A Survey on Time-Series Pre-Trained Models

Qianli Ma, *Member, IEEE*, Zhen Liu, Zhenjing Zheng, Ziyang Huang, Siying Zhu, Zhongzhong Yu, and James T. Kwok, *Fellow, IEEE*

(Survey Paper)

***Abstract*—Time-Series Mining (TSM) is an important research area since it shows great potential in practical applications. Deep learning models that rely on massive labeled data have been utilized for TSM successfully. However, constructing a large-scale well-labeled dataset is difficult due to data annotation costs. Recently, pre-trained models have gradually attracted attention in the time series domain due to their remarkable performance in computer vision and natural language processing. In this survey, we provide a comprehensive review of Time-Series Pre-Trained Models (TS-PTMs), aiming to guide the understanding, applying, and studying TS-PTMs. Specifically, we first briefly introduce the typical deep learning models employed in TSM. Then, we give an overview of TS-PTMs according to the pre-training techniques. The main categories we explore include supervised, unsupervised, and self-supervised TS-PTMs. Further, extensive experiments involving 27 methods, 434 datasets, and 679 transfer learning scenarios are conducted to analyze the advantages and disadvantages of transfer learning strategies, Transformer-based models, and representative TS-PTMs. Finally, we point out some potential directions of TS-PTMs for future work.**

***Index Terms*—Time-series mining, pre-trained models, deep learning, transfer learning, transformer.**

I. Introduction

As AN important research direction in the field of data mining, Time-Series Mining (TSM) has been widely utilized in real-world applications, such as finance [1], speech analysis [2], action recognition [3], and traffic flow forecasting [4], [5]. The fundamental problem of TSM is how to represent the time-series data [6]. Then, various mining tasks can be performed based on the given representations. Traditional time-series representations (e.g., shapelets [7]) are time-consuming due to heavy reliance on domain or expert knowledge. Therefore, it remains challenging to learn the appropriate time series representations automatically.

In recent years, deep learning models [8], [9], [10], [11] have achieved great success in a variety of TSM tasks. Unlike traditional machine learning methods, deep learning models do not require time-consuming feature engineering. Instead, they automatically learn time-series representations through a data-driven approach. However, the success of deep learning models relies on the availability of massive labeled data. In many real-world situations, it can be difficult to construct a large well-labeled dataset due to data acquisition and annotation costs.

To alleviate the reliance of deep learning models on large datasets, approaches based on data augmentation [12], [13] and semi-supervised learning [14], [15] have been commonly used. Data augmentation can enhance the size and quality of the training data, and has been used as an important component in many computer vision tasks [16]. However, different from image data augmentation, time-series data augmentation also needs to consider properties such as temporal dependencies and multi-scale dependencies in the time series. Moreover, design of the time-series data augmentation techniques generally relies on expert knowledge. On the other hand, semi-supervised methods employ a large amount of unlabeled data to improve model performance. However, in many cases, even unlabeled time-series samples can be difficult to collect (e.g., electrocardiogram time series data in healthcare [17]).

Another effective solution to alleviate the problem of insufficient training data is transfer learning [18], [19], which relaxes the assumption that the training and test data must be independently and identically distributed. Transfer learning usually has two stages: pre-training and fine-tuning. During pre-training, the model is pre-trained on some source domains that contain a large amount of data, and are separate but relevant to the target domain. On fine-tuning, the pre-trained model (PTM) is fine-tuned on the often limited data from the target domain.

Recently, PTMs, particularly Transformer-based PTMs, have achieved remarkable performance in various Computer Vision (CV) [20], [21] and Natural Language Processing (NLP) [22] applications. Inspired by these, recent studies consider the design of Time-Series Pre-Trained Models (TS-PTMs) for time-series

Received 29 April 2023; revised 18 August 2024; accepted 2 October 2024. Date of publication 7 October 2024; date of current version 13 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62272173, in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515010089 and under Grant 2022A1515010179, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2023A0505050106, and in part by the Fundamental Research Funds for the Central Universities under Grant 2024ZYGXZR104. Recommended for acceptance by Y. Shen. *(Qianli Ma and Zhen Liu are co-first authors.) (Corresponding author: Qianli Ma.)*

Qianli Ma, Zhen Liu, Zhenjing Zheng, Ziyang Huang, Siying Zhu, and Zhongzhong Yu are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: qianlima@scut.edu.cn; cszhenliu@mail.scut.edu.cn; 982360227@qq.com; stevenhuang12@outlook.com; 489531037@qq.com; yuzzhong2020@foxmail.com).

James T. Kwok is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR 999077, China (e-mail: jamesk@cse.ust.hk).

The source code is available at https://github.com/qianlima-lab/time-series-ptms

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2024.3475809, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2024.3475809

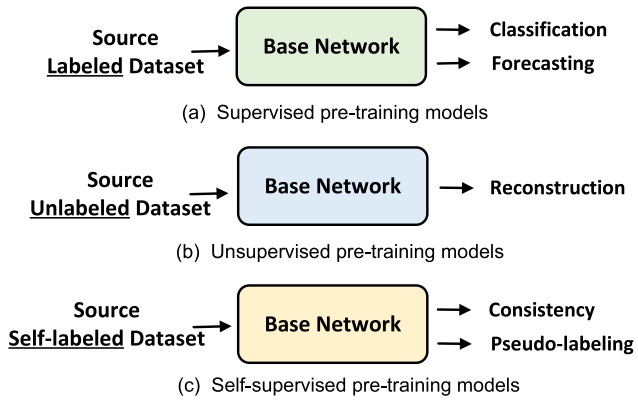

Fig. 1. Pre-training techniques for time series.

data. First, a time-series model is pre-trained by *supervised learning* [23], [24], *unsupervised learning* [25], [26], or *self-supervised learning* [27], [28], [29] so as to obtain appropriate representations. The TS-PTM is then fine-tuned on the target domain for improved performance on the downstream TSM tasks (e.g., time-series classification and anomaly detection).

Supervised TS-PTMs [23], [30] are typically pre-trained by classification or forecasting tasks. However, the difficulty to obtain massive labeled time-series datasets for pre-training often limits the performance of supervised TS-PTMs. In addition, unsupervised TS-PTMs utilize unlabeled data for pre-training, which further tackle the limitation of insufficient labeled data. For example, the reconstruction-based TS-PTMs [25] employ auto-encoders and a reconstruction loss to pre-train time-series models. Recently, self-supervised PTMs based on contrastive learning [31], [32] have shown great potential in CV. Therefore, some scholars [26], [33] have started exploring the design of consistency-based tasks and pseudo-labeling techniques for mining the inherent properties of time series. Nonetheless, the study of TS-PTMs remains a challenge.

In this survey, we provide a comprehensive review of TS-PTMs. Specifically, we first introduce various TSM tasks and deep learning models used in TSM. We then propose a taxonomy of TS-PTMs based on the pre-training techniques (Fig. 1). These include supervised pre-training techniques (leading to classification-based and forecasting-based PTMs), unsupervised pre-training techniques (reconstruction-based PTMs) and self-supervised pre-training techniques (consistency-based and pseudo-labeling-based PTMs). Note that some TS-PTMs may use multiple tasks (e.g., forecasting and reconstruction in [34]) for pre-training. To simplify the review, we classify the TS-PTM based on its core pre-training task.

While existing surveys have analyzed time series pre-training [35], [36], [37], [38], [39], [40], our work was made available online before the majority of these surveys. Our work specifically focuses on pre-training, whereas the majority of the aforementioned surveys primarily address representation learning. In addition, the key difference is that our work conducts extensive experiments to analyze the performance of various TS-PTMs in a uniform experimental setting (e.g., consistent dataset preprocessing and Python version). Specifically, our experiments involve 27 methods, 434 datasets, and 679 sets of transfer learning. The related experimental code and datasets have been open-sourced.[1] For a detailed comparison with existing surveys, please refer to Appendix A, available online.

The main contributions of our survey can be summarized as follows:

- We provide a taxonomy and a systematic review of existing TS-PTMs, ranging from early transfer learning methods to recent Transformer-based and consistency-based TS-PTMs. Specifically, we categorize TS-PTMs according to supervised, unsupervised, and self-supervised pre-training techniques, providing a detailed summary of each to guide future research.
- We perform extensive experiments to analyze the pros and cons of TS-PTMs. For time series classification, we find that transfer learning-based TS-PTMs perform poorly on the UCR time series datasets (containing many small datasets), but achieve excellent performance on other publicly-available large time series datasets. For time series forecasting and anomaly detection, we find that patch-based pre-training technique should be the focus of future research on TS-PTMs.
- We present potential future directions with a detailed and thorough discussion. In particular, we analyze the limitations of current TS-PTMs and suggest future directions under (i) datasets, (ii) deep learning models, (iii) inherent properties, (iv) adversarial attacks, (v) noisy labels, and (vi) pre-trained large language models.

The remainder of this paper is organized as follows. Section II provides background on the TS-PTM. A comprehensive review of the TS-PTMs is then given in Section III. Section IV presents experiments on the various TS-PTMs. Section V suggests some future directions. Finally, we summarize our findings in Section VI.

II. Background

In this section, we first describe the TSM tasks in Section II-A. Section II-B then introduces various deep learning models used in TSM. Finally, Section II-C discusses why we need to employ the PTMs.

A. Time-Series Mining Tasks

A time series can be represented as a T-dimensional vector $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T]$, where T is the length of the time series, $\boldsymbol{x}_t \in \mathbb{R}^M$ is the value at tth time step, and M is the number of variables.

1) Time-Series Classification: In time-series classification [41], a labeled time series dataset is used to train a classifier, which can then be used to classify unseen samples. A labeled time-series dataset with N samples can be denoted as $\mathcal{D} = \{(\boldsymbol{X}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{X}_t, \boldsymbol{y}_t), \ldots, (\boldsymbol{X}_N, \boldsymbol{y}_N)\}$, where $\boldsymbol{X}_t$ can be either a univariate or multivariate time series, and $\boldsymbol{y}_t$ is the corresponding one-hot label vector.

2) Time-Series Forecasting: Time-Series Forecasting (TSF) [42] aims to analyze the dynamics and correlations among historical temporal data to predict future behavior. TSF models usually

[1] https://github.com/qianlima-lab/time-series-ptms

need to consider the trend and seasonal variations in the time series, and also correlations between historical observed values. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T]$ be the historical observations, and H be the desired forecasting horizon, the problem is to predict the future values $[\boldsymbol{x}_{T+1}, \boldsymbol{x}_{T+2}, \ldots, \boldsymbol{x}_{T+H}]$.

3) Time-Series Clustering: Time-series clustering [43] aims to partition the time-series dataset $\mathcal{D}$ into a partition K clusters $\{c_1, \ldots, c_K\}$, such that both the similarities among samples from the same cluster and the dissimilarities between samples of different clusters are maximized. While clustering has been successfully used on static data, the clustering of time-series data is more difficult because of the presence of temporal and multi-scale dependencies. Time-series clustering helps to discover interesting patterns and enables the extraction of valuable information from massive time series datasets.

4) Time-Series Anomaly Detection: Time-series anomaly detection [44] aims to identify observations that significantly deviate from the other observations in the time series. It has to learn informative representations from the time series $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T]$, and then derive an anomaly score to determine whether a point $\boldsymbol{x}_t$ or a subsequence $S = [x_p, \ldots, x_{p+n-1}]$ (where $n \leq |T|$) is anomalous [45].

5) Time Series Imputation: Time-Series Imputation (TSI) [46] aims to replace missing values in a time series with realistic values so as to facilitate TSM tasks. Given a time series $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T]$ and a binary $\boldsymbol{M} = [m_1, \ldots, m_t, \ldots, m_T]$, $\boldsymbol{x}_t$ is missing if $m_t = 0$, and is observed otherwise. TSI imputes the missing values as

$$\boldsymbol{X}_{imputed} = \boldsymbol{X} \odot \boldsymbol{M} + \hat{\boldsymbol{X}} \odot (1 - \boldsymbol{M}),$$

where $\hat{\boldsymbol{X}}$ is the predicted values generated by the TSI technique. As a conditional generation model, TSI techniques have been studied and applied to areas such as gene expression [47] and healthcare [48].

6) Time Series Extrinsic Regression: Time-Series Extrinsic Regression (TSER) is a task designed to learn the relationship between time series data and a continuous scalar variable [49]. The TSER model, represented as a function $\mathcal{T} \rightarrow \mathcal{R}$, is trained on a dataset $\mathcal{D}$ comprising pairs of time series $\boldsymbol{X}_t$ and corresponding scalar values $\boldsymbol{r}_t$. Unlike time series classification, which predicts categorical labels, TSER produces numerical outputs [50]. TSER contrasts with traditional TSF tasks by focusing on the association between time series and an external variable sequence [51]. For instance, in smart city applications, TSER can integrate various sensor readings (e.g., temperature, humidity, rain, voltage) to predict a continuous value such as power consumption [52].

B. Deep Learning Models for Time Series

1) Recurrent Neural Networks: Recurrent Neural Networks (RNNs) [53] usually consist of an input layer, one or more recurrent hidden layers, and an output layer (Fig. 2(a)). In the past decade, RNNs and their variants (such as the long short-term memory network [54] and gated recurrent units [53]) have achieved remarkable success in TSM. For example, Muralidhar et al. [55] combined dynamic attention and a RNN-based sequence-to-sequence model for time-series forecasting. Ma et al. [56] employed a multi-layer dilated RNN to extract multi-scale temporal dependencies for time-series clustering.

