5] or by Gradient Reversal Layer [1]. Since then, this research field attracted a lot of interest from the research community and has been expanded to sub-domains such as Closed set [5], Open set [6], Partial [7] and Universal Domain Adaptation [8].

In recent years, this interest has been extended to the field of time series and all related applications. However, time series data introduces new challenges such as the lack of certain data points or dynamic reaction time [9]. Moreover, despite the fact that evaluation approaches have already been proposed [10], the datasets used do not allow carrying out a precise evaluation. Ragab et al. [11] have proposed an approach for evaluating domain adaptation methods across multiple datasets. This method has the particularity of standardizing architectures, making the focus on the method rather than the architecture employed. Even though this method is a building block towards a standardized evaluation method, it evaluates architectures on real datasets and hence does not lead to a fully controllable evaluation in terms of domain shift. Furthermore, during the literature investigation, we identified changes towards more specific domain shift definition [9], [12]. Nevertheless this definition does not fully expose the relationship between time series which is an essential feature for real world application.

The current literature did not deeply investigate the capability of proposed methods to close the gap between specific domain shift in time series. In reaction to that, we first propose a definition of domain shift for time series. Then, based on this definition, we introduce domain shifts with an increasing level of difficulty using a State Space Model (SSM). We then detail our approach to evaluate recently released domain adaptation models for comparison. Our dataset and the simulator code is available at https://github.com/Henri-Hoyez/ChemicalStateSpaceModel.

II. Related Work

A. Domain Adaptation in Time Series

Unsupervised Domain Adaptation (UDA) received great attention in computer vision tasks such as Medical Imaging [13] or Semantic Segmentation [14]. Since the first domain adaptation approaches, many publications contributed to the progression of the domain adaptation field for time series [8], [12], [15].

Cai et al. [9] argued that time series can vary not only in amplitude but also in time shift. They also suggest the idea that the causal relationships are stable across domains. They hence propose SASA, a method based on the extraction of sparse associative structure as a domain-invariant representation. To be robust against the varying time delay between input and

This work is supported by Fonds National de la Recherche (FNR), Luxembourg grant AFR-PPP, number 15411817.

output, they develop Adaptive Segment Summarization. Jin et al. [16] introduce a Domain Adaptation Forecaster (DAF) where the adaptation is made by sharing the attention key and value as domain-invariant and domain-specific vectors respectively. These two vectors can then be used by their multi-horizon forecasting model. Wang et al. [17] proposed to decompose the multivariate time series into endo- and exo-features. The endo-features represent local dependency, more precisely one sensor and its interactions, the exo-feature represents the sensors together. They align these features by using a contrastive learning approach on built sequential graphs.

Focusing on another representation, He et al. [18] suggest that domain shift can happen in the time and frequency domain and show that frequency features are more domain invariant than time features. Using a time-frequency feature extractor and the latest advancement of domain adaptation in images [8], RAINCOAT proposed a universal domain adaptation approach for time series. To further correct misalignments and detect unknown classes, they propose "align-then-correct". On the other hand, Liu et al. [2] note that the common feature extractors in MMD-based approaches fail to capture the most useful statistics of time series, such as non-stationarity. To address this issue, they developed a hybrid spectral kernel matching technique as an improved feature extractor. This method extracts more appropriate statistics for a reinforced MMD. AdvSKM is trained through an adversarial approach and also reduces the computational load.

In contrast to the work above, Ozyurt et al. [19] introduced CLUDA. They propose a contrastive learning algorithm that extracts the contextual representation of embedding which preserves the label information. To further align these contextual representations between domains, they use nearest-neighbor contrastive learning. Another work based on contrastive learning is CoTMix by Eldele et al. [20]. They state that better augmentation methods can be used to preserve the temporal dependency between variables. They hence introduce a new augmentation method called Temporal Mixup which generates a "source-dominant" and a "target-dominant" augmentation from the source and target domain respectively. This augmentation method preserves the temporal consistency for the time series and helps the features extractor to extract domain invariant representations.

III. Domain Shift Definitions

In this section, we will introduce a general definition of domain shift, then we will relate some contributions that introduce more specific domain shift definition.

Given a labeled source domain $\mathcal{D}_S = \{X_i^S, Y_i^S\}^{N_S}$ of size N_S and a target domain $\mathcal{D}_T = \{X_i^T, Y_i^T\}^{N_T}$ of size N_T. Where X and Y represent the input time series and the label, either from the source or the target domain. The label Y is defined by a task such as a classification or a regression. Using this notation, we can define a domain shift as a deviation between $\mathcal{D}_S$ and $\mathcal{D}_T$.

