Large Language Model Guided Knowledge Distillation for Time Series Anomaly Detection

Chen Liu[1], **Shibo He**[1*], **Qihang Zhou**[1], **Shizhong Li**[1] and **Wenchao Meng**[1]
[1]Zhejiang University
{liu777ch, s18he, zqhang, lisz, wmengzju}@zju.edu.cn

Abstract

Self-supervised methods have gained prominence in time series anomaly detection due to the scarcity of available annotations. Nevertheless, they typically demand extensive training data to acquire a generalizable representation map, which conflicts with scenarios of a few available samples, thereby limiting their performance. To overcome the limitation, we propose **AnomalyLLM**, a knowledge distillation-based time series anomaly detection approach where the student network is trained to mimic the features of the large language model (LLM)-based teacher network that is pretrained on large-scale datasets. During the testing phase, anomalies are detected when the discrepancy between the features of the teacher and student networks is large. To circumvent the student network from learning the teacher network's feature of anomalous samples, we devise two key strategies. 1) Prototypical signals are incorporated into the student network to consolidate the normal feature extraction. 2) We use synthetic anomalies to enlarge the representation gap between the two networks. AnomalyLLM demonstrates state-of-the-art performance on 15 datasets, improving accuracy by at least 14.5% in the UCR dataset.

1 Introduction

Time series anomaly detection (TSAD) aims to identify abnormal data whose patterns deviate from the majority of the data [Blázquez-García *et al.*, 2021]. It plays critical roles in applications such as industrial fault diagnosis, network intrusion detection, and health monitoring [Chen *et al.*, 2021].

The primary challenge for TSAD lies in the laborious process of acquiring annotations [Ruff *et al.*, 2021]. Consequently, most previous works follow the unsupervised setting where no labels are provided, and the majority of the data is assumed to be normal [Audibert *et al.*, 2020]. These methods can be broadly categorized into one-class classification-based methods [Ruff *et al.*, 2018; Shen *et al.*, 2020; Carmona *et al.*, 2021], density estimation-based methods [Dai and Chen, 2022; Zhou *et al.*, 2023a], and self-supervised methods [Jeong *et al.*, 2023]. With the advancement of representation learning, self-supervised methods have garnered growing attention and dominated the field [Zhang *et al.*, 2023]. They employ pretext tasks such as reconstruction [Su *et al.*, 2019; Xu *et al.*, 2021; Tuli *et al.*, 2022], forecasting [Deng and Hooi, 2021], imputation [Chen *et al.*, 2023], and contrastive learning [Yang *et al.*, 2023] to learn a representation map that distinguishes the normal and abnormal samples.

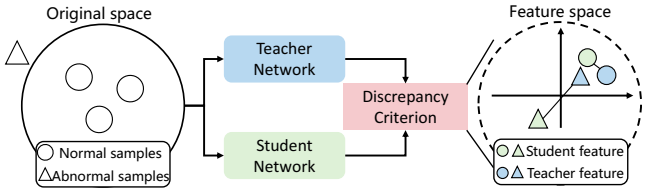


Figure 1: Knowledge distillation-based framework: the discrepancy between outputs of the student and teacher networks is expected to be small on normal samples while large on abnormal samples.

However, learning generalizable representations typically requires a vast amount of training data [Zhang *et al.*, 2023], which conflicts with scenarios of limited available samples, thereby limiting the performance of these self-supervised methods. To overcome this limitation, we introduce AnomalyLLM, a novel approach that integrates knowledge distillation and large language models (LLMs). The fundamental idea is to train a student network to mimic the output of a teacher network that is pretrained on a large-scale dataset. During the testing phase, anomalies are identified when a significant discrepancy exists between the outputs of the student and teacher networks, as shown in Figure 1. To this end, we need to address two key challenges.

How to pretrain the teacher network without large-scale time series datasets? Large-scale datasets for pretraining are abundant in computer vision (CV) and natural language processing (NLP), playing a critical role in generalizable representation learning. However, there is currently a lack of universal large-scale time series datasets, with the largest available dataset being less than 10GB, a size significantly smaller than that in CV and NLP [Godahewa *et al.*, 2021]. Consequently, pretraining the teacher network remains a challenge. Recent studies have explored the modal similarity between

*Corresponding author

language and time series, revealing the remarkable potential of pretrained LLMs in generating time series representations [Zhou *et al.*, 2023b]. LLMs can be fine-tuned on time series data under few-shot [Jin *et al.*, 2023a] or even zero-shot [Gruver *et al.*, 2023] settings. This observation motivates us to use a pretrained LLM as the teacher network. We follow the time series embedding layer with a pretrained LLM, adapting it to generate time series representations.

How to circumvent the student network from 'overlearning' representations produced by the teacher network? We anticipate the discrepancy between the outputs of the student and teacher networks to be small on normal samples but large on abnormal samples [Zhou *et al.*, 2022]. However, given the absence of abnormal samples to enlarge their representation gap, this can easily lead to overlearning of the student network, where the two networks consistently generate similar representations, even for abnormal samples. To circumvent this problem, we implement two designs. First, we incorporate prototypical signals into the student network, enabling its representations to focus more on the historical normal patterns [Song *et al.*, 2023]. Second, we employ data augmentations to produce synthetic anomalies [Sun *et al.*, 2023b], which are used to enlarge the representation discrepancy. Furthermore, the teacher network's representations of original and augmented samples are treated as positive pairs, and a contrastive loss is applied to bring them closer together, serving as a regularization term to encourage the teacher network to capture more general patterns. Comprehensive experiments are conducted to demonstrate the superiority of our method on 9 univariate datasets and 6 multivariate datasets.

