CATS: Contrastive learning for Anomaly detection in Time Series

Joël Roman Ky, Bertrand Mathieu, Abdelkader Lahmadi, Raouf Boutaba

CATS: Contrastive learning for Anomaly detection in Time Series

Joël Roman Ky*†, Bertrand Mathieu*, Abdelkader Lahmadi† and Raouf Boutaba‡
*Orange Innovation Lannion, {joelroman.ky, bertrand2.mathieu}@orange.com
†Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France, abdelkader.lahmadi@loria.fr
‡David R. Cheriton School of Computer Science, University of Waterloo, rboutaba@uwaterloo.ca

***Abstract*—Anomaly detection (AD) plays a critical role in a wide variety of big data applications, including cybersecurity, monitoring, and network systems. It consists in finding patterns in time series data that indicate unexpected events such as faults or defects. Traditional AD approaches, predominantly based on reconstruction techniques, often yield suboptimal performance, particularly when anomalies are present in the training set. Conversely, contrastive learning (CL) has shown significant performance in image processing tasks and is increasingly applied in time series data classification and forecasting. However, traditional CL frameworks are not well-adapted for time series AD due to two key challenges. First, AD is typically performed only on normal instances, and thus CL does not benefit from knowledge about anomalous instances. Second, the temporal nature of time series data is often neglected when computing time series similarity, thereby hindering the effective learning of time series representation.**

To overcome these limitations, we propose CATS, a novel approach that leverages a temporal similarity measure to learn time series representations. Moreover, through negative data augmentation, CATS generates a more realistic distribution of anomalies, which enables anomaly-informed CL. Extensive experiments conducted on six real-world datasets demonstrate that CATS outperforms existing AD methods. Our results highlight the efficacy of CATS in enhancing time series AD performance in big data environment across various application domains.

***Index Terms*—anomaly detection, time series, cloud gaming, contrastive learning**

I. Introduction

In various application domains such as networking or cybersecurity, large volumes of time series datasets are generated and crucial for monitoring systems performance. Accurately detecting anomalies within these datasets are essential for identifying networks faults, device/system malfunctions, security breaches or service degradations that can significantly impact operations and user quality of experience (QoE). However, labeling real-time series data for anomaly detection is challenging due to the large amount of time-series data and the scarcity of labeled anomalies.

Consequently, unsupervised time series anomaly detection (AD) has received significant attention in machine learning research community leading to various proposed techniques. These can be categorized into five groups: reconstruction-based [1]–[4], distance-based [5], one-class classification-based [6], [7], isolation-based [8], and generative-based [9], [10]. These techniques learn normal patterns from normal data and detect anomalies based on deviations from the learned normality. However, these techniques have some limitations as they can be affected by *data contamination* from unknown anomalies in training sets.

To enhance the performance of time series AD, contrastive learning (CL) [11] has emerged as a promising approach, increasingly applied to various classification and forecasting tasks [12]–[15]. CL learns data representations by contrasting positive and negative views through data augmentation. This process clusters similar views while repelling dissimilar ones, enhancing transformation-invariant properties.

However, in time series analysis tasks, the exploration of data augmentation techniques has not yet been as extensive as in the field of computer vision. Some previous works in this field have used CL for anomaly detection. Nevertheless, they did not utilize a similarity function specifically designed for time series data, resulting in an inefficient exploitation of the temporal aspect inherent in multivariate time series data, which are nevertheless crucial for modeling. To address these limitations, we propose in this paper an end-to-end method called contrastive learning for anomaly detection in time series (CATS). Our contributions can be summarized as follows:

- Using Dynamic Time Warping (DTW) similarity, we propose a novel DTW-based temporal contrastive learning loss to efficiently model multivariate time series.
- We use negative data augmentations to create synthetic anomalies, establishing a realistic out-of-order distribution that contrasts with normal instances in the training set.
- We conduct extensive empirical experiments on large real-world and popular benchmark time series datasets. Furthermore, we conduct experiments on time series datasets in the networking domain, collected from cloud gaming platforms. We demonstrate the effectiveness of our proposed framework and its generalization capabilities compared to previous AD techniques.
- We conduct ablation studies to evaluate the effectiveness of each component of the proposed method, data contamination and the influence of hyper-parameters.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III describes CATS method and Section IV presents the experimental results.

II. Related work

A. Unsupervised Anomaly Detection for Time Series

Anomaly detection in time series data has been extensively studied in the literature with various approaches such as statistical methods [16], distance-based methods [17], density-based methods [5], isolation-based methods [8] and one-class classification methods [7]. The emergence of deep learning led to many novel approaches including Deep-SVDD [6], GAN-based approaches [9], [10], [18], [19], reconstruction-based approaches [1]–[4], [10], [16] or transformers-based approaches [20], [21] where all of them, mainly rely on unsupervised learning due to the scarcity of labeled data.

Reconstruction-based models utilize autoencoders to train models with only normal data and detect anomalies by reconstructing a point or a sliding window from test data and comparing them to the actual data according to a reconstruction error. Variational autoencoders have been applied to this approach, assuming that the training data follows a Gaussian distribution. However, these models often struggle with generalization to other datasets that exhibit different data distributions. On the other hand, GAN-based approaches and one-class classification methods are prone to model collapse during training, leading to instability issues [22]. Other studies leveraged transformers to efficiently extract temporal information for time series anomaly detection but training transformers demand significant computing resources [23].

Compared to the aforementioned approach, in this work contrastive learning is adopted to enhance the anomaly detection performance and the robustness to data contamination.