Beyond this definition, Stojanov et al. [12] formally defined more specific domain shifts using the decomposition of the joint probability distribution formula.

$$P(X, Y) = P(X \mid Y)P(Y) \tag{1}$$

In Eq. 1, $P(X, Y)$ represents the joint probability distribution, $P(X \mid Y)$ defines the conditional distribution, and $P(Y)$ describes the output distribution. From this equation, they defined three domain shifts:

- **Target Shift:** Where $P(Y^S) \neq P(Y^T)$ but $P(X^S \mid Y^S) = P(X^T \mid Y^T)$.
- **Conditional Shift:** Where $P(X^S \mid Y^S) \neq P(X^T \mid Y^T)$ but $P(Y^S) = P(Y^T)$.
- **Conditional-target Shift:** Both joint probability distribution and conditionals changes independently.

This definition provided a strong foundation. However, as the majority of work in domain adaptation comes from images, it is normal to see image data as probability distributions due to the complexity of the data. Furthermore, chaotic time series such as financial data can be seen as stochastic processes such as a Wiener process. Nevertheless, time series are often recorded with sensors of particular industrial assets or humans in physiological monitoring. Hence, this representation does not enhance the temporal relationships and correlations present in all multivariate time series.

The second paper closely related to our work is from Ragab et al. [11], who state that the evaluation method for time series domain adaptation suffers from inconsistency. Based on this finding, they propose a unified evaluation protocol where an architecture normalization is adopted. Even though this modification shifts the focus of the experiments to the adaptation methodology, the evaluation method is conducted on real data which provides a less controllable domain shifts.

IV. Domain Shift Time Series

Time series can tell us the history of chemical reactions, physical or physiological states. These examples can be embedded in the form of a system, where an output is given in response to some inputs. More formally, a given system can be modeled as a State Space Model:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{2}$$

Where A is the state matrix, B is the input matrix, C is the output matrix, and D is the feed-forward matrix. $x(t)$, $u(t)$ and $y(t)$ are the state vector, the multivariate inputs and the output of the system respectively.

However, even if this form is able to introduce the time dependency between samples, this system is not able to model all types of real-world systems as many of them are non-linear. To solve this problem, we modify the Eq. 2 as follows.

$$\begin{cases} \dot{x}(t) = A(t)x(t) + Bu(t-\tau) \\ y(t) = h(Cx(t) + Du(t-\tau)) \end{cases} \tag{3}$$

Where h is a function which aims to introduce non-linearity, and τ introduces time delay between the input and the reaction of the system. $A(t)$ represents the fact that the state of the system is dynamic and can react differently through time.

Inspired by [12] and the field of Image-to-Image Translation, we define several sub-fields of domain shift:

- **Content Shift:** The content shift is defined as a change in the input and the output signals. For example, two same cars that drive at different speeds on the same road will not need the same inputs. The system is the same but operated differently, which defines a content shift. More formally, this can be seen as a change of $u(t)$ and $y(t)$ in Eq. 3.
- **Style Shift:** Similarly to Wang et al. [17], we define the style shift as a change in the relationship between all time series. If we take the vehicle example again, since a style shift is a change inside the system, this style shift could be a change between car and truck. More formally, the style shift is defined by a deviation in the matrices A, B, C, D, and τ in Eq. 3.
- **Content-Style Shift**: A Content-Style shift happens when there is both a style and a content shift.

In this given definition, we modify either the signals or the relationship between these. We hence make precise modification in $P(X)$ and $P(X \mid Y)$. Therefore, these 3 shifts belong to the conditional shift concept depicted in III.

V. Problem Definition

Unsupervised domain adaptation aims to extract a knowledge learned in a source domain in order to be robust in an unsupervised target domain. More precisely, given again a source domain $\mathcal{D}_S = \{X_i^S, Y_i^S\}^{N_S}$ and an unlabeled target domain $\mathcal{D}_T = \{X_i^T\}^{N_T}$, the aim of a domain adaptation method is to train a classifier $f(.)$ to perform well in both $\mathcal{D}_S$ and $\mathcal{D}_T$. Because the largest part of the literature focuses on closed-set domain adaptation, the scope of this paper is defined within closed-set domain adaptation, where the label space is the same between the source domain and the target domain. More specifically, this means that we want to challenge the domain adaptation literature on the *conditional-shift* by modifying the relationships between the variables $u(t)$ and $y(t)$ in Eq. 3.

