Rank Supervised Contrastive Learning for Time Series Classification

Qianying Ren
School of Computing
University of Connecticut
qianying.ren@uconn.edu

Dongsheng Luo
Knight Foundation School of Computing & Information Sciences
Florida International University
dluo@fiu.edu

Dongjin Song*
School of Computing
University of Connecticut
dongjin.song@uconn.edu

***Abstract*—Recently, various contrastive learning techniques have been developed to categorize time series data and have exhibited promising performance for real-world applications. A general paradigm is to utilize appropriate data augmentation methods and construct feasible positive samples such that the encoder can yield robust and discriminative representations by mapping similar data points closer together in the feature space while pushing dissimilar data points farther apart. Despite its efficacy, the fine-grained relative similarity (*e.g.*, rank) information of positive samples is not fully exploited, especially when labeled samples are limited. To this end, we present Rank Supervised Contrastive Learning (RankSCL) to perform time series classification. Different from conventional contrastive learning frameworks, RankSCL augments raw data in a targeted manner in the embedding space and selects more informative positive and negative pairs for the targeted sample. Moreover, a novel rank loss is developed to assign higher weights to more confident positive pairs and lower weights to less confident positive pairs, enabling the encoder to extract the same class's fine-grained information and produce a clear boundary among different classes. Thoroughly empirical studies on 128 UCR and 30 UEA datasets demonstrate that the proposed RankSCL can achieve state-of-the-art performance compared to existing baseline methods. Code is available at: https://github.com/UConn-DSIS/Rank-Supervised-Contrastive-Learning-for-Time-Series-Classification**

***Index Terms*—time series classification, representation learning, contrastive learning**

I. Introduction

Nowadays, time series data are becoming ubiquitous in numerous real-world applications. For instance, in a power plant [1], a large number of sensors can be employed to monitor the operation status in real time. With a fitness tracking device, a temporal sequence of actions [2], *e.g.*, walking for 5 minutes and sitting for 15 minutes, *etc*, can be recorded and detected with related sensors. With the huge amount of time series data, how to categorize and interpret the status becomes a critical issue to investigate.

Traditionally, one of the most popular time series classification approaches is to use the nearest neighbor (NN) classifier based on a distance measure [3]. For instance, Dynamic Time Warping (DTW) distance has been used together with an NN classifier (DTW-NN) to provide a strong baseline [4]. More recently, Collective Of Transformation-based Ensembles (COTE) combines the strengths of multiple approaches to handle various aspects of time series data can yield better classification accuracy. Lines et al. further extended COTE to create HIVE-COTE [5] by incorporating a hierarchical vote system. These approaches, however, involve high complexity for both training and inference.

Deep learning-based time series classification methods, like InceptionTime [6], have gained popularity for their strong performance. However, supervised learning often requires large amounts of labeled data, which can be challenging to acquire in large-scale time series applications where labeled data is often scarce or difficult to obtain [7]–[11]. Recently, contrastive learning techniques have used augmentations to create positive samples, allowing encoders to produce robust and discriminative representations by bringing similar data points closer and pushing dissimilar ones apart. For instance, TimCLR [12] employs DTW [13] to handle temporal variations and achieve promising performance. However, these methods do not fully utilize positive samples' fine-grained relative similarity (e.g., rank).

To this end, we propose Rank Supervised Contrastive Learning (RankSCL) to tackle this issue and yield more effective representations to facilitate time series classification. The key idea is to rank the importance of different positive samples to better understand the potential landscape of feature space. Specifically, we make full use of the information of positive samples by leveraging their relatively similarity information in terms of rank. We encode the rank by taking account of the number of triplets in which the distance of anchor-negative pairs is smaller than anchor-positive pairs. A targeted data augmentation technique is designed to generate designated samples, aiming to enrich the information for the same category and enhance the boundary from different categories. By combining these two techniques, our proposed RankSCL has been thoroughly evaluated on 128 UCR datasets and 30 UEA datasets. Our experiment results demonstrate that the proposed RankSCL can achieve state-of-the-art performance. Our main contributions include:

- We develop a novel rank supervised contrastive learning framework and present a novel rank loss that assigns different weights to different levels of positive samples.
- We propose a targeted data augmentation technique based on RSCL to generate designated positive samples that can enrich the information of samples from the same category

* denotes the corresponding author.