B. Contrastive Learning for Time-Series

Contrastive Learning has emerged as a prominent self-supervised learning technique demonstrating a great potential across various domains, including computer vision and natural language processing. Traditional contrastive learning models [24]–[26] construct positive sample pairs i.e. augmented views of the same instance and negative pairs i.e. augmented views of other instances or a dictionary queue to facilitate representation learning with data augmentation techniques. The effectiveness of these approaches relies on key factors such as data augmentation, efficient sampling of negative pairs, and large batch sizes [25]. Recent advancement in contrastive learning architectures such as [27], [28] have shown that meaningful representations can be learned without the explicit need of negative pairs or large batch sizes, achieving remarkable performance in downstream tasks.

In the domain of time series analysis, recent techniques have been developed to learn representations from time-series data. For instance, some methods introduced triplet loss and temporal negative sampling [13], while others leveraged local smoothness to define neighborhoods [29]. [12] utilized temporal and contextual contrastive learning on different views of time-series data generated with weak and strong data augmentation. Similarly, TS2Vec [14] employed a hierarchical contrastive learning approach, using augmented context views to capture multi-scale information. These aforementioned methods have demonstrated robust performance across various downstream tasks, notably in forecasting and classification.

CL techniques have also been applied to anomaly detection in time-series data. Some studies employed deterministic contrastive learning with learnable transformations [30], while others combined negative sample-free contrastive learning with one-class classification [31]. Other approaches [32]–[34] have utilized temporal transformations to generate anomalies from normal windows and have proposed frameworks that enhance anomaly detection through representation learning.

While some studies suggest that negative pairs are not essential for contrastive learning, we believe that negative pairs, constructed through negative data augmentations, bring some learning knowledge that may enhance AD tasks, unlike their impact on other downstream tasks. In contrast to the aforementioned AD approaches, our approach CATS considers temporal similarity using a temporal loss to better discriminate anomalous time series windows.

III. Proposed Method: CATS

In this section, we first formulate the multivariate time series anomaly detection problem. We then present the architecture of CATS and provide a detailed description of each of its components. In particular, we explain the data augmentation techniques employed in our work, introduce and describe the temporal contrastive loss, and the global contrastive loss. Finally, we define the computation of the anomaly score, which is utilized for the detection of time series anomalies.

A. Problem Formulation

Given a multivariate time series dataset $X = \{x_1, x_2, ..., x_T\}$, where T is the length of X and $x_t \in \mathbb{R}^m$ denotes a m-dimensional vector corresponding to the values of our m features at time t. The dataset is sliced into sequences of time series windows $W = \{w_1, w_2, ..., w_{T-p+1}\}$ with stride 1 where $w_t = \{x_t, x_{t+1}, ..., x_{t+p-1}\} \in \mathbb{R}^{p \times m}$, p being the window size.

Time series anomaly detection aims at training an unsupervised anomaly detection model $\mathcal{M}$ that given an unknown window time series $\tilde{w_t}$ at inference time will output an anomaly score $s(\tilde{w_t})$. By using this anomaly score and a threshold η, a binary label $\tilde{y_t} \in \{0, 1\}$ is computed which indicates whether a window is anomalous ($\tilde{y_t} = 1$) or not ($\tilde{y_t} = 0$).

B. Model Architecture

The overall architecture of CATS is shown in Fig. 1. Our architecture comprises the following components:

- A stochastic data augmentation module that transforms each input window time series w_t to three views: two positive views $\{w_i^+; w_{i+N}^+\}$ and a negative view w_i^-.
- A siamese network encoder f_θ that learn representations from augmented views of the input window time series $h_i = f_\theta(w_i) \in \mathbb{R}^{p \times d}$ with d the feature size.

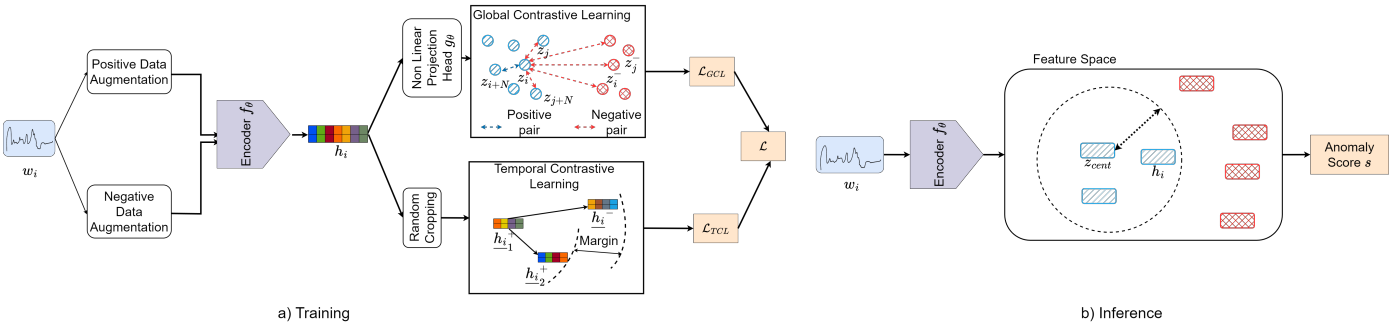

Fig. 1: The proposed architecture of CATS. (a) is the training phase which consists in time series representation learning with temporal contrastive learning (TCL) and global contrastive learning (GCL). (b) is the inference phase where given an unknown window time series $\tilde{w_i}$ its anomaly score is computed as the distance between its latent representation $\tilde{h_i}$ and the centroid of training samples z_{cent}.