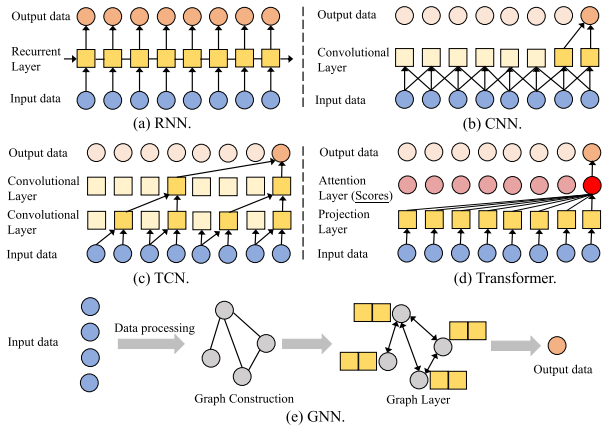

Fig. 2. Deep learning models used for time-series mining.

2) Convolutional Neural Networks: Convolutional Neural Networks (CNNs) [57] are originally designed for computer vision tasks. A typical CNN is shown in Fig. 2(b). To use CNNs for TSM, the data need to be first encoded in an image-like format. The CNN receives embedding of the value at each time step and then aggregates local information from nearby time steps using convolution. CNNs have been shown to be very effective for TSM [58]. For example, the multi-scale convolutional neural network [59] can automatically extract features at different scales by a multi-branch layer and convolutional layers. Kashiparekh et al. [60] incorporated filters of multiple lengths in all convolutional layers to capture multi-scale temporal features for time-series classification.

Unlike vanilla CNNs, Temporal Convolutional Networks (TCNs) [61] use a fully convolutional network [62] so that all the layers are of the same length, and employ causal convolutions with no information "leakage" from future to past. A typical TCNs is shown in Fig. 2(c). Compared to recurrent networks, TCNs have recently shown to be more accurate, simpler, and more efficient across a diverse range of sequence modeling tasks [63]. For example, Sen et al. [10] combined a local temporal network and a global matrix factorization model regularized by a TCN for time-series forecasting.

3) Transformers: Transformers [64], [65] integrate information from data points in the time series by dynamically computing the associations between representations with self-attention. A typical Transformer is shown in Fig. 2(d). Transformers have shown great power in TSM due to their powerful capacity to model long-range dependencies. For example, Zhou et al. [11] combined a self-attention mechanism (with $\mathcal{O}(L \log L)$ time and space complexities) and a generative decoder for long time-series forecasting.

4) Graph Neural Networks: Graph Neural Networks (GNNs) are highly effective at processing graph data consisting of nodes and edges [66]. Recently, some scholars have converted

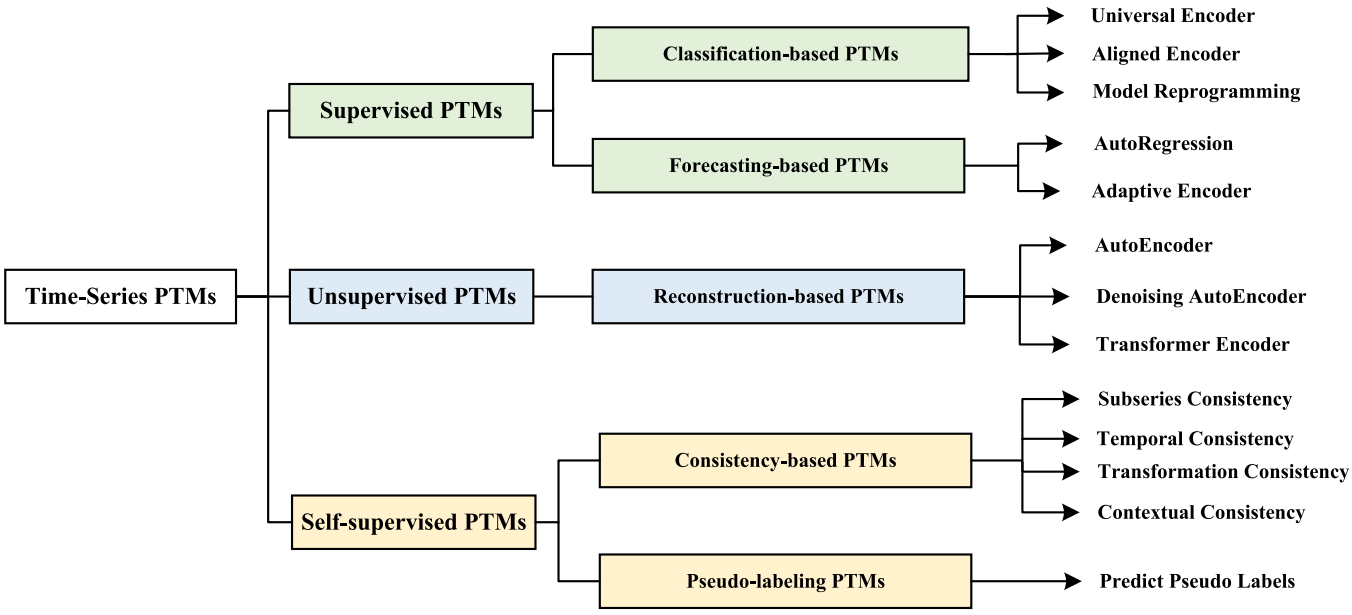

Fig. 3. The taxonomy of Pre-Trained Models for time-series mining.

time series data into graphs by examining both intra-sample and inter-sample relationships and applying GNNs to TSM (Fig. 2(e)). Intra-sample relationship approaches construct graphs by analyzing relationships between subsequences within a sequence [67] or between different variables [68]. For inter-sample relationships, graphs are constructed based on sample similarities or category information [69].

C. Why Pre-Trained Models?

With the rapid development of deep learning, deep learning-based TSM has received more and more attention. In recent years, deep learning models have been widely used in TSM and have achieved great success. However, as data acquisition and annotation can be expensive, the limited labeled time-series data often hinders sufficient training of deep learning models. For example, in bioinformatics, time series classification involves assigning each sample to a specific category, such as clinical diagnosis results. Nonetheless, obtaining accurate labels for all samples is a challenge as it requires expert knowledge to classify the samples. Therefore, pre-training strategies have been proposed to alleviate this data sparseness problem. The advantages of Pre-Trained Models (PTMs) for TSM can be summarized as follows:

- PTMs provide better model initialization for the downstream TSM tasks, which generally results in better generalization performance.
- PTMs can automatically obtain appropriate time series representations by pre-training on source datasets, thus avoiding over-reliance on expert knowledge.

III. Overview of TS-PTMs

In this section, we propose a new taxonomy of TS-PTMs, which systematically classifies existing TS-PTMs based on pre-training techniques. The taxonomy of TS-PTMs is shown in Fig. 3, and please refer to Appendix B-A for a literature summary of TS-PTMs, available online.

A. Supervised PTMs

The early TS-PTMs are inspired by transfer learning applications in CV. Many vision-based PTMs are trained on large labeled datasets such as the ImageNet [70]. The corresponding weights are then fine-tuned on the target dataset, which is usually small. This strategy has been shown to improve the generalization performance of deep learning models on many CV tasks. Naturally, some also investigated whether this strategy is effective in the time-series domain [23], [71]. Their experiments on the UCR time series datasets [72] show that transfer learning may improve or degrade downstream task performance, depending on whether the source and target datasets are similar [23].

1) Classification-Based PTMs: Time-series classification is the most common supervised learning task in TSM. The loss function is usually the cross-entropy, which is defined as

$$\mathcal{L}_{classification} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}).$$

Here, $\boldsymbol{y}_i = [y_{ij}]$ and $\boldsymbol{p}_i = [p_{ij}]$ are the ground-truth label vector and predicted label vector of the ith input time series, respectively, N is the number of samples and C is the number of categories. Recently, the CNN has also been used for time-series classification [73], [74], [75], and achieved better performance than traditional machine learning methods (such as the nearest neighbor classifier with dynamic time warping distance). However, deep learning models are prone to over-fitting on small-scale time-series datasets.

To prevent the over-fitting problem when training a new model from scratch on the target dataset, there have been attempts to employ classification-based PTMs that pre-train a classification base model on some labeled source datasets [23], [24], [76].

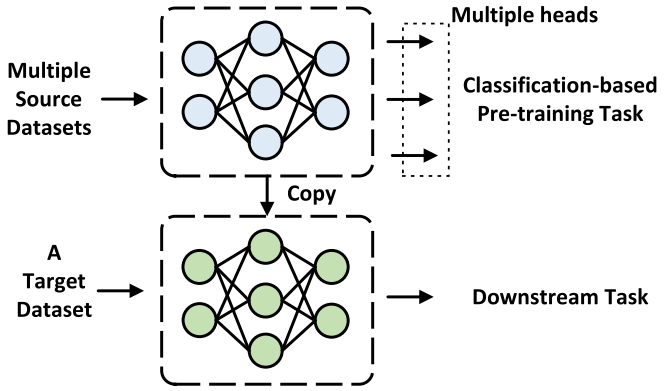

Fig. 4. Universal encoder aims to learn general time-series representations through pre-training on various source datasets. The universal encoder is then fine-tuned on the target dataset for downstream TSM tasks.

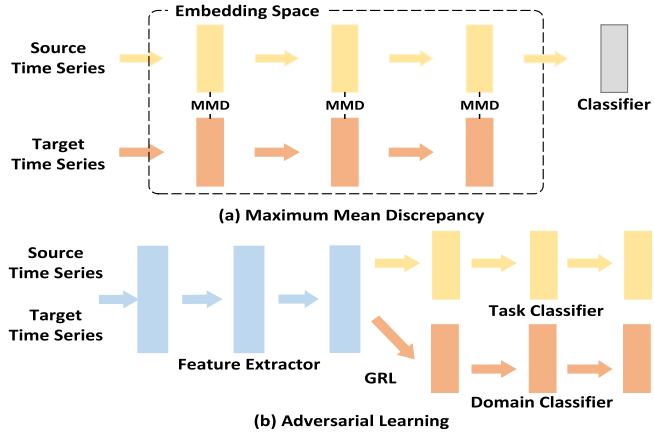

Fig. 5. Aligned encoder aims to learn domain-invariant representations.

Existing classification-based PTMs can be divided into three categories: (i) universal encoder, (ii) model reprogramming, and (iii) aligned encoder.

Universal Encoder: For TS-PTMs, a key issue is how to learn universal time-series representations [77] that can benefit a variety of downstream TSM tasks. A common approach is to design a universal encoder which can be quickly adapted to new tasks with or without fine-tuning. Serrà et al. [71] proposed a universal encoder that combines CNN with attention mechanism, and pre-trained the encoder using the supervised classification task. The encoder is jointly pre-trained on multiple time series datasets. Using multi-head outputs, each dataset has an extra fully-connected layer for classification. In this way, the encoder acts as a carrier to transfer knowledge from multiple related source domains to enhance learning in the target domain. Fig. 4 shows the schematic diagram of the universal encoder.

Fawaz et al. [23] considered transfer learning on time-series data from the shapelet perspective. Shapelets [78] are discriminative subsequences in the time series and can be used as an efficient time-series representation. Fawaz et al. [23] hypothesized that the learned shapelets can generalize to unseen datasets via transfer learning. For each dataset in the UCR archive [72], they trained a fully-convolutional network [73] and then fine-tuned it on all the other datasets. They found that pre-training can degrade (negative transfer) or improve (positive transfer) the encoder's performance on the target dataset, and the likelihood of positive transfer is greater when the source dataset is similar to the target dataset. Due to privacy and annotation issues, it may be difficult to obtain source datasets that are very similar to the target dataset. To alleviate this problem, Meiseles et al. [79] utilized the clustering property between latent encoding space categories as an indicator to select the best source dataset. The above studies mainly focus on univariate time series. Li et al. [76] proposed a general architecture that can be used for transfer learning on multivariate time series.

The aforementioned works employ CNN as backbone for time series transfer learning. However, vanilla CNNs have difficulty in capturing multi-scale information and long-term dependencies in the time series. Studies [59], [75] have shown that using different time scales or using LSTM with vanilla CNNs can further improve classification performance. For example, Kashiparekh et al. [60] proposed a novel pre-training deep CNN, in which 1-D convolutional filters of multiple lengths are used to capture features at different time scales. Mutegeki et al. [80] used CNN-LSTM as the base network to explore how transfer learning can improve the performance of time-series classification with few labeled time series samples.

Aligned Encoder: A universal encoder first pre-trains the model with the source dataset, and then fine-tunes the model using the target dataset. However, the difference between the source and target data distributions is often not considered. To address this issue, some recent works [81], [82] first map the source and target datasets to a shared feature representation space, and then prompt the model to learn domain-invariant representations during pre-training. The pre-training strategy of the aligned encoder is at the core of domain adaptation, and has been extensively validated studied on image data [83]. For time series data, extraction of domain-invariant representations is difficult due to distribution shifts among timestamps and the associative structure among variables. To this end, existing pre-training techniques for time series aligned encoder are based on either Maximum Mean Discrepancy (MMD) [84] or adversarial learning [85].