2374-8486/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDM59182.2024.00102

and distinguish the boundary among different categories.

- Our empirical studies on 128 UCR datasets and 30 UEA datasets demonstrate that the proposed RankSCL outperforms the state-of-the-art.

II. Related Work

A. Contrastive Time Series Representation Learning

In contrastive learning, enriching the representation space through the appropriate generation of positive and negative pairs is crucial. Traditionally, positive pairs are closely aligned, while negative pairs are distanced, as in SimCLR [14], where different augmentations of the same sample are positive, and different samples are negative. Recent works have introduced new pair designs to capture invariant features. For example, TimCLR [12] uses DTW [13] for phase-shift and amplitude-change augmentations suited to time series. TS2Vec [15] defines contrastive loss at both instance-wise and patch-wise levels, separating time series into patches, while TS-TCC [16] introduces a temporal task to predict future sequences from augmentations. CoST [17] applies contrastive loss across time and frequency domains to capture seasonal and trend patterns, and TF-C [18] focuses on optimizing time and frequency-based representations. InfoTS [19] maximizes fidelity and variety in augmentations through information-theoretic principles. However, these approaches rarely leverage fine-grained relative similarity (e.g., rank) of positive samples.

B. Time Series Classification

Time series classification is a rapidly advancing field. Traditional non-deep learning methods like TS-CHIEF [20], ROCKET [21], and DTW-NN [22] have been foundational. However, deep learning models now outperform these methods by pursuing better representations. InceptionTime [6] applied inception networks to time series to capture local patterns. MACNN [23] used attention mechanisms to enhance the classification performance of multi-scale CNNs. For multivariate TSC, CA-SFCN [24] incorporated variable and temporal attention modulation.

C. Time Series Data Augmentation

Data augmentation is critical for deep learning in time series. Traditional methods fall into time, frequency, and time-frequency domains. In the time-frequency domain, Yao et al. [25] applied short Fourier transforms (STFT) to generate features from sensor data, enhancing human activity classification using a deep LSTM network.

III. Method

A. Notations and Problem Definition

A time series instance x_i is represented by a $T \times F$ matrix, where T is the time step and F is the feature dimension. With $F = 1$, x is a univariate instance, otherwise x is known as a multivariate instance. Given a set of N time series instances $\mathbb{X} = \{x_1, x_2, x_3, ..., x_N\}$, the objective is to learn a nonlinear function f_θ that maps each x to a D dimensional vector $\mathbf{v} \in \mathbb{R}^D$, which preserves its semantics and $D \ll T \times F$. In supervised settings, we have a subset of $\mathbb{X}$, denoted by $\mathbb{X}_L$, where each instance x is associated with a label y.

B. Framework

The overall architecture is shown in Figure 1. Raw time series data is input into an *encoder network*, f_θ, to learn low-dimensional representations, which are then passed to a *projection head*. Each sample serves as an anchor, with positive samples defined by label similarity [26]; those with different labels are negative. Data augmentation is applied to embeddings to enrich intra-class information. A rank-supervised contrastive loss is used to train both the encoder and projection head. For classification, we follow contrastive learning frameworks [14], discarding the projection head and using the encoder's hidden representations.

C. Model Architecture

The key components of our method include the encoder network and projection head.

Encoder Network $f_\theta(\cdot)$. As shown in Figure 1, we use a 3-layer Fully Convolutional Network (FCN) to map the input time series x_i to the representation $r_i = f_\theta(x_i)$. Each layer consists of a convolutional layer, batch normalization, and ReLU activation, followed by global average pooling to reduce weights.

Projection Head $g_\theta(\cdot)$. Inspired by SimCLR [14], we use a small MLP projection head to transform r to $z_i = g_\theta(r_i)$, followed by a normalization function to map the representations on a unit hypersphere. The projection head is used only during training and discarded during inference.

D. Data Augmentation on the Embedding Space

The selection of a suitable augmentation strategy for contrastive learning frameworks stands as a crucial challenge. A common approach involves selecting augmented data points that are substantially divergent from the original data, thereby introducing increased variability to foster a more robust encoder. Yet, this strategy often leads to issues of distributional shifts, as highlighted in recent studies [15].