The main contributions are summarized as follows:

- As far as we know, AnomalyLLM is the first knowledge distillation-based time series anomaly detection method.
- We devise a teacher network that is adapted from the pretrained LLM, capable of learning a rich generalizable representation for time series after fine-tuning.
- To maintain the discrepancy between the teacher and student networks, we integrate prototypical signals into the student network and design a data augmentation-based training strategy.
- Extensive experiments show that the proposed model achieves SOTA performance on 15 real-world datasets.

2 Related Work

2.1 Time Series Anomaly Detection

Time series anomaly detection plays a pivotal role in various real-world applications [Chen *et al.*, 2021]. Early studies employ statistical methods or machine learning-based methods [Blázquez-García *et al.*, 2021], which fail to describe complex patterns of time series signals. In recent years, with the success of neural networks such as variational autoencoder [Park *et al.*, 2018] and generative adversarial network [Zhou *et al.*, 2019], numerous deep learning-based methods have emerged for time series anomaly detection. These methods can be roughly categorized into one-class classification-based methods [Ruff *et al.*, 2018; Shen *et al.*, 2020; Carmona *et al.*, 2021], density estimation-based methods [Dai and Chen, 2022; Zhou *et al.*, 2023a; Zhou *et al.*, 2024], and self-supervised-based methods [Jeong *et al.*, 2023]. With the rapid development of representation learning, self-supervised methods have dominated the field. Reconstruction is the most usual self-supervised method, where the reconstruction error indicates the outlyingness of the samples [Xu *et al.*, 2021; Tuli *et al.*, 2022; Li *et al.*, 2023; Song *et al.*, 2023]. Forecasting [Deng and Hooi, 2021], imputation [Chen *et al.*, 2023], and contrastive learning [Yang *et al.*, 2023; Wang *et al.*, 2023; Sun *et al.*, 2023b] also emerge as other pretext tasks for self-supervised anomaly detection. While they have achieved SOTA results on various datasets, the small data size used by these self-supervised methods hinders them from learning generalizable representations, thereby limiting their performance [Zhang *et al.*, 2023]. In this paper, we introduce a teacher network adapted from LLM, which has demonstrated a strong ability to generate generalizable time series representations. A student network is trained to mimic the output of the teacher network, and the discrepancy between their outputs serves as the anomaly score in the testing phase.

2.2 Large Language Model

Pretrained foundation models have proven excellent performance in NLP and CV, prompting its progress in time series analysis [Jin *et al.*, 2023b]. Despite the increasing interest in foundation models for time series [Garza and Mergenthaler-Canseco, 2023], it remains a significant challenge due to the limited availability of large-scale datasets. The largest time series dataset is currently less than 10GB, a size smaller than that of NLP datasets [Godahewa *et al.*, 2021]. However, recent studies suggest that pretrained LLMs can be adapted to time series analysis through fine-tuning on time series data [Zhou *et al.*, 2023b]. Gruver *et al.* [Gruver *et al.*, 2023] even argue that a pretrained LLM can serve as a zero-shot time series forecaster, thanks to its capability to model flexible distributions over sequences of numbers. Consequently, various studies leverage LLM for time series analysis, with a predominant focus on forecasting [Jin *et al.*, 2023a] and classification [Sun *et al.*, 2023a]. Notably, GPT4TS [Zhou *et al.*, 2023b] stands as the sole LLM-based time series anomaly detection method, employing an encoder-decoder reconstruction architecture with GPT2 as the encoder. However, the pretrained GPT2's strong generalization ability makes it prone to reconstructing abnormal signals and yielding false negatives. In contrast, we propose a novel knowledge distillation-based method, where the student network will not generalize to those unseen anomalies compared to GPT4TS.

2.3 Knowledge Distillation

Knowledge distillation is proposed by [Hinton *et al.*, 2015], aiming to push the student network to regress the output of the teacher network. It is first employed in vision anomaly detection by [Bergmann *et al.*, 2020]. The fundamental principle is that anomalies are identified when there is a substantial discrepancy between the outputs of the student and teacher networks. While this principle has been widely explored in vision anomaly detection, [Salehi *et al.*, 2021; Zhou *et al.*, 2022], its application in time series remains an unexplored territory. Our method can be seen as the first

attempt to introduce knowledge distillation into time series anomaly detection. Moreover, our method diverges from existing works in CV in two key aspects. First, unlike in CV where large datasets are commonly available for pretraining the teacher network, large time series datasets are scarce, impeding the effective pretraining. In this work, we employ the pretrained LLM as our teacher network and adapt time series signals to features LLM can understand by an input embedding layer. Second, to prevent the student network from overlearning the representation of the teacher network, we propose reminding the student network of the prototypical signals and devising a data augmentation-based training strategy. Through these efforts, we demonstrate that knowledge distillation can provide another approach for time series anomaly detection, which has not been explored previously.

3 Methodology

Given a D-dimension multivariate time series $X = [x_1, x_2, ..., x_L] \in \mathbf{R}^{D\times L}$ of length L, where $x_t \in \mathbf{R}^D$ denotes the data collected at the t-th time step and D denotes the number of variables, we aim to train an anomaly detector. During the testing phase, we utilize the trained detector to predict an unseen multivariate time series $\hat{X} = [\hat{x_1}, \hat{x_2}, ..., \hat{x_{L'}}]$ with $\hat{Y} = [\hat{y_1}, \hat{y_2}, ..., \hat{y_{L'}}]$, where $\hat{y_l} \in \{0, 1\}$ indicates whether anomalies occur at the l-th time step.

