CutAddPaste: Time Series Anomaly Detection by Exploiting Abnormal Knowledge

Rui Wang
School of Computer Science and Engineering, Beihang University
Beijing, China
ruiking@buaa.edu.cn

Xudong Mou
School of Computer Science and Engineering, Beihang University
Beijing, China
mxd@buaa.edu.cn

Renyu Yang*
School of Software, Beihang University
Beijing, China
renyuyang@buaa.edu.cn

Kai Gao
Institute of Future Cities, The Chinese University of Hong Kong
Hong Kong, China
kaigao@cuhk.edu.hk

Pin Liu
School of Information Engineering, China University of Geosciences
Beijing, China
liupin@cugb.edu.cn

Chongwei Liu
Kuaishou Inc.
Beijing, China
liuchongwei@kuaishou.com

Tianyu Wo
School of Software, Beihang University
Beijing, China
Zhongguancun Laboratory
Beijing, China
woty@buaa.edu.cn

Xudong Liu
School of Computer Science and Engineering, Beihang University
Beijing, China
Zhongguancun Laboratory
Beijing, China
liuxd@buaa.edu.cn

ABSTRACT

Detecting time-series anomalies is extremely intricate due to the rarity of anomalies and imbalanced sample categories, which often result in costly and challenging anomaly labeling. Most of the existing approaches largely depend on assumptions of normality, overlooking labeled abnormal samples. While anomaly assumptions based methods can incorporate prior knowledge of anomalies for data augmentation in training classifiers, the adopted random or coarse-grained augmentation approaches solely focus on pointwise anomalies and lack cutting-edge domain knowledge, making them less likely to achieve better performance. This paper introduces CutAddPaste, a novel anomaly assumption-based approach for detecting time-series anomalies. It primarily employs a data augmentation strategy to generate pseudo anomalies, by exploiting prior knowledge of anomalies as much as possible. At the core of CutAddPaste is cutting patches from random positions in temporal subsequence samples, adding linear trend terms, and pasting them into other samples, so that it can well approximate a variety of anomalies, including point and pattern anomalies. Experiments on standard benchmark datasets demonstrate that our method outperforms the state-of-the-art approaches.

CCS CONCEPTS

• **Computing methodologies → Anomaly detection**; Neural networks; • **Mathematics of computing →** *Time series analysis.*

KEYWORDS

Anomaly detection; abnormal knowledge; time series; data augmentation; anomaly-assumption

ACM Reference Format:
Rui Wang, Xudong Mou, Renyu Yang, Kai Gao, Pin Liu, Chongwei Liu, Tianyu Wo, and Xudong Liu. 2024. CutAddPaste: Time Series Anomaly Detection by Exploiting Abnormal Knowledge. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671739

1 INTRODUCTION

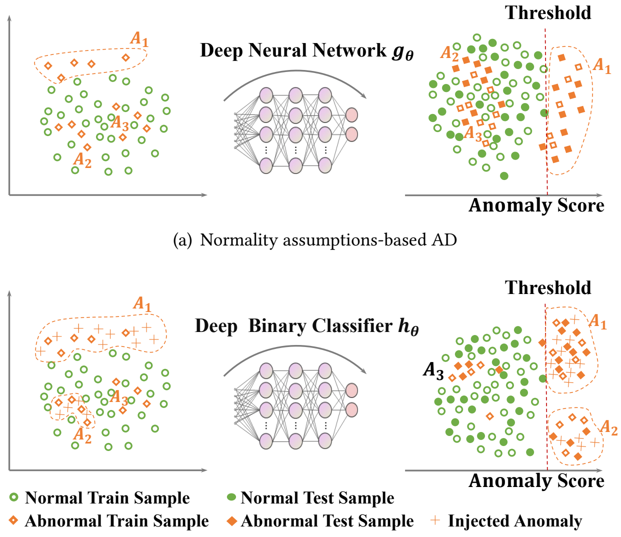
The primary task of anomaly detection (AD) in time-series data is detecting unusual time segments that are significantly dissimilar to the majority. Effective AD detection is the cornerstone of such applications as industrial devices, network intrusion detection, monitoring patient vital signs, etc. Different from binary classification, AD aims to identify events that rarely happen – in such circumstances, anomalous instances are rather sparse compared with normal ones. In addition, accurately labeled samples are far from sufficient, since labeling time-series anomalies usually requires a huge amount of expertise and effort that is expensive and impractical in anomaly detection tasks at scale such as data-intensive deep models.

Treating AD as an unsupervised learning procedure has therefore become a common practice. Most of the existing work – including autoencoder [1], one-class classification [2], and one-class

[*] Corresponding author (renyuyang@buaa.edu.cn)

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671739



(a) Normality assumptions-based AD

(b) Anomaly assumptions-based AD

Figure 1: Schematic of normality and anomaly assumptions used in AD approaches. Anomaly subgroups A_1, A_2, and A_3 exist. The red dotted line is the predefined threshold. a) is proficient in identifying anomalies out of distribution, like A_1. b) could identify anomalies mixed within the normal samples with the aid of augmentations, like A_2, but unable to do so when augmentations are absent, like A_3.

contrastive [3] – are established on the basis of *single/multiple normality assumptions*. The key insight is that none or only a few anomalous samples are engaged in the training set. As shown in Fig. 1(a), these methods treat all unlabeled training data as normal and depend on various pretext tasks to learn representations of normal samples. Obviously, inherent semantic information behind some abnormal instances (e.g., a small portion of anomalies annotated by domain experts) within the training set is under-exploited, and will lead to inferior detection effectiveness.

To capitalize on such labeled anomalies, there is a huge body of follow-up studies, including Outlier Exposure (OE) [4], CutPaste [5], and Deep SAD [6]. They are based on *anomaly assumptions* and, in practice, generated anomalous samples, guided by prior abnormal knowledge, to train deep classifiers (see Fig. 1(b)). However, all of them are dedicated to resolving problems in the image domain, and cannot be directly applied in time-series anomaly detection (TSAD) models, because the temporal dependencies and types of anomalies in time series are entirely different. NCAD [7] is among the very few attempts to support time-series AD. Inspired by OE, it incorporates contextual random point outliers to generate mocked anomalies. However, the existing anomaly assumptions-based techniques solely generate random anomalies or point anomalies by adopting a simplistic and rudimentary, overlooking sophisticated prior knowledge of anomalies. Such naive augmentations are unrealistic and can hardly increase the accuracy and efficacy of anomaly detection. In reality, time-series anomalies manifest in an ever more heterogeneous manner – in addition to random and point anomalies, there are many other types of anomalies, particularly the point-wise (global or contextual) and pattern-wise (shapelet, seasonal, or trend) anomalies[8], which have not yet been addressed by the existing approaches. As a special case of shapelet [9], the correlation of multivariant time series is also ignored.