- A projection head g_θ that projects the latent representations h_i to a projection space $z_i = g_\theta(h_i)$ due to its importance for contrastive learning [25].
- A global contrastive loss (GCL) that performs contrastive learning on projection space giving the set of (z_i) from the batch using an improvement of NT-Xent loss [35].
- Random cropping is applied on feature space representations h_i to built temporal pairs for the temporal contrastive learning.
- A temporal contrastive loss (TCL) to enforce temporal similarity between cropped versions ($\underline{h_i} \in \mathbb{R}^{k \times d}$) ($k$ the crop size) of latent vectors and a triplet loss.

The model is trained using a contrastive loss model combining the temporal and the global contrastive losses defined as follows:

$$\mathcal{L} = \alpha.\mathcal{L}_{TCL} + \beta.\mathcal{L}_{GCL} \tag{1}$$

where α and β are two hyper-parameters representing the relative weight of each loss.

At inference time, the projector g_θ is discarded and a anomaly score is obtained by computing the distance between an unseen window and the center of latent representations.

C. Data augmentation

Data augmentation plays a crucial role in contrastive learning, facilitating the learning process. While diverse data augmentation techniques are available in the computer vision domains, selecting appropriate data augmentation methods for time series remains challenging. Building upon previous works that have utilized or evaluated data augmentation for time series [25], [36], we define a set of positive data augmentations $\mathcal{D}^+_{aug} = \{d^+_1, \ldots, d^+_{n^+_{aug}}\}$ which are as follows, given a window time series $w = (x_1, ..., x_p)$:

- Jitter transformation adds a i.i.d Gaussian noise with zero mean and variance σ^2 to the time series.

$$d_{jitter}(w) = w + \mathcal{N}(0, \sigma^2) \tag{2}$$

- Scaling transformation changes the global magnitude of a portion P of the time series by multiplying all the values by a scaling factor α.

$$d_{scaling}(w) = \alpha_{scaling} * w_P \tag{3}$$

In the context of unsupervised learning for anomaly detection, the focus is typically on modeling normality using only normal instances during training, and anomalous instances are only encountered during inference. We use negative data augmentations to incorporate weak supervision during the representation learning stage, thus providing prior information about what does not constitute normal data [32]–[34], [37]. This approach enhances the learning process by incorporating knowledge of abnormal patterns during training. We thus define a a set of negative data augmentations $\mathcal{D}^-_{aug} = \{d^-_1, \ldots, d^-_{n^-_{aug}}\}$ which are as follows:

- Mask transformation consists in randomly masking some points of the time series.

$$d_{mask}(w) = (\widehat{x_1}, ..., \widehat{x_p}) \text{ with } \widehat{x_i} = 0 \tag{4}$$

- Trend transformation applies a linear drift w_{drift} to the time series to simulate a shift in the trend of the time series.

$$d_{trend}(w) = w + w_{drift} \tag{5}$$

Thus, given a time series window w_i from a batch $\mathcal{B}$ of size N, we generate two positive views w^+_i and w^+_{i+N} and one negative view w^-_i resulting in a batch of positive samples $\mathcal{B}^+ = (w^+_i)^{2N}_{i=1}$ and a batch of negative samples $\mathcal{B}^- = (w^-_i)^N_{i=1}$.

However, the range of anomalies present in time series data is usually large, and it is not feasible to expect negative data augmentations to generate all possible anomalies. Therefore, similar to previous studies [32]–[34], random data augmentation is employed in this paper. Specifically for each window w_t, and a selected data augmentation $d^-_i \in \mathcal{D}^-_{aug}$, diversity is enforced through the hyperparameters of d_i that control the augmentation process. These hyperparameters include the ratio of features n_{feat} that will be augmented and the ratio of time

points n_t that will be augmented per features. Given n_{feat} (resp. n_t), and a given time series window w_t, a random subset of features (resp. time steps) will be chosen for augmentation. This approach allows for increased variability and diversity in the generated augmented samples, enhancing the learning process for anomaly detection in multivariate time series data.

D. Temporal Contrastive Learning

One shortcoming of previous anomaly detection approaches using contrastive learning is that they do not exploit temporal dependencies for contrasting. We address this limitation by proposing a novel Temporal Contrastive Loss (TCL). TCL aims at learning temporal properties, and clusters time series window that are temporally similar while pushing away windows that are dissimilar. To achieve that representation in the feature space, we use DTW (Dynamic Time Warping) [38] as a time series similarity measure that is more suited for time series forecasting or clustering than classical Euclidean distance.

DTW similarity measure aims at minimizing the Euclidean distance between aligned time series under all possible temporal alignments. However, DTW measure is not differentiable and is not suitable for gradient-based algorithms. To overcome this limitation, Soft-DTW [39] was proposed to smooth DTW and make it differentiable everywhere and then can be used as a loss function or similarity measure. In this work, we choose a Soft-DTW variant called Soft-DTW divergence [40] that unlike the former is positive and minimized when the time series are equal. Specifically given two time series x_i and x_j, Soft-DTW divergence is defined as follows:

$$D^{\gamma}(x_i, x_j) = DTW^{\gamma}_{Soft}(x_i, x_j) - \frac{1}{2}(DTW^{\gamma}_{Soft}(x_i, x_i) + DTW^{\gamma}_{Soft}(x_j, x_j)) \tag{6}$$

with $DTW^{\gamma}_{Soft}(.)$ being the Soft-DTW measure and γ a smoothing parameter.

To enhance the temporal contrastive learning, we build triplets using cropped versions (subsets of time series windows as defined in [14]) of the two positive views and a crop version of the negative view. Specifically, we apply random cropping on a positive view $h_i^+ \in \mathbb{R}^{p \times d}$ to generate two positive subseries $\underline{h_i}_1^+$ and $\underline{h_i}_2^+ \in \mathbb{R}^{k \times m}$ where $k < p$. We do the same process on a negative view h_i^- to obtain a negative subseries $\underline{h_i}^-$. The rationale behind building triplets in this way, instead of using the views resulting from positive and negative data augmentations to build them, is to ensure that the two positive subseries will be *temporally* similar and *temporally* distant from the negative subseries. Since we consider only normal time series windows to build positive pairs, we avoid the limitation of the cropping strategy raised by [14].