3.1 Overall Architecture

Following previous works [Carmona *et al.*, 2021], we partition the entire time series into several time windows of fixed length T. Given a time window $\mathbf{w}_i \in \mathbf{R}^{D\times T}$, we aim to identify whether anomalies occur within it. Our method adopts the knowledge distillation architecture [Bergmann *et al.*, 2020]. The architecture consists of a student network $\phi : \mathbf{R}^{D\times T} \to \mathbf{R}^d$ and a teacher network $\varphi : \mathbf{R}^{D\times T} \to \mathbf{R}^d$, both transforming original signals into D-dimensional vectors. The time window is fed into the two networks, and outputs are denoted as $z_i = \phi(\mathbf{w}_i)$ and $c_i = \varphi(\mathbf{w}_i)$. We anticipate these two representations to be close for normal samples and distant for abnormal samples. The hypersphere classifier loss [Ruff *et al.*, 2020] is utilized to calculate the discrepancy between two representations:

$$\mathcal{L} = -(1 - y_i) \log \ell(z_i, c_i) - y_i \log(1 - \ell(z_i, c_i)), \quad (1)$$

where $\ell(z_i, c_i) = \exp(-\|z_i - c_i\|_2^2)$, and y_i denotes the ground truth label. In the unsupervised setting, all samples are assumed to be normal, and $y_i = 0$.

3.2 Prototype-based Student Network

To prevent the student network from learning overly generalizable representations like the teacher network, we guide it with prototypes. These prototypes represent characteristic segments in the entire time series and are trainable parameters in our method. We select prototypes that closely resemble the input time window to assist in generating the representation.

Prior study [Nie *et al.*, 2022] has demonstrated the effectiveness of channel independence in multivariate time series analysis. Therefore, we select the most similar prototype for each channel. First, we initialize a prototype pool $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_D\}$, where $\mathcal{M}_i = \{\mathbf{m}_i^1, \mathbf{m}_i^2, ..., \mathbf{m}_i^M\}$ represents the collection of M prototypes for the i-th channel. Given an input time window $\mathbf{w}$, we calculate the similarity between each channel and its corresponding prototypes, and then select the most similar prototype:

$$\mathbf{m}_j^s = \max_{i=1,2,...,M} sim(\mathbf{m}_j^i, \mathbf{w}^j), \quad (2)$$

where $\mathbf{w}^j$ is the j-th channel of input, and $sim(\cdot, \cdot) : \mathbf{R}^T \times \mathbf{R}^T \to \mathbf{R}^+$ represents the function to measure similarity. We use cosine similarity in the paper. The selected prototype for the entire time window is denoted as $\mathbf{m}_s$.

Instance normalization [Kim *et al.*, 2021] is employed to mitigate the distribution shift effect and patching [Nie *et al.*, 2022] is utilized to extract local semantic information. Both the input time window and prototype are addressed by these two techniques. It is noteworthy that the hyperparameters of the prototype normalization are the same as those of the input. Next, both the input and prototype are fed into an input embedding layer. The input embedding layer consists of linear probing which extracts the in-patch information and a positional embedding which records the position information of the sequences. The layer generates the input embedding $\mathbf{w}_e$ and prototype embedding $\mathbf{m}_e$.

To incorporate prototypical features into the original time window, we devise a prototype-based Transformer encoder. The traditional Transformer encoder is stacked by blocks, each consisting of an attention layer, a feed-forward layer, and layer normalization [Song *et al.*, 2023]. In this paper, we provide information about prototypes for each attention layer. Specifically, the queries and keys are produced for both prototypes and inputs. The correlations between the i-th patch of the input embedding $\mathbf{w}_e^i$ and other patches are calculated as follows:

$$\begin{aligned} s_{i,t}^w &= \frac{\exp(\langle \mathbf{q}_w^i, \mathbf{k}_m^t \rangle)}{\sum_{j=1}^n \exp(\langle \mathbf{q}_w^i, \mathbf{k}_w^j \rangle) + \sum_{j=1}^n \exp(\langle \mathbf{q}_w^i, \mathbf{k}_m^j \rangle)}, \\ s_{i,t}^m &= \frac{\exp(\langle \mathbf{q}_w^i, \mathbf{k}_m^t \rangle)}{\sum_{j=1}^n \exp(\langle \mathbf{q}_w^i, \mathbf{k}_w^j \rangle) + \sum_{j=1}^n \exp(\langle \mathbf{q}_w^i, \mathbf{k}_m^j \rangle)}, \end{aligned} \quad (3)$$

where $\mathbf{q}_w, \mathbf{q}_m, \mathbf{k}_w, \mathbf{k}_m$ represents the input patch query, prototype patch query, input patch key and prototype patch key, respectively, and $\langle , \rangle$ represents the inner product. n denotes the number of patches in the time window. Next, we calculate the representation of the input as follows:

$$\mathbf{o} = \sum_{t=1}^n s_{i,t}^w \mathbf{v}_w^t + \sum_{t=1}^n s_{i,t}^m \mathbf{v}_m^t, \quad (4)$$

where $\mathbf{v}_w^t, \mathbf{v}_m^t$ represent the value of input and prototype, respectively. The multi-head mechanism is also utilized to capture patterns of different scales. A flattened layer and a linear layer are employed to produce the final representations z.

3.3 LLM-based Teacher Network

The teacher network is expected to produce generalizable representations. Previous works have unveiled the potential of pretrained NLP models such as GPT2 in time series representation generation [Zhou *et al.*, 2023b]. As a result, we devise the teacher network based on the pretrained LLM.