MMD [86] is a standard metric on distributions, and has been employed to measure the dissimilarity of two distributions in domain adaptation [87]. Given a representation $f(\cdot)$ on source data $\boldsymbol{X}_s \in \mathcal{D}_s$ and target data $\boldsymbol{X}_t \in \mathcal{D}_t$, the empirical approximation of MMD is

$$\text{MMD}(\mathcal{D}_s, \mathcal{D}_t) = \left\| \frac{1}{|\mathcal{D}_s|} \sum_{\boldsymbol{X}_s \in \mathcal{D}_s} f(\boldsymbol{X}_s) - \frac{1}{|\mathcal{D}_t|} \sum_{\boldsymbol{X}_t \in \mathcal{D}_t} f(\boldsymbol{X}_t) \right\|.$$

MMD-based methods [88], [89], [90] learn domain-invariant representations by minimizing the MMD between the source and target domains in classification training (Fig. 5(a)). Khan et al. [91] used a CNN to extract features from the source and target domain data separately. The divergence of the source and target domains is reduced by minimizing the Kullback–Leibler divergence in each layer of the network. Wang et al. [88] proposed stratified transfer learning to improve the accuracy for

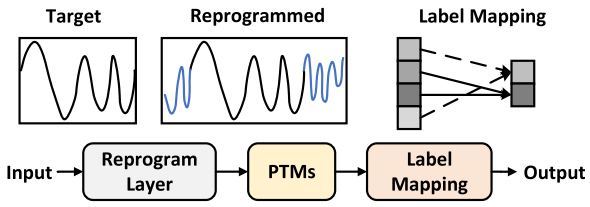

Fig. 6. Model reprogramming can adapt PTMs to a new task by reprogramming the time-series from the target domain and label mapping.

cross-domain activity recognition. Moreover, considering that time lags or offsets can influence the extraction of domain-invariant features, Cai et al. [81] designed a sparse associative structure alignment model which assumes that the causal structures are stable across domains. Li et al. [89] considered the compact causal mechanisms among variables and the variant strength of association, and used the Granger causality alignment model [92] to discover the data's causal structure. Ragab et al. [93] proposed an autoregressive domain discriminator to explicitly address the temporal dependencies during both representation learning and domain alignment. Despite all these advances, the use of MMD-based methods on time series data is still challenging due to the underlying complex dynamics.

Another common approach is to learn a domain-invariant representation between the source and target domains through adversarial learning [94], [95]. For example, Wilson et al. [82] proposed the Convolutional deep Domain Adaptation model for Time Series data (CoDATS), which consists of a feature extractor, Gradient Reversal Layer (GRL), task classifier, and domain classifier (Fig. 5(b)). The adversarial step is performed by the GRL placed between the feature extractor and domain classifier in the network. CoDATS first updates the feature extractor and task classifier to classify the labeled source data. The domain classifier is then updated to distinguish which domain each sample comes from. At the same time, the feature extractor is adversarially updated to make it more difficult for the domain classifier to distinguish which domain each sample comes from. Li et al. [96] argued that temporal causal mechanisms should be considered and proposed a time-series causal mechanism transfer network to obtain a domain-invariant representation. However, the exploitation of inherent properties in the time series (such as multi-scale and frequency properties) still need to be explored in adversarial training.

Model Reprogramming: The great success of PTMs in CV [31] and NLP [22] shows that using a large-scale labeled dataset can significantly benefit the downstream tasks. However, most time-series datasets are not large. Recently, a novel TS-PTM called Voice2Series [24] was proposed for time-series classification. Voice2Series is based on model reprogramming [97] (Fig. 6). It uses a large acoustic model pre-trained on massive human voice datasets (e.g., spoken-term recognition). The voice data can be considered as univariate time series, and therefore enormous voice data can be employed to pre-train an acoustic model. To make the acoustic model suitable for general time-series data, Voice2Series reprogrammed the model through input transformation learning and output label mapping. The input transformation of a time-series sample $\boldsymbol{X}$ is defined as

$$\boldsymbol{X}' = \mathrm{Pad}(\boldsymbol{X}) + \boldsymbol{M} \odot \boldsymbol{\theta},$$

where $\mathrm{Pad}(\cdot)$ is a zero-padding function, $\boldsymbol{M}$ is a binary mask, and $\boldsymbol{\theta}$ are reprogramming parameters for aligning the data distributions of the source and target domains. A random (but non-overlapping) many-to-one mapping between source and target labels is used as the output label mapping. Transformer-based attention mechanism [98] is used as the acoustic model. Note that model reprogramming not only uses more labeled training samples during pre-training, but also considers adapts the PTMs to the target tasks by constructing relationships between the source and target domain data.

Summary. The universal encoder first pre-trains a base network on the labeled source datasets, and then the base network is transferred to the target domain. This usually requires a large amout of labeled source samples for pre-training, and can be difficult to obtain in the time-series domain. Positive (resp. negative) transfer often occurs when the source and target datasets are similar (resp. dissimilar). Previous studies have explored how to select the source based on inter-dataset similarity or time series representations in the latent representation space. In addition, the aligned encoder based on domain adaption considers the differences between source and target data distributions. Voice2Serie [24] provides a new approach for classification-based PTMs. Some domain-specific time series data (e.g., voice data) is used to pre-train a base network, which is then applied to general time series through model reprogramming. Similarly, building on the success of pre-trained Large Language Models (LLMs) in time series modeling [99], [100], Time-LLM [101] converts time series data into text-like formats with patching strategy and use model reprogramming to transfer LLM knowledge to time series tasks. Nevertheless, how to construct a large-scale well-labeled time series dataset suitable for TS-PTMs remains an open challenge.

2) Forecasting-Based PTMs: Time Series Forecasting (TSF) aims to estimate the values at the future timesteps using observed values from the current and past timesteps. The problem of one-step-ahead forecasting is defined as

$$\hat{y}_{i,t+1} = f(y_{i,t-k:t}, \boldsymbol{x}_{i,t-k:t}),$$

where $\hat{y}_{i,t+1}$ is the model's predicted value of the ith sample at time $t+1$, $y_{i,t-k:t} = \{y_{i,t-k}, \ldots, y_{i,t}\}$ and $\boldsymbol{x}_{i,t-k:t} = \{\boldsymbol{x}_{i,t-k}, \ldots, \boldsymbol{x}_{i,t}\}$ are the observed values and exogenous inputs, respectively, over a look-back window of length k, and $f(\cdot)$ is the prediction function learnt by the model [42]. Unlike the classification task that employs manual labels as supervisory information, TSF utilizes the observed value in the future as supervisory information. In addition, the mean absolute error or mean squared error is often adopted as the loss function for the TSF task.

A unique property of time-series data is the presence of temporal dependencies. Forecasting can utilize a time series of past and present values to estimate future values, and it is naturally employed as a time series pre-training task. An intuitive approach to produce forecast values is the recursive strategy, which can be achieved by autoregression. In general, a PTM first pre-trains a

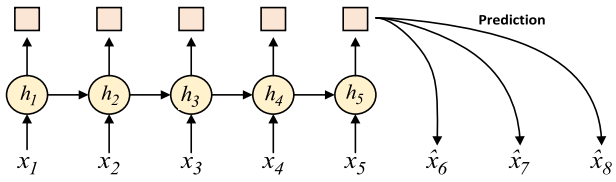

Fig. 7. The RNN-based forecasting architecture.

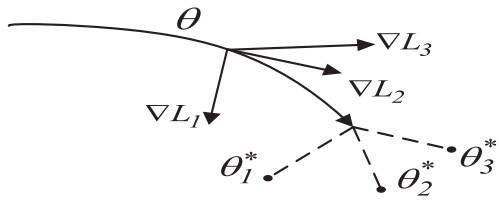

Fig. 8. Adaptive encoder aims to obtain a better initialization model parameters $\boldsymbol{\theta}$ so that the model can quickly generalize to new tasks using only a small number of samples (e.g., $\boldsymbol{\theta}_1^*$), where $\boldsymbol{\theta}_i^*(i \in 1, 2, 3)$ denote task adaptive parameters obtained using gradient descent on $\boldsymbol{\theta}$ (e.g., ∇L_1) based on the task adaptive data.

forecasting-based model on the source dataset. The weights of the base model are then fine-tuned on the target dataset.

AutoRegression. Deep learning models, including RNNs [55], TCNs [102] and the more recent Transformers [65], have been used for TSF. The RNN-based forecasting architecture is shown in Fig. 7. In this example, given the historical observations $[\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5]$, TSF tries to predict the future values $[\boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8]$. Thus, TSF does not require manual labels, but uses the values at future timesteps for supervision.

The dynamic nature and inherent temporal dependencies of the time series enable forecasting to assist TS-PTMs in obtaining robust representations [103], [104], [105]. For instance, Mallick et al. [106] developed a GNN-based model for pre-training short-term highway network forecasting models, employing a transfer learning strategy operation within a RNN. In addition, Du et al. [107] proposed Adaptive RNNs (AdaRNN) to solve the temporal covariate shift problem. The AdaRNN model consists of Temporal Distribution Characterization (TDC) and Temporal Distribution Matching (TDM) modules. The TDC divides the training time series into least-similar K different subsequences, which fully characterize the distribution information of the time series in each period. The K different subsequences are then used as source data to pre-train a generalized RNN model using the forecasting task. The TDM module can also be used with the Transformer architecture, further improving the TSF performance.

Some recent studies [108], [109], [110] combines the autoregressive forecasting strategy of TSF with contrastive learning [32] to obtain time series representations favorable for downstream tasks. For example, Oord et al. [108] proposed contrastive predictive coding by employing model-predicted timesteps as positive samples and randomly-sampled timesteps as negative samples. Eldele et al. [110] used the autoregressive model's predicted values as positive sample pairs for contrastive learning, thus enabling the model to capture temporal dependencies of the time series. In particular, they fed both strong and weak augmentation samples to the encoder, therefore using a cross-view TSF task as the PTM objective.

Adaptive Encoder: Unlike transfer learning which focuses on the current learning ability of the model, meta-learning [102], [111] focuses on the future learning potential of the model to obtain an *adaptive encoder* via task-adaptive pre-training paradigm (as shown in Fig. 8). Especially, transfer learning-based PTMs are prone to overfitting on downstream tasks when the number of samples in the target dataset is small. Inspired by the fact that humans are good at learning a priori knowledge from a few new samples, task-adaptive pre-training, or meta-learning, has also been used. Fig. 8 shows an adaptive encoder via a classic task-adaptive pre-training algorithm called model agnostic meta-learning [112], [113], [114]. Recently, task-adaptive pre-training strategies have been receiving increasing attention in the field of time series. Existing studies focus on how to perform cross-task learning using properties such as temporal and multivariate dependencies in the time series.

A task-adaptive pre-training paradigm based on time-series forecasting has been applied to TS-PTMs [115], [116]. For example, Oreshkin et al. [116] proposed a meta-learning framework based on deep stacking of fully-connected networks for zero-shot time-series forecasting. The meta-learning procedure consists of a meta-initialization function, an update function, and a meta-learner. The meta-initialization function defines the initial predictor parameters for a given task. The update function then iteratively updates the predictor parameters based on the previous value of the predictor parameters and the given task. The final predictor parameters are obtained by performing a Bayesian ensemble (or weighted sum) on the whole sequence of parameters. The meta-learner learns shared knowledge across tasks by training on different tasks, thus providing suitable initial predictor parameters for new tasks. Brinkmeyer and Rego [117] developed a few-shot multivariate time series forecasting model working across tasks with heterogeneous channels. Experimental results showed that it provides a good generalization for the target datasets. Also, Autoformer [64] and FEDformer [65] show that frequency domain information and Transformer architecture can improve time-series forecasting performance. Hence, it would be an interesting direction to explore task adaptive Transformer-based models for TS-PTMs.

Summary: TSF-based PTMs can exploit the complex dynamics in the time series and use that to guide the model in capturing temporal dependencies. Autoregression-based models use the dependencies between subseries and the consistency of future predicted values of the same time series, thus pre-training the time series data using TSF. Unlike classification-based PTMs that use manual labels for pre-training, avoiding sampling bias among subseries (e.g., outliers) [77] for pre-training based on the TSF task remain challenging. Meanwhile, the adaptive encoder based on meta-learning allows for scenarios with small time series samples in the target dataset. In addition, regression-based one-step forecasting models (e.g., RNNs) potentially lead to bad performance due to accumulated error [8]. Instead, some studies [11], [64] employ Transformer-based models to generate all predictions in one forward operation. Therefore, designing

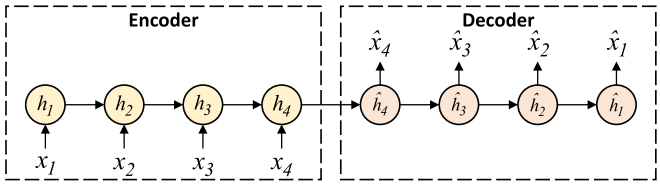

Fig. 9. The RNN-based encoder-decoder architecture.

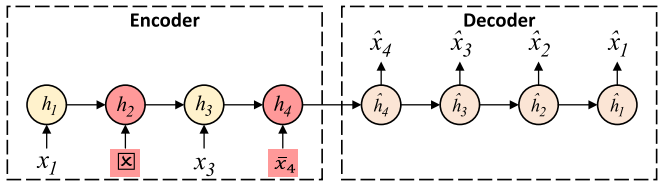

Fig. 10. The RNN-based denoising encoder-decoder architecture.

efficient TSF encoders would be a basis for studying TSF-based PTMs.

B. Unsupervised PTMs

This section introduces unsupervised TS-PTMs, which are often pre-trained by reconstruction techniques. Compared with supervised TS-PTMs, unsupervised TS-PTMs are more widely applicable since they do not require labeled samples.

1) Reconstruction-Based PTMs: Reconstruction is a common unsupervised task, and is usually implemented by an encoder-decoder architecture [56]. The encoder maps the original time series to a latent space of representations, which is then used by the decoder to reconstruct the input time series. The mean square error is often used as the reconstruction loss. For example, Castellani et al. [118] used reconstruction to learn robust representations for detecting noisy labels in the time series. Naturally, reconstruction is also utilized for TS-PTMs. In the following, we introduce TS-PTMs based on the (i) autoencoder, (ii) denoising autoencoder, and (iii) transformer encoder.