This paper presents, CutAddPaste, a novel anomaly assumption-based solution empowered by a series of data augmentation techniques with richer abnormal prior knowledge. The main goal is to create irregular subsequences that can act as an approximation of genuine point-wise and pattern-wise anomalies. To achieve this, we *cut* a patch from a random position within a time-series subsequence sample, *add* a linear trend component, and finally *paste* it into an arbitrary location within another sample. Afterward, the original time series and augmented counterparts are fed into a Temporal Convolution Network (TCN) layer and a projection layer to calculate a cross-entropy loss with the corresponding label. We conduct extensive experiments on four datasets, including AIOps, UCR, SWaT, and WADI, and employ Revised Point Adjusted (RPA) metrics [10] as the performance indicator. Experiment results show that our approach outperforms other baselines in TSAD tasks. Extensive results using other metrics such as Point Ajusted (PA) metrics [11] are detailed in the Appendix to justify the fairness of RPA.

In particular, this work makes the following contributions:

- To the best of our knowledge, CutAddPaste is the first work that can address complex anomaly augmentation for TSAD deep models, instead of simply using naive point injection.
- CutAddPaste is an elegantly simple yet potent data augmentation technique tailored for TSAD. Utilizing anomaly knowledge, CutAddPaste broadens the scope beyond point-wise anomalies to encompass pattern-wise anomalies like shapelets, seasonality, and trends.
- Experiments on 4 real-world datasets that cover both univariate and multivariate data demonstrate the supreme performance of our method. A comparison among six metrics also indicates the recommendation of using the RPA metric.

2 RELATED WORK

Unsupervised deep learning methods typically derive the foundation for anomaly detection from extensive datasets. Among these, the most prevalent approach involves extracting representations of normal samples, often referred to as the "normality assumption" [12]. Correspondingly, certain methods focus on discerning the distinctive features of anomalies. We categorize these techniques under the umbrella of "methods based on anomaly assumptions." A concise overview of these two categories follows.

Normality assumption based methods. Some approaches operate under the premise that the majority of collected training data are normal. They construct intricate representations of so-called *normality*, designating the data not aligning with these representations as abnormal. GANs-based [13, 14], autoencoder-based [1], one-class-based [2], and clustering-based [15] methods are the representatives. Most notably, autoencoder-based methods assume normal samples are better reconstructed from the latent space than anomalies. While they provide diverse insights into normality, they may entail certain limitations and biases in capturing its entirety. Researchers have endeavored to integrate these assumptions for better-encompassing normal features. Some methods, such as [16], adopt a two-stage mechanism – extracting a representation of the entire dataset and leveraging the intrinsic normality to identify anomalies. To eliminate the potential negative impact of features extraneous to AD, [3] and [17] combine some preceding assumptions into a unified process, constructing an all-encompassing portrayal of normality. Nevertheless, it remains challenging to ascertain whether this fusion captures the full extent of normality, and an increased amalgamation of assumptions will even worsen the model's convergence issue. Additionally, underutilizing valuable prior knowledge will hinder the effectiveness of model training, leaving the learning procedure from scratch.

Anomaly assumption based methods. While normality might prove elusive, researchers possess some insight into abnormal states. They harness random pseudo anomalies to train classifiers. Outlier Exposure (OE) [4] integrates data from external datasets during training, exposing the model to out-of-distribution instances to learn a more conservative concept of normal samples, and thus can boost the detection of uncharted anomaly patterns. Deep Semi-supervised Anomaly Detection (Deep SAD) [6], an extension of Deep SVDD [2], incorporates known anomalies in the model and can exhibit a higher cross-entropy within the latent distribution than normal samples. In the Computer Vision (CV) context, CutPaste [5] employs data augmentation techniques – e.g., involving the excision of an image patch and placement in a random location of a larger image and producing spatial irregularities – to offer a rudimentary simulation of image defects. [18] demonstrates that relatively few random OE samples are necessary to yield state-of-the-art detection performance. NCAD [7] adapts the OE approach to time series scenarios, by infusing mixed contextual and random point anomalies into the time series data, for establishing a discerning boundary between normal and anomalous classes. However, the prevailing methods merely generate pseudo-anomalies in a random or coarse manner, underscoring the importance of domain knowledge. This insight motivated us to improve the "CutPaste" methodology to the time series domain. To do so, we generate a spectrum of anomalies, encompassing anomalous seasons, trends, and shapelets, among other patterns. Table 1 detailed the comparison between our method and other related work from different perspectives.

3 OUR APPROACH

3.1 Problem Definition

Given an ordered time series $\mathcal{S} = \{x_1, x_2, \ldots, x_l\}$ collected during an l-length time period. $x_i \in \mathbb{R}^d$ is a d-dimensional vector collected at timestamp i. The time series will be denoted as univariate if $d = 1$, and as multivariate if $d > 1$. Conventional approaches typically split the long time series $\mathcal{S}$ into a set of time subsequences, i.e., $\mathcal{D} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ by sliding windows whose length is set to be t. A sample $\mathbf{X}_i = \{x_1, x_2, \ldots, x_t\}$ is the collection of points within a time subsequence, and N is the number of samples. Time step $\delta \leq t$ is the stride of sliding, where the samples are overlapping if $\delta < t$. To better describe the characteristics of pattern-wise anomalies, we adopt structural modeling [8] to represent a time-series sample as $X = \Gamma(2\pi\omega T) + \Theta(T)$, where $T = \{1, 2, \ldots, t\}$. Γ defines the basic shapelet function, which not only describes the shape of the time series but also includes relationships between various dimensions. These relationships can be depicted by a covariance matrix $cov(X, X)$. ω is the seasonality and Θ is the trend function describing the direction of X_i. Correspondingly, $\mathcal{D}$ has a set of labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ with $y_i \in \{0, 1\}$ indicating that the sample $\mathbf{X}_i$ is normal (0) or anomalous (1). The goal is to predict a label $\hat{y}_i \in \{0, 1\}$ given a time series $\mathbf{X}_i$. We calculate an anomaly score S_i instead of giving the binary labels directly, and a predicted label can be obtained by comparing S_i to a predefined threshold τ.