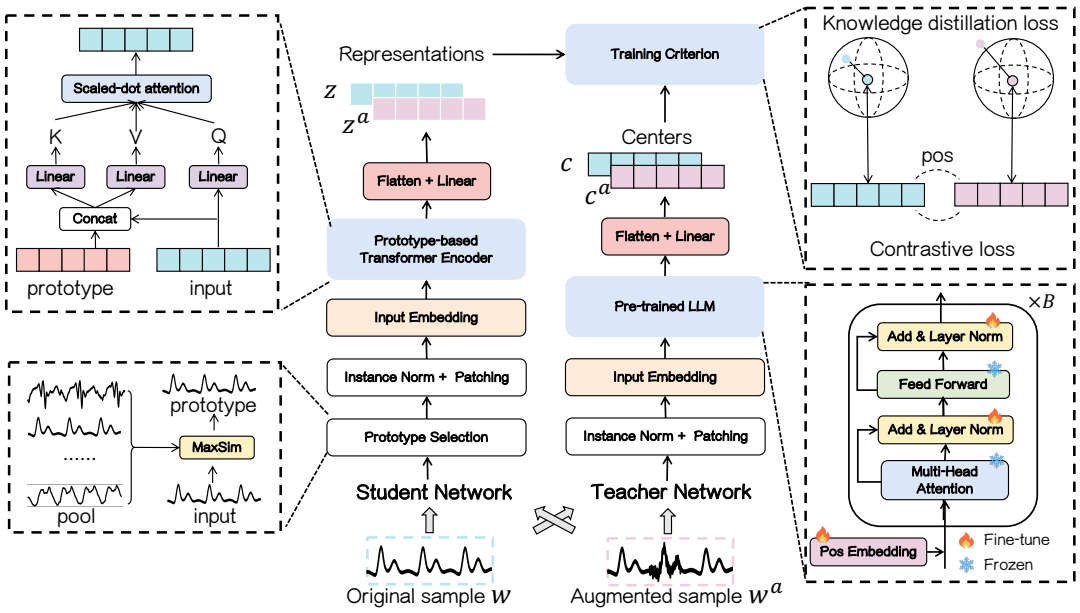


Figure 2: The framework of AnomalyLLM. It consists of three main components: prototype-based student network, LLM-based teacher network, and data augmentation-based training strategy.

The input time series undergoes normalization and patching before being fed into the input embedding layer. Notably, the input embedding layer consists only of linear embedding, as positional embedding is inherent in the pretrained LLM. The linear embeddings in the teacher network and the student network utilize different parameters. The linear embedding in the student network is designed to extract correlations within the time series, while that in the teacher network focuses on transforming original time series signals into a representation comprehensible to the language model.

Next, the preprocessed inputs are fed into a network that retains the positional embedding layer and B self-attention blocks of the pretrained LLM. We use GPT2 in this paper. To preserve the knowledge from pretrained LLM, we freeze the attention layer and the feed-forward layer which are crucial components for sequence modeling. The positional embedding layer and the layer normalization are fine-tuned on the input time series, adapting the LLM to understand the time series representations for anomaly detection. The output of the last self-attention block is fed into a flattened and linear layer to generate the eventual representation c.

3.4 Model Training

We aim to distinguish anomaly representations of the student and teacher networks. Given the absence of anomaly labels under unsupervised settings, we propose a data augmentation-based training strategy.

Firstly, we apply data augmentation to generate synthetic anomalies. Specifically, we randomly select a segment from the entire time window and apply augmentation to this segment. We use augmentation methods including jittering, scaling [Wang *et al.*, 2023], and warping [Sun *et al.*, 2023a]. The synthetic sample for the original sample $\mathbf{w}_i$ is denoted as $\mathbf{w}_i^a$.

Next, both original and synthetic samples are fed into the teacher and student networks. The generated representation pairs of original and synthetic samples are denoted as (z_i, c_i) and (z_i^a, c_i^a), respectively. We push away the representation pair of the synthetic sample while pulling together that of the original samples. This knowledge distillation loss is calculated based on Eq. 1:

$$\mathcal{L}_{kd} = \frac{1}{N}\sum_{i=1}^{N} \|z_i - c_i\|_2^2 - \log(1-\exp(-\|z_i^a - c_i^a\|_2^2)), \quad (5)$$

where N is the total number of training samples.

Additionally, to enable the teacher network to focus on more general patterns and produce representations robust to noise, we consider the teacher's representations of the original and synthetic samples as positive pairs and aim to minimize the distance between them. This negative-sample-free contrastive loss [Wang *et al.*, 2023] is defined as:

$$\mathcal{L}_{ce} = \frac{1}{N}\sum_{i=1}^{N} -\frac{c_i}{\|c_i\|_2} \cdot \frac{c_i^a}{\|c_i^a\|_2}. \quad (6)$$

Then, the complete loss function is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{kd} + \lambda \mathcal{L}_{ce}, \quad (7)$$

where λ is a hyperparameter that controls the weight of knowledge distillation loss and contrastive loss.

During the testing phase, we calculate the anomaly score of the given time window $\mathbf{w}$ according to:

$$A(\mathbf{w}) = \|\phi(\mathbf{w}) - \varphi(\mathbf{w})\|_2^2. \quad (8)$$

4 Experiment

4.1 Datasets

UCR Anomaly Archive (UCR). This archive comprises 250 diverse univariate time series signals spanning various

Dataset	Train	Test	Anomaly Ratio
ABP	1036746	1841461	0.53%
Acceleration	38400	62337	2.45%
AirTemperature	52000	54392	2.86%
ECG	1795083	6047314	0.53%
EPG	119000	410415	1.29%
Gait	1157571	2784520	0.43%
NASA	38500	86296	2.35%
PowerDemand	197149	311629	0.81%
RESP	868000	2452953	0.19%

Table 1: Details of univariate datasets.