AutoEncoder: For variable-length NLP sequences such as sentences, paragraphs, and documents, it has been very useful to first convert them to fixed-dimensional vectors [22]. Inspired by this, Malhotra et al. [25] proposed the time-series model Timenet, which encodes a univariate time series to a fixed-dimensional vector via the Sequence-to-Sequence (Seq2Seq) framework [119]. This consists of an encoder and a decoder. An example RNN-based encoder-decoder is shown in Fig. 9. The encoder converts the input sequence to a fixed-dimensional vector representation, and the decoder reconstructs it to another sequence as output. This process can be written as: $\boldsymbol{H} = f_{enc}(\boldsymbol{X})$ and $\hat{\boldsymbol{X}} = f_{dec}(\boldsymbol{H})$, where $\boldsymbol{H}$ is the representation, $\boldsymbol{X}$ is the input, $\hat{\boldsymbol{X}}$ is the reconstructed input, $f_{enc}(\cdot)$ and $f_{dec}(\cdot)$ are the encoder and decoder functions, respectively. By combining the pre-trained encoder with a classifier, Timenet has shown good performance on time-series classification. However, using autoencoder-based PTMs for other TSM tasks, such as forecasting and anomaly detection, has been less explored.

Denoising AutoEncoder: To learn more robust representations for the downstream tasks, the Denoising AutoEncoder (DAE) [120] has been used [121]. As shown in Fig. 10, the input time series $\boldsymbol{X}$ is corrupted to $\bar{\boldsymbol{X}}$ by adding noise or random masking. The DAE is then trained to reconstruct the original $\boldsymbol{X}$. Compared to the vanilla AutoEncoder, the DAE makes learning more difficult, thus enhancing robustness of the time-series representations. Since the DAE has achieved good performance on representation learning [120], [122], it is also used in the TS-PTMs [123], [124], [125].

A pioneering work that uses DAE in TS-PTM is the Audio Word2Vec [126]. Given variable-length audio segments, Audio Word2Vec obtains fixed-dimensional vectors, which are then used in applications such as query-by-example Spoken Term Detection (STD). To learn robust representations, Audio Word2Vec consists of two stages: offline and online [120]. In the offline phase, a pre-trained model is obtained using all the audio segments. This is then used to encode online language sequences to fixed-dimensional vectors. Inspired by Audio Word2Vec, Hu et al. [127] employed a DAE to pre-train the model for short-term wind speed time-series forecasting. Also, Shao et al. [128] integrated Graph Neural Networks (GNNs) with an unsupervised masked auto-encoder strategy for pre-training multivariate time series forecasting models. Specifically, they employed a series of Transformer blocks as the encoder-decoder architecture and used GNNs to capture the spatio-temporal relationships between different variables.

DAE's "mask and predict" mechanism has been successfully used in pre-training NLP [129] and CV models [21]. In the time-series domain, Ma et al. [122] proposed a joint imputation and clustering DAE-based model for incomplete time-series representation learning. They showed that the joint learning strategy enables the imputation task to be optimized in the direction favoring the downstream clustering task, resulting in better downstream clustering performance. Furthermore, Nie et al. [123] segment time series into patches and successfully apply the "mask and predict" mechanism for time series pre-training. Additionally, studies [130], [131], [132], [133] indicate that this mechanism can be effectively used for TS-PTMs.

Transformer Encoder: Transformers based on "mask and predict" training paradigms can be regarded as reconstruction-based models, which have been wildly applied in NLP for studying PTMs [129]. Transformers have received much attention because of their ability to capture input data's contextual (future-past) dependencies, making them a good choice for modeling sequence data [134], [135]. Inspired by Transformer-based PTMs in NLP [136], Zerveas et al. [26] proposed a multivariate time-series unsupervised representation learning model called Time-Series Transformer (TST). Specifically, TST employs the original transformer encoder [137] to learn multivariate time series representations through a designed masking strategy. Further, Shi et al. [138] proposed an unsupervised pre-training method based on the self-attention mechanism of Transformers. In particular, the authors [138] utilize a denoising pretext task that captures the local dependencies and changing trends in the time series by reconstructing corrupted time series. Unlike the above studies, Zhang et al. [29] proposed a cross reconstruction transformer pre-trained by a cross-domain dropping

reconstruction task, thereby modeling temporal-spectral relations of time series.

In addition, Transformer-based PTMs have been applied to traffic flow time series, tabular time series, and speech series data. For example, Hou et al. [139] proposed a token-based PTM for traffic flow time series. Concretely, the authors [139] designed a random mask tokens strategy, and then the transformer encoder was pre-trained to predict these tokens. Further, Zhao et al. [140] designed a bidirectional Transformer encoder to learn relationships between two-time intervals across different scales. For tabular time-series data, Padhi et al. [141] proposed hierarchical tabular BERT [129] as the backbone to pre-train by predicting masked tokens. Also, Shankaranarayana et al. [142] utilized a 1-D convolution model combined with the transformer as the backbone, which was pre-trained using the "mask and predict" mechanism. In terms of speech series data, Liu et al. [143] proposed an unsupervised speech representation learning method based on a multilayer transformer encoder, pre-trained by recovering the random 15% zero-masked input frames. Unlike the PTMs mentioned above, Liu et al. [135] proposed a novel PTM called Transformer Encoder Representations from Alteration (TERA) for speech series data. Specifically, TERA utilized three self-supervised training schemes (time alteration, frequency alteration, and magnitude alteration) based on a transformer encoder through reconstructing altered counterpart speech data for pre-training.

Summary: DAE-based TS-PTMs add noise or masks to the original time series for pre-training, and have recently been gaining attention compared to autoencoder-based PTMs. Using DAE to study PTMs [21] demonstrates the potential of denoising strategies in pre-training. However, designing DAE-based PTMs applicable to time series is still in the exploratory stage. Meanwhile, Transformer-based PTMs inherit the advantages of unsupervised training from the denoising strategy and have achieved good pre-training results in NLP in recent years [136]. Nevertheless, existing Transformer-based TS-PTMs mainly focus on the time series classification and forecasting tasks, and their performance on other downstream tasks warrants further exploration. At the same time, the time series of different domains may vary widely, resulting in poor model transferability across different domain datasets. Therefore, how to design reconstruction-based unsupervised learning mechanisms (i.e, mask and predict) applicable to different domains is a challenge for studying TS-PTMs.

C. Self-Supervised PTMs

This section presents self-supervised TS-PTMs based on consistency and pseudo-labeling training strategies which are commonly utilized in self-supervised learning. Compared with unsupervised learning (e.g., reconstruction), self-supervised learning employs self-provided supervisory information (e.g., pseudo-labels) during the training process.

1) Consistency-Based PTMs: Self-supervised learning strategies based on the consistency of different augmented (transformed) views from the same sample have been studied for PTMs in CV [144] with great success. Naturally,

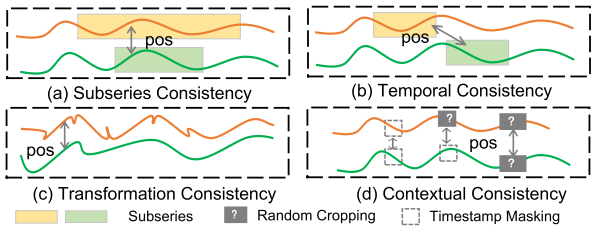

Fig. 11. Positive pair selection strategies in contrastive learning.

consistency-based strategies [108], [145], [146], [147] have recently attracted attention in the time series field. Specifically, consistency-based PTMs keep the distances of different view representations from the same sample (positive pairs) close to each other, while keeping the distances of view representations (negative pairs) from different samples far from each other. The above learning strategy motivates the model to learn representations beneficial for downstream tasks. Based on the above idea, contrastive learning [31], [32] is commonly employed as a training loss for PTMs, which is defined as

$$\mathcal{L}_{CL} = -\frac{1}{N}\sum_{i=1}^{N}\log \frac{\exp(f(\boldsymbol{X}_i)^{\mathrm{T}} f(\boldsymbol{X}_i^p))}{\exp(f(\boldsymbol{X}_i)^{\mathrm{T}} f(\boldsymbol{X}_i^p)) + \sum_{j=1}^{B-1}\exp(f(\boldsymbol{X}_i)^{\mathrm{T}} f(\boldsymbol{X}_i^j))}, \quad (1)$$

where N denotes the number of training sample pairs and each anchor sample ($\boldsymbol{X}_i$) has 1 positive sample ($\boldsymbol{X}_i^p$) and $B-1$ negative samples. $\boldsymbol{X}_i^j$ denotes the jth negative sample of the ith anchor sample.

Consistency-based PTMs need to consider how to construct positive and negative samples so that the model can effectively exploit the complex dynamic properties of time series data. It is worth noting that there is no uniform large-scale well-labeled time series dataset for pre-training. Most existing consistency-based PTMs first pre-train the encoder using the target dataset via self-supervised representation learning, and then fine-tune the pre-trained encoder using the supervised information from the target dataset or directly utilize the representations for downstream tasks. Using contrastive learning for time-series consistency-based PTMs can be roughly divided into four categories [77]: subseries consistency, temporal consistency, transformation consistency, and contextual consistency.

Subseries Consistency: Two subseries belonging to an inclusion relationship sampled from the same time series sample are utilized as positive pair, which is called subseries consistency, as shown in Fig. 11(a). Utilizing Word2Vec [148] as an analogy, Franceschi et al. first [149] employed a sufficiently long and non-stationary subseries in a time series sample as the context. Then, a subseries selected from the context corresponds to a word (positive subseries), while a subseries collected from a different context in another time series sample represents a random word (negative subseries). Further, the authors [149] employed the Triplet Loss (T-Loss) to keep the context and positive subsequences close, while making the context and negative subsequences far away for representation learning of time series.

Experimental results on UCR and UEA archives indicated that the representations obtained by T-Loss can be beneficial for the downstream classification task.

Moreover, Fan et al. [150] proposed a self-supervised representation learning framework named SelfTime, by exploring the inter-sample and intra-temporal relations of time series. In terms of the inter-sample relationships, unlike SimCLR [32], the authors employed the cross-entropy loss to guide the reasoning of different levels of entity relationships. As for the intra-temporal relationships, the authors attempted to capture the temporal pattern by reasoning the multi-level relation among subseries sampled from the same sample, thus obtaining representations for the downstream classification task.

Temporal Consistency: Two adjacent subseries from the same time series sample are selected as a positive pair, which is called temporal consistency, as shown in Fig. 11(b). Based on the assumption of temporal consistency, Tonekaboni et al. [151] proposed a representation learning framework named Temporal Neighborhood Coding (TNC). TNC learns time series representations by ensuring that the distribution of adjacent signals in the coding space is distinguishable from the distribution of non-neighboring signals. Experimental results on multiple time series datasets demonstrate that TNC performs better on downstream classification and clustering tasks compared with the T-Loss [149].

Further, Woo et al. [152] proposed a contrastive learning pre-training method for long sequence time series forecasting, called CoST. Specifically, CoST utilizes temporal consistencies in the time domain to learn the discriminative trend of a sequence, while transforming the data to the frequency domain to discover the seasonal representation. Deldari et al. [153] proposed a self-supervised time series change point detection method based on contrast predictive coding. Particularly, the authors exploit local correlations in the time series to drive the maximization of shared information across consecutive time intervals while minimizing shared information across time intervals. Luo et al. [154] applied the objective of contrastive learning to both the local subseries and global instance levels, thus obtaining feature representations favorable to downstream time series forecasting and classification tasks.

Transformation Consistency: Two augmented sequences of the same time series by different transformations are selected as positive pair, which is called transformation consistency, as shown in Fig. 11(c). Inspired by the successful application of transformation consistency in CV, Eldele et al. [110] proposed a Time-Series representation learning framework via Temporal and Contextual Contrasting (TS-TCC). TS-TCC first transforms the original time series data in two different ways to obtain weak augmentation (jitter-and-scaling) and strong augmentation (permutation-and-jitter) variants, respectively. Then, the temporal and contextual contrastive learning modules are designed in TS-TCC for learning discriminative representations.

In addition, Hyvärinen et al. [155] analyzed the non-stationarity of time series by performing a nonlinear mixture transformation of subseries from different time windows to ensure that the same subseries are consistent. Unlike [155], Hyvarinen and Morioka [156] developed a method based on a logistic regression estimation model to learn the dependencies of time dimensions by distinguishing the differences between subseries of the original time series and those randomly transformed time points. Lately, Zhang et al. [28] utilized a time-based augmentation bank (jittering, scaling, time-shifts, and neighborhood segments) and frequency-based augmentation methods (adding or removing frequency components) for time series self-supervised contrastive pre-training via time-frequency consistency.

Contextual Consistency: Two augmented contexts with the same timestamp inside the same time series are selected as a positive pair, which is called contextual consistency, as shown in Fig. 11(d). By designing a timestamp masking and random cropping strategy to obtain augmented contexts, Yue et al. [77] proposed a universal framework for learning time series representations named TS2Vec. Specifically, TS2Vec distinguishes positive and negative samples from the temporal dimensions and instance level hierarchically. In terms of temporal consistency, the augmented contexts with the same timestamp within the same time series are treated as positive pairs, while the augmented contexts in different timestamps are treated as negative pairs. Unlike temporal consistency, instance level consistency treats the augmented contexts of other instances within a batch as negative pairs. Experiments on various time series datasets indicate that TS2Vec achieves excellent performance on downstream classification, forecasting, and anomaly detection tasks.

Recently, Yang et al. [157] proposed a novel representation learning framework for time series, namely Bilinear Temporal-Spectral Fusion (BTSF). Unlike TS2Vec [77], BTSF utilizes instance-level augmentation by simply applying a dropout strategy to obtain positive/negative samples for unsupervised contrastive learning, thus preserving the global context and capturing long-term dependencies of time series. Further, an iterative bilinear temporal-spectral fusion module is designed in BTSF to iteratively refine the representation of time series by encoding the affinities of abundant time-frequency pairs. Extensive experiments on classification, forecasting, and anomaly detection tasks demonstrate the superiority of BTSF.

