ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446381

DCS: DEBIASED CONTRASTIVE LEARNING WITH WEAK SUPERVISION FOR TIME SERIES CLASSIFICATION

Rongyao Cai[1, 2], Linpeng Peng[1], Zhengming Lu[1], Kexin Zhang[1, 2,], Yong Liu[1, 2,*]*

[1] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China
[2] Huzhou Institute of Zhejiang University, Huzhou, China

ABSTRACT

Self-supervised contrastive learning (SSCL) has performed excellently on time series classification tasks. Most SSCL-based classification algorithms generate positive and negative samples in the time or frequency domains, focusing on mining similarities between them. However, two issues are not well addressed in the SSCL framework: the sampling bias and the task-agnostic representation problems. Sampling bias indicates fake negative sample selection in SSCL, and task-agnostic representation results in the unknown correlation between the extracted feature and downstream tasks. To address the issues, we propose **D**ebiased **C**ontrastive learning with weak **S**upervision framework, abbreviated as **DCS**. It employs the clustering operation to remove fake negative samples and introduces weak supervisory signals into the SSCL framework to guide feature extraction. Additionally, we propose a channel augmentation method that allows the DCS to extract features from local and global perspectives simultaneously. The comprehensive experiments on the widely used datasets show that **DCS** achieves performance superior to state-of-the-art methods on the widely used popular benchmark datasets.

Index Terms— Time series classification, weak supervision, contrastive learning, data augmentation

1. INTRODUCTION

With the rapid growth of the Internet of Things and other monitoring systems, there has been an enormous increase in time series data[1]. Time series classification (TSC), which comprehensively recognizes system patterns, has recently become a popular research topic. TSC plays a crucial role in different application domains, e.g., action recognition [2], transportation [3] and healthcare [4]. Self-supervised contrastive learning (SSCL) has achieved great success in TSC problems in the absence of labeled data [5]. SSCL utilizes consistency constraints between different views of one sample to extract a semantic representation of the sample, named the pretraining step. Then SSCL maps the extracted representations to the downstream tasks with the guidance of a few annotated data, named as fine-tuning step. This strategy refines the semantic information of original data and improves the efficiency in using labels [6].

However, there are two challenges when using SSCL for feature extraction: 1) distribution issue: Conventional SSCL typically results in features being evenly distributed on a hypersphere, which is clearly at odds with the goal of TSC tasks to have features exhibit a clustered distribution in hidden space [7]; 2) task-agnostic representation issue: The pretraining and fine-tuning are two separate steps, which result in the absence of a constraint between the extracted representations and downstream tasks. There is a risk of emerging sub-optimal representation, which can consequently impact model performance in downstream tasks.

Sampling bias is the primary cause of the distribution issue. In the traditional SSCL, only the augmented view of a sample is regarded as a positive sample, while the rest are treated as negatives. This results in each sample being treated as an independent class in SSCL, leading to a uniform feature distribution. Modifying the conventional Info-NCE loss, which enhances the discriminative capability of the loss function for actual positive samples, represents a viable approach [8]. Furthermore, to tackle the task-agnostic representation issue, it is imperative to introduce weak supervisory signals that guide the feature extractor's training, ensuring the extracted representation's relevance to downstream classification tasks.

Based on the above analysis, we propose a novel framework for the TSC tasks named **D**ebiased **C**ontrastive learning with weak **S**upervision framework, abbreviated as **DCS**. The main contributions are summarized as follows:

- DCS introduces supervisory signals into SSCL via weight dual-updating scheme (WDUS) to tackle task-agnostic representation issue. And cluster-wise SSCL is employed to deal with sampling bias.
- Channel augmentation is proposed to fuse the time and frequency information, enabling the framework to extract features from both local and global perspectives.
- A comprehensive evaluation compared with several

* Corresponding authors

This work was supported by the National Key R&D Program of China under Grant 2021YFB2012300.

state-of-the-art TSC models on UCR/UEA benchmarks, DCS releases prior performance.