VI. Evaluation Methodology

A. Dataset:

To evaluate the representative works of the current literature on their ability to adapt to specific domain shifts, we generate a dataset that simulates a simple chemical reaction between two chemical components that come into contact.

$$3u_1 + u_2 \xrightarrow{k_f} 2y_1 + y_2 \tag{4}$$

As depicted in Eq. 4, the simulated dataset is composed of two input concentrations, u_1 and u_2, to compute the output concentrations of y_1 and y_2. In our simulation, u_1 and u_2 are entered at a given velocity $v_c(t)$ and temperature $T(t)$. This velocity and temperature control the chemical reaction intensity k_f as defined in Eq. 5. The coefficient k_f introduces dynamic time shift and step response as depicted by [9].

$$k_f(v_c, T) = e^{-[\alpha_2 \beta v_c - \alpha_1 (1-\beta)(T - T_{max})]} \tag{5}$$

In Eq. 5, the importance of a change in T or v_c for k_f is controlled by β. α_1 and α_2 control the slope of $k_f(v_c, T)$ when T or v_c varies. This reaction intensity will be denoted as k_f for clarity. Then, from this generated data, we define a classification task where the goal is to classify the range of temperature where the chemical reaction occurred.

For the purpose of introducing content shifts, the inputs of the system are defined as sinusoidal functions in Eq. 6:

$$u_{i=\{1,2\}}(t) = a \sin(2\pi f t) + b + \epsilon(t) \tag{6}$$

Where a is the amplitude of the input, b is the mean of the input signal, f represents the frequency and $\epsilon(t) \sim \mathcal{N}(\mu, \sigma)$ is a noise.

Since our domain shift definition is based on SSMs, we can then define the dynamics of the Eq. 3 using the following matrices:

$$A = \begin{bmatrix} 3\rho^2 k_f - 3k_f & 0 \\ 0 & \rho^2 k_f - k_f \end{bmatrix}; B = \begin{bmatrix} \rho \\ \rho \end{bmatrix};$$

$$C = \begin{bmatrix} 2k_f - 2\rho^2 k_f & 2k_f - 2\rho^2 k_f \\ k_f - \rho^2 k_f & k_f - \rho^2 k_f \end{bmatrix}; u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}; \tag{7}$$

In Eq. 7, ρ is the material flow rate, $u_1(t)$ and $u_2(t)$ are our input concentrations. These equations define the dynamics of the output concentrations $y_1(t)$ and $y_2(t)$.

In this simulation, parameters a, b, f, v_c, T, $\epsilon(t)$, and τ control the input of our system. Moreover, the inner reaction of the system is influenced by the parameter k_f, and hence β, α_1, and α_2. Since all these parameters contribute to the time series relationships, this set of parameters defines its own domain. Furthermore, we introduce several controlled domain shifts including content and style domain shifts.

- **Input amplitudes shift (content):** An amplitude shift between two simulations is defined when the amplitude factor a in the input Eq. 6 differs between the source and the target simulations.
- **Input mean shift (content):** This shift happens when the coefficient b in the Eq. 6 differs between two simulations source and target.
- **Input frequencies shift (content):** Defined by a modification of the frequency coefficient f in the Eq. 6 between two simulations.
- **Noises shift (content/style):** This noise can occur in input, output signal or both. For the input noise, we modify the noise variance of $\epsilon(t)$ in the Eq. 6. For the output noise we add a gaussian white noise term to the function h in the Eq. 3.
- **Reaction-time shift (style):** The reaction time shift is defined as the time delay between the action and the

reaction of the system. This is modeled by a modification of the factor τ in the Eq. 3 between two simulations.

- **Non-linearity (style):** To introduce non-linearity in our system, we modify the h function of the Eq. 3.
- **Causal shift (style):** Causal shift is defined by a modification of the reaction of the system due to a change of the matrix of the SSM. In our case, this shift is implemented by changing the parameters β in Eq. 5 which influences the reaction intensity k_f in the matrix $A(t)$ in Eq. 3. In this way, we modify the inner relations in the system by making it more sensitive to a change of temperature T or gas velocity v_c.
- **Shift in unit (content):** It defines a modification of the input signal which does not affect the reaction of the system. For example, an input can be defined as a material flow or can be normalized by a time varying production. We implemented this causal shift with a progressive shift of parameter B in the Eq. 3.