Diverging from traditional contrastive learning approaches that apply data augmentation directly to raw time series data, our method addresses these challenges within the embedding space. For each training instance and its corresponding label (x_i, y_i) in the labeled dataset $\mathbb{X}_L$, we utilize the encoder network coupled with a projection head to generate a compact embedding z_i of x_i within a unit hypersphere. We introduce jittering with scale α, denoted by t_α as a selected augmentation operation. An augmented instance is then obtained through $z_i' = t_\alpha(z)$.

To further enhance diversity, we consider a set of two jittering operations with distinct scales, $\mathcal{T} = \{t_{\alpha_1}, t_{\alpha_2}\}$. In practice, we set $\alpha_1 = 0.03$ and $\alpha_2 = 0.05$. As indicated in prior research [15], augmentations performed in the hidden space effectively preserve label information. Consequently, both the augmented instances and their original labels are incorporated into the current training batch.

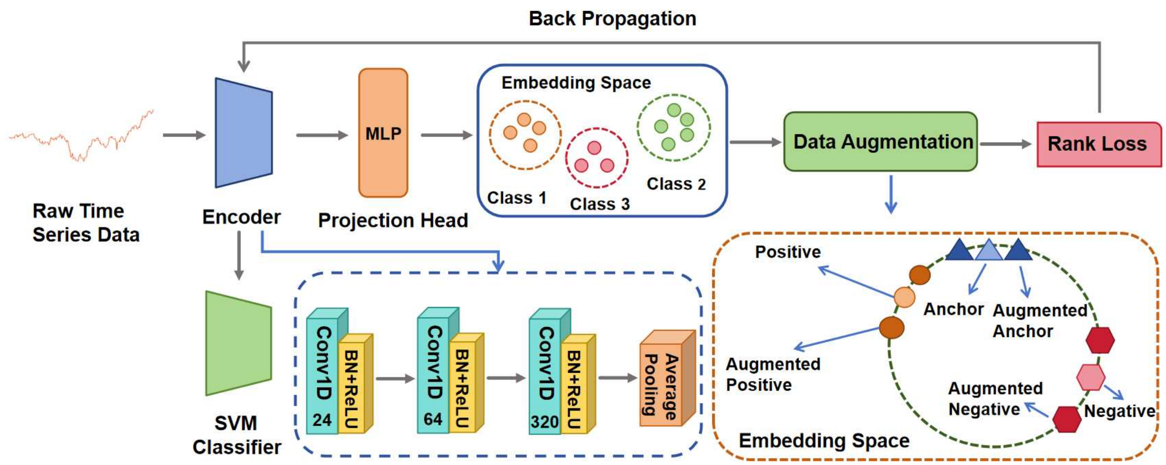


Fig. 1: Overview of RankSCL framework, consisting of three components: (1) a Fully Convolutional Network that captures the embeddings of raw time series instances, (2) targeted data augmentation that generates more samples in the embedding space, (3) selection of valid triplets and calculation of rank loss to train the encoder network. Even though this figure shows a univariate time series instance as an example, the architecture supports multivariate instances.

By applying small-scale jittering to embeddings in a compact unit hypersphere, our augmentation technique not only enriches valid information but also mitigates potential issues related to distribution drift or the creation of outliers.

E. Selection of Valid Triplet Pairs

In contrastive learning, the primary goal is to maximize the similarity between positive samples while keeping negative samples distinct. Early models used a single positive and negative pair per minibatch. Recent advancements, incorporating multiple positive and negative pairs [26], [27], have led to progress in fields like computer vision [26] and NLP [28]. However, these approaches often require considering all anchor-negative pairs, which can be computationally intensive.

To optimize this process, we propose the novel concept of a "valid hard negative pair" for contrastive learning. In this context, for an anchor instance x_a and a corresponding positive sample x_p, a negative sample x_n is deemed "valid hard" if the distance $dist(x_a, x_n)$ between the anchor x_a and the negative sample x_n is less than the distance between the anchor and the positive sample, i.e., $dist(x_a, x_n) < dist(x_a, x_p)$. A triplet formed by these criteria, (x_a, x_p, x_n), is then defined as a *valid triplet pair*. This approach focuses on more challenging and informative negative examples during training, thereby enhancing the discriminative power of the resultant representations.

