TimeCSL: Unsupervised Contrastive Learning of General Shapelets for Explorable Time Series Analysis

Zhiyu Liang
Harbin Institute of Technology
Harbin, China
zyliang@hit.edu.cn

Chen Liang
Harbin Institute of Technology
Harbin, China
23B903050@stu.hit.edu.cn

Zheng Liang
Harbin Institute of Technology
Harbin, China
lz20@hit.edu.cn

Hongzhi Wang*
Harbin Institute of Technology
Harbin, China
wangzh@hit.edu.cn

Bo Zheng
CnosDB Inc.
Beijing, China
harbour.zheng@cnosdb.com

ABSTRACT

Unsupervised (a.k.a. Self-supervised) representation learning (URL) has emerged as a new paradigm for time series analysis, because it has the ability to learn generalizable time series representation beneficial for many downstream tasks without using labels that are usually difficult to obtain. Considering that existing approaches have limitations in the design of the representation encoder and the learning objective, we have proposed Contrastive Shapelet Learning (CSL), the first URL method that learns the general-purpose shapelet-based representation through unsupervised contrastive learning, and shown its superior performance in several analysis tasks, such as time series classification, clustering, and anomaly detection. In this paper, we develop TimeCSL, an end-to-end system that makes full use of the general and interpretable shapelets learned by CSL to achieve explorable time series analysis in a unified pipeline. We introduce the system components and demonstrate how users interact with TimeCSL to solve different analysis tasks in the unified pipeline, and gain insight into their time series by exploring the learned shapelets and representation.

PVLDB Reference Format:
Zhiyu Liang, Chen Liang, Zheng Liang, Hongzhi Wang, and Bo Zheng. TimeCSL: Unsupervised Contrastive Learning of General Shapelets for Explorable Time Series Analysis. PVLDB, 17(12): 4489-4492, 2024.
doi:10.14778/3685800.3685907

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/real2fish/CSL.

1 INTRODUCTION

A time series is one (a.k.a. univariate) or a group (multivariate) of variables observed over time. Time series analysis by discovering dependencies of the time-evolving variables can offer important insights into the underlying phenomena, which is useful for real-world applications in various scenarios, such as manufacturing [6], medicine [11] and finance [2].

A major challenge in modeling time series is the lack of labels, because the underlying states required for labeling these time-dependent data are difficult to understand, even for domain specialists [15]. For this reason, recent studies focus on unsupervised (a.k.a. self-supervised) representation learning (URL) of time series [3, 4, 15, 20, 21], which aims to train a neural network (called encoder) without accessing the ground-truth labels to embed the data into feature vectors. The learned features (a.k.a. representation) can then be used for training models to solve a downstream analysis task, using ***little annotated data*** compared to traditional supervised methods [21]. Not only that, the features are more ***general-purpose*** since they can benefit several tasks.

Unfortunately, existing URL approaches for time series have two limitations. First, these methods focus on the representation encoders based on the convolutional neural network (CNN) [5, 16] and the Transformer [18]. However, these architectures are originally designed for domains such as computer vision and natural language processing, and have been shown to face many difficulties in modeling time series, due to the lack of capability to deal with the characteristics specific to time series [14, 23]. Second, some existing approaches are based on domain-specific assumptions. For example, Franceschi et al. [4] and Tonekaboni et al. [15] assume that the subseries distant in time are dissimilar, but it is easily violated in periodic time series [13]. As a result, these methods cannot be well generalized to different scenarios.

To address the above issues, we have proposed ***Contrastive Shapelet Learning (CSL)*** [10], a brand new unsupervised representation learning framework for multivariate (and also univariate) time series. To the best of our knowledge, ***CSL is the first general-purpose URL method based on shapelet***, an interpretable pattern specifically designed for time series that represents *the discriminative subsequence*. Unlike traditional approaches that learn the shapelets for a specific analysis task, such as time series classification [8, 9, 19] or clustering [22], the shapelets of the proposed CSL are learned with unsupervised contrastive learning, which has shown superiority in many downstream analysis tasks [10], including classification, clustering, segment-level anomaly detection, and long time series representation. We summarize the performance of CSL against the competing methods in Figure 1, which is extensively

*Corresponding author.