Table 1: Comparison of related work. N and A represent whether the method uses normal and abnormal samples. G and C are global and contextual point anomalies. SL, SE, and T are shapelet, seasonal, or trend pattern anomalies.

Method	Domain	N	A	TS Anomaly Types				
				G	C	SL	SE	T
Deep SVDD [2]	Image	✓	×	×	×	×	×	×
OE [4]	Image	✓	✓	×	×	×	×	×
CutPaste [5]	Image	✓	✓	×	×	×	×	×
DeepSAD [6]	Image	✓	✓	×	×	×	×	×
COCA [3]	TS	✓	×	×	×	×	×	×
NCAD [7]	TS	✓	✓	✓	✓	×	×	×
CutAddPaste	TS	✓	✓	✓	✓	✓	✓	✓

3.2 Overview

The core idea of our approach is to enhance a window-based anomaly detection approach through data augmentation. Fig. 2 shows the proposed model architecture, consisting of a CutAddPaste augmentation module for anomaly injection and a deep TSAD model that has an encoder for feature extraction and a projector for vector mapping. CutAddPaste augmentation is devised for the task of TSAD, which, to the best of our knowledge, is the first attempt at injecting all five types of anomalies in the time series. Inspired by CutPaste [5] in the image domain, we extend and achieve it in a window-based manner for coping with time series more efficiently. In addition to the pattern anomalies (shape and seasonal), we introduce a vital trend term to generate trend, correlation, and point-wise anomalies.

In particular, the raw time series are normalized to enable a zero mean and unit variance. We use a t-sized time window to split the time series into several sub-sequences. Some of them are fed into the prior-based data augmentation module CutAddPaste where a set of operations are enforced to generate the pseudo anomalies, e.g., $\mathbf{X}_a$ in Fig. 2. Technically, we cut patches from random positions in temporal subsequence samples, add linear trend terms, and paste them into other samples. The collection of pseudo anomalous samples $\mathcal{D}_P$, together with the real samples $\mathcal{D}_R$ and the labels $\mathcal{Y}$, can be therefore used for further model training and evaluation. The details of CutAddPaste are discussed in Section 3.3.

Afterward, a TCN $f_\theta : \mathcal{D} \mapsto C$ encoder is devised to convert t-sized $\mathbf{X_i}$ into a fixed-size vector representation $\mathbf{z_i}$. It is achieved by three temporal convolutional blocks and each block is composed of a Conv1D layer, a Batch Normalization (BN) layer, a ReLU activation function, and a MaxPool1D layer. The first block contains an additional Dropout layer. The representation $\mathbf{z_i}$ is fed into a

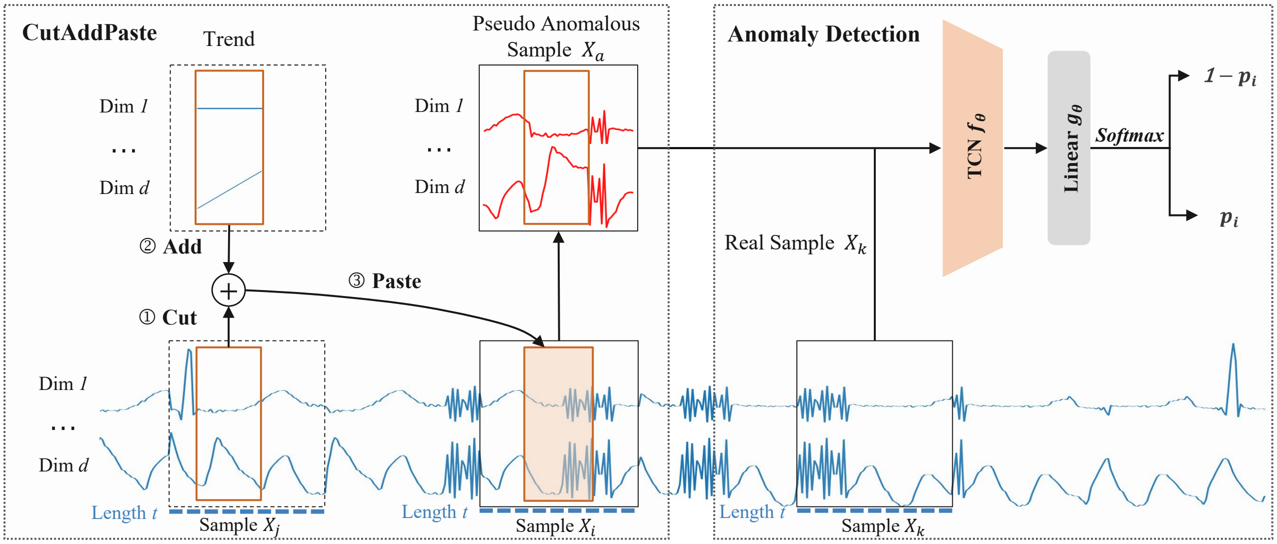


Figure 2: The overview of CutAddPaste. It generates the pseudo anomaly X_a by firstly cutting a patch from sample X_j, adding trend terms on some dimensions of the patch, and finally pasting it on another arbitrary X_i. All the real samples, e.g. X_k, and pseudo anomalous samples are fed into TCN to learn the probability scores of being abnormal.

learnable nonlinear projector $g_\theta : C \mapsto Q$ to output the projection q_i. The projector applies an MLP with one hidden layer that uses BN and ReLU activation functions. It maps representations obtained from the feature encoder into a 2-dimensional projection space. This mapping is guided by calculating the cross-entropy loss, which will be detailed in Section 3.4

3.3 Data Augmentation

Scoping. CutAddPaste aims to augment the most recent types of time series and exploit them as prior knowledge in the downstream detection tasks. Following the definitions of temporal anomalies presented in [8], we delve into both point-wise and pattern-wise abnormal behaviors. Point-wise anomalies refer to the points that significantly deviate from the *global* points or *contextual* points, while pattern-wise anomalies encompass *shapelet*, *seasonal*, and *trend* anomalies. More specifically, *shapelet* anomalies include *shape* anomalies that exhibit noticeable shape disparities compared with the normal subsequences and *correlation* anomalies that refer to abnormal dependencies among the variables. Seasonal and trend outliers present unusual seasonality or trend.