Summary: The aforementioned studies demonstrate that consistency-based PTMs can obtain robust representations beneficial for TSM tasks. Although subseries and temporal consistency strategies have achieved good performance on the classification task, sampling biases between subseries (e.g., outliers and pattern shifts) [77] tend to introduce false positive pairs. Meanwhile, the transformation consistency strategy relies on effective instance-level data augmentation techniques. However, designing uniform data augmentation techniques for time series from different domains remains a challenge [13], [158]. One alternative is to utilize expert features of time series [159] instead of the commonly used data transformation for contrastive learning. The contextual consistency strategy utilizes a mask (or dropout) mechanism [77], [157] to obtain contrastive pairs by capturing temporal dependencies, which can alleviate the problem of sampling bias among subsequences and achieve excellent performance on forecasting, classification, and anomaly detection tasks. Nevertheless, designing consistency-based strategies using the multi-scale property [75] of time series has not been fully explored.

2) Pseudo-Labeling PTMs: In addition to consistency-based PTMs mentioned above, some other self-supervised TS-PTMs have been explored to improve the performance of TSM. We briefly review these methods in this section.

Predict Pseudo Labels: A large amount of correctly labeled information is the basis for the success of deep neural network models. However, labeling time series data generally requires manual expert knowledge assistance, resulting in high annotation costs. Meanwhile, representation learning using pseudo-label-guided models has yielded rich results in CV [160]. In addition, some studies [161], [162] employed self-supervised learning as an auxiliary task to help the primary task learn better by training the auxiliary task alongside the primary task. In the auxiliary task, when given the transformed sample, the model predicts which transformation is applied to the input (i.e., predicts pseudo labels). This idea has been applied in a few TS-PTMs [150]. For example, Fan et al. [150] randomly selected two length-L subsequences from the same time series and assigned a pseudo-label based on their temporal distance, and then the proposed model is pre-trained by predicting the pseudo-label of subsequence pairs. Also, Zhang et al. [33] incorporated expert features to create pseudo-labels for time series self-supervised contrastive representation learning. Despite these progresses, predicting pseudo labels inevitably contain incorrect labels. How to mitigate the negative impact of incorrect labels will be the focus of studying TS-PTMs.

IV. Experimental Results and Analysis

In this section, following [77], [157], we evaluate TS-PTMs on three TSM tasks, including classification, forecasting, and anomaly detection. Like [77], we select a range of time-series benchmark datasets employed in the corresponding TSM tasks for evaluation. We first analyze the performance of TS-PTMs on the classification task using UCR [163] and UEA [164] archives time series datasets. Also, following [28], we select four time series scenarios datasets for transfer learning analysis. Second, the performance of TS-PTMs and related baselines on the forecasting task are compared using nine benchmark datasets [11], [165]. Finally, we analyze the performance of TS-PTMs on the anomaly detection by using the univariate Yahoo [166], KPI [167], UCR anomaly detection archive [168], and seven multivariate datasets [145]. For information about datasets, baselines, and implementation details, please refer to Appendix B, available online.

A. Performance of PTMs on Time-Series Classification

The datasets in the UCR and UEA archives do not specify a validation set for hyperparameter selection, while some datasets have a much higher number of test set samples than the training set. As suggested by [163], we merge the training and test sets for each dataset in UCR and UEA archives. Then, we adopt the five-fold cross-validation strategy to divide the training, validation, and test sets in the ratio of 60%-20%-20%. For the four independent time series scenarios datasets, we utilize the datasets processed by [28] for experimental analysis. Finally, we use the average accuracy on the test set as the evaluation metric.

1) Comparison of Transfer Learning PTMs Based on Supervised Classification and Unsupervised Reconstruction: Due to space constraints, we report the transfer learning classification results in Table I. The P-value is calculated for the transfer classification results between the supervised classification and unsupervised reconstruction transfer strategy. As shown in Table I, the supervised classification transfer (SUP CLS) strategy has the best average Acc and the number of positive transfer results on the minimum, medium and maximum target datasets. The above results indicate that the SUP CLS strategy is more suitable for time series pre-training than the unsupervised transfer strategy. On the smallest target datasets, the overall performance of the SUP CLS strategy and the unsupervised reconstruction strategy utilizing the symmetric FCN decoder is insignificant (P-value greater than 0.05). Also, the unsupervised reconstruction strategy utilizing the symmetric FCN decoder is better than the supervised classification transfer strategy. The above results indicate that a symmetric FCN decoder is more suitable for transfer learning in the time-series classification task than the asymmetric RNN decoder. Overall, the number of positive transfer classification results obtained on the target datasets is unsatisfactory, which may be related to the small number of samples in the UCR source datasets (the number of samples in most source datasets is less than 8000, and please refer to Table VI in the Appendix), available online.

Further, we select four independent time series datasets for transfer learning PTMs analysis. The number of samples in each source dataset is large, and the test classification results are shown in Table II. Compared with the supervised approach (FCN) without pre-training, the transfer learning strategy based on supervised classification (Sup CLS) achieves significant positive transfer performance in the neural stage detection and mechanical device diagnosis scenarios. In addition, the transfer learning strategy based on unsupervised RNN Decoders achieves obvious positive transfer performance in the activity recognition scenario. However, in the physical status monitoring scenario, the three transfer learning strategies all result in negative transfer, which may be related to the small number of EMG samples. Compared with using UCR time series datasets, transfer learning has a better pre-training effect on independent time series datasets with large source datasets.

2) Comparison of TS-PTMs for Classification: We select one general time series model and seven TS-PTMs for performance comparison of the downstream classification task. TimesNet [165] is a general model based on CNNs for time series analysis. TST [26] is a transformer-based PTM. T-loss [149], Selftime [150], TS-TCC [110], and TS2Vec [77] are consistency-based PTMs. PatchTST [123] utilizes patches based on the transformer for time series modeling, employing a reconstruction strategy for pre-training. GPT4TS [99] adopts the same patch strategy in PatchTST to fine-tune pre-trained LLMs for time series analysis. Related work in CV [32] and NLP [129] generally utilizes a selected backbone combined with a linear classifier as a supervised method to analyze the classification performance compared with PTMs. Therefore, the FCN combined with a linear classifier is selected as a

TABLE I
TEST CLASSIFICATION RESULTS OF TRANSFER LEARNING ON 675 SETS (15 SOURCE DATASETS MULTIPLIED BY 45 TARGET DATASETS) UCR TIME SERIES DATASETS

	15 Minimum Target Datasets				15 Medium Target Datasets				15 Maximum Target Datasets			
Transfer Strategy	Avg. Acc	Avg. Rank	P-value	Positive	Avg. Acc	Avg. Rank	P-value	Positive	Avg. Acc	Avg. Rank	P-value	Positive
Sup CLS	**0.7720 (0.0738)**	1.75	-	**48 (225)**	**0.7196 (0.0398)**	**1.45**	-	**9 (225)**	**0.7885 (0.0231)**	**1.73**	-	**17 (225)**
Unsup FCN Decoder	0.7616 (0.0756)	**1.59**	5.53E-02	41 (225)	0.7001 (0.0380)	1.92	7.64E-08	4 (225)	0.7715 (0.0240)	2.02	1.27E-03	15 (225)
Unsup RNN Decoder	0.7292 (0.0823)	2.16	2.28E-9	29 (225)	0.6655 (0.0355)	2.20	1.87E-15	2 (225)	0.7532 (0.0264)	2.20	9.38E-07	14 (225)

Positive represents the classification performance is better than direct classification without using a transfer strategy.
The best results are in bold and underlined.

TABLE II
TEST CLASSIFICATION RESULTS OF TRANSFER LEARNING ON FOUR INDEPENDENT TIME SERIES SCENARIOS

Scenario	Source Dataset	Target Dataset	Supervised (FCN)	Sup CLS	Unsup FCN Decoder	Unsup RNN Decoder
Neurological Stage Detection	SleepEEG	Epilepsy	0.7766 (0.0552)	**0.8021 (0.0010)**	0.6813 (0.2483)	0.6813 (0.2486)
Mechanical Device Diagnosis	FD-A	FD-B	0.5781 (0.0292)	**0.6718 (0.0783)**	0.6541 (0.0547)	0.6541 (0.0543)
Activity Recognition	HAR	Gesture	0.3904 (0.0281)	0.3796 (0.0257)	0.3517 (0.0363)	**0.4933 (0.0196)**
Physical Status Monitoring	ECG	EMG	**0.9756 (0.0322)**	0.4634 (0.0010)	0.8439 (0.0892)	0.8731 (0.0103)

Supervised (FCN) means that the FCN encoder is directly used for supervised classification training on the target dataset.
The best results are in bold and underlined.

TABLE III
COMPARISONS OF CLASSIFICATION TEST ACCURACY ON DIFFERENT TS-PTMS (STANDARD DEVIATIONS ARE IN PARENTHESES)

Models	128 UCR datasets				30 UEA datasets			
	Avg. Acc	Avg. Rank	P-value	Training Time (hours)	Avg. Acc	Avg. Rank	P-value	Training Time (hours)
Supervised (FCN)	0.8296 (0.0520)	4.15	1.24E-02	4.87	0.7718 (0.0722)	3.07	4.80E-02	0.73
T-Loss	0.8325 (0.0302)	5.07	1.36E-02	37.56	0.5863 (0.0459)	7.07	2.78E-09	22.03
Selftime	0.8017 (0.0339)	6.59	9.69E-07	-	-	-	-	-
TS-TCC	0.7807 (0.0430)	6.59	8.79E-12	9.45	0.7059 (0.0432)	5.27	5.64E-05	34.35
TST	0.7755 (0.0365)	6.39	4.53E-13	196.72	0.6921 (0.0382)	5.37	3.29E-05	52.1
TS2Vec	**0.8691 (0.0265)**	**3.28**	1.28E-01	**0.87**	0.7101 (0.0411)	4.27	1.04E-03	**2.07**
TimesNet	0.8367 (0.0687)	4.4	4.78E-05	336.62	0.7570 (0.0833)	4.13	2.58E-03	137.72
PatchTST	0.8265 (0.0706)	4.16	2.60E-03	11.94	0.7504 (0.0898)	3.9	3.13E-03	120.47
GPT4TS	0.8593 (0.0761)	3.73	-	86.02	**0.8355 (0.0978)**	**2.7**	-	16.84

The best results are in bold and underlined.

baseline without pre-training. The averaged results on UCR and UEA archives are reported in Table III. Detailed results are in Appendix C-B, available online. Also, we provide the visualization analysis in Appendix C-B, available online. Since the training of Selftime on the multivariate time series UEA datasets is too time-consuming, we only report its performance on the univariate UCR datasets. Also, we spend about a month on four 1080Ti GPUs to obtain the classification results of SelfTime on the UCR archive.

As shown in Table III, the classification performance of TS2Vec and GPT4TS achieve the best and second-best on the 128 UCR datasets. Also, the P-value is calculated to compare the classification results between GPT4TS and other methods. In terms of the P-value for significance tests, TS2Vec is insignificant compared to GPT4TS (P-value greater than 0.05). However, GPT4TS requires more training time for fine-tuning LLMs compared to TS2Vec. The above results show that both consistency-based TS2Vec and LLM-based GPT4TS can effectively learn robust representations beneficial for the time-series classification task on the UCR time series classification archive. On the 30 UEA datasets, GPT4TS achieves the best performance. The model using FCN for direct classification is the second-best, while TS2Vec is inferior in terms of average accuracy and rank. Also, PatchTST outperforms TS2Vec, highlighting the effectiveness of the patch strategy combined with a vanilla Transformer for multivariate time series classification. Additionally, GPT4TS shows faster training times on the UEA datasets compared to most other methods, such as PatchTST, due to its ability to converge quickly with fewer training epochs. In contrast, TimesNet requires significantly more training time on both the UCR and UEA datasets. In summary, consistency-based strategies, patch-based strategies, and LLM-based fine-tuning demonstrate significant potential for pre-training in time series classification tasks.

B. *Performance of PTMs on Time-Series Forecasting*

We further evaluate the performance of PTMs including TS2Vec [77], CoST [152], GPT4TS [99], and TEMPO [100] on time-series forecasting. Experiments on state-of-the-art direct forecasting methods are also conducted for comparison. These approaches include five Transformer based models, LogTrans [171], Informer [11], Autoformer [64], PatchTST [123], iTransformer [172], one simple liner layer-based model named DLinear [173], Temporal Convolutional Network (TCN) [61], and TimesNet [165]. We employ the Mean Square Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics following [152]. Also, we follow [77], [165] to preprocess the datasets. Table IV presents the average test results with different prediction lengths on nine public datasets for multivariate time-series forecasting. For detailed results of Table IV, please refer to Appendix C-C, available online.