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685907

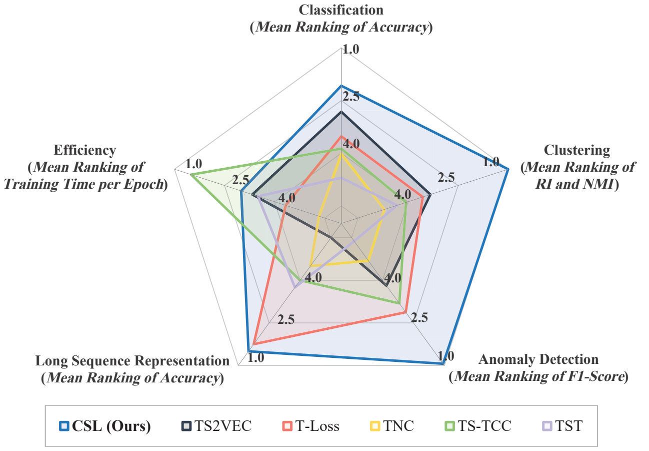


Figure 1: Overall performance of CSL against the competitors (smaller is better) regarding classificaton, clustering, anomaly detection, long sequence representation and training efficiency. See §5.2, §5.7 and §5.8 in [10] for the details.

evaluated using 38 datasets from various real-world scenarios [10]. The results can also be reproduced using the UniTS system [7].

In this paper, ***we demonstrate* `TimeCSL`*, a novel system that makes full use of CSL to achieve explorable time series analysis for various tasks in a unified process.*** `TimeCSL` includes an end-to-end ***unified pipeline*** that first learns the general shapelets of multiple scales and (dis)similarity metrics without any annotation by running the CSL algorithm [10]. Then, it addresses different time series analysis tasks by building arbitrary task-oriented analyzers (e.g., SVM for classification and K-Means for clustering) on top of the general-purpose shapelet-based features. The pipeline has shown superior performance compared to that of the complex task-specific approaches, and significantly outperforms the traditional supervised methods when there are few available labels. We refer the interested readers to our research paper [10] for more details.

`TimeCSL` provides flexible and intuitive ***visual exploration*** of the raw time series, the learned shapelets, and the shapelet-based time series representation, offering a useful tool for interpreting the analysis results. Users can experiment with the system using their own data to explore the learned shapelet-based features, which are usually more insightful and intuitive-to-understand than the complex raw time series. This "explorable" analysis can help to explain the decisions made by the task-oriented analyzer.

2 THE TIMECSL SYSTEM

As depicted in Figure 2, `TimeCSL` is comprised of two components, including *Unsupervised Contrastive Shapelet Learning* and *Explorable Time Series Analysis*. These components work as follows.

2.1 Unsupervised Contrastive Shapelet Learning

The goal of this component is to learn general shapelets from the training time series, which transforms the raw time series into shapelet-based representation to facilitate different downstream analysis tasks. This is achieved using our proposed CSL method [10].

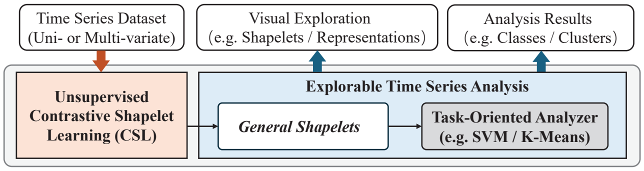

Time Series Dataset
(Uni- or Multi-variate)
Visual Exploration
(e.g. Shapelets / Representations)
Analysis Results
(e.g. Classes / Clusters)
Unsupervised Contrastive Shapelet Learning (CSL)
Explorable Time Series Analysis
General Shapelets
Task-Oriented Analyzer
(e.g. SVM / K-Means)

Figure 2: The `TimeCSL` pipeline.