Key Technique. As shown in Fig. 2, we perform a window-based augmentation that is oriented to temporal data after $\mathcal{S}$ is cut into many equal-length samples by the sliding windows. We generate a new patch of anomalous samples and substitute the patch for a random position of sample X_i, i.e., the original values of X_i are replaced. Particularly, we devise the following steps to generate samples with all five types of anomalies.

- **Cut:** we cut a patch of randomly sized $r \geq \zeta$ subsequence from another arbitrarily chosen sample X_j, where hyperparameter $\zeta \in (0, t)$ controls the minimum length of the patch since the length has to be long enough for generating an anomaly.
- **Add:** we add a series of incremental or decremental values V_{trend} deriving from a linear function with a random slope to the patch. For multi-dimensional time series, we selectively choose and aggregate some of these dimensions to form the random trends and add them into the cut patch.
- **Paste:** we embed the patch into a random position of sample X_i. The pre-existing value within this location is replaced by the value from the patch.

Analysis. Given the paste (destination) sample $\mathbf{X}_i = \Gamma_i(2\pi\omega_i T) + \Theta_i(T)$ and the cut (source) sample $\mathbf{X}_j = \Gamma_j(2\pi\omega_j T) + \Theta_j(T)$, where $T = \{1, 2, \ldots, t\}$. $\mathbf{X}_i$ and $\mathbf{X}_j$ come from $\mathcal{S}$, which has been standardized. The trend item $V_{trend} = \rho \cdot m \cdot R$, where $R = \{1, 2, \ldots, r\}$. $m = (m_1, m_2, \ldots, m_d)$ is a d-dimensional random slope vector, where $-1 < m_i < 1$. Hyperparameter $\rho > 0$ controls the degree of the trend term. After the above augmentation operation, the pseudo-abnormal sample $\mathbf{X}_a$ is defined as:$\{x_b | b \in T\}$, and the formulation of X_a involving $\mathbf{X}_i$, $\mathbf{X}_j$, and $\mathbf{V}_{trend}$ is depicted as follows:

$$x_b = \begin{cases} \Gamma_i(2\pi\omega_i b) + \Theta_i(b), & b \in \{1, \ldots, k_i - 1\} \\ \Gamma_j(2\pi\omega_j(b+k)) + \Theta_j(b+k) & \\ \qquad\qquad +\rho m(b - k^{'}), & b \in \left\{k_i, \ldots, k^{'} + r\right\} \\ \Gamma_i(2\pi\omega_i b) + \Theta_i(b), & b \in \{k_i + r, \ldots, t\}, \end{cases} \quad (1)$$

where $k = k_j - k_i$, $k^{'} = k_i - 1$. $k_i < t - r$ and $k_j < t - r$ are the random pasting position and cutting position respectively. As illustrated in Fig. 3, considering the subsequence $X[t_1 : t_2]$, i.e., $\left\{x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}\right\}$, we formulate the following relevant five types of anomalies onto X_a:

- **A1-S: Shape.** Generally, Γ_i is different from Γ_j because X_j is chosen randomly. Therefore, an arbitrary sub-sequence $X_a[t_1 : t_2]$ from $t_1 \in [1 : k_i)$ to $t_2 \in [k_i + 1, T]$ contains at least one hop transition of the shape function Γ, forming the shape anomalies.

- **A1-C: Correlation.** For X_i, the original relationships among the multiple variables are defined as $cov(X_i, X_i) = \mathrm{E}[(X_i - \mathrm{E}[X_i])([(X_i - \mathrm{E}[X_i])^T]$. Because the i-th slope m_i is an independently generated random number, $\mathrm{E}[X_a]$ will deviate from $\mathrm{E}[X_i]$, further causing $cov(X_a, X_a) \neq cov(X_i, X_i)$, i.e., correlation anomalies.
- **A2: Seasonality.** The seasonality parameter of the patch $X_a[k_i : k_i + r - 1]$ is ω_j rather than the expected ω_i.
- **A3: Trend.** After the augmentation, the trend function of the patch $\Theta_a = \Theta_j + V_{trend}$, which is different from the expected Θ_i.
- **A4 & A5: Point-wise.** When $b = k' + r$, $x_b = X_j[k_j + r - 1] + \rho m r$. If $m_i \neq 0$, such as $Dim1$ and $Dim3$ in Fig. 3, x_b is proportional to r. Therefore, when r is large enough, $|x_b - \mu|$ will exceed the 3σ to create a global (A4) or contextual (A5) anomaly, where μ and σ are the mean and standard deviation of the whole time series $\mathcal{S}$ or the neighborhood points.

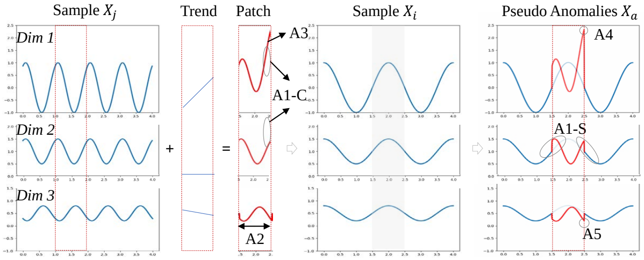


Figure 3: Example of generating an anomalous sample over sine waves. The original $Dim2 = 0.5 \cdot Dim1 + 1$. There are pattern-wise anomalies in terms of shapelet (A1-S: shape, A1-C: correlation), seasonality (A2), and trend (A3). Point-wise anomalies include global (A4) and contextual (A5) ones.