F. Rank Supervised Contrastive Learning Loss

In traditional contrastive learning, a key overlooked aspect is the variable distances of different positive samples from the anchor. Treating all positive samples equally, without considering their proximity to the anchor, may not effectively capture the nuanced potential representations of a class. Additionally, this approach is susceptible to the influence of outliers, potentially introducing noise into the learning process.

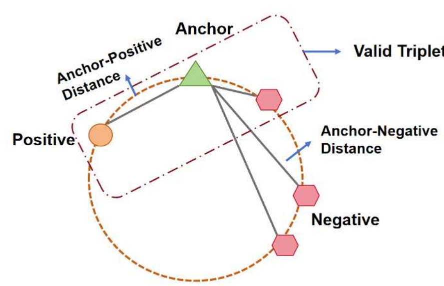


Fig. 2: Valid Triplet Selection

To address this issue, we introduce the concept of ranking different positive samples and propose a novel rank-supervised contrastive learning loss function. This function differentially weights positive samples based on their utility in training the model. We first define the set of valid hard negative samples for an anchor-positive pair (x_a, x_p) as:

$$\mathbb{X}_{ap}^{(n)} = \{x_n | dist(x_a, x_n) < dist(x_a, x_p)\} \tag{1}$$

The intuition behind this approach is that a positive sample with fewer valid hard negative samples is likely closer to the anchor and therefore more informative for the class representation. Such samples should be given more weight. Conversely, a positive sample with a larger $\mathbb{X}_{ap}^{(n)}$ might be distant from the class centroid and treated with lower weight, as it could potentially be an outlier introducing noise. To implement this, we rank positive samples based on the count of their valid hard negative samples. In this framework, distinct positive samples receive different weights during the learning process. A high-rank positive sample, indicating a larger number of valid hard negatives, may offer less valuable information for the class and is more likely to be an outlier. In contrast, a low-rank positive sample, with fewer valid hard negatives, is weighted more heavily, suggesting it provides more relevant information for the class.

Formally, we define the rank of a pair (x_a, x_p) based on the size of its valid hard negative sample set:

$$R(x_a, x_p) = \sum_{n}^{|\mathbb{X}_{ap}^{(n)}|} \mathbb{1}(dist(x_a, x_n) \leq dist(x_a, x_p)). \quad (2)$$

Here, $\mathbb{1}(\cdot)$ is an indicator function. This rank-based approach aims to enhance the model's ability to discern between more and less informative positive samples, thereby refining the training process and improving the quality of representations.

The discrete nature of the indicator function in the rank function $R(x_a, x_p)$ presents a challenge for optimization during the training process. To address this, we replace the indicator function with a continuous approximation using the sigmoid function $\sigma(\cdot)$. Consequently, the revised rank function becomes:

$$R(x_a, x_p) = \sum_{n}^{|\mathbb{X}_{ap}^{(n)}|} \sigma(dist(x_a, x_p) - dist(x_a, x_n)), \quad (3)$$

where $\sigma(k) = \frac{1}{1+e^{-k}}$ is the sigmoid function.

We introduce a novel objective function that assigns differentiated weights to positive samples based on their ranks, penalizing poorly ranked samples more heavily. Lower-ranked samples, which have more negative samples closer to the anchor, receive less weight as they may be outliers and contribute noisy information. We use the arctan function to map ranks, ensuring the loss increases less for higher-ranked samples as R grows. Our final objective function is:

$$\mathcal{L}(R(x_a, x_p)) = \sum_{a} \sum_{p} \arctan(R(x_a, x_p)) \quad (4)$$

Remark. The choice of mapping functions significantly affects our model's performance. We compared $\log(1 + z)$ with $\arctan(x)$ and found $\arctan(x)$ performed better. While our loss function concept is similar to [29], we uniquely apply the inverse tangent function to adjust the weights of positive samples.

IV. EXPERIMENTS

A. Datasets and Baselines

The classic two different kinds of benchmark datasets are used for the evaluation[1]. For the multivariate time series classification task we use the UEA archive [30] consists of 30 multivariate datasets, while the UCR archive [31] has 128 univariate time series datasets for the univariate time series classification task. The extensive experiments are conducted compared with other SOTAs, such as InfoTS [19], TimesNet [32], TS2Vec [15], T-Loss [33], TNC [34], TS-TCC [16], TST [35] and DTW [13]. To better test the quality of the representations, we chose more comprehensive metrics and all of the baseline models are measured by the same metrics.

