Contrastive Time Series Anomaly Detection by Temporal Transformations

1st Bin Li
Department of Computer Science
TU Dortmund University
Dortmund, Germany
bin.li@cs.tu-dortmund.de

2nd Emmanuel Müller
Department of Computer Science
TU Dortmund University
Dortmund, Germany
emmanuel.mueller@cs.tu-dortmund.de

***Abstract*—Detecting anomalies in time series data is challenging due to their complex and volatile temporal features. Some anomalies only show deviating patterns to their local context instead of the overall distribution. Additionally, the biased sample distribution between normal and abnormal classes hinders the efficient usage of the available data labels. Self-supervised approaches are practically efficient for anomaly detection, in which only normal data is used during the training. However, they often fail to detect contextual anomalies in high-dimensional time series data, while the representation learning of such complex data patterns is sub-optimal.**

This paper introduces ContrastAD, a novel self-supervised framework for time series anomaly detection. Specifically, we employ the contrastive learning process with anomaly-induced temporal transformations. Targeting the point and contextual anomalies appearing in time series data, we develop corresponding transformations to enforce the model to learn discrepant representations for normal and abnormal data in the latent space. With extensive experiments, we show that our approach outperforms baseline anomaly detectors on various benchmark datasets. Our empirical results indicate that ContrastAD improves anomaly detection performance on noisy and high-dimensional time series datasets, even without common repeating patterns.

***Index Terms*—Contrastive learning, Time series, Anomaly detection**

I. Introduction

Time series anomaly detection (TSAD) has attracted vast attention in recent machine learning research and has been widely developed in numerous real-world applications, e.g., health care, manufacturing, astronomy [1]–[3]. Various algorithms solve the problem from different perspectives depending on the use case requirement. Classical database anomaly detection approaches [4]–[7] usually work efficiently on low-dimensional data and detecting *point anomalies*. Meanwhile, temporal information is a major aspect when considering *contextual anomalies* in time series. Time series forecasting-based models can detect anomalies without losing historical information [8]. Reconstruction-based models [9] belong to another thread of approaches that use the reconstruction error to indicate the likelihood of anomaly without accessing labels. However, those models often fail when the complexity and dimensionality of the time series increase. A sub-optimal representation of the data hinders anomaly detection tasks.

Recently, multiple deep models have achieved promising performance on standard benchmark datasets. The deep models capture long-term temporal information in the time series sequences with their deep structures. Learning informative representation with deep models facilitates anomaly detection enormously. Recurrent Neural Networks (RNNs) are used in both forecasting-based [10] and reconstruction-based models [11], [12]. The recurrent units are supposed to extract important information from history aggregately. Temporal Convolutional Networks (TCNs) model the temporal data with convolution kernels [13]. Transformers are also applied to time series data where the attention mechanism extracts temporal dependencies between timestamps from the time series data [14]–[16]. Despite the powerful modeling capacity of the deep models, they usually focus on intra-instance information (e.g., within a sliding window) while neglecting the semantic relation between instances. Inter-instance information is especially important for contextual anomaly detection, which requires instance-level behavior analysis.

Contrastive learning (CL) conducts a self-supervised learning paradigm to learn underlying representation from unlabeled data [17]. It has achieved notable breakthroughs in multiple computer vision tasks [18]–[20]. The main idea is to augment the original data with multiple transformations and discriminatively learn to distinguish between them. Previous works commonly follow the strategy of augmenting the input data with different transformations, then deriving positive and negative pairs from the augmented data. The learning objective encourages relevant data (*positive pairs*) to stay together while irrelevant data (*negative pairs*) to be apart from each other. This procedure allows the model to learn to differentiate features between instances. We are motivated to apply CL to TSAD tasks to enable the model to recognize the imperceptible contextual anomalies in time series. In the CL-based anomaly detection models, data transformations facilitate learning useful features for detecting novelties [20]. A downstream auxiliary classification task is often used to predict the anomaly score [19], [21]–[23]. It has been shown that the contrastive loss can also be used directly as the anomaly score if no label is available to train a classifier [24], [25]. The anomaly data are supposed to cause larger contrastive loss due to sub-optimally distinguishing the original and transformed data.

CL has already been employed in many image [19], [21], [22] but few in time series [25], [26] anomaly detection tasks. Thanks to the nature of image data, it is straightforward to apply transformations to the raw data and further define contrastive pairs for the learning process. Common transformations include geometric transformations (e.g., rotation, flipping, reflection) [20] and jittering (i.e., adding noise) [27]. CL has yet to be widely developed for time series data, while finding informative transformation for temporal data is not trivial. Existing works apply an autoregressive model for latent space forecasting [28] or learn transformations by a dedicated neural network [25]. However, they are not designed for TSAD and explicitly target representing the normal patterns [1] as well as the highly contrasting point and contextual anomalies. The contribution of such transformations to time series data with context-dependent anomalies is unclear.