Given a dataset containing N time series as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \in \mathbb{R}^{N\times D\times T}$, where each time series $\boldsymbol{x}_i \in \mathbb{R}^{D\times T}$ has D variables ($D \geq 1$) and T observations ordered by time, CSL embeds $\boldsymbol{x}_i$ into the shapelet-based representation $\boldsymbol{z}_i \in \mathbb{R}^{D_{repr}}$ using the proposed *Shapelet Transformer* f, i.e. $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$, where f contains the learnable shapelets of various (dis)similarity measures and lengths (a.k.a. scales). CSL learns f using an unsupervised contrastive learning algorithm, which iteratively optimizes the proposed *Multi-Grained Contrasting* and *Multi-Scale Alignment* objectives in an end-to-end manner with stochastic gradient descent.

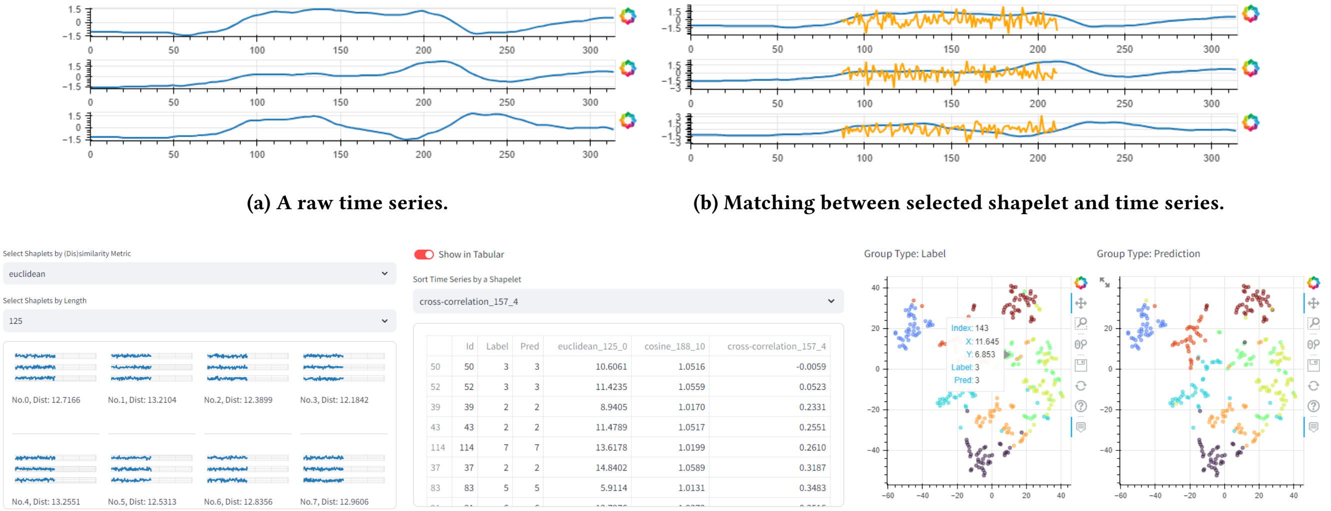
Using the Shapelet Transformer f (i.e. all the shapelets) learned by CSL, the `TimeCSL` system transforms all input time series into the shapelet-based features as $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$, and performs the downstream analysis tasks on top of the representation $\boldsymbol{z}_i$. It is noteworthy that $\boldsymbol{z}_i$ represents the (dis)similarity (e.g., the minimum Euclidean norm or the maximum cosine similarity) between the subsequences of $\boldsymbol{x}_i$ and each of the shapelets, and therefore is fully interpretable and explainable.

2.2 Explorable Time Series Analysis

By making full use of the general-purpose and explainable shapelet-based features learned by CSL, this component not only offers a unified and flexible way to perform different time series analysis tasks (e.g. classification, clustering, and anomaly detection), but also the intuitive visual exploration of the raw time series, the learned shapelets, and the shapelet-based representation, so that the users can gain useful insights into their data to understand the decision basis of the analysis results (e.g. the predicted classes or clusters).