2. METHODOLOGY

Given two datasets: labeled sub-dataset $\mathcal{X}^{super} \in \mathbb{R}^{M\times L\times C}$ and unlabeled sub-dataset $\mathcal{X}^{self} \in \mathbb{R}^{N\times L\times C}$. Samples in $\mathcal{X}^{super}$ contain the data and label, defined as $\mathcal{X}^{super} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M\}$, where $\mathbf{X}_m = (x_m, y_m)$. In contrast, samples in $\mathcal{X}^{self}$ only contain the data, i.e., $\mathcal{X}^{self} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, where $\mathbf{X}_n = (x_n)$. In most cases, $M << N$ and we define sampling factor $p = \frac{M}{N}$. Our goal is to obtain the robust feature extractor based on $\mathcal{X}^{super}$ and $\mathcal{X}^{self}$, capable of transforming raw time series data into representations suitable for downstream tasks.

As illustrated in Figure.1, our framework consists of the self-supervised and supervised branches. We aim to obtain the robust feature extractor, Self-Trunk, by introducing supervisory signals into the SSCL model to alleviate task-agnostic representation issue. Meanwhile, we employ the clustering-based pseudo-label generation mechanism to eliminate the sampling bias. After the pretraining stage, we train a simple classifier, Self-Classifier, that can map the features extracted by Self-Trunk to the downstream tasks.

2.1. Sample and Channel Augmentation

We generate the weak augmented views $\mathcal{X}_w$ and strong augmented views $\mathcal{X}_s$ of original samples $\mathcal{X}$ through jittering and permutation, respectively. On the one hand, strong augmentation involves distorting the chronological order of time series, thereby affecting the original characteristics. On the other hand, weak augmentation introduces some slight variations to the time series without significantly altering their shape.

In DCS, we generate $\mathcal{X}^{super}$ and $\mathcal{X}^{self}$ by dropping the labels of the original data $\mathcal{X}$ and sampling p percent of annotated data simultaneously. $\mathcal{X}^{super}$ and $\mathcal{X}^{self}$ are transformed to $\hat{\mathcal{X}}^{super}$, $\hat{\mathcal{X}}_w^{self}$ and $\hat{\mathcal{X}}_s^{self}$ by data augmentation, i.e.,

$$\hat{\mathcal{X}}^{super} = \mathcal{C}(\mathcal{X}^{super}, \mathcal{T}_w(\mathcal{X}^{super}), \mathcal{T}_s(\mathcal{X}^{super})), \quad (1)$$

$$\hat{\mathcal{X}}_w^{self} = \mathcal{T}_w(\mathcal{X}^{self}), \quad (2)$$

$$\hat{\mathcal{X}}_s^{self} = \mathcal{T}_s(\mathcal{X}^{self}), \quad (3)$$

where B is the batch size, p is the sampling factor of $\mathcal{X}^{super}$, L and C represent the length and the number of channels of a time series sample, respectively. $\hat{\mathcal{X}}^{super} \in \mathbb{R}^{3B*p\times L\times C}$ is the full augmented data of $\mathcal{X}^{super}$, $\hat{\mathcal{X}}_w^{self} \in \mathbb{R}^{B\times L\times C}$ and $\hat{\mathcal{X}}_s^{self} \in \mathbb{R}^{B\times L\times C}$ are the weakly and strongly augmented views of $\mathcal{X}^{self}$, respectively. $\mathcal{C}(\cdot)$ is a concatenation operation, $\mathcal{T}_w(\cdot)$ and $\mathcal{T}_s(\cdot)$ are weak and strong augmentations, respectively.

Generally, the time and frequency domains provide distinct yet equivalent descriptions of the same signal with identical semantics. The time domain primarily captures the local and dynamic properties. In contrast, the frequency domain emphasizes global and stable features. To effectively integrate time and frequency information, we propose a channel augmentation method that utilizes the amplitude and phase of frequency information as a new channel into the data. This expansion increases the dataset dimension $\mathbb{R}^{B\times L\times C}$ to $\mathbb{R}^{B\times L\times 3C}$ as follow:

$$\mathcal{R} + \mathcal{I}j = FFT(\mathcal{X}), \quad (4)$$

$$\mathcal{X} = \mathcal{C}(\mathcal{X}, \mathcal{R}, \mathcal{I}), \quad (5)$$

where $\mathcal{R}$ and $\mathcal{I}$ are the real and imaginary part of frequency respectively, $FFT(\cdot)$ is the fast fourier transform. After sample and channel augmentation operation, we acquire $\hat{\mathcal{X}}^{super} \in \mathbb{R}^{3B*p\times L\times 3C}$ and $\hat{\mathcal{X}}^{self} \in \mathbb{R}^{B\times L\times 3C}$.