3.4 Model Training Objective

Finally, the loss is obtained through the calculation of the projection q_i and the label y_i. Inspired by hypersphere classification (HSC) [4, 18], we use the cross-entropy loss as the training objective. The projector g_θ outputs the 2-dimensional projections $Q = \{q_1, ..., q_i, ..., q_N\}$, and $softmax$ maps q_i to a pair of probabilities $[1 - p_i, p_i]$, where p_i and $1 - p_i$ represent the probability of being anomalous and being normal, respectively. We use $softmax$ instead of $sigmoid$ to obtain the two probabilities, separately. The cross-entropy loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)], \tag{2}$$

where y_i is the label in $\mathcal{Y}$. For the training set with labels, $\mathcal{Y}$ contains the original labels annotated by domain experts and our labels corresponding to the pseudo-anomalous samples. It is worth mentioning that our method does not require the training set to contain anomalies. As will be shown in Table 3, CutAddPaste performs well on UCR, SWaT, and WADI, all of which are generic training sets without anomalies.

In the test phase, we consider the probability p_i of the subsequence sample $\mathbf{X}_i$ as the anomaly score $S_i \in [0, 1]$. Then, we use the following standard to determine whether $\mathbf{X}_i$ can be classified as anomalous:

$$x_i = \begin{cases} anomaly, & S_i > \tau \\ normal, & S_i \leq \tau \end{cases}, \tag{3}$$

where τ is a predefined threshold. The comprehensive algorithm is outlined in Appendix A.

4 EVALUATION

In this section, we outline the experiment setup, implementation specifics, primary outcomes, ablation research, visualization, and hyperparameter analysis. The code is available at https://github.com/ruiking04/CutAddPaste.

4.1 Experiment Setup

This part introduces the experimental setup, including the datasets, evaluation metrics, and baselines.

Datasets. In light of [19], the evaluation is conducted on AIOps, UCR, SWaT, and WADI datasets, eschewing flawed time-series AD datasets like Numenta [20], Yahoo [21], NASA (SMAP and MSL) [10], and SMD [22]. Dataset details are as follows.

- **AIOps** [23] comprises 29 univariate time-series sub-datasets of well-maintained business cloud key performance indicators from prominent Internet companies. Some references may identify it as "KPI". It contains more point-wise anomalies compared with pattern-wise ones.
- **UCR** [19] contains 250 univariate time-series sub-datasets spanning various domains. Each sub-dataset contains only 1 anomaly segment. Most of the anomaly segments are pattern-wise anomalies.
- **SWaT** [24] encompasses multivariate data from 51 sensors of the Secure Water Treatment (SWaT) testbed under normal and attacked behavioral modes. It contains a number of long-term pattern-wise anomalies.
- **WADI** [25] is an extension of SWaT with 123 sensors and actuators. The pattern-wise anomalies are shorter than those on SWaT.

As demonstrated in Table 2, each time series is segmented into length-t sequences using a sliding window with a time-step δ. The table additionally presents the sequence count in the training, validation, and testing sets and the proportion of anomalous samples.

Metrics. At present, there is no unified evaluation plan for anomaly detection in the time series. Some metrics may have high or low deviations, leading to a confusing situation. In this paper, we will quantify and compare several well-known metrics, and, illustrate choosing RPA [10] metric is the best option for the sake of fairness.

As common time series anomalies, especially pattern-wise ones, are often continuous rather than isolated points, traditional Point-Wise (PW) precision, recall, and F1 score calculations tend to underestimate the detection capability. To address this issue, many alternative metrics have been proposed, such as the NAB Score [20] and the Point Adjusted (PA) metrics [11]. However, PA metrics are more widely accepted due to their simpler calculation compared to NAB scores. As illustrated in Fig. 4, PA metrics assume that if any

Table 2: Summary of TSAD datasets. Tra-Ano is the proportion of anomalies in the training set.

	AIOps	UCR	SWaT	WADI
Subsets	29	250	1	1
Variables	1	1	51	127
Domain	Cloud KPIs	Various	Waterworks	Waterworks
Length	32	64	32	32
Time step	32	16	16	16
Training	93864	330583	29699	49034
Validation	18251	175564	5624	2160
Testing	91201	877355	28118	10799
Anomalies	3.59%	0.53%	5.96%	1.06%
Tra-Ano	3.86%	0%	0%	0%

point in an anomaly window is identified as anomalous, all points in the window are considered true positives. Nevertheless, [10], [26], and [27] successively criticize that PA has a great possibility of overestimating AD performance, and hence propose RPA, PA%K, and affiliation metrics, respectively.

Upon reproducing the results of [3] and [17], it was found that affiliation metrics also tend to overestimate, and even random anomaly scores could lead to high affiliation F1 scores. For the PA%K metric, K% is a predefined threshold. As shown in Fig. 4, when the portion of abnormal points in the window exceeds K%, all the points are considered anomalous. Therefore, PW and PA are special cases where K is 100 and 0. Moreover, determining a suitable K for PA%K leads to increased time complexity. The RPA metrics treat the entire abnormal segment as a single sample, aligning with our intuition about the number of anomalies.

In this study, we choose the RPA F1-score as a fair assessment. For a better comparison, we also report the results of the PA F1-score. Note that this paper reports on the metrics for the entire dataset, which is a weighted average of the RPA or PA F1-score for each sub-dataset:

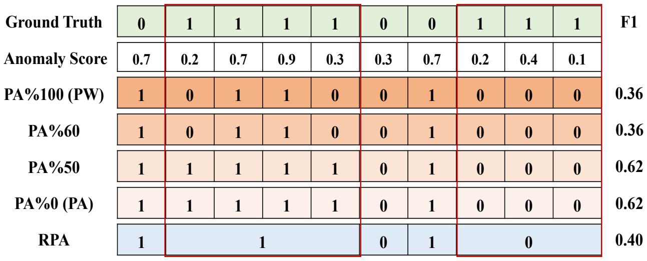
$$\mathrm{F1}_{\mathrm{entire}} = \sum_{i=1}^{M} \frac{e_i}{E} \mathrm{F1}_i, \tag{4}$$

where M is the number of sub-datasets, E is the total number of anomaly segments for the entire dataset, and e_i is the number of anomaly segments of the i-th sub-dataset. Please refer to Appendix C for the performance of our approach and the baselines under the other metrics.

Baselines. The proposed approach is compared against the following traditional, normality assumptions-based, and anomaly assumptions-based AD methods.