Dataset	Dimension	Train	Test	Anomaly Ratio
SMD	38	708377	708393	4.16%
MSL	55	58317	73729	10.50%
SMAP	25	135183	427617	12.79%
PSM	25	132481	87841	27.75%
GECCO	10	60000	60000	1.25%
SWAN	39	69260	69261	23.80%

Table 2: Details of multivariate datasets.

domains [Wu and Keogh, 2021]. Following [Goswami *et al.*, 2022], we partition the complete UCR archive into 9 separate datasets based on the domain to which each signal belongs, (1) Arterial Blood Pressure, ABP, (2) Acceleration, ACC, (3) Air Temperature, AirTem, (4) Electrocardiogram, ECG, (5) Electrical Penetration Graph, EPG, (6) Gait, (7) NASA, (8) Power Demand, and (9) Respiration, RESP.

Previous benchmarks. We also evaluate our method on 6 previously commonly used multivariate datasets, including SMD, MSL, SMAP, PSM, WaQ, and SWAN. Details are listed in Table 1 and Table 2.

4.2 Settings

Evaluation metrics. Traditional metrics for anomaly detection include precision, recall, and F1 score. Point adjustment and revised point adjustment are also utilized to post-process the metrics. However, previous works have demonstrated that these metrics might lead to an overestimation of method performance [Kim *et al.*, 2022]. In this paper, we adopt affiliation metrics to assess the performance from an event-wise perspective [Huet *et al.*, 2022]. Precision, recall, and F1-score are calculated based on the affiliation between ground truth and prediction sets. For UCR datasets, we also employ accuracy as a metric [Wang *et al.*, 2023], indicating the probability of correctly predicting subdatasets.

Hyperparameters. The baselines are implemented based on the hyperparameters reported in previous literature. For our model, we use the pretrained GPT2 with 6 layers as our teacher network. Regarding the student network, we use a pool with 32 prototypes and an attention mechanism with an intermediate dimension of 64 and a head number of 8. During the training stage, we utilize an Adam optimizer with a learning rate of 0.0001 and a batch size of 32. All experiments are conducted on a single RTX 3090.

Baselines. We compare our method with 10 baselines for evaluation, including the one-class classification-based methods: DeepSVDD [Ruff *et al.*, 2018], THOC [Shen *et al.*, 2020], NCAD [Carmona *et al.*, 2021]; the density estimation-based method: MTGFlow [Zhou *et al.*, 2023a]; the self-supervised methods: AnoTrans [Xu *et al.*, 2021], MEMTO [Song *et al.*, 2023], TS-TCC [Eldele *et al.*, 2021], COCA [Wang *et al.*, 2023], DCdetector [Yang *et al.*, 2023]; the LLM-based method: GPT4TS [Zhou *et al.*, 2023b].

4.3 Model Comparison

Table 3 presents the performance of all methods on the UCR datasets. Our method consistently achieves the highest accuracy and affiliated F1 (AF1) score across all domains, demonstrating an average improvement of 22.2% in accuracy and 10.0% in AF1 compared to the second-best method. Notably, our method detects 82% of anomalies in total. Furthermore, three key observations can be made. First, while conventional DeepSVDD struggles to deliver satisfactory performance, one-class classification methods like THOC and NCAD show great potential in time series anomaly detection when they incorporate temporal information within the time window. Second, contrastive learning-based methods (TS-TCC and COCA) outperform reconstruction-based methods (AnoTrans, MEMTO), suggesting that approaching anomaly detection from a representation perspective has advantages over using original signals.

Results on other datasets are presented in Table 4 and Table 5. Notably, most previous methods report results after employing the point adjustment strategy. To ensure a fair comparison, we also include the adjusted metrics for these datasets. For datasets such as SMD, MSL, SMAP, and PSM which contain a large number of obvious anomalies [Wu and Keogh, 2021], our method demonstrates comparable performance with SOTA methods. In the case of WaQ and SWAN, which present diverse and challenging anomalies, our method outperforms DCdetector significantly, achieving a 62% F1 score compared to 47% on GECCO and an 80% F1 score compared to 73% on SWAN.

4.4 Model Analysis

Anomaly score visualization. We present three case studies in Figure 3 to illustrate the functionality of our method. Anomaly scores are normalized and plotted for each sub-dataset. Anomalies are identified by assigning a high anomaly score, indicative of the disparity between the output of the student network and the LLM-based teacher network. In the UCR 113 dataset, the original signals exhibit non-stationary characteristics, with varying mean values across different stages. Our method demonstrates robustness to this domain shift and successfully identifies the most anomalous segment, characterized by a distinct shape.

Prototype visualization. To highlight the efficiency of the prototype-based mechanism, we present a visualization case in Figure 6, where 8 learned prototypes for the Acceleration dataset are depicted. First, all prototypes are similar to certain segments of original signals. Second, anomalous segments exhibit a different shape compared to all prototypes.

Ablation study. In this section, our goal is to investigate the role of each component in our method.

1) LLM-based teacher network: To assess the significance of the LLM in representation generation, we replace it with