B. Experimental Settings:

In our experiments, we first consider a generated source domain with a fixed set of parameters. From this set of parameters, we simulate deviations by modifying specific parameters in the source domain to generate a target domain. Therefore we vary a parameter of the SSM in 10 steps to define target systems with specific domain shift of different intensity. For each domain shifts, each tested method is trained on the source to target scenario. Each method has been trained 10 times on each domain adaptation problem. In order to focus on the literature challenges, we will present our benchmark on the *Causal shift*, *Input/Output noise*, and *Reaction-time shift*.

- **Causal shift:** In this experiment, we want to simulate two different systems. To make these variations, we progressively modify β inside the Eq. 5 in the range 0 to 0.9 with a step of 0.1.
- **Input/Output Noise:** We want to know which models are more robust to noises. These noises can be present in the input or the output. To accomplish this, we apply a variation by applying a modification on the input noise $\epsilon(t)$ in the Eq. 6. For the output noise we modify h in the Eq. 3.
- **Reaction-time shift:** Here, we want to know the ability of the literature to adapt to a system with different reaction times. To do this, we apply modifications on τ in the Eq. 3.

C. Compared Methods:

In this paper, we want to specifically investigate the capability of domain adaptation methods to handle simulated domain shifts. For this purpose, we test four domain adaptation methods. CLUDA [19], CoTMix [20], AdvSKM [2] and RAINCOAT [18]. These domain adaptation methods will have to classify between 3 ranges of temperatures where the simulated reaction occurs. For evaluation and visualization purposes, methods will be compared thanks to the accuracy

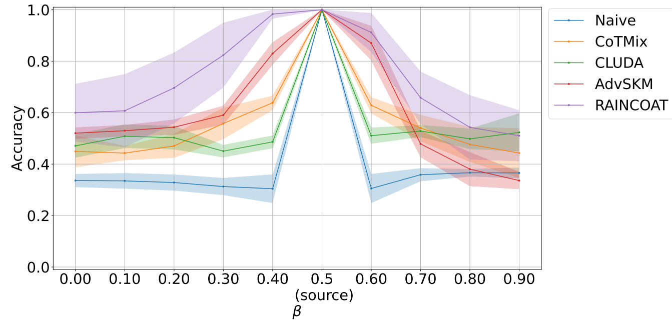

Fig. 1. Overall accuracy of evaluated literature on *Causal shift*. Where β is influencing k_f in the Eq. 5. The area represents the standard deviation of the metric.

metric, which is the number of correctly classified samples divided by the total number of samples.

Inspired by [11] and [18], we modified the paper's implementation to normalize the architecture of the feature extractors and classifier in terms of number of trainable weights. We compare all these approaches against a source-only classifier, which is a simple 1D convolutional classifier trained on source data only. For the presented results, each model has been trained on a source-target scenario with the goal of being as robust as possible on the target dataset.

VII. Results

Fig. 1 is representing the accuracy on the *Causal Shift* domain adaptation task. In this test, we progressively change β in the Eq. 5. This variation change the coefficients inside the $A(t)$ matrix in Eq. 3 and hence its internal workings. The source dataset has a $\beta = 0.5$. Firstly, we can observe that all domain adaptation methods smoothed out the curve of the source only model. Especially AdvSKM and RAINCOAT are shown to degrade more slowly with the strength increase of the domain shift. Moreover, RAINCOAT is the best performing model outperforming the literature in the majority of β. This can be attributed to the fact that RAINCOAT uses a frequency representation in the latent space that is less sensitive to domain shift.

In our second test, we test the literature on its ability to adapt to input and output noise. In the Fig. 2, we can see that RAIN-

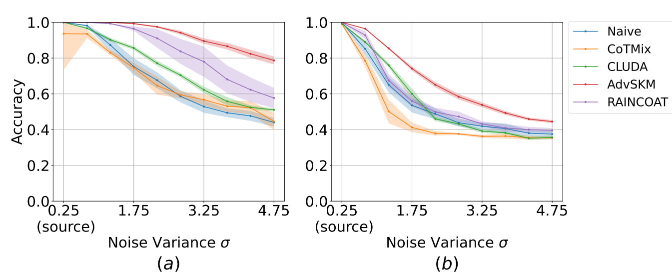

Fig. 2. *(a)*, Overall accuracy of the tested approaches on the *Input Noise* domain shift. *(b)*, Overall accuracy of the tested approaches on the *Output Noise* domain shift. The area represents the standard deviation for the accuracy.