[1]https://www.cs.ucr.edu/~eamonn/time_series_data/

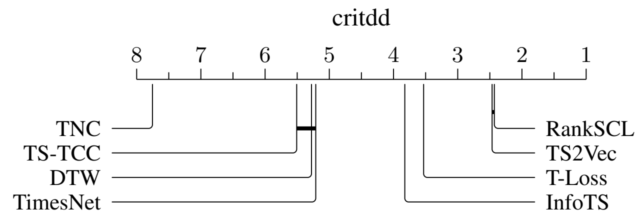


Fig. 3: Critical Difference (CD) diagram of Univariate Time series classification task

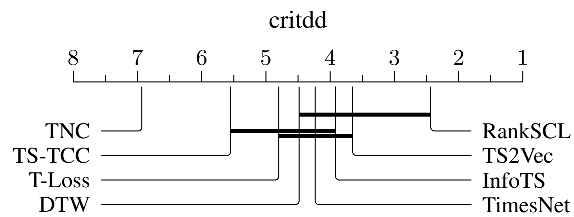


Fig. 4: Critical Difference (CD) diagram of Multivariate Time series classification task

B. Univariate Time Series Results

Table II summarizes the experiment results from the UCR datasets. RankSCL outperforms other baselines in terms of Precision score, F1 score, and Recall value. It improves 0.1% classification precision score, 0.4% F1 score, and 0.3% Recall value. The results of all the experiments are the results obtained by taking the mean value according to the 5 different seeds.

In particular, the RankSCL model achieves a similar accuracy score over the best baseline TS2Vec on all 128 UCR datasets. In terms of ACC, our method outperforms TS2Vec in 72 out of 128 UCR datasets, InfoTS [19] in 96 datasets, and T-loss [33] in 88 datasets out of 125 datasets. Figure 3 shows the Critical Difference diagram of different methods on 128 UCR datasets. This result indicates that RankSCL has similar results with the TS2Vec [15] model in terms of Accuracy scores, while significantly outperforming the other methods.

C. Multivariate Time Series Results

Table II presents the results on UEA multivariate time series datasets, where RankSCL outperforms all baselines across all metrics. Specifically, RankSCL surpasses the best baseline, TS2Vec [15], by 2.0% in Accuracy, 3.2% in Precision, 3.3% in F1 score, and 3.5% in Recall. These results highlight the effectiveness of RankSCL's augmentation techniques and rank loss function, particularly for multivariate time series data.

D. Ablation Study

We evaluate different variants of the RankSCL model on the PigCVP dataset, as shown in Table III, to assess the effectiveness of our method. To assess the effectiveness of

Dataset	RankSCL	InfoTS	TS2Vec	T-Loss	TS-TCC	DTW	TimesNet	TNC
UCR	**2.430**	3.824	2.465	3.531	5.508	5.277	5.211	7.754
UEA	**2.433**	3.917	3.650	4.800	5.550	4.483	4.233	6.933

TABLE I: Average Rank values for 128 UCR datasets and 30 UEA datasets

	128 UCR				30 UEA			
Method	ACC ↑	Prec. ↑	F1 ↑	Recall ↑	ACC ↑	Prec. ↑	F1 ↑	Recall ↑
RankSCL	0.821	**0.817**	**0.803**	**0.789**	**0.715**	**0.719**	**0.705**	**0.692**
InfoTS	0.733	0.723	0.705	0.688	0.669	0.672	0.657	0.643
TS2Vec	**0.822**	0.816	0.799	0.783	0.695	0.687	0.672	0.657
TS-TCC	0.685	0.603	0.566	0.533	0.617	0.573	0.533	0.499
T-Loss	0.782	0.750	0.743	0.737	0.581	0.572	0.545	0.520
DTW	0.679	0.672	0.668	0.646	0.654	0.645	0.624	0.605
TimesNet	0.688	0.675	0.649	0.625	0.676	0.664	0.639	0.616
TNC	0.406	0.305	0.291	0.279	0.345	0.311	0.286	0.265