2.2. Cluster-wise SSCL

The SSCL method plays a pivotal role in extracting valuable information from positive pairs while accentuating the differences in features among negative pairs [9]. However, the current implementation of SSCL employs a strategy where only two views, both augmented from the same sample, are utilized as positive pairs. In contrast, the remaining views are categorized as negative pairs. This approach treats each sample as an independent class, leading to a uniform distribution of features among samples with the same actual label, i.e., sampling bias. A uniform feature distribution is suboptimal for downstream tasks like classification or anomaly diagnosis. In the context of classification tasks, it proves advantageous to possess distinct clustering distributions of features derived from various samples [10].

To deal with the sampling bias, we perform a clustering operation on the features F to generate pseudo-labels denoted as $\mathcal{C}^{psd}$ and determine the relationship of F based on $\mathcal{C}^{psd}$. F with identical $\mathcal{C}^{psd}$ are considered positive, while the others are considered negative.

In the clustering process, we begin by specifying the desired number of clusters n and then randomly select n features as the initial cluster centers. Each feature is assigned $\mathcal{C}^{psd}$ based on its proximity to the nearest cluster center, determined by the Euclidean distance. Subsequently, we recalculate the cluster centers by averaging the features with the same $\mathcal{C}^{psd}$. This process is repeated iteratively until the label of each feature remains unchanged, signifying convergence.

The debiased cluster-wise InfoNCE loss is as follow:

$$\begin{aligned}\mathcal{L}_{Debiased} &= -\mathbb{E}[\frac{e^{sim(F,F^+)/\tau}}{\log(\sum_{n\in N(i)} e^{\text{sim}(F,F_n^-)})}] \\ &= \frac{1}{|I|}\sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{e^{sim(F_i,F_p^+)/\tau}}{\sum_{n\in N(i)} e^{sim(F_i,F_n^-)/\tau}},\end{aligned} \quad (6)$$

where F^+ and F^- are positive and negative features of F respectively, I is the set of features, $P(i)$ and $N(i)$ are the set