Traditional AD Baselines. Five commonly used traditional AD methods are adopted: One-Class SVM (OC-SVM) [28], Isolation Forest (IF) [29], Robust Random Cut Forest (RRCF) [30], Spectral Residual (SR) [31], and DAMP [32]. Inspired by [26], we design two simple baselines: the Randomized Anomaly Score (RAS) and the Absolute value of the input itself as the Anomaly Score (AAS).

Normality Assumptions-based AD Baselines. Then, three single and three multiple deep normality assumptions-based AD methods are compared: LSTM Encoder-decoder (LSTM-ED) [1], Deep one-class (Deep SVDD) [2], Anomaly Transformer [33], COCA [3], AOC [17], and TCC [16, 34].



Ground Truth	0	1	1	1	1	0	0	1	1	1	F1
Anomaly Score	0.7	0.2	0.7	0.9	0.3	0.3	0.7	0.2	0.4	0.1	
PA%100 (PW)	1	0	1	1	0	0	1	0	0	0	0.36
PA%60	1	0	1	1	0	0	1	0	0	0	0.36
PA%50	1	1	1	1	1	0	1	0	0	0	0.62
PA%0 (PA)	1	1	1	1	1	0	1	0	0	0	0.62
RPA	1	1				0	1	0			0.40

Figure 4: Schematic diagram of metrics. The red boxes are two abnormal segments. The threshold of anomaly score is 0.5 to decide whether each point is normal (0) or abnormal (1). The right column is the F1 score obtained by each metric.

Anomaly Assumptions-based AD Baselines. Finally, an anomaly assumptions-based method is set: NCAD [7]. NCAD offers both supervised and unsupervised settings for the KPIs dataset, but the former yields better AD performance and serves as the baseline for this paper.

It's worth noting that while Deep SVDD is designed for AD in the image data, their underlying concepts remain significant. Therefore, previous work [3, 7, 17] has adapted it to the temporal domain by employing Conv1D to implement its autoencoder architecture. Regarding TCC based on [16], we conduct representation learning during the pre-training phase using TS-TCC [34] and anomaly detection during fine-tuning with the principles of Deep SVDD.

4.2 Implementation Details

The 1D-CNN of the TCN encoder has a dropout rate of 0.45. We adopt a learning rate from $1e-4$ to $5e-4$, weight decay of $5e-4$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$ in an Adam optimizer. Since in UCR, the time series each has only one anomaly segment, we choose the sample with the largest anomaly score as where the anomaly lies. In addition, we perform the early stopping strategy on UCR, as the time series from different domains vary in epochs to convergence. For the other datasets, the obtained raw anomaly scores are converted into Z-scores and we search for the optimal anomaly threshold $\tau \in [-3, 3]$ according to the 3σ-rule. To ensure robustness, each method is executed 10 times with distinct random seeds to obtain the mean and standard deviation of the metrics. The models are implemented using PyTorch 1.7 and Merlion 1.1.1 [35], and trained on an NVIDIA Tesla V100 GPU. See more details about hyperparameters in Appendix B.

4.3 Main Results

We report the AD performance of the baselines and our method in Table 3, in terms of RPA and PA F1.

Metric selection. We design two straightforward baselines as comparison targets to emphasize the importance of selecting a fair metric: RAS and AAS. RAS generates random decimals as anomaly scores over several epochs and then chooses the best one. AAS regards the absolute value of the normalized input as the anomaly score. They may look ridiculous to detect anomalies but receive surprise results, even beating several classical approaches such as SR and TCC. This observation, well aligned with the discussions

Table 3: Average RPA F1-score(%), and PA F1-score(%) with standard deviation for baselines, our method, and variants on AIOps, UCR, SWaT, and WADI datasets over 10 runs. Notably, PA is the metric we don't recommend. The best results are in bold. SR does not support multivariate time series anomaly detection. SameCAP is a variant method only for multivariate time series.

Methods	RPA F1				PA F1			
	AIOps	UCR	SWaT	WADI	AIOps	UCR	SWaT	WADI
OC-SVM	7.02	16.42	0.02	0.03	58.58	33.69	30.60	17.66
IF	3.86±0.07	4.51±0.44	12.79±1.94	0.87±0.16	57.03±0.67	10.76±0.93	86.90±0.59	75.65±1.56
RRCF	2.89±0.06	5.91±0.67	0.99±0.05	0.95±0.09	62.26±0.18	16.04±1.56	**93.88±0.38**	**86.88±1.45**
SR	8.52	22.00	–	–	76.19	36.51	–	–
DAMP	2.72	39.03	0.70	0.18	64.23	53.59	35.93	36.51
RAS	5.46±0.83	20.88±3.06	10.30±2.00	9.77±3.55	18.57±4.07	31.06±4.62	83.55±1.70	50.06±13.59
AAS	16.07	18.04	24.56	7.50	34.84	31.04	82.33	28.75
LSTM-ED	14.05±0.52	24.64±1.11	5.48±0.04	3.86±0.52	74.69±0.18	42.38±1.83	14.05±0.52	18.66±0.02
Deep SVDD	25.53±7.38	48.34±2.74	8.88±6.97	6.14±2.09	54.11±8.90	63.97±3.90	79.65±5.12	59.35±5.34
COCA	40.59±8.11	56.29±1.74	15.83±6.81	4.27±2.04	64.20±5.52	68.82±1.55	81.24±4.17	46.91±7.20
AOC	41.40±3.28	21.39±0.80	27.59±0.02	0.36±0.01	57.42±2.55	41.48±0.97	82.89±0.19	11.09±0.03
TCC	3.03±0.66	2.53±0.75	2.06±0.66	5.02±3.79	16.49±3.17	3.31±1.29	75.30±2.80	34.89±14.16
Anomaly Transformer	0.38±0.13	6.01±1.26	32.91±1.19	1.68±0.02	29.95±6.49	11.35±3.07	92.61±0.35	52.90±0.04
NCAD	41.06±3.32	22.24±2.99	7.54±2.48	6.84±2.53	67.37±2.10	40.35±3.83	82.34±2.93	46.65±6.82
CutAddPaste	**77.41±0.96**	**68.22±1.55**	**45.86±2.64**	**26.58±3.47**	**84.68±0.67**	**81.95±1.25**	92.35±1.03	65.20±10.15
Cutout	59.94±1.42	53.67±1.20	34.24±3.63	11.87±8.70	75.12±1.19	70.37±1.15	87.12±1.24	35.59±11.05
PointAdd	64.69±1.70	54.07±2.86	24.31±5.42	7.18±4.11	77.50±0.71	70.93±2.62	85.15±1.68	34.04±16.24
TrendAdd	71.55±1.05	58.82±3.77	25.32±3.29	14.65±5.81	82.78±0.61	76.98±2.37	84.96±2.30	76.31±5.15
Same-CutPaste	71.59±3.48	67.28±2.70	26.95±3.51	9.60±8.10	79.54±2.02	80.44±1.72	84.86±1.73	34.29±14.39
Diff-CutPaste	61.67±0.95	67.72±2.92	44.23±3.59	23.36±5.83	77.59±0.94	80.77±2.02	92.30±1.20	59.57±5.43
SameCAP	–	–	33.90±10.03	19.83±7.61	–	–	87.97±5.80	50.46±16.40