Given this similarity measure, and a triplet of latent representations $\{\underline{h_i}_1^+; \underline{h_i}_2^+; \underline{h_i}^-\}$ TCL is defined as follows:

$$\mathcal{L}_{TCL} = \frac{1}{N}\sum_{i=1}^{N} max(D^{\gamma}(\underline{h_i}_1^+ - \underline{h_i}_2^+) - D^{\gamma}(\underline{h_i}_1^+ - \underline{h_i}^-) + m; 0) \tag{7}$$

where $D^{\gamma}(.)$ is the Soft-DTW divergence measure, m the margin (minimum distance that must be kept between positive samples and negative samples). One advantage of Soft-DTW divergence measure is that it can be applied to time series of different sizes and then our TCL can be computed using the cropped versions of the latent vectors. However, the computation of Soft-DTW has a quadratic time complexity which can increase the training time.

E. Global Contrastive Learning

We define a Global Contrastive Loss to learn representations at the instance level. This loss improves the NT-Xent loss by considering more negative pairs for contrastive learning. Traditionally, given an instance z_i in a batch of size N, CL models using this loss contrast one positive pair (z_i, z_i^+) i.e., two views from the same instance to $N-1$ negative pairs (z_i, z_j^+) $\forall j \neq i$ i.e., pairs with views of different instances.

However, in anomaly detection tasks that are performed on normality assumptions, the training data mostly belong to the same class, i.e., normal data. To enhance, the representation learning, we include views coming from negative data augmentation. Specifically, we form a negative pairs by computing the similarity between an instance and all negative views of all the instances in the batch (z_i, z_j^-) $\forall j \in [|1; N|]$. Consequently, we get one positive pair and $2N-1$ negative pairs for each z_i in the batch. Then, the GCL is expressed as follows:

$$\mathcal{L}_{GCL} = \frac{1}{2N}\sum_{i \in \mathcal{B}^+} log\frac{exp(sim(z_i, z_{i+N})/\tau)}{\sum_{j \in \mathcal{B} \text{ and } j \neq i} exp(sim(z_i, z_j)/\tau)} \tag{8}$$

with $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$, N the batch size, τ the temperature hyperparameter and $sim(.)$ the cosine similarity.

F. Anomaly score

After model training, we discard the projection head g_θ and we only use the encoder f_θ for our downstream task since h_i feature have more information for contrastive learning than z_i [25]. To identify anomalies, we follow the same assumptions as one-class classifiers and we expect the normal data to be clustered and anomalies to lies away from that cluster. Hence, given a window from the test set $\tilde{w_t}$, we define the anomaly score $s(.)$ as follows:

$$s(\tilde{w_t}) = \mathcal{D}(f_\theta(\tilde{w_t}), z_{cent}) = \mathcal{D}(\tilde{z_t}, z_{cent}) \tag{9}$$

$$z_{cent} = \frac{1}{N_{train}}\sum_{i=1}^{N_{train}} w_i \tag{10}$$

where $\mathcal{D}(.)$ is the L2-distance measure function, z_{cent} is the centroid of latent features of the training set and N_{train} is the size of the training set.

TABLE I: Datasets summary.

	Dataset	Train	Test	Dimensions	Anomalies (%)
Benchmark	SMD	708405	708420	28*38	4.16
	SMAP	135183	427617	55*25	13.13
	MSL	58317	73729	27*55	10.72
Cloud Gaming	STD	80486	169706	14	52.57
	GFN	27415	22667	14	55.36
	XC	83611	17918	14	24.32

One advantage of using Eq. 9 as anomaly score is that z_{cent} can be computed offline and stored allowing lower inference time.

IV. Experiments

This section begins by outlining the experimental setup, including the datasets and anomaly detection (AD) models used for comparison. The experiments focus on evaluating CATS's accuracy in anomaly detection against other models across different datasets. Additionally, we conduct an ablation study on CATS's components, examine its robustness to data contamination, and analyze the effects of hyper-parameters.

A. Dataset description

We evaluate the performance of our technique on three public anomaly detection benchmark datasets to confirm that our approach maintains its effectiveness when applied to them. The selected datasets are listed as follows: i) SMD dataset (Server Machine Dataset) which consists of 38 sensors continuously monitored during 10 days collected on 28 servers. ii) MSL dataset (Mars Science Laboratory) and iii) SMAP (Soil Moisture Active Passive) are two datasets collected from a monitoring system. The benchmark datasets (SMD, MSL and SMAP) are multi-entity datasets (contain different subdatasets)

We also evaluate the model on case study datasets specifically collected for this purpose. The case study datasets are multivariate time-series features collected on a cloud gaming testbed for QoE degradation detection in low-latency applications [41]. These datasets consist of QoE (Quality of Experience) and QoS (Quality of Service) time series collected on three cloud gaming platforms: iv) STD (Stadia from Google), v) GFN (GeForceNow from NVIDIA) and vi) XC (Xbox Cloud from Microsoft).

B. Benchmark AD models

We compare our models against state-of-the-art algorithms or traditional algorithms mostly used in previous studies for anomaly detection in time series data.

- **Shallow machine learning algorithms**: we use Isolation Forest (IF) [8] that isolate anomalies using features values.
- **Unsupervised time-series anomalies detection**: We select i) Deep-SVDD [6], ii) Auto-Encoder (AE), iii) UnSupervised Anomaly Detection (USAD) [1] that reconstruct normal data and use reconstruction error to detect anomalies.
- **Contrastive learning algorithms**: We use the following contrastive learning architectures from i) SimCLR [25], ii) SimSiam [27], iii) TS2Vec [14].