Here, we address the two major challenges in the TSAD tasks: **(1) point and contextual anomalies are hard to detect in time series data** and **(2) high-dimensional and noisy time series data are difficult to represent**. We employ deep models to capture complex time series data, and we use self-supervised CL in the TSAD task with multiple point and contextual anomaly-induced temporal transformations. We are different from the classical CL approaches, which usually define the original data instances and their transformations as the *positive pairs*, while data instances with transformations of other instances as *negative pairs*. This approach is practical for learning not task-specific representations, however, is not target the complex anomaly patterns in time series. Thus, we adapt positive and negative pairs based on the artificial temporal transformations. By artificially augmenting positive and negative temporal transformations, we aim to explicitly encourage the model to learn a robust representation of normal data and to differentiate the normal data from anomalies.

Our contribution is threefold:

1) we propose a CL framework for TSAD;
2) we propose multiple temporal time series transformations for the anomaly detection task;
3) we demonstrate the effectiveness of the proposed framework with extensive empirical experiments on common real-world TSAD benchmark datasets.

II. Related works

A. Time series anomaly detection

Classical anomaly detection approaches have shown their efficacy in detecting highly deviating points from their neighborhoods in the same data collection, e.g., PCA [29], density-based LOF [5] and the one-class classifiers [7], [30]. Recent deep models expand the anomaly detector to more high-dimensional and complex data. They include extended one-class approach DeepSVDD [31], reconstruction-based approaching using autoencoders [32], [33] and generative approaches [34], [35].

[1]In this manuscript, the term "normal" only refers to the opposite of "abnormal" in the anomaly detection context, instead of normal distribution.

A major challenge in TSAD is detecting contextual anomalies. To capture contextual information, one thread of works is based on time series forecasting and uses the prediction error to indicate anomalies [10], [36]. Several deep models are also used to capture the temporal information in time series data. Malhotra et al. [11] use LSTMs build Autoencoders to reconstruct time series data. Similarly, Hundman et al. [1] use LSTMs for forecasting-based anomaly detection. Su et al. [3] employ GRUs, which are supposed to be easier to train than LSTMs due to their fewer parameters. Convolutional neural networks are also used for time series data. Bai. et al. [13] have shown empirical results that TCNs outperform LSTMs and RNNs in sequence modeling. Thill et al. [37] construct Autoencoders with TCNs for anomaly detection. Finally, attention mechanism-based Transformer models are capable of long-term sequences. Tuli et al. [15] propose a Transformer-based anomaly detector with an adversarial training procedure that can amplify reconstruction error. Xu et al. [14] compose a Transformer network with prior- and series-association to learn the temporal dependency.

B. Contrastive learning

Data augmentation is a key step in CL. In computer vision tasks, geometric transformations enrich the image data without dramatically changing the semantic, e.g., flipping [18], rotation [19], reflection [20], permutation [21], jittering [38], cropping [22] and changing brightness [22]. To enrich the transformation beyond the original image into a more general case, Bergman et al. [39] employ affine transformation. By manipulating the affine parameters, the affine transformations allow for dimensionality reduction, non-distance preservation, and random transformation. Mistra et al. [40] extend the application to the video domain and transform the data by shuffling the frames.

A common training strategy is the SimCLR [38], which encourages close embedding of positive pairs and penalizes nearly embedding of negative pairs [21], [22]. The InfoICE loss [28], [41] manipulates the mutual information in the latent space. Incorporated with the downstream classification tasks, the constrictive objective is also combined with cross entropy [24], triplet center loss [39] and one-class classification loss [19].

Under the self-supervised setting, CL also shows its strength in various anomaly detection tasks with the biased class distribution. Winkens et al. [22] enhance Out-of-Distribution detection with the contrastive objective. Sohn et al. [19] incorporate negative-sample-free CL with deep one-class classification for anomaly detection. Wang et al. [27] further proposed using distribution augmentation to overcome the class collision problem in the one-class setting.

Despite the recent vast development of CL, the application domain still needs to be expanded beyond computer vision. Time series data still faces the challenge of getting proper transformations to form contrastive pairs. Shenkar et al. [24] transform tabular data by masking consecutive feature subsets; however, they do not consider the temporal dependency. Ana-