	ABP		ACC		AirTem		ECG		EPG		Gait		NASA		Power		RESP		Total	
	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1	Acc	AF1
DeepSVDD	0.333	0.564	0.714	0.743	0.385	0.726	0.297	0.587	0.360	0.700	0.242	0.495	0.182	0.435	0.182	0.418	0.000	0.225	0.288	0.555
AnoTrans	0.357	0.645	0.286	0.580	0.538	0.759	0.297	0.655	0.400	0.755	0.364	0.683	0.636	0.819	0.455	0.660	0.117	0.667	0.348	0.679
Dcdetector	0.571	0.556	0.571	0.493	0.846	0.681	0.220	0.303	0.640	0.717	0.424	0.438	0.909	0.736	0.455	0.492	0.117	0.588	0.424	0.479
MEMTO	0.405	0.655	0.714	0.750	0.615	0.750	0.319	0.616	0.440	0.733	0.364	0.673	0.364	0.664	0.455	0.680	0.059	0.598	0.368	0.656
MTGflow	0.500	0.558	0.714	0.904	0.462	0.687	0.286	0.602	0.520	0.661	0.364	0.572	0.364	0.749	0.182	0.574	0.235	0.513	0.372	0.608
GPT4TS	0.476	0.681	0.429	0.504	0.462	0.686	0.330	0.608	0.360	0.759	0.212	0.461	0.364	0.849	0.182	0.598	0.353	0.544	0.348	0.623
TS-TCC	0.690	0.754	0.286	0.549	**1.000**	0.969	0.637	0.784	0.880	0.931	0.697	0.794	0.364	0.511	0.545	0.763	0.412	0.560	0.656	0.770
THOC	0.762	0.815	0.714	0.776	**1.000**	0.971	0.604	0.760	0.880	0.908	0.636	0.784	0.909	0.896	0.455	0.775	0.294	0.389	0.671	0.780
NCAD	0.680	0.794	0.846	0.849	0.714	0.758	0.593	0.735	0.760	0.789	0.848	0.858	0.818	0.861	0.545	0.723	0.353	0.613	0.663	0.767
COCA	0.714	0.740	0.428	0.545	**1.000**	0.946	0.637	0.767	0.640	0.785	0.545	0.699	0.818	0.841	0.364	0.632	0.235	0.562	0.620	0.742
Our	**0.857**	**0.920**	**1.000**	**0.956**	**1.000**	**0.974**	**0.758**	**0.787**	**0.920**	**0.933**	**0.878**	**0.871**	**1.000**	**0.961**	**0.818**	**0.886**	**0.471**	**0.736**	**0.820**	**0.858**

Table 3: Overall results on the UCR datasets.

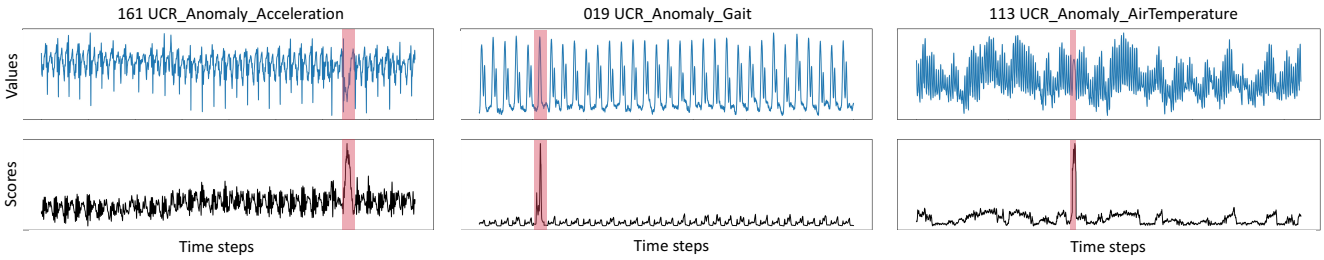


Figure 3: Case studies of anomaly score visualization.

	SMD	MSL	SMAP	PSM
IForest	0.536	0.665	0.555	0.835
DAGMM	0.573	0.746	0.685	0.801
DeepSVDD	0.791	0.836	0.690	0.907
THOC	0.850	0.897	0.907	0.895
LSTMVAE	0.823	0.826	0.781	0.810
BeatGAN	0.781	0.875	0.696	0.920
DAEMON	**0.963**	0.910	0.953	0.928
OmniAnomaly	0.852	0.877	0.869	0.828
TranAD	0.961	0.892	0.949	0.967
AnomalyTrans	0.912	0.906	0.962	0.977
AnomalyBERT	0.925	0.914	0.585	0.950
DCdetector	0.872	**0.966**	**0.970**	0.979
MEMTO	0.935	0.944	0.966	0.983
Our	0.958	0.956	0.965	**0.997**

Table 4: F1 score on SMD, MSL, SMAP, and PSM.

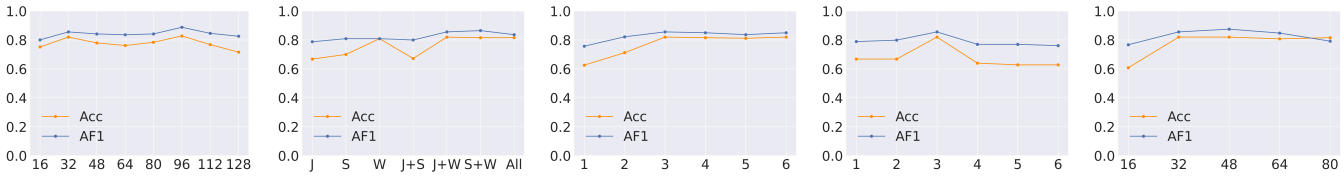
	WaQ			SWAN		
	P	R	F1	P	R	F1
OCSVM	0.021	0.341	0.040	0.193	0.001	0.001
MatrixProfile	0.046	0.185	0.074	0.167	0.175	0.171
GBRT	0.175	0.140	0.156	0.447	0.375	0.408
LSTM-RNN	0.343	0.275	0.305	0.527	0.221	0.312
Autoregression	0.392	0.314	0.349	0.421	0.354	0.385
IForest	0.439	0.353	0.391	0.569	0.598	0.583
AutoEncoder	0.424	0.340	0.377	0.497	0.522	0.509
AnomalyTrans	0.257	0.285	0.270	0.907	0.474	0.623
MTGFlow	0.333	0.125	0.182	**1.000**	0.494	0.662
DCdetector	0.383	0.597	0.466	0.955	0.596	0.734
Ours	**0.511**	**0.793**	**0.620**	0.873	**0.745**	**0.804**

Table 5: Overall results on the NIPS benchmark.

three main transformer-based blocks (Informer, Reformer, and Autoformer). The blocks are trained from scratch on each dataset. The F1-score of variants with these replaced blocks fluctuates by less than 2.6%, with an average AF1 score of 73.6%. Our method utilizes GPT2 with pretrained parameters and only fine-tunes the positional embedding and layer normalization, resulting in an enhanced AF1 score of 12%.