C. Implementation details

We choose as encoder f_θ, the same encoder architecture as TS2Vec, which consists of a dilated CNN module with ten residual blocks of 1D convolutional layers. The projection head g_θ is a three-layer MLP with ReLU activation. The embedding size and projection size are 100 and 50 for use-case datasets and 128 and 128 for benchmark datasets respectively. CATS is trained for 100 epochs with a batch size of 512, using Adam optimizer with a learning rate of 10^{-3}, a weight decay of 10^{-5} and the learning rate is decayed using cosine decay schedule. GCL loss temperature parameter τ is set to 0.1, TCL margin to 5 and TCL smooth parameter γ to 1 in all experiments. We use $\alpha = \beta = 0.5$. The positive data augmentations are *jitter* and *scaling* and the negative data augmentations used are *trend* and *mask*.

To allow a fair comparison with contrastive learning benchmark methods, we adopt the same encoder to learn representations, the same positive data augmentation and the same anomaly score procedure for all benchmarks (cf. in Section III-F). All deep learning architectures are trained using the same aforementioned hyperparameters.

All experiments are performed on a workstation with the following specifications: Ubuntu 22.04, Intel(R) Xeon(R) W-2235 CPU @ 3.8GHz with 32 GB of RAM, NVIDIA GeForce RTX 3090 Ti with 24GB, Python 3.10.6, PyTorch 2.2.0 and CUDA 12.1. The datasets and the code to reproduce all the experiments are provided.[1]

D. Evaluation metrics

The performance of unsupervised anomaly detection models is usually assessed using well-known Precision (P), Recall (R) and F1-Score (F1) metrics. Since the values of those metrics are threshold-dependent, some studies prefer threshold-independant metrics like AUC (Area Under the Curve) or AUPR (Area Under Precision-Recall curve). Due to the limitations of F1-score [42] for binary classification, we also include MCC (Matthews Coefficient Correlation) metric in our evaluation.

We then report the performance results using AUC, F1 and MCC. Point Adjust (PA) approach is not used in our evaluation, despite being very popular and used in several time series AD studies, since it was shown [32], [34], [41] to overestimate the effectiveness of time series models.

E. Performance comparison

Table II presents the anomaly detection performance of CATS in comparison to other methods, evaluated using the AUC, F1, and MCC metrics.

[1] https://github.com/joelromanky/cats

TABLE II: Performance comparison on the datasets. Mean and standard deviation computed over all entities for benchmark datasets and over five runs for case-study datasets. Bold values indicate best results.