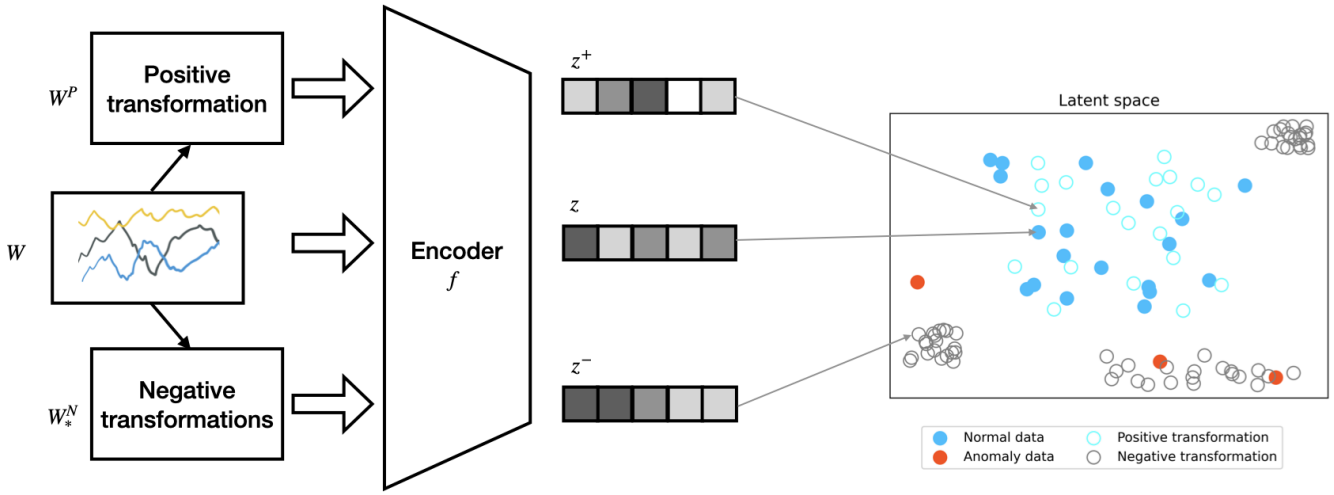

Fig. 1: **Architecture overview**: For each input window W, we augment it with multiple anomaly-induced positive (W^P) and negative (W_*^N) transformations. After the augmentation, all data are fed to a time series encoder. The embeddings (z,z^+ and z^-'s) are used for the contrastive learning process in the latent space.

log to jittering for image data, Wang et al. [27] transform time series data by adding noise and manipulating the sequence magnitude. Another implicit solution is to learn transformations with neural networks [25], [26]. However, they do not explicitly consider the context-relevant anomalies in the time series data during their transformation processes. Considering the unique nature of point and contextual anomalies, we extend the artificial time series transformation (e.g., jittering and pattern-wise magnitude change) by Wang et al. [27] with extensive anomaly-induced positive and negative transformations; and design a novel self-supervised CL framework for TSAD.

III. Method

A. Problem statement

Let $X = \{X_t\}_{t\in\mathbb{Z}}$ be a D-dimensional time series ($X_t \in \mathbb{R}^D$). $W = \{X_{t+1}, ..., X_{t+L}\}$ is a sliding window over X with length L. We aim to detect both point and contextual anomalies from the sliding window. A point anomaly occurs at a single timestamp $T \in [t+1, t+L]$, e.g., a spike. A contextual anomaly event describes a consecutive subsequence of W in time period $[T, T+l] \subset [t+1, t+L]$ that makes W deviate from the common sliding windows. For a given sliding window W, we need to predict one anomaly score a, indicating the likelihood of W being anomalous. As a post-step, users can apply a threshold over a to receive a binary prediction of being anomalous. The selection of such a threshold is out of our scope.

B. ContrastAD

*1) **Model architecture**:* The ContrastAD model consists of two stages. Firstly, we apply positive and negative temporal transformations to each input sliding window W (Section III-B2). Secondly, after the original and transformed data windows are encoded into the latent space, we conduct the CL process using our contrastive objective (Section III-B3). The final anomaly score is calculated based on the contrastive loss (Section III-B4). Figure 1 shows an overview of the model architecture.

*2) **Temporal transformations**:* We follow the common assumption in unsupervised anomaly detection tasks where pure normal data is available for the training and anomalies only appear in the test phase. To meet the nature of the TSAD problem, we carry out temporal transformations of sliding windows considering the local context within each window. Specifically, we augment the normal time series data windows with the *jittering*-based positive transformation and multiple anomaly-induced negative transformations. The model is generalized by augmenting the normal training data to learn noisy and slightly perturbed normal patterns. The negative temporal transformations contain both point and contextual anomaly-induced artificial abnormal patterns. The contrastive objective pushes the normal cluster away from negative transformations in the latent space.

In the *jittering*-based positive transformation, for data window $W \in \mathbb{R}^{L\times D}$, we add random noise $\epsilon \in \mathbb{R}^{L\times D}$ to W, specifically,

$$W^P := W + \epsilon$$

for $\epsilon_{i,j} \in \mathcal{N}(0, \sigma_{jitter})$ with $i \in [1, L]$ and $j \in [1, D]$. σ_{jitter} controls the strength of the noise.