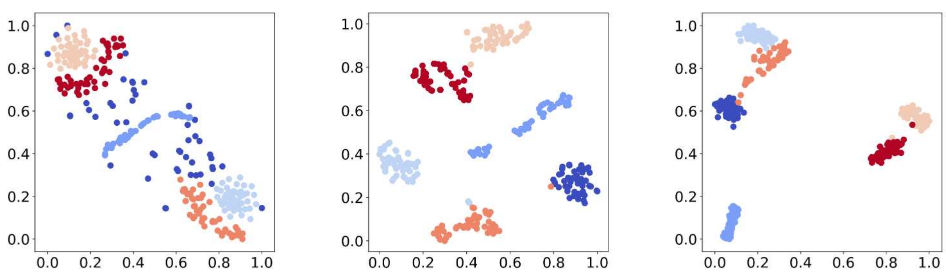
TABLE II: Classification results on 128 UCR datasets and 30 UEA datasets

	Avg. Accuracy ↑	Avg. Precision ↑	Avg. F1 ↑	Avg. Recall ↑
RankSCL	**0.797**	**0.834**	**0.793**	**0.756**
w/o Data Augmentation	0.636	0.663	0.62	0.582
w/ Data Augmentation (Raw Data)	0.519	0.550	0.500	0.458
w/o FCN (Resnet Backbone)	0.318	0.319	0.293	0.271
w/o Rank Loss (CE)	0.636	0.663	0.620	0.582
w/o Rank Loss (CTL)	0.676	0.710	0.663	0.622
w/o Rank Loss (TL)	0.506	0.532	0.509	0.488

TABLE III: Ablation studies on PigCVP dataset

Number	Avg. Accuracy ↑	Avg. Precision ↑	Avg. F1 ↑	Avg. Recall ↑
0	0.636	0.663	0.62	0.582
5	**0.797**	**0.834**	**0.793**	**0.756**
10	0.517	0.544	0.503	0.468
15	0.520	0.560	0.504	0.458

TABLE IV: Data Augmentation Analysis on PigCVP dataset



(a) SyntheticControl (b) TS2Vec (c) RankSCL

Fig. 5: The t-SNE visualization of representations of SyntheticControl dataset (with 6 classes). (Best viewed in color)

the data augmentation, we evaluate different numbers of augmented positive samples. The results demonstrate that without augmented positive samples, accuracy drops by 16.1%, with further performance decline in other metrics. Increasing the number of augmented samples beyond 5 reduces performance, indicating that more augmentations do not always improve results. Augmenting raw time series data (**w/o** Data Augmentation (Raw Data)) is less effective than augmenting in the embedding space, supporting our method. Comparing encoder architectures, FCN outperforms other options (**w/o** FCN (Resnet Backbone)). Additionally, our rank-based arctan loss function proves superior to cross-entropy (CE), contrastive loss (CTL), and triplet loss (TL).

E. Qualitative Evaluation

Using t-SNE, we project the SyntheticControl dataset (6 classes) from raw space to two dimensions (Figure 5(a)) and compare it with the embeddings from TS2Vec and RankSCL. Both TS2Vec and RankSCL show well-separated clusters, but TS2Vec forms 7 clusters, while RankSCL correctly forms 6, matching the number of classes. This indicates that RankSCL preserves clearer class boundaries.

V. Conclusion

In this paper, we propose RankSCL, a supervised contrastive learning framework for time series classification. We introduce a new data augmentation method to generate targeted positive samples, enriching intra-class information. A mining rule reduces computational complexity by capturing valid triplet pairs. Additionally, we propose a rank loss function to optimize representation learning. Evaluation results demonstrate that with effective data augmentation, valid triplet pairs, and enriched intra-class information, RankSCL learns robust representations applicable to future tasks and other data modalities.

ACKNOWLEDGMENT

This project was supported by the National Science Foundation (NSF) Grant No. 2338878 and UConn Research Excellent Program (REP) Award.

REFERENCES

[1] P. Prickett, G. Davies, and G. R., "A scada based power plant monitoring and management system," *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 6883, pp. 433–442, 2011.

[2] J. Parkka, M. Ermes, K. P., M. J., and P. J., "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 6883, pp. 119–128, 2006.

[3] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565–592, 2015.