2) Prototype-based student network: To validate the effectiveness of our student network, we experiment with different choices (TCN, TimesNet, and TST). TCN is a classical 1D CNN-based feature extractor commonly used in time series analysis and has been applied in previous one-class classification methods such as DeepSVDD, THOC, and NCAD. TimesNet, proposed by [Wu *et al.*, 2022], rearranges time series data as a 2D tensor using Fast Fourier Transform (FFT) and leverages 2D CNN to extract features. TST is a SOTA Transformer-based feature extractor proposed by [Nie *et al.*, 2022]. The results show that TST alone does not significantly improve upon TCN, as it applies the attention mechanism like the teacher network does, which can easily result in similar representations of these two networks. In contrast, we incorporate prototypes into TST, enabling it to focus more on those historical frequent patterns and resulting in a notable improvement of 14.7% in AF1 score.

3) Training strategy: Finally, we explore different training strategies. The first strategy 'NonAug' excludes augmentation techniques and trains the model solely using original data, resulting in the poorest performance. The second strategy 'W/O CT' incorporates data augmentation but omits the contrastive loss between the teacher networks' representations in Eq. 6. This strategy improves AF1 by 4.2% but still falls behind our strategy. We force the teacher network to learn the representations that are robust to the noises by employing a contrastive regularization term. The third strategy 'W/ CS' introduces an additional contrastive loss aiming to maximize the discrepancy between the student network's



(a) Window sizes. (b) Augmentations. (c) GPT2 layers. (d) Student layers. (e) Prototype sizes.

Figure 4: Parameter sensitivity studies of main hyperparameters in AnomalyLLM.

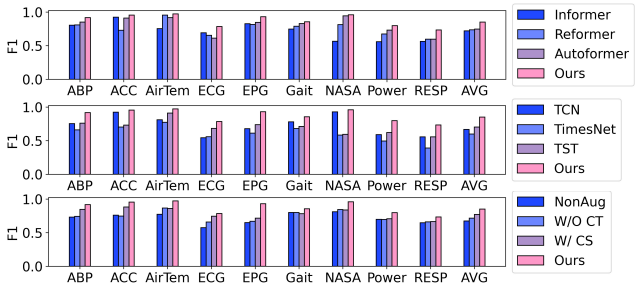


Figure 5: Ablation studies. **Top:** It depicts the performance of the model with different center predictors. **Middle:** It depicts the performance of the models with different projectors. **Bottom:** It depicts the performance of different training strategies.

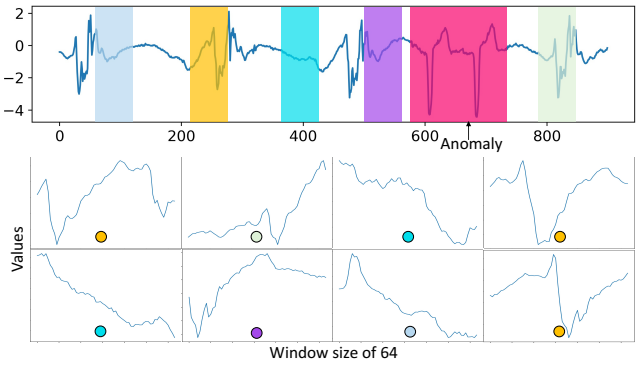


Figure 6: Visulization of prototypes. Segments similar to the learned prototypes can be found in original time series signals.

representations of the original and augmented samples. This additional loss leads to an 8.0% reduction in AF1 score, suggesting that our loss, which prioritizes the discrepancy between the representations of the student and teacher network, is more effective.

Parameter sensitivity. Figure 4(a) depicts the performance under various window sizes, showing that the performance remains relatively stable as the window size increases, maintaining an accuracy of over 72% and an AF1 score of over 80%. Figure 4(b) showcases the performance with different augmentation methods (Jittering, scaling, and warping). We can find that the introduction of warping leads to an increase in performance. The performance under different numbers of layers in GPT2 is shown in Figure 4(c). The performance stabilizes when the number is above 3, indicating that GPT2 with 3 frozen layers is sufficient to describe the distribution of time series signals. Figure 4(d) shows the results under different numbers of layers in the projector. It can be seen that the performance reaches its peak at 3. Finally, we find that the 48 prototype provides the most useful information for our projector to extract the representations, as shown in Figure 4(e). Complete results are reported in the appendix.

	100%	50%	20%	10%	zero-shot
TSTCC	0.763	0.577	0.488	0.213	0.139
THOC	0.775	0.686	0.570	0.473	0.269
COCA	0.632	0.584	0.562	0.478	0.150
Our	0.886	0.817	0.784	0.781	0.650

Table 6: F1-scores under few-shot and zero-shot settings.

Performance under few-shot settings. We also explore the performance of our methods under few shot and zero settings. Table 6 presents a case study on the PowerDemadn dataset. TSTCC relies on a large number of samples to pretrain, so its performance drops quickly when the sample number decreases. THOC and COCA have a large performance gap when transferred to different scenes because they rely on a fixed center representation decided by the training data. Our method is more generalizable when trained on a limited number of samples or across different scenes due to the generalizability of LLM. The LLM will adapt representations under different scenes, which guides the representations given by the student network.