Task solving. As mentioned above, `TimeCSL` solves all different time series analysis tasks using the shapelet-based representation learned by CSL. This is achieved by building a task-oriented analyzer (e.g., SVM for classification or K-Means for clustering) that takes the shapelet-based feature vector $\boldsymbol{z}_i$ as input and outputs the corresponding analysis results (e.g., classes or clusters). `TimeCSL` provides two modes to build the analyzer, including the *freezing mode* and the *fine-tuning mode*, which differ in whether to fine-tune the parameters of the Shapelet Transformer f that is pre-trained by the Unsupervised Contrastive Shapelet Learning component.

- *Freezing mode*. In this basic mode, the task-oriented analyzer is built by directly using the pre-trained Shapelet Transformer f to extract the general-purpose shapelet-based features, while the parameters of f are kept unchanged during the building. Therefore, any standard analyzer (e.g. the many popular classifiers such as SVM, logistic regression, GBDT, etc) can be seamlessly integrated to facilitate different application scenarios of the users.
- *Fine-tuning mode*. This is an advanced mode that allows users to fine-tune the parameters of the pre-trained Shapelet Transformer



(a) A raw time series.

(b) Matching between selected shapelet and time series.

Select Shaplets by (Dis)similarity Metric: euclidean

Select Shaplets by Length: 125

Show in Tabular

Sort Time Series by a Shapelet: cross-correlation_157_4

	Id	Label	Pred	euclidean_125_0	cosine_188_10	cross-correlation_157_4
50	50	3	3	10.6061	1.0516	-0.0059
52	52	3	3	11.4235	1.0559	0.0523
39	39	2	2	8.9405	1.0170	0.2331
43	43	2	2	11.4789	1.0517	0.2551
114	114	7	7	13.6178	1.0199	0.2610
37	37	2	2	14.8402	1.0589	0.3187
83	83	5	5	5.9114	1.0131	0.3483

(c) Learned shapelets grouped by length and (dis)similarity metric.

(d) Shapelet-based representation of a dataset shown in a tabular form.

(e) Shapelet-based representation of a dataset visualized using two-dimensional t-SNE [17].

Figure 3: The `TimeCSL` interface.

f for specific tasks to further improve the accuracy of the analysis. In this mode, `TimeCSL` uses a task-specific neural network g (e.g., a linear classifier) as the analyzer, which is appended to the pre-trained f to map the shapelet-based features to the analysis results, i.e., $\hat{y}_i = g(z_i) = g(f(x_i))$ where $\hat{y}_i$ is a prediction (e.g. a predicted class) for the time series $\boldsymbol{x}_i$. It simultaneously adjusts the parameters of f and g by minimizing the task-oriented loss function (e.g. the cross-entropy loss widely used for classification) using the backpropagation algorithm [12]. Afterwards, the fine-tuned f and g are used for extracting the shapelet-based features and predicting the analysis results, respectively.

The fine-tuning mode of `TimeCSL` is ***especially effective in the semi-supervised scenarios***, where only a small portion of the data is annotated. Users can pre-train the Shapelet Transformer f through Unsupervised Contrastive Shapelet Learning to make full use of all time series. Subsequently, they only need to fine-tune f and g using the available labeled data to achieve competitive performance. For example, our study on a real-world gesture recognition problem (see §5.5 in [10]) shows that, when the proportion of labeled data is less than 20%, the fine-tuned method (using the Shapelet Transformer f and a linear layer g) that takes advantage of the unsupervised pre-training of CSL, is 7% to 10% more accurate than the state-of-the-art customized time series classification approach.

Visual exploration. `TimeCSL` provides the visual exploration of the following two types of data for gaining insight into the time series and understanding the analysis results.