TABLE IV
COMPARISON OF AVERAGE TEST RESULTS FOR TIME-SERIES FORECASTING WITH PREDICTION LENGTHS OF $S \in \{24, 48, 168, 336, 720\}$ FOR ETTH1, ETTH2, AND ELECTRICITY DATASETS; $S \in \{24, 48, 96, 288, 672\}$ FOR ETTM1 DATASET; $S \in \{24, 36, 48, 60\}$ FOR ILI DATASET; AND $S \in \{96, 192, 336, 720\}$ FOR OTHER DATASETS

Models	LogTrans		TCN		Informer		Autoformer		TS2Vec		CoST		TimesNet		PatchTST		DLinear		GPT4TS		TEMPO		iTransformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.9166	0.7521	0.8815	0.7098	0.9174	0.7317	0.4574	0.4639	0.8061	0.6543	0.6482	0.5843	0.4758	0.4635	0.5121	0.4804	0.4697	0.4633	0.4655	0.4602	0.5341	0.4972	**0.4179**	**0.4259**
ETTh2	3.0287	1.3299	4.1595	1.5241	3.3405	1.4479	0.4126	0.4412	1.5138	0.9413	1.2838	0.8533	0.3896	0.4060	0.3676	0.3989	0.4929	0.4816	0.3917	0.4172	0.4542	0.4544	**0.3363**	**0.3759**
ETTm1	1.0399	0.6569	0.7323	0.6077	1.0390	0.6837	**0.2911**	**0.3415**	0.6834	0.5717	0.5957	0.5253	0.4029	0.4049	0.3672	0.3852	0.3693	0.3876	0.3260	0.3621	0.3698	0.3907	0.3538	0.3784
ETTm2	1.2367	0.8489	0.9831	0.7286	1.4381	0.8539	0.3137	0.3565	1.1120	0.7532	0.9154	0.6846	0.3256	0.3493	0.2943	0.3367	0.3550	0.4041	0.2974	0.3508	0.3032	0.3492	**0.2919**	**0.3342**
Electricity	0.3921	0.4210	0.4998	0.4889	0.6151	0.5550	0.2401	0.3370	0.3622	0.4206	0.1916	0.2845	0.1915	0.2891	0.2136	0.2913	0.2750	0.3661	**0.1497**	**0.2409**	-	-	0.2005	0.2818
Traffic	0.9500	0.5396	0.7040	0.4360	0.7291	0.4089	0.6398	0.3942	-	-	-	-	0.6417	0.3362	0.6308	0.3937	0.7918	0.4835	**0.3953**	**0.2677**	0.4194	0.3035	0.4518	0.3002
Weather	0.3074	0.3730	0.2543	0.3169	0.7977	0.6144	0.3508	0.3853	0.2620	0.3288	**0.2347**	0.3026	0.2777	0.2992	0.2791	0.2973	0.2662	0.3184	0.2353	**0.2755**	0.2453	0.2802	0.2625	0.2825
Exchange	1.5922	1.0458	1.0117	0.7443	1.2631	0.9041	0.5071	0.5011	0.6890	0.6061	0.7597	0.6614	0.5506	0.4878	0.3856	0.4242	**0.2943**	**0.4053**	0.4430	0.4525	-	-	0.3724	0.4136
ILI	7.2016	1.9188	7.2306	1.9251	5.3215	1.5495	3.3649	1.3129	3.2403	1.1139	**2.0877**	**0.9073**	2.2351	0.9849	2.7266	1.1015	3.9794	1.4557	2.9957	1.2795	4.3664	1.5527	2.1335	1.0102

"-" indicates that the results could not be obtained due to memory errors or excessive training time.
The best results are in bold and underlined.

TABLE V
COMPARISIONS TEST RESULTS OF TIME-SERIES ANOMALY DETECTION

Models	Yahoo									KPI								
	F1	P	R	Aff-P	Aff-R	R_A_R	R_A_P	V_ROC	V_PR	F1	P	R	Aff-P	Aff-R	R_A_R	R_A_P	V_ROC	V_PR
SPOT	0.5037	0.4972	0.5105	0.9413	0.5425	0.5758	0.4389	0.5828	0.4420	0.1911	0.7628	0.1092	**0.9310**	0.4090	0.5431	0.1211	0.5535	0.1314
DSPOT	0.3952	0.3222	0.5109	**0.9468**	0.2889	0.5500	0.2814	0.5585	0.2890	0.1250	0.0889	0.2106	0.9092	0.3696	0.5224	0.1136	0.5243	0.1154
LSTM-VAE	0.5386	0.4764	0.6196	0.7533	0.7818	0.6791	0.6662	0.6709	0.6392	0.1833	0.3715	0.1216	0.5488	0.6642	**0.6619**	**0.4229**	0.6506	0.3939
DONUT	0.4352	0.3795	0.5101	0.6429	0.6930	**0.7100**	**0.6834**	**0.6978**	**0.6413**	0.4044	0.5423	0.3225	0.5411	**0.7006**	0.6615	0.4200	**0.6516**	**0.3943**
SR*	0.5630	0.4510	0.7470	-	-	-	-	-	-	0.6220	0.6570	**0.5980**	-	-	-	-	-	-
AT	0.7529	**0.8173**	0.6980	-	-	-	-	-	-	0.4444	0.7272	0.3200	-	-	-	-	-	-
TS2Vec	**0.7574**	0.7543	**0.7605**	0.9331	**0.8788**	0.6150	0.6056	0.6207	0.6026	**0.6904**	**0.9221**	0.5517	0.8730	0.5922	0.5659	0.3753	0.5709	0.3782

Aff-P and Aff-R refer to the precision and recall of the affiliation metric [169]. R_A_R and R_A_P represent range-AUC-ROC and range-AUC-PR [170], indicating scores based on label transformation under the ROC and PR curves, respectively. V_ROC and V_RR denote the volumes under the surfaces created by the ROC and PR curves [170], respectively.
The best results are in bold and underlined.

As can be seen in Table IV, iTransformer and GPT4TS generally outperform other methods. Also, CoST, Autoformer, and DLinear demonstrate notable improvements, with better long-term robustness on two, one, and one datasets, respectively. The performance of GPT4TS suggests that fine-tuning pre-trained LLMs with patching process data for time series forecasting is a promising strategy. Furthermore, iTransformer and Autoformer demonstrate the potential of Transformer frameworks for time series pre-training. Note that CoST achieves the best performance on the Weather and ILI datasets. We credit it to the modeling of trend and seasonal features with self-supervised pre-training, which is also verified to be effective in the decomposition architecture of Autoformer. The above empirical results demonstrate that designing an efficient TS-PTM is a promising paradigm for time series forecasting.

C. Performance of PTMs on Time-Series Anomaly Detection

For time-series anomaly detection, we follow the settings of [77], [167] to determine whether the last time point x_t of a sequence fragment $[x_1, x_2, \ldots, x_t]$ is an anomaly. Also, we select various evaluation metrics for a comprehensive comparison, including the commonly-used F1-Score (F1), Precision (P), and Recall (R) metrics [77], as well as recently proposed evaluation measures: f1-score Point Adjustment (PA)%K [174], affiliation precision/recall pair [169], and volume under the surface [170]. The TS2Vec [77], TimesNet [165], GPT4TS [99], DCdetector [145], and the benchmark methods SPOT [175], DSPOT [175], LSTM-VAE [176], DONUT [177], Spectral Residual (SR) [167], and Anomaly Transformer (AT) [178] are employed as comparison methods. The test results on Yahoo and KPI datasets are shown in Table V. The authors of SR do not provide open-source code, so we use the anomaly detection results (F1, P, and R) reported in the original SR article [167] for comparison. Also, the "-" in the AT method indicates that the metric could not be obtained due to excessive computation time.

As shown in Table V, on the traditional F1, P, and R metrics, TS2Vec and AT achieve the best performance compared to other baselines. However, on the other six metrics, the overall performance of DOUNT is better than TS2Vec. The above results suggest that there are still challenges in designing suitable PTMs over traditional methods for the latest evaluation metrics on Yahoo and KPI datasets. The test results on the UCR anomaly detection archive in Table VI indicate that TS2Vec outperforms DOUNT on the F1, P, F1-PA-10, F1-PA-50, and F1-PA-90 metrics, while DOUNT excels in the other six metrics. Overall, the consistency-based DCdetector achieves the best average performance across all twelve metrics. Due to space constraints, the detailed test results of Table VI and results for seven multivariate datasets are provided in Appendix C-D, available online. For the seven multivariate datasets, DCdetector performs best on five datasets, while TS2Vec and GPT4TS each lead in one dataset. Notably, DCdetector uses a patch strategy similar to PatchTST to preprocess raw time series data before inputting it into the Transformer for pre-training. In short, the patch-based strategy and Transformer framework present promising avenues for further exploration in pre-training for time series anomaly detection tasks.

V. FUTURE DIRECTIONS

A. Large-Scale Time Series Datasets

Large-scale benchmark datasets are critical to the success of PTMs in CV and NLP. Most existing TS-PTMs are pre-trained on datasets from archives such as UCR [163] and UEA [164], most of which have small sample sizes (only a few thousand or even hundreds of samples). Although these time-series benchmark datasets have contributed significantly to the development of the time-series community, it is challenging to utilize them to pre-train deep learning models due to limitations of sample sizes and generality.

TABLE VI
COMPARISON OF OPTIMAL DATASET COUNT FOR TIME-SERIES ANOMALY DETECTION IN THE UCR ANOMALY DETECTION ARCHIVE WITH 250 DATASETS

Models	F1	P	R	F1-PA-10	F1-PA-50	F1-PA-90	Aff-P	Aff-R	R_A_R	R_A_P	V_ROC	V_PR	Avg. Count
SPOT	7	10	10	68	73	70	16	6	4	4	4	4	23.00
DSPOT	6	9	12	**71**	**92**	**77**	12	9	5	7	5	7	26.00
LSTM-VAE	21	19	**169**	49	31	34	10	**110**	53	**84**	52	78	59.17
DONUT	21	18	**169**	46	28	31	7	87	33	79	31	66	51.33
AT	23	34	17	-	-	-	81	4	20	9	19	10	24.11
TS2Vec	73	67	108	62	82	74	13	28	14	22	13	19	47.92
TimesNet	23	36	19	-	-	-	54	8	13	6	15	6	20.00
GPT4TS	16	22	17	-	-	-	**83**	4	13	5	9	5	19.33
DCdetector	**103**	**99**	146	59	49	69	6	45	**127**	66	**134**	**87**	**82.50**

F1-PA-10, F1-PA-50, and F1-PA-90 represent the F1-scores for PA%K [174] with k set to 10, 50, and 90, respectively. "-" indicates that the metric could not be obtained due to excessive computation time.
The best results are in bold and underlined.

Recently, Yang et al. [24] proposed a TS-PTM called Voice2Series that pre-trains deep learning models using a large-scale sequence dataset from the speech domain. Then, the pre-training models are transferred to the general time-series datasets (UCR time-series archive) through model reprogramming and label mapping. Experimental results indicate that pre-training with large-scale sequence datasets in the speech domain can achieve state-of-the-art on the time-series classification task. Further, several studies [103], [104], [133] have constructed large-scale time series datasets by integrating sequences from various existing time series datasets across multiple domains. These synthesized datasets have been effectively employed for the purpose of time series pre-training. However, the construction of large-scale generic time-series datasets like ImageNet [70] is a crucial focus, which will significantly facilitate further research on TS-PTMs.

B. Inherent Properties of Time Series

Time-series representation learning has attracted much attention in recent years. Existing studies have explored the inherent properties of time series for representation learning, such as using CNNs to capture multi-scale dependencies, RNNs to model temporal dependencies, and Transformers to model long-term temporal dependencies. Also, the context-dependencies and frequency domain (or seasonal-trend [152]) information of time series have been explored in recent contrastive learning studies. Although mining the inherent properties of time series can learn representations beneficial for downstream TSM tasks, the transferability of time series from different domains remains weakly interpretable.

Due to the inherent properties (i.e., frequency domain information) of time series, the pre-training techniques applied to image data are difficult to transfer directly to time series. Compared to NLP, TS-PTMs are challenging to learn universal time-series representations due to the absence of a large-scale unified semantic sequence dataset. For example, each word in a text sequence dataset has similar semantics in different sentences with high probability. Therefore, the word embeddings learned by the model can transfer knowledge across different text sequence data scenarios. However, time series datasets are challenging to obtain subsequences (corresponding to words in text sequences) that have consistent semantics across scenarios, making it difficult to transfer the knowledge learned by the model. Hence, exploiting the inherent properties of time series to mine transferable segments in time series data is a challenge for future research on TS-PTMs.

C. Deep Learning Models in Time Series

Recently, deep learning models such as CNNs and Transformers [165], [179] have garnered significant attention in time series mining. In particular, Transformer-based models have achieved excellent pre-training performance in fields like NLP and CV, prompting their exploration in time series studies [103]. Leveraging multi-head attention mechanisms to capture long-term dependencies, Transformers have been successfully employed to design large foundational pre-training models for time series forecasting tasks [104], [105]. However, there are relatively few Transformer models with competitive advantages for time series classification, possibly because this task focuses more on capturing discriminative subseries (e.g., shapelets) or multiscale features [75]. For classification tasks, the superior performance of consistency-based PTMs with TCNs and patching-based strategies on UCR and UEA archives validates their advantages. Furthermore, our experimental results across various downstream tasks suggest that the patch strategy [123] and the iTransformer architecture [172] offer promising research avenues.

Although significant work has applied transformers to time series pre-training [104], [123], [133], some CNN-based models also deserve attention [180]. Notably, Eldele et al. [131] demonstrated that their CNN-based model outperforms existing Transformer-based approaches on various time series downstream tasks. Additionally, the Mamba model, based on state space models, shows strong sequence learning capabilities for pre-training [181], outperforming Transformers in many CV domain tasks [182]. Consequently, researchers in time series analysis are increasingly exploring Mamba for time series mining [183], [184]. Compared to these deep learning models, GNNs offer the ability to learn spatio-temporal dependencies in time series pre-training, especially when combined with models like Transformers [128]. In summary, selecting the appropriate deep learning architectures will be crucial for future designs of time series pre-training models.

D. Adversarial Attacks on Time Series

Adversarial example attacks have recently received extensive attention in various fields because they have significant

security risks. Naturally, scholars in the time-series domain have also begun to consider the impact of adversarial sample attacks on time-series models [185], [186]. For example, Karim et al. [187] utilized a distilled model as a surrogate that simulated the behavior of the attacked time-series classification model. Then, an adversarial transformation network is applied to the distilled model to generate time-series adversarial examples. Experiments on 42 UCR datasets show they are vulnerable to adversarial examples.