Moreover, oppositely, we augment the data window W with multiple anomaly-induced negative transformations to encourage the normal windows to be embedded apart from deviating patterns. Specifically, we incorporate *spike* to simulate point anomalies and *shuffle*, *trend* as well as *scale* to simulate contextual anomalies. So that we end up with the negative

transformation set

$$W^N := \{W^N_{spike}, W^N_{shuffle}, W^N_{trend}, W^N_{scale}\}$$

The *spike* transformation simulates point anomalies by extreme value. The transformation W^N_{spike} is achieved by replacing the value $W_{T,d}$ at random timestamp T and feature d with an extreme value deviating from the mean of the feature d

$$W^{'}_{T,d} = \mu_d + \lambda_{spike} \cdot \sigma_d \tag{1}$$

where μ_d and σ_d are the mean and standard deviation on feature d in the sliding window. λ_{spike} is a hyperparameter that controls the spike amplitude.

The *shuffle* transformation simulates an interrupted temporal context by shuffling the first and second half of the window. The data window still contains the local temporal context. However, the global order within the whole window is shuffled.

$$W^N_{shuffle} = [W_{\lceil \frac{L}{2} \rceil : L}; W_{1:\lceil \frac{L}{2} \rceil - 1}] \tag{2}$$

The *trend* transformation simulates abnormal data windows caused by irregular trends within the window. W^N_{trend} scales each timestamp with an incremental factor so that for value $W_{T,d}$ at timestamp T on feature d,

$$W^{'}_{T,d} = (\frac{L+T}{L}) \times W_{T,d} \tag{3}$$

where L is the window length.

The *scale* transformation stretches the sequence on its feature dimensions. The transformation scales the whole window W by a random factor $\lambda_{scale} \in \mathcal{N}(2, \sigma_{scale})$:

$$W^N_{scale} = W \otimes \lambda_{scale} \tag{4}$$

where σ_{scale} is a hyperparamter controls the scaling strength.

Figure 3 gives a visual example of the temporal transformations on one normal and one abnormal data window from the univariate **ECG** [42] dataset.

3) ***Contrastive objective****:* All the positive, negative temporal transformations and the original time series windows are fed into the encoder to get latent space representations. The encoder f can be one of the time series representation learning models that capture contextual information. We give a further discussion on the existing encoder models in Section III-B5. After the encoding process, we get the latent space representations $z = f(W)$ for the original input window W, $z^+ = f(W^P)$ for the positive-transformed window W^P and $z^-_* = f(W^N_*)$ for each negative-transformed window $W^N_* \in W^N$.

We define contrastive pairs in each mini-batch $\mathcal{B}$ of data windows. Figure 2 visualize an overview of all pairs derived from the two data windows $W(i)$ and $W(j)$ in mini-batch $\mathcal{B}$. Our intuition is to pull the normal windows and their positive transformations together and push positive transformations of

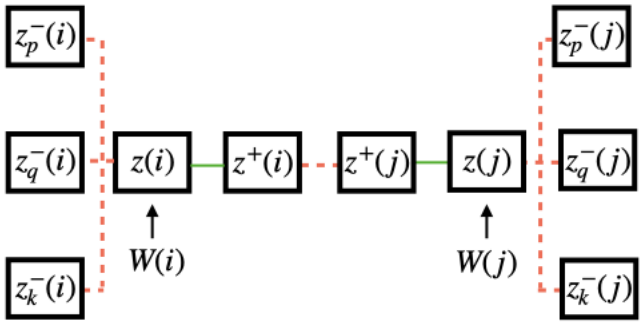

Fig. 2: **Contrastive pairs**: $z(i)$ and $z(j)$ are window embeddings of $W(i)$ and $W(j)$. z^+ and z^-_* are corresponding positive and negative transformations. The green solid lines connect positive pairs, and red dashed lines connect negative pairs in contrastive learning.

different data windows as well as data windows with their negative transformations apart from each other. Concretely, we define positive pairs as the embeddings of a window $W(i) \in \mathcal{B}$ and its own positive transformation. So that the positive loss component is

$$\mathcal{L}^P = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{\mathcal{B}} exp(sim(z(i), z^+(i)))/\tau) \tag{5}$$

where τ is the temperature parameter, $sim(\cdot, \cdot)$ denotes the cosine similarity.

Furthermore, in each mini-batch $\mathcal{B}$, we define the negative pairs and corresponding negative loss components as

(1). the embeddings of the positive transformations of two different data windows $W(i)$ and $W(j)$

$$\mathcal{L}_{WW} = \frac{2}{|\mathcal{B}|^2 - |\mathcal{B}|} \sum_{i=1}^{\mathcal{B}} \sum_{j=i+1}^{\mathcal{B}} exp(sim(z^+(i), z^+(j))/\tau) \tag{6}$$

(2). the embeddings of each data window $W(i)$ and its every negative transformation $W^N_*(i) \in W^N$

$$\mathcal{L}_{WN} = \frac{1}{|\mathcal{B}| \times |W^N|} \sum_{i=1}^{\mathcal{B}} \sum_{p=1}^{|W^N|} exp(sim(z(i), z^-_p(i))/\tau) \tag{7}$$