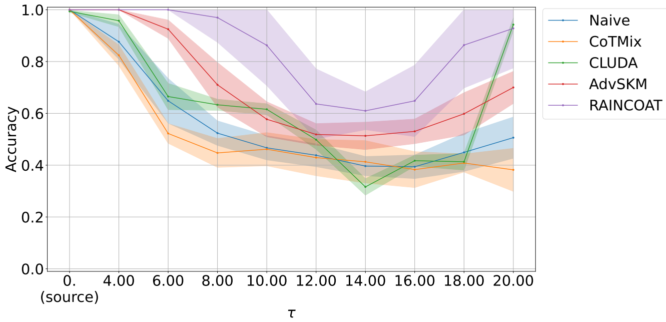

Fig. 3. Accuracy on the time shift domain adaptation task, where τ represents the shift in unit of data points. The area represents the accuracy standard deviation between runs.

COAT is not outperforming the literature being overtaken by AdvSKM in both input and output noise domain shift. This is because white noise has a value in all frequencies and this may disturb the frequency feature extractor of RAINCOAT. In this domain shift, AdvSKM is performing better. This could be explained because the inner working of AdvSKM is based on spectral features extraction.

The performance of the literature under a reaction-time shift is displayed in Fig. 3. In this domain adaptation task, we can see that the phase extraction in its time encoder makes RAINCOAT outperform the literature. Furthermore, the accuracy increases after $\tau = 14$ because the periodicity of our input signal.

Overall, despite the fact that contrastive learning methods enable automatic domain invariant feature extraction, these extracted features are not the most optimal. Presented results suggest that frequency-based features extraction is a big advantage in the domain adaptation scenarios. However, these frequency features can be fooled by a noisy domain adaptation task. Here, a simpler approach such as AdvSKM is preferable. Nevertheless, these frequency features extractor are subject to the Shannon law and hence limit the minimum sequence length for smaller problems.

VIII. Conclusion

In this paper, we first provided a definition of domain shift adapted to time series and dynamic systems. From this definition we designed a simulator based on SSMs and modified its parameters to construct domain adaptation scenarios with an increasing level of complexity. With these simulations, we made a controlled benchmark of recent domain adaptation methods over specific types of domain shifts. The results suggest that methods like RAINCOAT or AdvSKM based on frequency feature extraction tend to extract a more general representation and hence perform better than contrastive learning algorithms, even if the non periodic settings have not been investigated yet. We have released the simulator code in order to provide a controlled evaluation benchmark to the community.

References

[1] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning (ICML)*, 2015.

[2] Q. Liu and H. Xue, "Adversarial spectral kernel matching for unsupervised time series domain adaptation," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2021.

[3] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, 2016.

[4] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 443–450.

[5] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, 2018.

[6] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[7] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[8] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[9] R. Cai, J. Chen, Z. Li, W. Chen, K. Zhang, J. Ye, Z. Li, X. Yang, and Z. Zhang, "Time series domain adaptation via sparse associative structure alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[10] Y. Shi, X. Ying, and J. Yang, "Deep unsupervised domain adaptation with time series sensor data: A survey," *Sensors*, 2022.

[11] M. Ragab, E. Eldele, W. L. Tan, C.-S. Foo, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Adatime: A benchmarking suite for domain adaptation on time series data," *ACM Transactions on Knowledge Discovery from Data*, 2023.

[12] P. Stojanov, Z. Li, M. Gong, R. Cai, J. Carbonell, and K. Zhang, "Domain adaptation with invariant representation learning: What transformations to learn?" *Advances in Neural Information Processing Systems*, 2021.

[13] H. Shin, H. Kim, S. Kim, Y. Jun, T. Eo, and D. Hwang, "Sdcuda: Volumetric unsupervised domain adaptation framework for slice-direction continuous cross-modality medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[14] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[15] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, 2021.

[16] X. Jin, Y. Park, D. Maddix, H. Wang, and Y. Wang, "Domain adaptation for time series forecasting via attention sharing," in *International Conference on Machine Learning*, 2022.

[17] Y. Wang, Y. Xu, J. Yang, Z. Chen, M. Wu, X. Li, and L. Xie, "Sensor alignment for multivariate time-series unsupervised domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[18] H. He, O. Queen, T. Koker, C. Cuevas, T. Tsiligkaridis, and M. Zitnik, "Domain adaptation for time series under feature and label shifts," in *International Conference on Machine Learning*, 2023.

[19] Y. Ozyurt, S. Feuerriegel, and C. Zhang, "Contrastive learning for unsupervised domain adaptation of time series," *ICLR*, 2023.

[20] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Contrastive domain adaptation for time-series via temporal mixup," *IEEE Transactions on Artificial Intelligence*, 2023.