in existing work [19, 26], indicates that the most recent advancements are highly likely to be misleading due to dataset changes and evaluation metric shifts.

The phenomenon is even more noticeable under the PA metric – F1 scores of RAS and AAS achieve 83.55% and 82.33%, respectively, on SWaT. We attribute this overestimation to the high portion of long-term sequence (pattern) anomalies in this dataset. When evaluating the performance of a method, PA regards each point in an anomalous segment as an anomaly. However, it unreasonably assumes that all of the predictions are correct throughout the entire anomaly segment once it encounters a correctly predicted point. Hence, almost all these methods attain a PA F1 score of over 80% on SWaT and could hardly be distinguished. By comparison, for such datasets as AIOps dominated by point anomalies, the PA is relatively fair, e.g., the random RAS is 18.57%. As a result, considering that the fairness of metrics is influenced by the types of anomalies, this article cautiously adopts RPA as the metric, and the following "F1-score" are RPA ones. We further advocate the use of RPA instead of PA. The performance of CutAddPaste and baselines under the other metrics are provided in Appendix C, while our method still outperforms others.

TSAD performance. From the perspective of the datasets in Table 3, there are following two key observations. Firstly, there is significant variation in the AD performance of methods like OC-SVM, AAS, and COCA across the AIOps and UCR datasets. For instance, the RPA F1 scores of COCA differ by 16% between the two univariate datasets. This divergence can be attributed to the predominant anomaly types in each dataset: point-wise anomalies for AIOps and pattern-wise ones for UCR. As a result, these methods exhibit varying sensitivity to specific types of anomalies. In contrast, our proposed method effectively strikes a relative balance between these two anomaly types, as indicated by its RPA F1 exceeding 68% on the two datasets. Secondly, multivariate TSAD is much more thorny as indicated by the results on SWaT and WADI. Most methods fail to match their capabilities shown on univariate data sets. Even the method designed for multivariate data, such as Anomaly Transformer, can only achieve 32.91% on SWaT and performs much poorer on WADI as the dimension increases a lot. On the other hand, Anomaly Transformer does not perform well on univariate data sets, demonstrating that there are different emphases in univariate and multivariate TSAD tasks. Our approach obtains 45.86% and 26.58% F1 scores on SWaT and WADI, surpassing Anomaly Transformer by a large margin, indicating that the CutAddPaste augmentation plays an important role in the progress.

Based on the comparison of the methods in the table, the following conclusions can be drawn. Firstly, among the traditional machine learning methods, DAMP and SR get RPA F1 of 39.03% and 22% on UCR, even surpassing some deep approaches. It indicates that the shallow methods could also work well in some specific cases such as univariate pattern-wise anomalies. Nevertheless, they are defective in dealing with high-dimensional multivariate time series. Secondly, AOC and COCA exhibit better performance than other baselines based on normality assumptions for TSAD. This implies that techniques involving reconstruction and incorporating multiple normality assumptions are more aligned with the nature of normal samples. In addition, TCC's unsatisfying results confirm the argument that pre-training limits the capability of the two-staged methods. Lastly, comparing the anomaly assumption-based methods, NCAD trails behind CutAddPaste in anomaly detection on all four datasets. On AIOps, NCAD manages to achieve a respectable F1 score exceeding 41% because it generates point-wise anomalies, consistent with the nature of the dataset. This reaffirms the potential of anomaly assumptions-based AD methods as a promising avenue. However, its gap with our CutAddPaste on the other datasets suggests that the injection of point anomalies and OE might struggle to handle complex pattern-wise anomalies effectively.

In summary, the proposed CutAddPaste method successfully balances point-wise and pattern-wise anomalies, generating highly discriminative anomaly scores, and outperforming all baselines on all four datasets. This outcome underscores the effectiveness and robustness of AD based on CutAddPaste augmentation.

4.4 Ablation Study

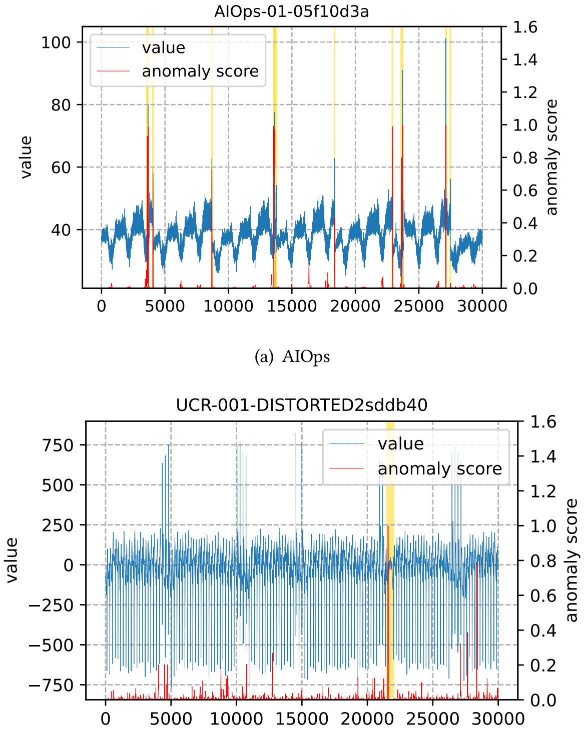
In this section, we study six variants, from only cutting, adding, our transplanted CutPaste, to our CutAddPaste with the same trends. The RPA and PA F1 scores of these variants on the four datasets are shown in Table 3. The results reveal some insights into the effectiveness of our proposed method's components.