The final contrastive objective is

$$\mathcal{L} = -log \frac{\mathcal{L}^P}{\mathcal{L}^P + \mathcal{L}_{WW} + \mathcal{L}_{WN}} \tag{8}$$

4) ***Anomaly score****:* We define the anomaly score of each data window based on the contrastive loss [25] without access to the labels. Specifically, we adapt the positive and negative loss components to extract each data window's positive and negative contributions. Concretely, for data window $W(i)$, the positive contribution is

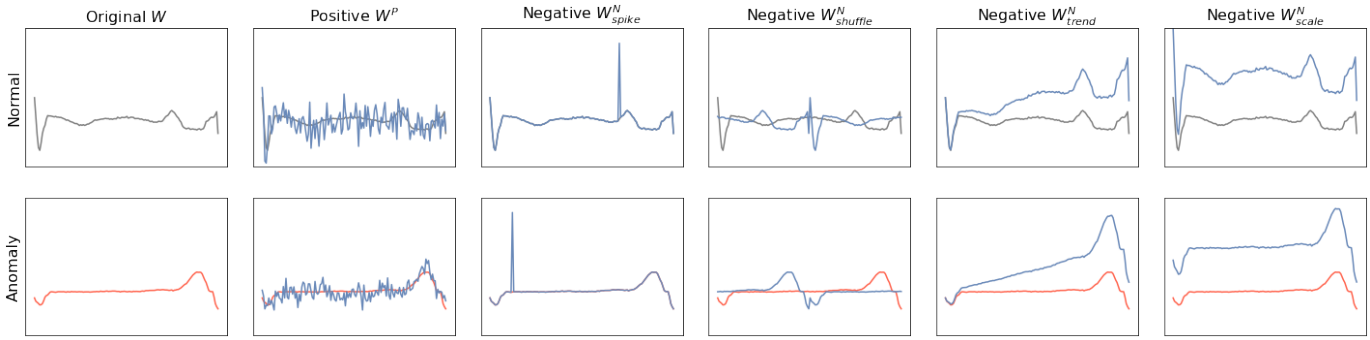

Fig. 3: **An example of the temporal transformations on ECG**: The gray lines and red lines are two examples of the original normal and abnormal data. The blue lines are the temporal transformations.

$$a^P(i) = exp(sim(z(i), z^+(i)))/\tau) \tag{9}$$

and the negative contributions are

$$a_{WW}(i) = \frac{1}{|\mathcal{B}|-1} \sum_{j\in[1,|\mathcal{B}|],j\neq i} exp(sim(z^+(i), z^+(j))/\tau) \tag{10}$$

$$a_{WN}(i) = \frac{1}{|W^N|} \sum_{p=1}^{|W^N|} exp(sim(z(i), z_p^-(i))/\tau) \tag{11}$$

The final anomaly score is

$$a(i) = -log\frac{a^P(i)}{a^P(i) + a_{WW}(i) + a_{WN}(i)} \tag{12}$$

*5) **Time series encoders**:* In ContrastAD, the encoder f learns representations of the original data windows and their temporal transformations. Several existing time series representation learning models can act as the encoder. In our experiment, we primarily use the autoregressive model-based Contrastive Predictive Coding (CPC) [28], which is effective in multiple sequential data representation learning tasks [41], [43]. This is especially important for high-dimensional and long-span time series to keep the global structure, while classical unimodal approaches are often not powerful enough, and conditional generative models are computationally intense [28]. We compare different time series encoders in Section IV-F.

IV. Experiments

A. Datasets

We conduct experiments on multiple TSAD benchmark datasets. **ECG** [42] is a univariate dataset describing 5000 patient heartbeats in 5 classes. We consider the two smallest classes as anomalies and the other three as normal. **SMAP** and **MSL** are high-dimensional spacecraft telemetry data with pre-labeled point and contextual anomalies [1]. These datasets are noisy and do not contain common repeating patterns. **SWaT** [44] is collected from a water treatment testbed, where artificial attacks are labeled as anomalies. Finally, **UCR** [45] is a challenging univariate dataset from various real-world applications. **SMAP**, **MSL** and **UCR** contain subsets collected from different devices. We train a dedicated model for each subset. In our experiments, we use 5 subsets from **UCR** ($001 \sim 005$), **SMAP** ($P-1$, $S-1$, $E-1$, $E-2$, $E-3$) and **MSL** ($M-6$, $M-1$, $M-2$, $S-2$, $P-10$) respectively based on the order in the original datasets, which cover both point and contextual anomalies. The train and test sets are specified in the original datasets, so the train sets only contain normal data. The evaluation results are averaged over all subsets.

B. Baseline models

We compare ContrastAD with multiple common baseline anomaly detection models. We include classical anomaly detection approaches Local Outlier Factor (**LOF**) [5], One-Class SVM (**OCSVM**) [7] with RBF kernel and Isolation Forest (**IF**) [4]. Furthermore, we also compare with the recent reconstruction-based deep models LSTM-Autoencoder (**LSTMAE**) [11] and **DAGMM** [32]. The implementation of baseline models is taken either from Scikit-learn [2] or online resources [3].