- *Time series and shapelets.* Since the shapelets represent the salient approximate subsequences of the raw time series which are of the same number of variates, `TimeCSL` shows both the raw time series and the learned shapelets in the line chart, where each line indicates the values of one variate.
- *Shapelet-based features.* By definition, a shapelet-based feature represents the (dis)similarity (e.g., Euclidean norm) between a shapelet and the most similar subsequence of a time series. Therefore, given a time series and a shapelet, `TimeCSL` visualizes them, matches the shapelet with its most similar subsequence, and displays the feature value measured by the corresponding (dis)similarity metric. In addition to this instance-level exploration, `TimeCSL` can also show the shapelet-based representation of a dataset in a tabular form, allowing flexible sorting of the time series according to each of the shapelets (equivalent to the features), and visualize the high-dimensional time series representation in a two-dimensional graph using t-SNE [17].

The user can interact with `TimeCSL` to iteratively use the features extracted with different subsets of the shapelets in the explorable time series analysis. Given the interpretability of the shapelet-based representation, it is feasible to gain an understanding of the factors that influence the analysis results.

3 DEMONSTRATION

The demonstration[1] illustrates two key capabilities of the `TimeCSL` system. Firstly, it showcases how the user performs different time series analysis tasks using the unified pipeline. Secondly, it demonstrates how users gain insight into their time series by exploring the interpretable shapelet-based representation, which helps to understand the underlying decision basis of the analysis results.

We prepare the UEA archive [1] for the audience to interact with our system, which contains 30 time series datasets from various applications. Users can also analyze and explore their own data.

Generally, a user takes the following four steps to achieve explorable time series analysis using the `TimeCSL` system.

[1] https://youtu.be/hjdKy4Zr2Zw

Step 1: Configuring the Shapelet Transformer. A user can flexibly configure the Shapelet Transformer to decide the number, length, and (dis)similarity metric of the shapelets to be learned, or directly start with our recommended configuration that uses Euclidean norm, cosine similarity, and cross-correlation function to measure the (dis)similarity, and adaptively sets the number and lengths of the shapelets based on the time series (see §4.2 in [10]).

Step 2: Unsupervised Contrastive Learning of the General Shapelets. After configuration, a user clicks on the "Run CSL" button to begin the unsupervised contrastive learning. Then, `TimeCSL` automatically learns the general and interpretable shapelets by running the CSL algorithm. A user can diagnose the model performance through the learning curve plotted in our GUI that records the training or validation loss over time.

Step 3: Performing the Analysis Tasks. `TimeCSL` transforms the time series of different lengths and numbers of variates into a unified vector representation using the learned shapelets. Therefore, a user can easily perform an analysis task by specifying a mode and a task-oriented analyzer. `TimeCSL` has integrated several commonly used analyzers. In the freezing mode, to name only a few, it integrates SVM, K-Means, and Isolation Forest using scikit-learn[2], for classification, clustering, and anomaly detection, respectively. It also implements a linear classifier using PyTorch[3] to deal with the (semi-supervised) classification problem in the fine-tuning mode. Using the GUI, a user can select one of the tasks, modes, and analyzers, and then click "Run Analyzer" to get the analysis results.

Step 4: Exploring the Data. Figure 3 shows part of the GUI for analyzing the time series of the prepared UWaveGestureLibrary dataset with the default configuration of the system. A user can explore the time series (Figure 3a) and the learned shapelets (Figure 3c). To see the feature of a time series extracted using a specific shapelet, a user only needs to select them and click on the "Match" button. `TimeCSL` will visually match the shapelet with the most similar subsequence of the time series to illustrate this shapelet-based feature (Figure 3b). `TimeCSL` also provides two types of exploration of the high-dimensional shapelet-based representation of the dataset. A user can select a set of interested shapelets, and click "Show in Tabular" to see the features extracted using these shapelets in a table and sort the time series according to each of the shapelets or features (Figure 3d). The user can also click "Show in t-SNE" to investigate these features in a two-dimensional t-SNE representation, where the points are clickable so that the user can see details about the corresponding time series (Figure 3e).