[4] A. J. Bagnall, A. Bostrom, J. Large, and J. Lines, "The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version," *CoRR*, vol. abs/1602.01711, 2016. [Online]. Available: http://arxiv.org/abs/1602.01711

[5] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 52:1–52:35, 2018.

[6] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[7] K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J. Y. Zhang, Y. Liang, G. Pang, D. Song *et al.*, "Self-supervised learning for time series analysis: Taxonomy, progress, and prospects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[8] Y. Jiang, Z. Pan, X. Zhang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song, "Empowering time series analysis with large language models: A survey," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, K. Larson, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2024, pp. 8095–8103, survey Track. [Online]. Available: https://doi.org/10.24963/ijcai.2024/895

[9] Z. Pan, Y. Jiang, D. Song, S. Garg, K. Rasul, A. Schneider, and Y. Nevmyvaka, "Structural knowledge informed continual multivariate time series forecasting," *arXiv preprint arXiv:2402.12722*, 2024.

[10] Y. Jiang, W. Yu, D. Song, L. Wang, W. Cheng, and H. Chen, "Fedskill: Privacy preserved interpretable skill learning via imitation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1010–1019.

[11] Z. Pan, Y. Jiang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song, "$\mathcal{S}$2 ip-llm: Semantic space informed prompt learning with llm for time series forecasting," in *Forty-first International Conference on Machine Learning*, 2024.

[12] X. Yang, Z. Zhang, and R. Cui, "Timeclr: A self-supervised contrastive learning framework for univariate time series representation," *Know.-Based Syst.*, vol. 245, no. C, jun 2022. [Online]. Available: https://doi.org/10.1016/j.knosys.2022.108606

[13] M. Muller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[15] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8980–8987.

[16] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," *CoRR*, vol. abs/2106.14112, 2021. [Online]. Available: https://arxiv.org/abs/2106.14112

[17] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. C. H. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," *CoRR*, vol. abs/2202.01575, 2022. [Online]. Available: https://arxiv.org/abs/2202.01575

[18] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. 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.

[19] D. Luo, W. Cheng, Y. Wang, D. Xu, J. Ni, W. Yu, X. Zhang, Y. Liu, Y. Chen, H. Chen *et al.*, "Time series contrastive learning with information-aware augmentations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4534–4542.

[20] A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, "Ts-chief: a scalable and accurate forest algorithm for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 3, pp. 742–775, 2020.

[21] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

[22] B. K. Iwana, V. Frinken, and S. Uchida, "Dtw-nn: A novel neural network for time series recognition using dynamic alignment between inputs and weights," *Knowledge-Based Systems*, vol. 188, p. 104971, 2020.

[23] P.-R. Lai and J.-S. Wang, "Multi-stage attention convolutional neural networks for hevc in-loop filtering," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2020, pp. 173–177.

[24] Y. Hao and H. Cao, "A new attention mechanism to classify multivariate time series," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020.

[25] S. Yao, A. Piao, W. Jiang, Y. Zhao, H. Shao, S. Liu, D. Liu, J. Li, T. Wang, S. Hu, L. Su, J. Han, and T. F. Abdelzaher, "Stfnets: Learning sensing signals from the time-frequency perspective with short-time fourier neural networks," *CoRR*, vol. abs/1902.07849, 2019. [Online]. Available: http://arxiv.org/abs/1902.07849

[26] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[27] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[28] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=cu7IUiOhujH

[29] D. Song, N. Xia, W. Cheng, H. Chen, and D. Tao, "Deep r-th root of rank supervised joint binary embedding for multivariate time series retrieval," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2229–2238.

[30] A. J. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. J. Keogh, "The uea multivariate time series classification archive, 2018," *CoRR*, vol. abs/1811.00075, 2018. [Online]. Available: http://arxiv.org/abs/1811.00075

[31] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," *CoRR*, vol. abs/1810.07758, 2018. [Online]. Available: http://arxiv.org/abs/1810.07758

[32] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The eleventh international conference on learning representations*, 2022.

[33] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *Advances in neural information processing systems*, vol. 32, 2019.

[34] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," *CoRR*, vol. abs/2106.00750, 2021. [Online]. Available: https://arxiv.org/abs/2106.00750

[35] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," *CoRR*, vol. abs/2010.02803, 2020. [Online]. Available: https://arxiv.org/abs/2010.02803