C. Evaluation metric

Our model calculates the anomaly score based on the contrastive loss. The anomaly score indicates the likelihood that a window is an anomaly. We do not include a specific thresholding technique to receive a binary prediction. The anomaly detector should deliver a good performance of multiple threshold settings, and the user can select the threshold depending on the concrete use cases. Following [20], [25], we use the area under the Receiver Operating Characteristic curve (AUC) to evaluate the anomaly detector performance.

D. Experiment setup

For sliding window construction, we set window size $L = 140$ for **ECG** defined by the data source. For the other datasets without prior knowledge, we generally set $L = 100$ except

[2]https://scikit-learn.org
[3]https://github.com/KDD-OpenSource/DeepADoTS

TABLE I: Overall performance

	ECG	SMAP	MSL	SWaT	UCR
LOF	0.487	0.348	0.702	0.435	0.502
OCSVM	0.505	0.268	0.789	0.617	0.526
IF	0.500	0.307	0.500	0.469	0.481
LSTMAE	0.566	0.253	0.786	**0.791**	0.535
DAGMM	**0.643**	0.576	0.745	0.659	0.567
ContrastAD	0.500	**0.619**	**0.813**	0.729	**0.734**

TABLE II: Ablation study: Negative temporal transformations

	ECG	SMAP	MSL	SWaT	UCR
w/o spike	**0.535**	0.565	0.705	0.626	0.464
w/o shuffle	0.530	0.474	0.632	0.650	0.455
w/o trend	0.500	0.624	0.710	0.682	0.484
w/o scale	0.500	**0.639**	0.702	0.588	0.553
ContrastAD	0.500	0.619	**0.813**	**0.729**	**0.734**

$L = 50$ for the smaller dataset **MSL**. We slide the window forward without overlap. For the time series transformation, we set hyperparameters by default $\sigma_{jitter} = 0.2$ for W^P, $\lambda_{spike} = 5$ for W^N_{spike} and $\sigma_{scale} = 0.8$ for W^N_{scale}. With these, the spike transformation generates a significant point anomaly, while the *jittering* transformation only augments the data with minor perturbation. Beyond the selected default parameter configuration, an extensive parameter sensitivity analysis is provided in Section IV-G. For ContrastAD, we train with the Adam optimizer [46] for 50 epochs with learning rate 0.001 and batch size $|\mathcal{B}| = 8$. We use 20% of the training data for validation. In the contrastive objective, we set the temperature parameter $\tau = 0.2$. As the time series encoder, in addition to the default CPC model [28], we also compare to a three-layer bidirectional LSTM model, a TCN model [13] with kernel size 5 and a Transformer model [47], all with 80 hidden units.

E. Overall performance

The overall performance (AUC score) of ContrastAD and the baseline models has been summarized in Table I. Our model outperforms the baseline model on three benchmark datasets and performs on par with the baseline models on **SWaT**. Specifically, on the **ECG** data, the classical deep models LSTMAE and DAGMM show better performance than ContrastAD. This may indicate that the artificial transformations in ContrastAD do not bring many benefits to the dataset containing fixed repeating normal patterns and specific abnormal patterns. In this case, we recommend defining data-specific artificial negative transformations based on prior knowledge of the datasets.

F. Ablation study

We conduct two ablation studies to examine the importance of the negative transformation functions (Table II) and encoder models (Table III). ContrastAD with all four negative transformation functions *spike*, *shuffle*, *trend*, and *scale* shows the best performance on **MSL**, **SWaT** and **UCR** and is on par with the two deep models on **SMAP**. Especially on **UCR**, which is claimed to be a challenging dataset [14], even removing a single transformation function will cause a significant drop in the AUC score. The **ECG** dataset shows results in the opposite direction. Removing *trend* or *scale* does not impact the performance, while removing *spike* or *shuffle* is even beneficial. One possible reason is that **ECG** contains neither severe point anomalies like *spike* nor contextual anomalies like *shuffle*; those two functions may confuse the model with unrealistic transformations. In the other high-dimensional datasets with more general and noisy patterns, the negative transformations benefit the learning procedure.

TABLE III: Ablation study: Time series encoders

	ECG	SMAP	MSL	SWaT	UCR
LSTM	0.430	0.349	0.503	0.365	0.473
TCN	0.449	0.447	0.516	0.670	0.369
Transformer	**0.540**	**0.718**	0.688	0.708	0.477
CPC	0.500	0.619	**0.813**	**0.729**	**0.734**

Table III shows the model performance when alternating the encoder CPC with another time series representation learning model. The CPC encoder shows dominating performance on **MSL**, **SWaT** and **UCR**, while is on par with other encoders on **ECG** and **SMAP**. The classical time series modeling approaches LSTM and TCN, however, do not show convincing results.