A user can redo the analysis task (i.e., Step 3) using the selected shapelets of interest to explore their influence on the analysis results. For example, if a user runs an SVM analyzer on UWaveGestureLibrary by selecting the shapelets of length 31 learned as above, s/he can get a classification accuracy of 0.75. If the user restarts Step 3 using the learned shapelets of larger lengths, the resulting accuracy scores are, to name a few, 0.85 for length 97 and 0.89 for length 188, which is even comparable to the accuracy of 0.91 that is achieved by using all the learned shapelets. The user can infer from the results that, in this scenario, the categories of the time series can be better distinguished by focusing on longer salient subsequences.

[2] https://scikit-learn.org/stable/
[3] https://pytorch.org/

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their time and the valuable comments. This paper was supported by the NSFC grant (62232005, 62202126), the National Key Research and Development Program of China (2021YFB3300502), and the Postdoctoral Fellowship Program of CPSF (GZC20233457).

REFERENCES

[1] Anthony J. Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn J. Keogh. 2018. The UEA multivariate time series classification archive, 2018. *CoRR* abs/1811.00075 (2018).

[2] Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. 2022. Detection and clustering of lead-lag networks for multivariate time series with an application to financial markets. (2022).

[3] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. In *IJCAI*. 2352–2359.

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

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[6] Eunji Kim, Sungzoon Cho, Byeongeon Lee, and Myoungsu Cho. 2019. Fault detection and diagnosis using self-attentive convolutional neural networks for variable-length sensor data in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing* 32, 3 (2019), 302–309.

[7] Zhiyu Liang, Chen Liang, Zheng Liang, Hongzhi Wang, and Bo Zheng. 2024. UniTS: A Universal Time Series Analysis Framework Powered by Self-Supervised Representation Learning. In *SIGMOD*. 480–483.

[8] Zhiyu Liang and Hongzhi Wang. 2021. Efficient class-specific shapelets learning for interpretable time series classification. *Inf. Sci.* 570 (2021), 428–450.

[9] Zhiyu Liang and Hongzhi Wang. 2024. FedST: secure federated shapelet transformation for time series classification. *The VLDB Journal* (July 2024).

[10] Zhiyu Liang, Jianfeng Zhang, Chen Liang, Hongzhi Wang, Zheng Liang, and Lujia Pan. 2023. A Shapelet-based Framework for Unsupervised Multivariate Time Series Representation Learning. *PVLDB* 17, 3 (2023), 386–399.

[11] Ann Riley and Elvira Nica. 2021. Internet of things-based smart healthcare systems and wireless biomedical sensing devices in monitoring, detection, and prevention of COVID-19. *American Journal of Medical Research* 8, 2 (2021), 51–64.

[12] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533–536.

[13] Saeid Sanei and Jonathon A Chambers. 2013. *EEG signal processing.* John Wiley & Sons.

[14] Wensi Tang, Guodong Long, Lu Liu, Tianyi Zhou, Michael Blumenstein, and Jing Jiang. 2022. Omni-Scale CNNs: a simple and effective kernel size configuration for time series classification. In *ICLR*. OpenReview.net.

[15] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. 2021. Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In *ICLR*. OpenReview.net.

[16] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. *CoRR* abs/1609.03499 (2016).

[17] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[19] Akihiro Yamaguchi, Ken Ueo, and Hisashi Kashima. 2022. Learning Evolvable Time-series Shapelets. In *ICDE*. IEEE, 793–805.

[20] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *AAAI*, Vol. 36. 8980–8987.

[21] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *SIGKDD*. 2114–2124.

[22] Qin Zhang, Jia Wu, Peng Zhang, Guodong Long, and Chengqi Zhang. 2019. Salient Subsequence Learning for Time Series Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 9 (2019), 2193–2207.

[23] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *ICML*, Vol. 162. 27268–27286.