		Models	IForest	Deep-SVDD	AE	USAD	SimCLR	SimSiam	TS2Vec	CATS
Benchmark datasets	SMD	AUC	77.10$_{(\pm 11.9)}$	75.31$_{(\pm 14.5)}$	81.33$_{(\pm 13.2)}$	81.08$_{(\pm 12.5)}$	80.83$_{(\pm 14.7)}$	77.26$_{(\pm 14.9)}$	74.25$_{(\pm 16.6)}$	**82.21$_{(\pm 14.3)}$**
		F1	29.88$_{(\pm 20.6)}$	34.75$_{(\pm 21.5)}$	46.00$_{(\pm 24.3)}$	46.62$_{(\pm 26.3)}$	46.51$_{(\pm 25.7)}$	41.82$_{(\pm 25.3)}$	43.18$_{(\pm 25.9)}$	**50.65$_{(\pm 23.6)}$**
		MCC	29.62$_{(\pm 20.8)}$	36.25$_{(\pm 22.0)}$	47.00$_{(\pm 24.1)}$	47.98$_{(\pm 25.1)}$	48.06$_{(\pm 24.2)}$	43.24$_{(\pm 24.9)}$	44.95$_{(\pm 24.7)}$	**50.85$_{(\pm 23.6)}$**
	MSL	AUC	56.94$_{(\pm 14.1)}$	61.38$_{(\pm 17.1)}$	62.30$_{(\pm 16.1)}$	63.31$_{(\pm 14.3)}$	61.09$_{(\pm 15.4)}$	62.07$_{(\pm 14.3)}$	63.95$_{(\pm 15.0)}$	**64.98$_{(\pm 15.7)}$**
		F1	21.24$_{(\pm 21.4)}$	27.93$_{(\pm 25.6)}$	26.02$_{(\pm 22.9)}$	27.16$_{(\pm 23.0)}$	25.72$_{(\pm 23.1)}$	23.78$_{(\pm 23.2)}$	28.43$_{(\pm 24.5)}$	**29.15$_{(\pm 24.2)}$**
		MCC	11.09$_{(\pm 21.8)}$	19.24$_{(\pm 29.2)}$	16.49$_{(\pm 24.4)}$	17.33$_{(\pm 24.8)}$	16.30$_{(\pm 25.1)}$	14.11$_{(\pm 24.0)}$	19.86$_{(\pm 24.8)}$	**20.14$_{(\pm 27.8)}$**
	SMAP	AUC	56.98$_{(\pm 17.3)}$	62.52$_{(\pm 19.1)}$	**64.30$_{(\pm 19.6)}$**	61.11$_{(\pm 19.4)}$	63.99$_{(\pm 17.7)}$	62.12$_{(\pm 17.1)}$	61.42$_{(\pm 20.3)}$	64.07$_{(\pm 18.6)}$
		F1	22.80$_{(\pm 27.2)}$	29.20$_{(\pm 33.0)}$	28.93$_{(\pm 33.5)}$	**30.10$_{(\pm 33.1)}$**	28.23$_{(\pm 32.2)}$	27.46$_{(\pm 33.2)}$	28.26$_{(\pm 33.2)}$	29.07$_{(\pm 29.07)}$
		MCC	11.38$_{(\pm 29.0)}$	23.93$_{(\pm 33.1)}$	23.96$_{(\pm 34.0)}$	23.66$_{(\pm 34.9)}$	22.52$_{(\pm 32.2)}$	21.44$_{(\pm 32.5)}$	23.5$_{(\pm 32.8)}$	**24.28$_{(\pm 32.7)}$**
Case-study datasets	STD	AUC	74.57$_{(\pm 1.63)}$	91.19$_{(\pm 1.08)}$	96.04$_{(\pm 0.27)}$	96.09$_{(\pm 0.08)}$	95.78$_{(\pm 0.39)}$	75.65$_{(\pm 11.3)}$	95.63$_{(\pm 1.94)}$	**97.93$_{(\pm 0.13)}$**
		F1	75.79$_{(\pm 1.42)}$	87.18$_{(\pm 1.24)}$	90.35$_{(\pm 0.51)}$	90.02$_{(\pm 0.24)}$	90.15$_{(\pm 0.52)}$	74.21$_{(\pm 9.22)}$	92.83$_{(\pm 1.92)}$	**94.06$_{(\pm 0.45)}$**
		MCC	39.56$_{(\pm 3.66)}$	71.83$_{(\pm 2.77)}$	78.93$_{(\pm 1.14)}$	77.89$_{(\pm 0.36)}$	78.48$_{(\pm 1.17)}$	39.31$_{(\pm 19.8)}$	84.33$_{(\pm 4.12)}$	**86.72$_{(\pm 0.88)}$**
	GFN	AUC	61.97$_{(\pm 0.87)}$	71.78$_{(\pm 3.41)}$	74.05$_{(\pm 0.84)}$	74.84$_{(\pm 0.42)}$	78.50$_{(\pm 1.95)}$	67.07$_{(\pm 3.25)}$	74.91$_{(\pm 4.32)}$	**84.35$_{(\pm 1.23)}$**
		F1	74.12$_{(\pm 0.71)}$	75.51$_{(\pm 2.11)}$	74.05$_{(\pm 0.84)}$	77.80$_{(\pm 0.38)}$	81.20$_{(\pm 2.61)}$	74.25$_{(\pm 2.93)}$	76.76$_{(\pm 2.71)}$	**82.88$_{(\pm 0.96)}$**
		MCC	17.07$_{(\pm 1.27)}$	24.26$_{(\pm 6.56)}$	28.08$_{(\pm 0.14)}$	31.40$_{(\pm 1.22)}$	37.46$_{(\pm 3.87)}$	17.86$_{(\pm 6.27)}$	28.19$_{(\pm 8.39)}$	**47.27$_{(\pm 1.49)}$**
	XC	AUC	78.71$_{(\pm 1.13)}$	67.32$_{(\pm 6.52)}$	89.18$_{(\pm 2.31)}$	89.97$_{(\pm 0.26)}$	85.81$_{(\pm 3.17)}$	83.35$_{(\pm 10.6)}$	**96.96$_{(\pm 1.36)}$**	96.10$_{(\pm 0.41)}$
		F1	63.33$_{(\pm 1.18)}$	50.83$_{(\pm 7.69)}$	75.94$_{(\pm 3.30)}$	77.59$_{(\pm 0.58)}$	70.58$_{(\pm 3.45)}$	69.09$_{(\pm 13.4)}$	**89.60$_{(\pm 2.03)}$**	86.69$_{(\pm 0.83)}$
		MCC	43.42$_{(\pm 2.43)}$	27.40$_{(\pm 11.4)}$	63.95$_{(\pm 4.63)}$	65.35$_{(\pm 0.72)}$	56.59$_{(\pm 4.89)}$	52.30$_{(\pm 21.3)}$	**84.07$_{(\pm 2.94)}$**	79.67$_{(\pm 0.11)}$

1) Benchmark datasets: Since the benchmark datasets consist of multiple sub-datasets, we report the average performance of each model across all sub-datasets. On average, CATS demonstrates superior performance on SMD and MSL, and ranks second-best on the SMAP. The results in this study are reported using the standard F1-score instead of the more commonly used F1-PA metric, which has a tendency to inflate performance estimates [32], [34], [41]. Despite yielding lower values, the standard F1-score offers a more reliable and consistent basis for comparison across datasets. Consequently, our reported scores are below those documented in previous works on time series anomaly detection which used F1-PA metric. However, they are consistent with those of other studies that have used the standard F1-score [34]. Moreover, the benchmark datasets contain sub-datasets with varying levels of anomaly detection difficulty (e.g., models report a standard F1-score of 0 on some sub-datasets, while achieving scores as high as 90% on others). Consequently, averaging results across all sub-datasets leads to overall scores with large standard deviations.

2) Case-study datasets: The reported outcomes on the case study datasets represent the average of five independent runs, ensuring fair comparisons across methods using the same evaluation metrics as previously mentioned. CATS achieves the best performance on all datasets, except for the XC dataset, where it ranks second.

The experimental results show that our model performs on average better than the other methods on all datasets. It also suggests that traditional contrastive learning only is not sufficient for AD since unsupervised models like AE and USAD perform on average better than SimCLR or SimSiam. However, by considering temporal dependencies and introducing knowledge of the anomaly class, the AD performance is enhanced as shown through CATS results.