G. Parameter sensitivity

We show the results of parameter sensitivity analysis in Figure 4. Since the contrastive pairs are built within mini-batches, the batch size is supposed to be an important factor in ContrastAD. We evaluate the model performance under the batch sizes $\mathcal{B} \in \{1, 2, 4, 8, 16, 32, 64, 128\}$. We train one model per subset (some datasets do not allow batch sizes larger than 32). The results are shown with mean and standard deviations over three runs. Our experimental results show that large batch sizes do not directly bring better results. Rather, there is a drop in the AUC score on **UCR** when the batch size increases from 16 to 128. Most datasets show increasing performance when batch size increase from 1 to 16. This indicates that a proper middle size of batches helps to learn features among local instances.

Furthermore, Figure 4 also shows the result of sensitivity analysis of the hyperparameters σ_{jitter}, λ_{spile} and σ_{scale} in the negative temporal transformations. The parameters are used to determine the strength of the transformation effect. We examine both small values $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and large values $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ for σ_{jitter} and λ_{spile}. For σ_{scale}, we try values in $\{0, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9\}$. We observe that ContrastAD is generally not sensitive to all parameters in the value ranges we have examined. For the σ_{jitter} in the positive transformation, a small value will add random noise to the input signal, which helps the model to learn robust representations of the normal pattern. However,

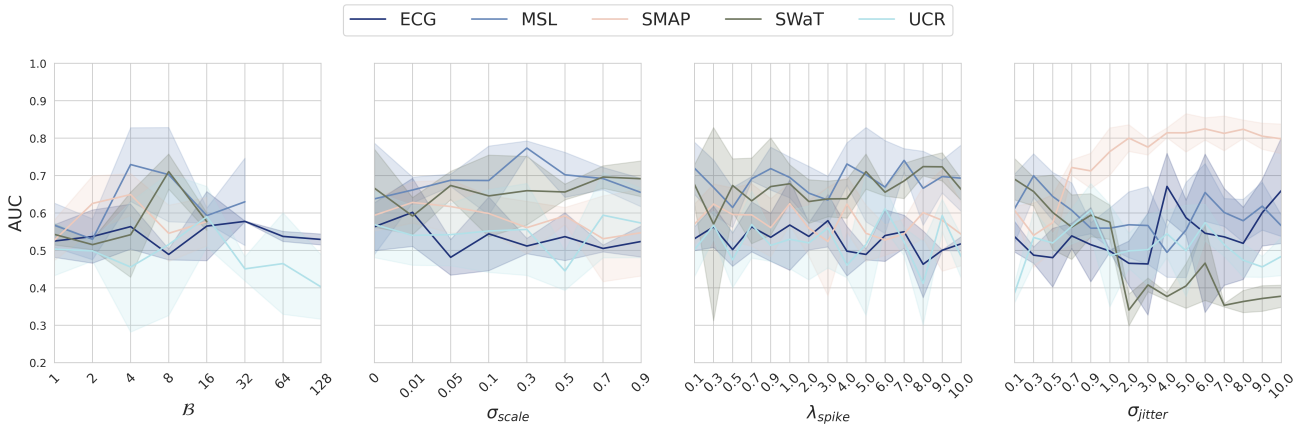

Fig. 4: **Parameter sensitivity analysis**. Batch size $\mathcal{B}$. Positive transformation parameter σ_{jitter}. Negative transformation parameter λ_{spike}. Negative transformation parameter σ_{scale}.

a very large σ_{jitter} will lead a noisy normal data becoming an anomaly, therefore, harming the performance (e.g., on **SWaT**). On the opposite side, a tiny spike is hard to be distinguished by the model. Therefore we recommend to select values $\sigma_{jitter} \in [0, 1]$ and $\lambda_{spile} \in [1, 10]$. The *scale* transformation scales the data window with some random factors $\lambda_{scale} \in \mathcal{N}(2, \sigma_{scale})$. In our experiments, we fix the average factor value to 2, so it doubly stretches the data window vertically. The hyperparameter σ_{scale} does not significantly impact performance.

V. Conclusion

We presented a novel TSAD framework ContrastAD by temporal transformation-based CL. Specifically, we defined multiple point and contextual anomaly-induced temporal transformations when constructing contrastive pairs. Our experimental results indicate that ContrastAD performs better or on par with common baseline models. In the extensive analysis, we discovered that ContrastAD brings more benefits to high-dimensional and noisy datasets without common repeating patterns.

Currently, we include four negative transformations to simulate the most common point and contextual anomalies in time series data. These contribute to the anomaly detector even though the anomalies in the datasets are not directly the same as what we generated. However, we believe few real anomaly data are necessary for the model to learn precise negative transformations for datasets with some common repeating patterns or limited types of anomalies, e.g., **ECG**. To this end, we plan to develop the temporal transformation procedure in a semi-supervised manner, with a few labeled anomaly patterns as guidance for the negative temporal transformations.