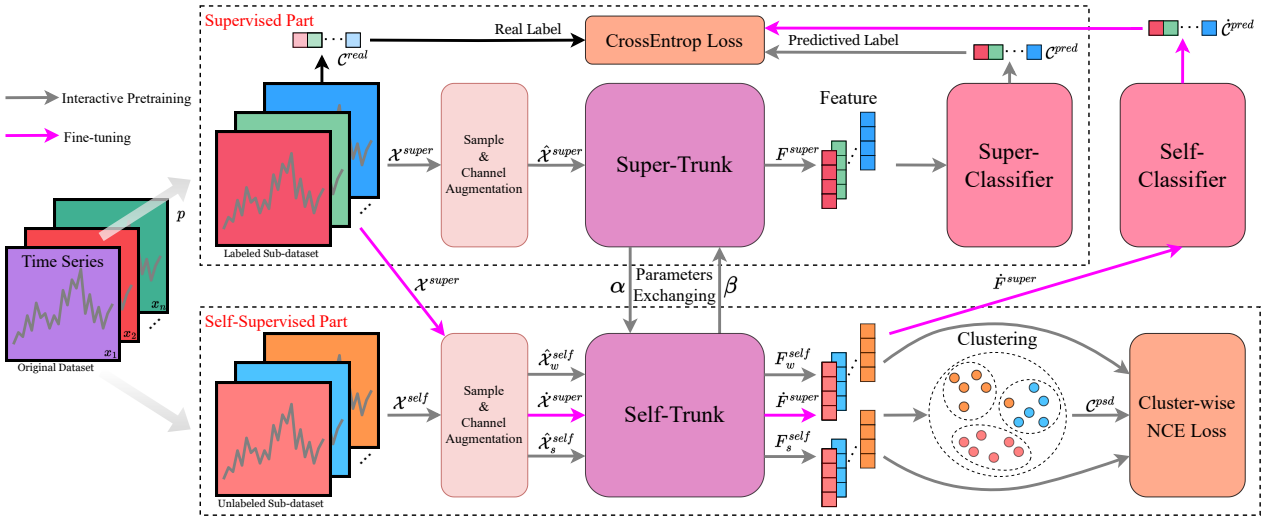

Fig. 1. Overview of the proposed framework. DCS consists of self-supervised and supervised branches, which are trained simultaneously. Sub-datasets $\mathcal{X}^{super}$ and $\mathcal{X}^{self}$ are sampled from original dataset $\mathcal{X}$. In supervised branch, $\mathcal{X}^{super}$ is augmented to $\hat{\mathcal{X}}^{super}$ and then extracts features F^{super} by Super-Trunk. Super-Classifier predicts the labels $\mathcal{C}^{pred}$ of F^{super}. We employ cross-entropy loss constraints supervised branch. In self-supervised branch, $\mathcal{X}^{self}$ generates the views $\hat{\mathcal{X}}_w^{self}$ and $\hat{\mathcal{X}}_s^{self}$. Then assigning labels $\mathcal{C}^{psd}$ to $\hat{\mathcal{X}}_w^{self}$ and $\hat{\mathcal{X}}_s^{self}$ via clustering operation. Finally, a cluster-wise NCE loss is utilized to constrain the self-supervised branch. When the two branches are trained in parallel, we introduce supervisory signals generated by the supervised branch to the self-supervised branch via WDUS.

of F^+ and F^- for F_i, $sim(\cdot,\cdot)$ denotes cosine similarity, τ is the temperature coefficient.

2.3. Weight Dual-updating Scheme

To generate effective supervisory signals, we construct the supervised branch in Figure. 1. Self-supervise branch and supervised branch are trained independently, resulting in the Self-Trunk and Super-Trunk, which are feature extractors of each branch. The Super-Trunk incorporates supervisory signals, while the Self-Trunk contains self-supervised information. Then, WDUS is employed to bidirectional exchange information between them.

WDUS treats Super-Trunk's parameters as supervisory signals and transposes the signal to Self-Trunk through weight updating. Simultaneously, Self-Trunk's parameters are also transposed to Super-Trunk through the same weight updating manner to narrow the gap between them and prevent parameter divergence. The mathematical expression of WDUS is as follows:

$$\mathcal{P}_i^{self} = \alpha * \mathcal{P}_i^{self} + (1-\alpha) * \mathcal{P}_i^{super}, i = 1, 2, ..., n, \tag{7}$$

$$\mathcal{P}_i^{super} = \beta * \mathcal{P}_i^{super} + (1-\beta) * \mathcal{P}_i^{self}, i = 1, 2, ..., n, \tag{8}$$

where $\mathcal{P}_i$ is the i^{th} corresponding parameter in the Self-Trunk and Super-Trunk, α and β are the weighting factors, n denotes the number of parameters in feature extractor.

3. EXPERIMENTS

3.1. Preparation

We comprehensively evaluate our proposed framework by adopting 28 widely-used time series datasets from popular archives. Specifically, we utilize 21 univariate datasets (DistalPhalanxOutlineAgeGroup $\sim$ ProximalPhalanxTW) from UCR archive [11] and 7 multivariate datasets (Epilepsy $\sim$ StandWalkJump) from UEA archive [12]. Each dataset has been separated as train sub-dataset and test sub-dataset.

We compare the performance of DCS with several SOTA time series classification algorithms, i.e., TS2Vec [5], TS-TCC [13], TNC [14], TST [15], T-Loss [16], DTW [17]. Due to space limitations, we only show the experimental results with accuracy (ACC) as an indicator.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where TP and TN represent the true positive and true negative, FP and FN indicate the false positive and false negative, respectively.