5 Conclusion

In this paper, we propose the first knowledge distillation-based TSAD method, named AnomalyLLM. Anomaly scores are determined by the representation discrepancy between the student and teacher networks. The teacher network is fine-tuned from a pretrained LLM to generate generalizable representations for time series signals when only limited samples are available. The student network incorporates prototypical signals to produce more domain-specific representations. Besides, we propose a data augmentation-based training strategy to enhance the representation gap on anomalous samples. AnomalyLLM surpasses SOTA approaches on 9 univariate datasets and 6 multivariate datasets, highlighting the remarkable potential of knowledge distillation and LLM in time series anomaly detection. Future research should explore lightweight versions of the teacher network within our framework, tailored for deployment in scenarios with limited computational and memory resources.

Acknowledgments

This work was supported in part by the Nature Science Foundation of China (NSFC) under Grant No. U23A20326, and the Fundamental Research Funds for the Central Universities 226-2023-00111, 226-2024-00004.

References

[Audibert *et al.*, 2020] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3395–3404, 2020.

[Bergmann *et al.*, 2020] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192, 2020.

[Blázquez-García *et al.*, 2021] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

[Carmona *et al.*, 2021] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.

[Chen *et al.*, 2021] Xuanhao Chen, Liwei Deng, Feiteng Huang, Chengwei Zhang, Zongquan Zhang, Yan Zhao, and Kai Zheng. Daemon: Unsupervised anomaly detection and interpretation for multivariate time series. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2225–2230. IEEE, 2021.

[Chen *et al.*, 2023] Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection. *arXiv preprint arXiv:2307.00754*, 2023.

[Dai and Chen, 2022] Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. *arXiv preprint arXiv:2202.07857*, 2022.

[Deng and Hooi, 2021] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4027–4035, 2021.

[Eldele *et al.*, 2021] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

[Garza and Mergenthaler-Canseco, 2023] Azul Garza and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

[Godahewa *et al.*, 2021] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

[Goswami *et al.*, 2022] Mononito Goswami, Cristian Ignacio Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. Unsupervised model selection for time series anomaly detection. In *The Eleventh International Conference on Learning Representations*, 2022.

[Gruver *et al.*, 2023] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. *arXiv preprint arXiv:2310.07820*, 2023.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Huet *et al.*, 2022] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 635–645, 2022.

[Jeong *et al.*, 2023] Yungi Jeong, Eunseok Yang, Jung Hyun Ryu, Imseong Park, and Myungjoo Kang. Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme. *arXiv preprint arXiv:2305.04468*, 2023.

[Jin *et al.*, 2023a] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[Jin *et al.*, 2023b] Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, et al. Large models for time series and spatio-temporal data: A survey and outlook. *arXiv preprint arXiv:2310.10196*, 2023.

[Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

[Kim *et al.*, 2022] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7194–7201, 2022.

[Li *et al.*, 2023] Yuxin Li, Wenchao Chen, Bo Chen, Dongsheng Wang, Long Tian, and Mingyuan Zhou. Prototype-oriented unsupervised anomaly detection for multivariate time series. In *International Conference on Machine Learning*, pages 19407–19424. PMLR, 2023.

[Nie *et al.*, 2022] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[Park *et al.*, 2018] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

[Ruff *et al.*, 2018] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

[Ruff *et al.*, 2020] Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft. Rethinking assumptions in deep anomaly detection. *arXiv preprint arXiv:2006.00339*, 2020.

[Ruff *et al.*, 2021] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

[Salehi *et al.*, 2021] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021.

[Shen *et al.*, 2020] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33:13016–13026, 2020.

[Song *et al.*, 2023] Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. Memto: Memory-guided transformer for multivariate time series anomaly detection. *arXiv preprint arXiv:2312.02530*, 2023.

[Su *et al.*, 2019] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837, 2019.

[Sun *et al.*, 2023a] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

[Sun *et al.*, 2023b] Yuting Sun, Guansong Pang, Guanhua Ye, Tong Chen, Xia Hu, and Hongzhi Yin. Unraveling theanomaly'in time series anomaly detection: A self-supervised tri-domain solution. *arXiv preprint arXiv:2311.11235*, 2023.

[Tuli *et al.*, 2022] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, 2022.

[Wang *et al.*, 2023] Rui Wang, Chongwei Liu, Xudong Mou, Kai Gao, Xiaohui Guo, Pin Liu, Tianyu Wo, and Xudong Liu. Deep contrastive one-class time series anomaly detection. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 694–702. SIAM, 2023.

[Wu and Keogh, 2021] Renjie Wu and Eamonn Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Wu *et al.*, 2022] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[Xu *et al.*, 2021] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

[Yang *et al.*, 2023] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. *arXiv preprint arXiv:2306.10347*, 2023.

[Zhang *et al.*, 2023] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2306.10125*, 2023.

[Zhou *et al.*, 2019] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. Beatgan: Anomalous rhythm detection using adversarially generated time series. In *IJCAI*, volume 2019, pages 4433–4439, 2019.

[Zhou *et al.*, 2022] Qihang Zhou, Shibo He, Haoyu Liu, Tao Chen, and Jiming Chen. Pull & push: Leveraging differential knowledge distillation for efficient unsupervised anomaly detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[Zhou *et al.*, 2023a] Qihang Zhou, Jiming Chen, Haoyu Liu, Shibo He, and Wenchao Meng. Detecting multivariate time series anomalies with zero known label. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4963–4971, 2023.

[Zhou *et al.*, 2023b] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. *arXiv preprint arXiv:2302.11939*, 2023.

[Zhou *et al.*, 2024] Qihang Zhou, Shibo He, Haoyu Liu, Jiming Chen, and Wenchao Meng. Label-free multivariate time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.