Acknowledgement

This work was supported by the Research Center Trustworthy Data Science and Security, an institution of the University Alliance Ruhr.

References

[1] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.

[2] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 3009–3017.

[3] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. 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*, 2019, pp. 2828–2837.

[4] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[7] B. Scholkopf, R. Williamson, A. Smola, and J. Shawe-Taylor, "Sv estimation of a distribution's support," *Advances in Neural Information Processing Systems*, vol. 41, 01 2000.

[8] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005.

[9] H. Hoffmann, "Kernel pca for novelty detection," *Pattern recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[10] P. Malhotra, L. Vig, G. Shroff, P. Agarwal *et al.*, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, vol. 89, 2015, pp. 89–94.

[11] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[12] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 2014, pp. 4–11.

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

[14] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," *arXiv preprint arXiv:2110.02642*, 2021.

[15] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *arXiv preprint arXiv:2201.07284*, 2022.

[16] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2114–2124.

[17] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[18] H. Cho, J. Seol, and S.-g. Lee, "Masked Contrastive Learning for Anomaly Detection," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 1434–1441. [Online]. Available: https://www.ijcai.org/proceedings/2021/198

[19] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, "Learning and Evaluating Representations for Deep One-class Classification," Mar. 2021, arXiv:2011.02578 [cs]. [Online]. Available: http://arxiv.org/abs/2011.02578

[20] I. Golan and R. El-Yaniv, "Deep Anomaly Detection Using Geometric Transformations," Nov. 2018, arXiv:1805.10917 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1805.10917

[21] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances," Oct. 2020, arXiv:2007.08176 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2007.08176

[22] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, T. Cemgil, S. M. A. Eslami, and O. Ronneberger, "Contrastive Training for Improved Out-of-Distribution Detection," Jul. 2020, arXiv:2007.05566 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2007.05566

[23] V. Sehwag, M. Chiang, and P. Mittal, "SSD: A Unified Framework for Self-Supervised Outlier Detection," Mar. 2021, arXiv:2103.12051 [cs]. [Online]. Available: http://arxiv.org/abs/2103.12051

[24] T. Shenkar and L. Wolf, "ANOMALY DETECTION FOR TABULAR DATA WITH INTERNAL CONTRASTIVE LEARNING," p. 26, 2022.

[25] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural Transformation Learning for Deep Anomaly Detection Beyond Images," Feb. 2022, arXiv:2103.16440 [cs]. [Online]. Available: http://arxiv.org/abs/2103.16440

[26] T. Schneider, C. Qiu, M. Kloft, D. A. Latif, S. Staab, S. Mandt, and M. Rudolph, "Detecting Anomalies within Time Series using Local Neural Transformations," Feb. 2022, arXiv:2202.03944 [cs]. [Online]. Available: http://arxiv.org/abs/2202.03944

[27] R. Wang, C. Liu, X. Mou, K. Gao, X. Guo, P. Liu, T. Wo, and X. Liu, "Deep Contrastive One-Class Time Series Anomaly Detection," Oct. 2022, arXiv:2207.01472 [cs]. [Online]. Available: http://arxiv.org/abs/2207.01472

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

[29] R. Paffenroth, K. Kay, and L. Servi, "Robust pca for anomaly detection in cyber networks," *arXiv preprint arXiv:1801.01571*, 2018.

[30] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[31] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," *arXiv preprint arXiv:1906.02694*, 2019.

[32] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[33] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[34] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[35] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision*. Springer, 2018, pp. 622–637.

[36] R. Greis, T. Reis, and C. Nguyen, "Comparing prediction methods in anomaly detection: an industrial evaluation," 2018.

[37] M. Thill, W. Konen, H. Wang, and T. Bäck, "Temporal convolutional autoencoder for unsupervised anomaly detection in time series," *Applied Soft Computing*, vol. 112, p. 107751, 2021.

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

[39] L. Bergman and Y. Hoshen, "Classification-Based Anomaly Detection for General Data," May 2020, arXiv:2005.02359 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2005.02359

[40] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European conference on computer vision*. Springer, 2016, pp. 527–544.

[41] P. de Haan and S. Löwe, "Contrastive predictive coding for anomaly detection," *arXiv preprint arXiv:2107.07820*, 2021.

[42] Y. Chen, Y. Hao, T. Rakthanmanon, J. Zakaria, B. Hu, and E. Keogh, "A general framework for never-ending learning from time series streams," *Data mining and knowledge discovery*, vol. 29, no. 6, pp. 1622–1664, 2015.

[43] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International conference on machine learning*. PMLR, 2020, pp. 4182–4192.

[44] A. P. Mathur and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ics security," in *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 2016, pp. 31–36.

[45] E. Keogh, D. R. Taposh, U. Naik, and A. Agrawal, "Multi-dataset time-series anomaly detection competition," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://compete. hexagonml. com/practice/competition/39*, 2021.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," *arXiv preprint arXiv:2106.14112*, 2021.