Adversarial examples are generated by adding perturbations to the original examples to cross the classification boundaries of the classifier. In general, adding random perturbations is difficult to generate adversarial examples. Also, adversarial examples are not easy to generate when each cluster is far from the classification boundary. Recently, Hendrycks et al. [188] found that self-supervised learning could effectively improve the robustness of deep learning models to adversarial examples. Therefore, improving the robustness of time series models to adversarial examples by utilizing TS-PTMs is a direction worth exploring.

E. Pre-Training Models for Time-Series Noisy Labels

The acquisition cost of large-scale labeled datasets is very expensive. Therefore, various low-cost surrogate strategies are provided to collect labels automatically. For example, many weakly labeled images can be collected with the help of search engines and crawling algorithms. In the time-series domain, Castellani et al. [118] employed sensor readings to generate labels for time series. Although these strategies enable obtaining large-scale labeled data, they also inevitably lead to label noise. Training deep learning models efficiently with noisy labels is challenging since deep learning models have a high capacity for fitting noisy labels [189]. To this end, many studies on learning with label noise [190] have emerged. As an effective representation learning method, PTMs can be effectively employed to solve the problem of noisy labels [191], which has been studied in CV. However, only a few studies are investigating the time-series noisy label [192], and PTMs for time-series noisy labels have not been studied.

F. Pre-Trained LLMs for Time-Series Mining

Pre-trained LLMs have achieved significant success in NLP [136], particularly with GPT-based models like ChatGPT, which are widely used in real life. Scholars [38], [40], [101] have demonstrated that LLMs can be effectively applied to time series analysis, with GPT4TS [99] showing for the first time that combining the patching mechanism and fine-tuning some parameters of the GPT2 model can yield strong performance across various time series tasks. However, the reasons behind the effectiveness of fine-tuning LLMs directly on time series data require further investigation. Notably, Tan et al. [193] found that pre-trained LLMs do not outperform models trained from scratch in time series forecasting tasks, and a simple model with patching and attention as an encoder can achieve comparable results.

Unlike approaches that directly fine-tune LLMs on time series tasks, we tend to think that the pre-training and fine-tuning paradigm using time series datasets for designing TS-PTMs will remain the dominant research approach [103], [104]. This paradigm aligns with human learning processes and has been validated in CV and NLP fields. However, the potential of LLMs in time series mining should not be overlooked. To better leverage LLMs for time series analysis, converting time series into textual descriptions for fine-tuning might more effectively transfer the knowledge acquired during LLM pre-training to time series tasks [194]. Also, integrating textual descriptions with the time series data for multi-modal pre-training [195], [196] is a promising direction worth exploring.

VI. Conclusion

In this survey, we provide a systematic review and analysis of the development of TS-PTMs. In the early research on TS-PTMs, related studies were mainly based on CNN and RNN models for transfer learning on PTMs. In recent years, Transformer-based, consistency-based models and patching-based strategies have achieved remarkable performance in time-series downstream tasks, and have been utilized for time series pre-training. Hence, we conducted a large-scale experimental analysis of existing TS-PTMs, transfer learning strategies, Transformer-based time series methods, and related representative methods on the three main tasks of time-series classification, forecasting, and anomaly detection. The experimental results suggest that LLM-based fine-tuning PTMs, when combined with patching strategies and Transformer-based models, hold significant potential for time-series classification and forecasting tasks. Additionally, consistency-based PTMs using patching strategies demonstrate promising results for time-series anomaly detection. Meanwhile, the pre-training strategy involving the selection of an appropriate deep learning model, such as CNNs or Mamba, represents a promising direction for the future development of TS-PTMs.

Acknowledgment

The author wants to thank Professor Eamonn Keogh from UCR and all the people who have contributed to the UCR&UEA time series archives and other time series datasets. The authors would like to thank Professor Garrison W. Cottrell from UCSD, and Chuxin Chen, Xidi Cai, Yu Chen, and Peitian Ma from SCUT for the helpful suggestions.

References

[1] Y. Wu, J. M. Hernández-Lobato, and G. Zoubin, "Dynamic covariance models for multivariate financial time series," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2013, pp. 558–566.
[2] N. Moritz, T. Hori, and J. Le, "Streaming automatic speech recognition with the transformer model," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6074–6078.
[3] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2891–2900.
[4] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3681–3700, Aug. 2022.
[5] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 04, pp. 1544–1561, Apr. 2022.

[6] T.-C. Fu, "A review on time series data mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, 2011.

[7] G. Li, B. K. K. Choi, J. Xu, S. S. Bhowmick, K.-P. Chun, and G. L. Wong, "Efficient shapelet discovery for time series classification," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 03, pp. 1149–1163, Mar. 2022.

[8] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, 2018.

[9] M. H. Tahan, M. Ghasemzadeh, and S. Asadi, "Development of fully convolutional neural networks based on discretization in time series classification," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6827–6838, Jul. 2023.

[10] R. Sen, H.-F. Yu, and I. S. Dhillon, "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–10.

[11] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.

[12] Q. Wen et al., "Time series data augmentation for deep learning: A survey," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, 2021, pp. 4653–4660.

[13] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *PLoS One*, vol. 16, no. 7, 2021, Art. no. e0254841.

[14] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[15] Z. Liu, Q. Ma, P. Ma, and L. Wang, "Temporal-frequency co-training for time series semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 8923–8931.

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[17] L. Yang, S. Hong, and L. Zhang, "Spectral propagation graph network for few-shot time series classification," 2022, arXiv *2202.04769*.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[19] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021, arXiv *2111.06377*.

[22] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, pp. 1872–1897, 2020.

[23] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," in *Proc. 2018 IEEE Int. Conf. Big Data*, 2018, pp. 1367–1376.

[24] C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, "Voice2Series: Reprogramming acoustic models for time series classification," in *Proc. 38th Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 11808–11819.

[25] P. Malhotra, V. Tv, L. Vig, P. Agarwal, and G. Shroff, "TimeNet: Pretrained deep recurrent neural network for time series classification," 2017, arXiv *1706.08838*.

[26] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 2114–2124.

[27] S. Deldari, H. Xue, A. Saeed, J. He, D. V. Smith, and F. D. Salim, "Beyond just vision: A review on self-supervised representation learning on multimodal and temporal data," 2022, arXiv *2206.02353*.

[28] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–16.

[29] W. Zhang, L. Yang, S. Geng, and S. Hong, "Cross reconstruction transformer for self-supervised time series representation learning," 2022, arXiv *2205.09928*.

[30] R. Ye and Q. Dai, "A novel transfer learning framework for time series forecasting," *Knowl.-Based Syst.*, vol. 156, pp. 74–99, 2018.

[31] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 1597–1607.

[33] H. Zhang, J. Wang, Q. Xiao, J. Deng, and Y. Lin, "SleepPriorCL: Contrastive representation learning with prior knowledge-based positive mining and adaptive temperature for sleep staging," 2021, arXiv *2110.09966*.

[34] N. Laptev, J. Yu, and R. Rajagopal, "Reconstruction and regression loss for time-series transfer learning," in *Proc. Special Int. Group Knowl. Discov. Data Mining 4th Workshop Mining Learn. Time Ser.*, 2018, pp. 1–8.

[35] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Label-efficient time series representation learning: A review," *IEEE Trans. Artif. Intell.*, early access, Jul. 17, 2024, doi: 10.1109/TAI.2024.3430236.

[36] K. Zhang et al., "Self-supervised learning for time series analysis: Taxonomy, progress, and prospects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 10, pp. 6775–6794, Oct. 2024.

[37] Q. Meng, H. Qian, Y. Liu, Y. Xu, Z. Shen, and L. Cui, "Unsupervised representation learning for time series: A review," 2023, arXiv *2308.01578*.

[38] M. Jin et al., "Large models for time series and spatio-temporal data: A survey and outlook," 2023, arXiv *2310.10196*.

[39] M. Jin et al., "Position paper: What can large language models tell us about time series analysis," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–17.

[40] Y. Liang et al., "Foundation models for time series analysis: A tutorial and survey," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2024, pp. 6555–6565.

[41] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discov.*, vol. 33, no. 4, pp. 917–963, 2019.

[42] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos. Trans. Roy. Soc. A*, vol. 379, no. 2194, 2021, Art. no. 20200209.

[43] B. Lafabregue, J. Weber, P. Gançarski, and G. Forestier, "End-to-end deep representation learning for time series clustering: A comparative study," *Data Mining Knowl. Discov.*, vol. 36, pp. 29–81, 2022.

[44] F. Liu et al., "Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 06, pp. 2626–2640, Jun. 2022.

[45] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–33, 2021.

[46] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional recurrent imputation for time series," 2018, arXiv *1805.10572*.

[47] M. C. De Souto, I. G. Costa, D. S. De Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: A comparative study," *BMC Bioinf.*, vol. 9, no. 1, pp. 1–14, 2008.

[48] J. de Jong et al., "Deep learning for clustering of multivariate clinical patient trajectories with missing values," *GigaScience*, vol. 8, no. 11, 2019, Art. no. giz134.

[49] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, "Time series extrinsic regression: Predicting numeric values from time series data," *Data Mining Knowl. Discov.*, vol. 35, no. 3, pp. 1032–1060, 2021.

[50] Q. Xu, K. Wu, M. Wu, K. Mao, X. Li, and Z. Chen, "Reinforced knowledge distillation for time series regression," *IEEE Trans. Artif. Intell.*, vol. 5, no. 6, pp. 3184–3194, Jun. 2024.

[51] B. Li et al., "DifFormer: Multi-resolutional differencing transformer with dynamic ranging for time series analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13586–13598, Nov. 2023.

[52] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, "Monash University, UEA, UCR time series extrinsic regression archive," 2020, arXiv *2006.10996*.

[53] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, arXiv *1412.3555*.

[54] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[55] N. Muralidhar, S. Muthiah, and N. Ramakrishnan, "DyAt nets: Dynamic attention networks for state forecasting in cyber-physical systems," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3180–3186.

[56] Q. Ma, J. Zheng, S. Li, and G. W. Cottrell, "Learning representations for time series clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3781–3791.

[57] J. Gu et al., "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, 2018.

[58] M. Liu et al., "SCINet: Time series modeling and forecasting with sample convolution and interaction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 5816–5828.

[59] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," 2016, arXiv *1603.06995*.

[60] K. Kashiparekh, J. Narwariya, P. Malhotra, L. Vig, and G. Shroff, "ConvTimeNet: A pre-trained deep convolutional neural network for time series classification," in *Proc. 2019 Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

[61] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, arXiv *1803.01271*.

[62] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[63] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, 2020.

[64] J. Xu et al., "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–12.

[65] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," 2022, arXiv *2201.12740*.

[66] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[67] Z. Cheng et al., "Time2Graph: Bridging time series and graph representation learning via multiple attentions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 2078–2090, Feb. 2023.

[68] Y. Wang et al., "Graph-aware contrasting for multivariate time-series classification," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 15725–15734.

[69] D. Zha, K.-H. Lai, K. Zhou, and X. Hu, "Towards similarity-aware time-series classification," in *Proc. 2022 SIAM Int. Conf. Data Mining*, SIAM, 2022, pp. 199–207.

[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[71] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," in *Proc. Artif. Intell. Res. Develop.*, 2018, pp. 120–129.

[72] Y. Chen et al., "The UCR time series classification archive," Jul. 2015. [Online]. Available: www.cs.ucr.edu/eamonn/time_series_data/

[73] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. 2017 Int. Joint Conf. Neural Netw.*, 2017, pp. 1578–1585.

[74] H. Ismail Fawaz et al., "InceptionTime: Finding alexnet for time series classification," *Data Mining Knowl. Discov.*, vol. 34, no. 6, pp. 1936–1962, 2020.

[75] W. Tang, G. Long, L. Liu, T. Zhou, M. Blumenstein, and J. Jiang, "Omni-scale CNNs: A simple and effective kernel size configuration for time series classification," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–17.

[76] F. Li, K. Shirahama, M. A. Nisar, X. Huang, and M. Grzegorzek, "Deep transfer learning for time series data based on sensor modality classification," *Sensors*, vol. 20, no. 15, pp. 4271–4296, 2020.

[77] Z. Yue et al., "TS2Vec: Towards universal representation of time series," in *Proc. Advance. Artif. Intell. Conf. Artif. Intell.*, 2022, pp. 8980–8987.

[78] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 947–956.

[79] A. Meiseles and L. Rokach, "Source model selection for deep learning in the time series domain," *IEEE Access*, vol. 8, pp. 6190–6200, 2020.

[80] R. Mutegeki and D. S. Han, "Feature-representation transfer learning for human activity recognition," in *Proc. 2019 Int. Conf. Inf. Commun. Technol. Convergence*, 2019, pp. 18–20.

[81] R. Cai et al., "Time series domain adaptation via sparse associative structure alignment," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6859–6867.

[82] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1768–1778.

[83] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," 2017, arXiv *1702.05374*.

[84] C. Chen et al., "HoMM: Higher-order moment matching for unsupervised domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3422–3429.

[85] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 5423–5432.

[86] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2272–2281.

[87] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2720–2729.

[88] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, "Stratified transfer learning for cross-domain activity recognition," in *Proc. 2018 IEEE Int. Conf. Pervasive Comput. Commun.*, 2018, pp. 1–10.

[89] Z. Li, R. Cai, T. Z. Fu, and K. Zhang, "Transferable time-series forecasting under causal conditional shift," 2021, arXiv *2111.03422*.

[90] Q. Liu and H. Xue, "Adversarial spectral kernel matching for unsupervised time series domain adaptation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 2744–2750.