F. Ablation studies

We conduct two ablation studies. First, we assess the effectiveness of the two loss components $\mathcal{L}_{TCL}$ and $\mathcal{L}_{GCL}$. The results are shown in Table III. We compare our $\mathcal{L}_{GCL}$ to NT-Xent loss $\mathcal{L}_{NTXent}$ that performs contrastive learning without using negative augmentations and also compare the performance by removing random cropping $\mathcal{L}_{w/o-crop}$

The results show that each loss individually achieves lower score than the combined loss in CATS. On the one hand, GCL outperforms the NT-Xent loss on AD tasks as indicated by the MCC score which fairly reflects the model's ability to distinguish well between normal instances and anomalous instances. On the other hand, TCL individually achieves slightly lower performance than GCL, but when combined with GCL contributes to an overall performance enhancement. This disparity may be attributed to TCL having only one positive pair and one negative pair for each window time series, whereas GCL incorporates one positive pair and $2N-1$ negative pairs. Using multiple negative pairs to enhance the discriminative capacity of TCL comes at the expense of longer training times ($\mathcal{O}(n^4)$) due to the quadratic time complexity of the Soft-DTW divergence. Thus, we opt for using only one negative pair to maintain reasonable training times. Moreover, the results show that removing the random cropping stage negatively impacts the AD performance.

G. Data contamination impact

To assess the robustness of our model to data contamination, we introduce various levels of anomalies, denoted as $c \in \{0, 4, 8, 12, 20\}$, into the training set. These contamination

TABLE III: Ablation study on loss components.

	GFN		XC	
Loss	**F1**	**MCC**	**F1**	**MCC**
$\mathcal{L}_{NTXent}$	$81.20_{(\pm 2.61)}$	$37.46_{(\pm 3.87)}$	$70.58_{(\pm 3.45)}$	$56.59_{(\pm 4.89)}$
$\mathcal{L}_{GCL}$	$82.52_{(\pm 1.77)}$	$40.73_{(\pm 2.32)}$	$85.68_{(\pm 2.52)}$	$78.31_{(\pm 3.67)}$
$\mathcal{L}_{TCL}$	$79.93_{(\pm 2.69)}$	$38.12_{(\pm 8.32)}$	$76.57_{(\pm 5.68)}$	$65.71_{(\pm 8.34)}$
$\mathcal{L}_{w/o-crop}$	$80.44_{(\pm 2.01)}$	$33.45_{(\pm 1.96)}$	$85.68_{(\pm 2.52)}$	$78.31_{(\pm 3.67)}$
$\mathcal{L}_{GCL} + \mathcal{L}_{TCL}$	$\mathbf{82.88}_{(\pm \mathbf{0.96})}$	$\mathbf{47.27}_{(\pm \mathbf{1.49})}$	$\mathbf{86.69}_{(\pm \mathbf{0.83})}$	$\mathbf{79.67}_{(\pm \mathbf{0.11})}$

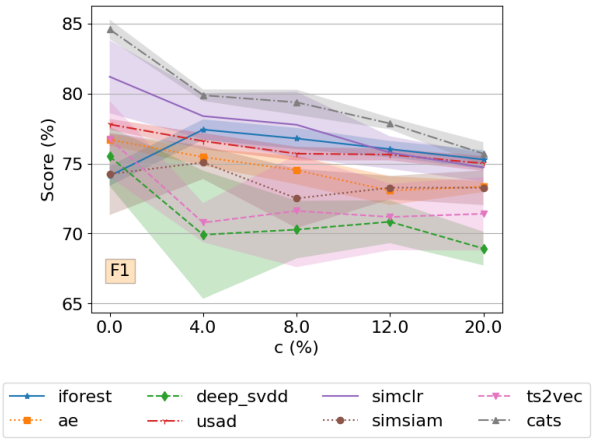

Fig. 2: CATS robustness to data contamination.

rates are chosen to reflect realistic scenarios. Fig. 2 illustrates the F1-score of CATS in comparison to other models on the GFN dataset, considering different levels of data contamination.

Among the compared models, with the exception of the iForest model, the performance of all models deteriorates as the data contamination rate increases. Although CATS is slightly more affected than some models, it consistently demonstrates the highest performance across all contamination levels when compared to the other models even with high contamination rates of up to 20%.

We attribute CATS's behavior to the use of negative data augmentation during the learning process, which helps the model recognize anomalies. However, the extent of this improvement is limited due to the use of synthetic anomalies. Further experiments should be conducted in future work to explore the potential enhancements in robustness that can be achieved by incorporating more data-specific synthetic anomalies.

H. Hyperparameters sensitivity

We study here how some CATS hyperparameters, namely the temperature τ, Soft-DTW smoothing parameter γ, the margin m, the embedding size, the projection size and the batch size may impact the performance. We illustrate that with Fig. 3 on GFN dataset. Our experimental results reveal that the temperature parameter τ and the margin m are the most influential hyperparameters. Lower τ values enable the

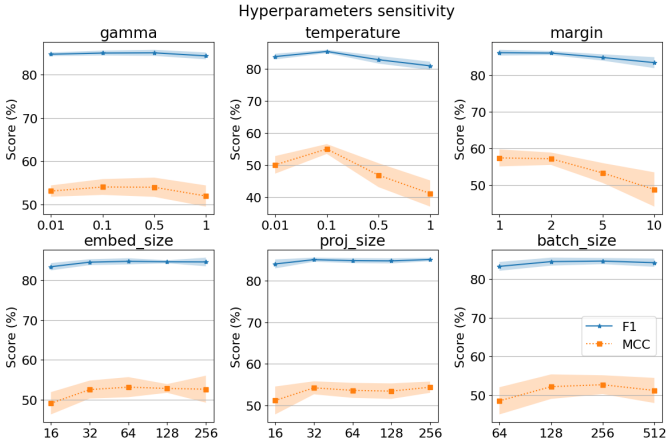

Fig. 3: CATS sensitivity evaluation to its hyperparameters using F1-score and MCC metrics.

model to enhance its learning by focusing on hard-negatives i.e., negative samples that are closer to the positive samples [25]. Conversely, the results show that larger margins lead to suboptimal performance. This is because larger margins penalize negatives that are not distant enough from the anchors and the positives.