3.2. Implementation Details

Temporal convolutional network (TCN) [18] with six layers is employed as our main structure of Self-Trunk and Super-Trunk. Additionally, both Self-Classifier and Super-Classifier are implemented as two-layer MLP architecture. Considering that both UCR and UEA are small datasets with multi-classes,

Table 1. Experimental results on time series classification task. The best results are highlighted in bold, and the second-best results are underlined. The average rank describes the comprehensive performance of frameworks in various datasets.

Dataset	TS2Vec	TS-TCC	TNC	TST	T-Loss	DTW	DCS	Dataset	TS2Vec	TS-TCC	TNC	TST	T-Loss	DTW	DCS
DistalPhalanx OutlineAgeGroup	0.727	0.727	0.741	0.755	0.741	**0.770**	0.755	MiddlePhalanx OutlineAgeGroup	0.636	0.656	0.643	0.630	0.617	0.500	**0.662**
DistalPhalanx OutlineCorrect	0.775	0.775	0.754	0.754	0.728	0.717	**0.786**	MiddlePhalanx OutlineCorrect	0.838	0.825	0.818	0.818	0.753	0.698	**0.839**
DistalPhalanxTW	0.698	0.676	0.669	0.676	0.568	0.590	**0.705**	MiddlePhalanxTW	0.591	0.591	0.571	0.610	0.506	0.506	**0.630**
Earthquakes	0.748	0.748	0.748	0.748	0.748	0.719	**0.755**	MoteStrain	0.863	0.851	0.825	0.843	0.768	0.835	**0.878**
ECG200	0.920	**0.940**	0.830	0.880	0.830	0.770	0.890	ProximalPhalanx OutlineAgeGroup	0.844	0.844	0.854	0.839	0.854	0.805	**0.888**
ECG5000	0.935	0.933	0.937	**0.941**	0.928	0.924	0.938	ProximalPhalanx OutlineCorrect	**0.900**	0.859	0.866	0.873	0.770	0.784	0.893
FaceAll	0.805	0.786	0.766	**0.813**	0.504	0.808	0.808	Proximal PhalanxTW	**0.824**	0.771	0.810	0.800	0.780	0.761	0.820
FordA	0.948	0.928	0.902	0.930	0.568	0.555	**0.950**	Epilepsy	0.964	**0.971**	0.957	0.957	0.949	0.964	0.949
FordB	0.807	0.793	0.733	0.815	0.507	0.620	**0.826**	FingerMovements	0.480	0.580	0.470	0.460	0.560	0.530	**0.660**
GunPoint	0.987	0.980	0.967	**0.993**	0.827	0.907	0.933	Heartbeat	0.683	0.741	0.746	0.751	0.746	0.717	**0.756**
Ham	0.724	0.724	0.752	0.743	0.524	0.467	**0.829**	MotorImagery	0.510	0.580	0.500	0.610	0.500	0.500	**0.660**
Herring	0.641	0.594	0.594	0.594	0.594	0.531	**0.688**	SelfRegulationSCP1	0.812	0.843	0.799	0.823	0.754	0.775	**0.904**
InsectWingbeat Sound	**0.630**	0.597	0.549	0.415	0.266	0.355	0.525	SelfRegulationSCP2	0.578	0.539	0.550	0.533	0.550	0.539	**0.611**
ItalyPower Demand	0.961	0.954	0.928	0.955	0.845	0.950	**0.969**	StandWalkJump	0.467	0.333	0.400	0.333	0.267	0.200	**0.600**
Total Average Rank	2.679	3.179	3.786	3.107	5.000	5.071	1.607								

Table 2. Ablation study of DCS on MoteStrain. We randomly mask (expressed as ✓) three main components in DCS.

Dataset	Masked Component			ACC
	Channel Augmentation	Supervisory signal	Cluster-wise Constrastive loss	
MoteStrain				**0.880 ± 0.013**
	✓			0.855 ± 0.012
	✓	✓		0.671 ± 0.146
	✓		✓	0.813 ± 0.011
	✓	✓	✓	0.530 ± 0.088
		✓		0.832 ± 0.019
		✓	✓	0.805 ± 0.010
			✓	0.833 ± 0.006

we choose the supervised sampling factor $p = 0.2$ to obtain available supervisory signals. To reduce the impact of random initialization, each ACC value in Table.1 is the average of the results of multiple experiments. In the clustering process, the number of clusters n is larger than the actual number of categories c in the dataset, $n = c + 2$ [19]. According to orthogonal experiments, α and β are set to 0.6 and 0.8.