[91] M. A. A. H. Khan, N. Roy, and A. Misra, "Scaling human activity recognition via deep learning-based domain adaptation," in *Proc. 2018 IEEE Int. Conf. Pervasive Comput. Commun.*, 2018, pp. 1–9.

[92] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. B. Fox, "Neural granger causality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4267–4279, Aug. 2022.

[93] M. Ragab, E. Eldele, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Self-supervised autoregressive domain adaptation for time series data," 2021, arXiv *2111.14834*.

[94] P. R. d. O. da Costa, A. Akçay, Y. Zhang, and U. Kaymak, "Remaining useful lifetime prediction via deep domain adaptation," *Rel. Eng. Syst. Saf.*, vol. 195, 2020, Art. no. 106682.

[95] W. Garrett, R. D. Janardhan, and J. C. Diane, "CALDA: Improving multi-source time series domain adaptation with contrastive adversarial learning," 2021, arXiv *2109.14778*.

[96] Z. Li et al., "Causal mechanism transfer network for time series domain adaptation in mechanical systems," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 2, pp. 1–21, 2021.

[97] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–14.

[98] C.-H. H. Yang et al., "Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6523–6527.

[99] T. Zhou et al., "One fits all: Power general time series analysis by pretrained LM," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 43322–43355.

[100] D. Cao et al., "TEMPO: Prompt-based generative pre-trained transformer for time series forecasting," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 1–33.

[101] M. Jin et al., "Time-LLM: Time series forecasting by reprogramming large language models," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 1–24.

[102] X. Jiang, R. Missel, Z. Li, and L. Wang, "Sequential latent variable models for few-shot high-dimensional time-series forecasting," in *Proc. 11th Int. Conf. Learn. Representations*, 2023, pp. 1–21.

[103] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: Generative pre-trained transformers are large time series models," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–31.

[104] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified training of universal time series forecasting transformers," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–25.

[105] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–21.

[106] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Transfer learning with graph neural networks for short-term highway traffic forecasting," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 10367–10374.

[107] Y. Du et al., "AdaRNN: Adaptive learning and forecasting of time series," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 402–411.

[108] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, arXiv *1807.03748*.

[109] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 3465–3469.

[110] E. Eldele et al., "Time-series representation learning via temporal and contextual contrasting," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, 2021, pp. 2352–2359.
[111] O. Vinyals et al., "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.
[112] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1126–1135.
[113] B. Lu, X. Gan, W. Zhang, H. Yao, L. Fu, and X. Wang, "Spatio-temporal graph few-shot learning with cross-city knowledge transfer," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 1162–1172.
[114] R. Wang, R. Walters, and R. Yu, "Meta-learning dynamics forecasting using task inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 21640–21653.
[115] T. Iwata and A. Kumagai, "Few-shot learning for time-series forecasting," 2020, arXiv *2009.14379*.
[116] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "Meta-learning framework with applications to zero-shot time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9242–9250.
[117] L. Brinkmeyer, R. R. Drumond, J. Burchert, and L. Schmidt-Thieme, "Few-shot forecasting of time-series with heterogeneous channels," 2022, arXiv *2204.03456*.
[118] A. Castellani, S. Schmitt, and B. Hammer, "Estimating the electrical power output of industrial devices with end-to-end time-series classification in the presence of label noise," in *Proc. Eur. Conf. Mach. Learn.*, 2021, pp. 1–32.
[119] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.
[120] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.
[121] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," 2019, arXiv *1910.05453*.
[122] Q. Ma, C. Chen, S. Li, and G. W. Cottrell, "Learning representations for incomplete time series clustering," in *Proc. Advance. Artif. Intell. Conf. Artif. Intell.*, 2021, pp. 8837–8846.
[123] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. 11th Int. Conf. Learn. Representations*, 2023, pp. 1–24.
[124] J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long, "SimMTM: A simple pre-training framework for masked time-series modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 29996–30025.
[125] K. Zhang, C. Li, and Q. Yang, "TriD-MAE: A generic pre-trained model for multivariate time series with missing values," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 3164–3173.
[126] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio Word2Vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," 2016, arXiv *1603.00982*.
[127] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–95, 2016.
[128] Z. Shao, Z. Zhang, F. Wang, and Y. Xu, "Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 1567–1577.
[129] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.
[130] J. Dong et al., "TimeSiam: A pre-training framework for siamese time-series modeling," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–25.
[131] E. Eldele, M. Ragab, Z. Chen, M. Wu, and X. Li, "TSLANet: Rethinking transformers for time series representation learning," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–20.
[132] Y. Zhang, M. Liu, S. Zhou, and J. Yan, "UP2ME: Univariate pre-training to multivariate fine-tuning as a general-purpose framework for multivariate time series analysis," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–24.
[133] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, "Moment: A family of open time-series foundation models," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–38.
[134] R. R. Chowdhury, X. Zhang, J. Shang, R. K. Gupta, and D. Hong, "TARNet: Task-aware reconstruction for time-series transformer," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, Washington, DC, USA, 2022, pp. 14–18.
[135] A. T. Liu, S.-W. Li, and H.-Y. Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2351–2366, 2021.
[136] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS: A survey of transformer-based pretrained models in natural language processing," 2021, arXiv *2108.05542*.
[137] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[138] P. Shi, W. Ye, and Z. Qin, "Self-supervised pre-training for time series classification," in *Proc. 2021 Int. Joint Conf. Neural Netw.*, 2021, pp. 1–8.
[139] L. Hou, Y. Geng, L. Han, H. Yang, K. Zheng, and X. Wang, "Masked token enabled pre-training: A task-agnostic approach for understanding complex traffic flow," *IEEE Trans. Mobile Comput.*, early access, Apr. 18, 2024, doi: 10.1109/TMC.2024.3390941.
[140] L. Zhao, M. Gao, and Z. Wang, "ST-GSP: Spatial-temporal global semantic representation learning for urban flow prediction," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 1443–1451.
[141] I. Padhi et al., "Tabular transformers for modeling multivariate time series," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 3565–3569.
[142] S. M. Shankaranarayana and D. Runje, "Attention augmented convolutional transformer for tabular time-series," in *Proc. 2021 Int. Conf. Data Mining Workshops*, 2021, pp. 537–541.
[143] A. T. Liu, S.-W. Yang, P.-H. Chi, P.-C. Hsu, and H.-Y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6419–6423.
[144] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3024–3033.
[145] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "DCdetector: Dual attention contrastive representation learning for time series anomaly detection," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2023, pp. 3033–3045.
[146] M. Hu et al., "Self-supervised pre-training for robust and generic spatial-temporal representations," in *Proc. 2023 IEEE Int. Conf. Data Mining*, 2023, pp. 150–159.
[147] J. Liu and S. Chen, "TimesURL: Self-supervised contrastive learning for universal time series representation learning," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 13918–13926.
[148] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–9.
[149] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.
[150] H. Fan, F. Zhang, and Y. Gao, "Self-supervised time series representation learning by inter-intra relational reasoning," 2020, arXiv *2011.13548*.
[151] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–17.
[152] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–18.
[153] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time series change point detection with self-supervised contrastive predictive coding," in *Proc. Web Conf.*, 2021, pp. 3124–3135.
[154] D. Luo et al., "Information-aware time series meta-contrastive learning," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–23.
[155] A. Hyvarinen and H. Morioka, "Unsupervised feature extraction by time-contrastive learning and nonlinear ICA," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.
[156] H. M. Aapo Hyvarinen, "Nonlinear ICA of temporally dependent stationary sources," in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2017, pp. 460–469.
[157] L. Yang, S. Hong, and L. Zhang, "Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion," 2022, arXiv *2202.04770*.
[158] Q. Wen et al., "Time series data augmentation for deep learning: A survey," 2020, arXiv *2002.12478*.

[159] M. T. Nonnenmacher, L. Oldenburg, I. Steinwart, and D. Reeb, "Utilizing expert features for contrastive learning of time-series representations," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 16969–16989.
[160] Y. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–22.
[161] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–17.
[162] Y.-A. Chung and J. Glass, "Speech2Vec: A sequence-to-sequence framework for learning word embeddings from speech," 2018, arXiv *1803.08976*.
[163] H. A. Dau et al., "The UCR time series archive," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, Nov. 2019.
[164] A. Bagnall et al., "The UEA multivariate time series classification archive 2018," 2018, arXiv *1811.00075*.
[165] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," in *Proc. 11th Int. Conf. Learn. Representations*, 2023, pp. 1–23.
[166] Y. B. Nikolay Laptev and S. Amizadeh, "A benchmark dataset for time series anomaly detection," 2015. [Online]. Available: https://yahooresearch.tumblr.com/post/114590420346/a-benchmark-dataset-for-time-series-anomaly
[167] H. Ren et al., "Time-series anomaly detection service at Microsoft," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 3009–3017.
[168] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2421–2429, Mar. 2023.
[169] A. Huet, J. M. Navarro, and D. Rossi, "Local evaluation of time series anomaly detection algorithms," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 635–645.
[170] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. Elmore, and M. J. Franklin, "Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection," *Proc. VLDB Endowment*, vol. 15, no. 11, pp. 2774–2787, 2022.
[171] S. Li et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.
[172] Y. Liu et al., "iTransformer: Inverted transformers are effective for time series forecasting," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 1–25.
[173] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 11121–11128.
[174] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, "Towards a rigorous evaluation of time-series anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7194–7201.
[175] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 1067–1075.
[176] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.
[177] H. Xu et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proc. 2018 World Wide Web Conf.*, 2018, pp. 187–196.
[178] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–20.
[179] Y. Chen, K. Ren, Y. Wang, Y. Fang, W. Sun, and D. Li, "ContiFormer: Continuous-time transformer for irregular time series modeling," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 47143–47175.
[180] D. Luo and X. Wang, "ModernTCN: A modern pure convolution structure for general time series analysis," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 1–43.
[181] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2023, arXiv *2312.00752*.
[182] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–11.
[183] B. N. Patro and V. S. Agneeswaran, "SiMBA: Simplified mamba-based architecture for vision and multivariate time series," 2024, arXiv *2403.15360*.
[184] Z. Wang et al., "Is mamba effective for time series forecasting?," 2024, arXiv *2403.11144*.
[185] T. Wu, X. Wang, S. Qiao, X. Xian, Y. Liu, and L. Zhang, "Small perturbations are enough: Adversarial attacks on time series prediction," *Inf. Sci.*, vol. 587, pp. 794–812, 2022.
[186] L. Liu, Y. Park, T. N. Hoang, H. Hasson, and L. Huan, "Robust multivariate time-series forecasting: Adversarial attacks and defense mechanisms," in *Proc. 11th Int. Conf. Learn. Representations*, 2023, pp. 1–18.
[187] F. Karim, S. Majumdar, and H. Darabi, "Adversarial attacks on time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3309–3320, Oct. 2021.
[188] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.
[189] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8135–8153, Nov. 2023.
[190] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–11.
[191] J. Li, R. Socher, and S. C. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–14.
[192] Z. Liu, P. Ma, D. Chen, W. Pei, and Q. Ma, "Scale-teaching: Robust multi-scale training for time series classification with noisy labels," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 33726–33757.
[193] M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, "Are language models actually useful for time series forecasting?," 2024, arXiv *2406.16964*.
[194] H. Xue and F. D. Salim, "PromptCast: A new prompt-based learning paradigm for time series forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6851–6864, Nov. 2024.
[195] Z. Liu, W. Pei, D. Lan, and Q. Ma, "Diffusion language-shapelets for semi-supervised time-series classification," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 14079–14087.
[196] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 8748–8763.

Qianli Ma (Member, IEEE) received the PhD degree in computer science from the South China University of Technology, Guangzhou, China, in 2008. He is a professor with the School of Computer Science and Engineering, South China University of Technology. From 2016 to 2017, he was a visiting scholar with the University of California, San Diego. His current research interests include machine learning algorithms, data-mining methodologies, and their applications. He is an associate editor of *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. He has been recognized among the World's Top 2% Scientists for 2023 and 2024.

Zhen Liu received the bachelor's degree in software engineering from South-Central Minzu University, Wuhan, China, in 2018. He is currently working toward the PhD degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include machine learning, deep learning, and time-series analysis.

Zhenjing Zheng received the bachelor's and master's degrees in computer science from the South China University of Technology, Guangzhou, China, in 2019 and 2022, respectively. His current research interests include machine learning, deep learning, and time-series analysis.

Ziyang Huang received the bachelor's degree from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, in 2023. His current research interests include machine learning, deep learning, and time-series analysis.

Siying Zhu received the bachelor's degree in computer science from Southwest University, Chongqing, China, in 2020, and the master's degree from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, in 2023. Her current research interests include machine learning, deep learning, and time-series forecasting.

Zhongzhong Yu received the bachelor's and master's degrees in computer science from the South China University of Technology, Guangzhou, China, in 2019 and 2022, respectively. His current research interests include machine learning, deep learning, and time-series anomaly detection.

James T. Kwok (Fellow, IEEE) received the PhD degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 1996. He is currently a professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. His research interests include machine learning, deep learning, and artificial intelligence. He is serving/served as an associate editor for the *IEEE Transactions on Neural Networks and Learning Systems*, *Neural Networks*, *Neurocomputing*. He is also serving/served as senior area chairs/area chairs of major conferences including NeurIPS, ICML, ICLR, IJCAI, and AAAI. He is on the IJCAI Board of Trustees. He is recognized as the Most Influential Scholar Award Honorable Mention for "outstanding and vibrant contributions to the field of AAAI/IJCAI between 2009 and 2019". He is the IJCAI-2025 program chair.