Furthermore, our findings indicate that lower embedding sizes, projection sizes and batch sizes reduce the efficacy of CATS model to learn a good representation for AD. These results highlight the importance of the hyperparameters selection for CATS model.

V. Conclusion and Future Work

This paper introduces CATS, a comprehensive end-to-end contrastive learning approach for anomaly detection in time series data. CATS addresses the limitations of traditional contrastive learning methods by incorporating two key components: temporal similarity and negative data augmentation. By employing negative data augmentation, synthetic anomalies are introduced, significantly enhancing the model's capability to effectively detect anomalies. Additionally, CATS introduces a novel DTW-based temporal loss, enabling efficient time series representation learning and capturing the temporal patterns inherent in time series data.

Empirical evaluations conducted on benchmark datasets and use-case datasets demonstrate the significant improvements achieved by CATS in anomaly detection compared to competing unsupervised models. Notably, CATS exhibits superior performance even in scenarios involving data contamination.

In future research, we aim to explore further performance and robustness enhancements in anomaly detection by incorporating multiple negative pairs along with the temporal contrastive loss without incurring higher time complexity and dataset-specific data augmentation.

Acknowledgements

This work is partially funded by a ANR - French government grant under the France 2030 program, project SPIREC of

PEPR Cloud (ANR-23-PECL-0006) and the French National Research Agency (ANR) MOSAICO project, under grant No ANR-19-CE25-0012.

REFERENCES

[1] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[2] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1544–1551, 2018.

[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," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[4] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," *Proceedings of the 2018 World Wide Web Conference*, 2018.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *SIGMOD '00*, 2000.

[6] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *ICML*, 2018.

[7] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *NIPS*, 1999.

[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[9] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. R. Chandrasekhar, "Adversarially learned anomaly detection," *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 727–736, 2018.

[10] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "Beatgan: Anomalous rhythm detection using adversarially generated time series," in *International Joint Conference on Artificial Intelligence*, 2019.

[11] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[12] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," in *International Joint Conference on Artificial Intelligence*, 2021.

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

[14] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *AAAI Conference on Artificial Intelligence*, 2021.

[15] X. Zheng, X.-Y. Chen, M. Schurch, A. Mollaysa, A. Allam, and M. Krauthammer, "Simts: Rethinking contrastive representation learning for time series forecasting," *ArXiv*, vol. abs/2303.18205, 2023.

[16] R. C. Paffenroth, K. Kay, and L. D. Servi, "Robust pca for anomaly detection in cyber networks," *ArXiv*, vol. abs/1801.01571, 2018.

[17] W. A. Chaovalitwongse, Y. J. Fan, and R. C. Sachdeo, "On the time series k-nearest neighbor classification of abnormal brain activity," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, pp. 1005–1016, 2007.

[18] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*, 2019.

[19] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "Tadgan: Time series anomaly detection using generative adversarial networks," in *2020 ieee international conference on big data (big data)*. IEEE, 2020, pp. 33–43.

[20] S. Qin, J. Zhu, D. Wang, L. Ou, H. Gui, and G. Tao, "Decomposed transformer with frequency attention for multivariate time series anomaly detection," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 1090–1098.

[21] 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.

[22] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *ArXiv*, vol. abs/1701.04862, 2017.

[23] B. Zhuang, J. Liu, Z. Pan, H. He, Y. Weng, and C. Shen, "A survey on efficient training of transformers," *arXiv preprint arXiv:2302.01107*, 2023.

[24] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 9912–9924.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2019.

[27] X. Chen and K. He, "Exploring simple siamese representation learning," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 745–15 753, 2020.

[28] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[29] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *International Conference on Learning Representations*, 2021.

[30] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural transformation learning for deep anomaly detection beyond images," in *International conference on machine learning*. PMLR, 2021, pp. 8703–8714.

[31] 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," in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 2023, pp. 694–702.

[32] H. Kim, S. Kim, S. Min, and B. Lee, "Contrastive time-series anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[33] B. Li and E. Müller, "Contrastive time series anomaly detection by temporal transformations," *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2023.

[34] Z. Z. Darban, G. I. Webb, S. Pan, and M. Salehi, "Carla: A self-supervised contrastive representation learning approach for time series anomaly detection," *arXiv preprint arXiv:2308.09296*, 2023.

[35] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Neural Information Processing Systems*, 2016.

[36] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *PLoS ONE*, vol. 16, 2020.

[37] A. Sinha, K. Ayush, J. Song, B. Uzkent, H. Jin, and S. Ermon, "Negative data augmentation," in *International Conference on Learning Representations*, 2021.

[38] H. Sakoe, "Dynamic-programming approach to continuous speech recognition," in *1971 Proc. the International Congress of Acoustics, Budapest*, 1971.

[39] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *International Conference on Machine Learning*, 2017.

[40] M. Blondel, A. Mensch, and J.-P. Vert, "Differentiable divergences between time series," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3853–3861.

[41] J. R. Ky, B. Mathieu, A. Lahmadi, and R. Boutaba, "Ml models for detecting qoe degradation in low-latency applications: A cloud-gaming case study," *IEEE Transactions on Network and Service Management*, vol. 20, pp. 2295–2308, 2023.

[42] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, 2020.