3.3. Performance Comparisons

It is clear that the DCS demonstrates its capability to achieve optimal performance in both univariate and multivariate datasets and outperforms the rest of the baselines by a great margin. Compared with the previous best algorithm, TS2Vec, the DCS framework exhibits a notable performance improvement on univariate datasets; meanwhile, the improvement becomes even more significant when applied to multivariate datasets. Compared to other algorithms, excluding TS2Vec, the DCS framework demonstrates superior performance, achieving an average increase of 2-3 percent in ACC.

3.4. Ablation Study

MoteStrain dataset is utilized to evaluate the feature extraction capability of DCS when dealing with datasets with limited samples, shown in Table 2. A basic model (strategy 5) only with Self-Trunk and classical InfoNCE is employed to compare the effectiveness of each component. When comparing strategy 5 with 3, 4, and 7, it is evident that each component contributes significantly to improving the performance of the basic model, respectively, with an increase in ACC of over 10 percent. We also compare the synergy between components, e.g., strategy 2 with 3. The pairwise fusion between components can improve its overall effectiveness to a certain extent. Finally, DCS achieves state-of-the-art performance in the MoteStrain dataset.

4. CONCLUSION

This article proposes a debiased contrastive learning framework with weak supervision (DCS) for time series classification tasks. Although SSCL has succeeded, the distribution and task-agnostic representation issues must be better investigated. We employ cluster-wise InfoNCE and WDUS with supervisory signals to fill the gaps. Besides, we propose a channel augmentation method to fuse time and frequency domain information. Based on these considerations, the proposed DCS framework can deal with the challenges of various TSC tasks. We obtain better performance on widely used datasets than the last classification algorithms. Extensive experimental studies on UCR and UEA benchmark datasets reveal that our DCS obtains a state-of-the-art performance. In the future, we plan to simplify our DCS framework and search for more effective supervisory signal generator.

5. REFERENCES

[1] Christos Faloutsos, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, and Yuyang Wang, "Forecasting big time series: Theory and practice," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 3209–3210.

[2] Vittorio Mazzia, Simone Angarano, Francesco Salvetti, Federico Angelini, and Marcello Chiaberge, "Action transformer: A self-attention model for short-time pose-based human action recognition," *Pattern Recognition*, vol. 124, pp. 108487, 2022.

[3] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32.

[4] Hyohyeong Kang and Seungjin Choi, "Bayesian common spatial patterns for multi-subject eeg classification," *Neural Networks*, vol. 57, pp. 39–50, 2014.

[5] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu, "Ts2vec: Towards universal representation of time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 8980–8987.

[6] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3988–4003, 2022.

[7] Tongzhou Wang and Phillip Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.

[8] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka, "Debiased contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 8765–8775, 2020.

[9] Philip Bachman, R Devon Hjelm, and William Buchwalter, "Learning representations by maximizing mutual information across views," in *Advances in Neural Information Processing Systems*, 2019, vol. 32.

[10] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng, "Class-aware contrastive semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14421–14430.

[11] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.

[12] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[13] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan, "Time-series representation learning via temporal and contextual contrasting," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021, pp. 2352–2359.

[14] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," *arXiv preprint arXiv:2106.00750*, 2021.

[15] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2114–2124.

[16] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi, "Unsupervised scalable representation learning for multivariate time series," *Advances in neural information processing systems*, vol. 32, 2019.

[17] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo EAPA Batista, "Dtw-d: time series semi-supervised learning from a single example," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 383–391.

[18] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018.

[19] Qianli Ma, Zhenjing Zheng, Jiawei Zheng, Sen Li, Wanqing Zhuang, and Garrison W Cottrell, "Joint-label learning by dual augmentation for time series classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 8847–8855.