Cutout. This variant chooses a random subsequence from the sample X_i and replaces the values on it with zero to simulate that the patch is cut off. It appears that a pure masking method helps to improve the performance, most likely because masking eliminates the original pattern.

PointAdd and TrendAdd. Both of the variants add anomalies in the sample directly. *PointAdd* injects abnormal point x_j ($|x_j - \mu| > 3\sigma$) at random locations of the sample X_i, where μ and σ are the mean and standard deviation of X_i. *TrendAdd* retains the "Add" component which adds a patch of trend outliers that significantly alter the mean of the sample X_i. *PointAdd* improves little over AD performance except on AIOps compared to *Cutout*, confirming the intuition that only point-wise augmentation is insufficient. Even on the AIOps dataset, which contains more point anomalies, *TrendAdd* performs better than *PointAdd*, indicating that the anomalies produced by the Add component include both trend and point-wise anomalies.

CutPaste variants. *Same-* and *Diff-CutPaste* are the window-based CutPaste method achieved by us. For the patch to be pasted to sample X_i, the former cuts it from X_i itself, while the latter cuts it from another X_j. Both of them outperform the *Cutout* variants, demonstrating the significance of the Paste component. *Diff-CutPaste* yields better results than *Same-CutPaste*, which suggests that drawing from different samples enhances the model's exposure to abnormal instances, improving both performance and robustness. Meanwhile, the resulting gap of *Diff-CutPaste* to our CutAddPaste highlights the significance of the Add term in our approach again.

CutAddPaste with the same trends (SameCAP). We further study the influence of the different trends that are added to the patch. SameCAP adds the same trends on each dimension of the patch, which underperforms CutAddPaste by over 7% on each multivariate dataset. This indicates that the correlation anomalies introduced by different trends work.



(a) AIOps

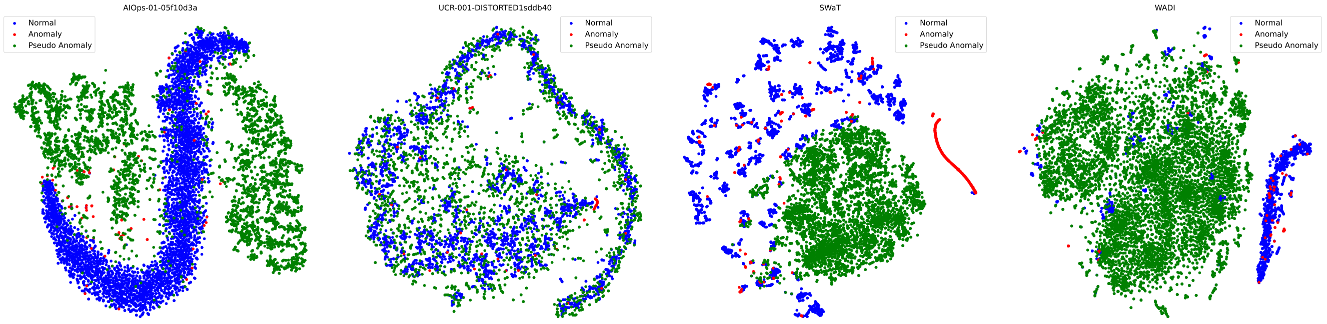
(b) UCR

Figure 5: The visualization of CutAddPaste AD results on AIOps and UCR datasets. The titles are the number (ID in the dataset) of them, the x-axles are timestamps, and the y-axles are signal values. The original data is present as the blue curves. The yellow areas are ground-truth anomalies, including point- and pattern-wise ones. The red curves represent the anomaly scores that our method predicts.

Overall, CutAddPaste outperforms the other six alternatives, indicating both the effectiveness and importance of each component in our approach.

4.5 Visualization

To provide a more intuitive evaluation, the visualizations of our method on AIOps and UCR datasets are conducted, in Fig. 5. Qualitative analysis indicates that the model predicts a sequence of anomaly scores with better distinction, i.e., it's more sensitive to each type of anomaly. In Fig. 5(a), which represents the AIOps dataset, the time series contains several distinct point-wise anomalies, making anomaly detection relatively straightforward. The high anomaly scores generated by our CutAddPaste method closely align with the ground-truth anomaly regions, confirming that our approach is sensitive to point-wise anomalies. On the other hand, in Fig. 5(b), which depicts a time series from the UCR dataset, each sequence contains only one subtle pattern-wise anomaly segment. These types of anomalies are typically harder to detect due to their inconspicuous nature. Despite the challenge, our method's anomaly scores sharply peak around the ground-truth anomaly region, effectively identifying the anomaly. While the model does produce a few



(a) AIOps (b) UCR (c) SWaT (d) WADI

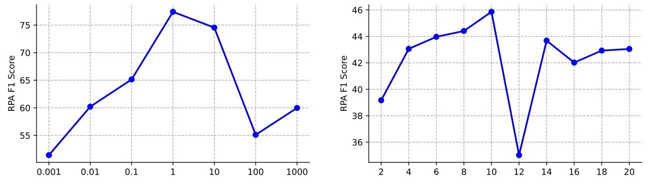
Figure 6: Illustration of t-SNE embeddings of original and generated samples on four datasets. The top titles are the dataset name and the ID within the dataset, where (c) and (d) contain no ID as SWaT and WADI only have one subset. Blue dots are the normal samples, red are the anomalies, and green are the pseudo anomalies generated by CutAddPaste.

other high-score segments, the overall performance is impressive. In short, CutAddPaste performs well on both subsequences, further illustrating that our model can detect both simple point-wise anomalies and complex pattern-wise anomalies.

In addition, to further analyze the differences between the generated pseudo-anomalies and the original time series, their t-SNE embeddings are shown in Fig. 6. Overall, pseudo-anomalous and normal samples are divided by relatively clear decision boundaries, as shown in Fig. 6(a), 6(c), and 6(d). In Fig. 6(b), although the pseudo anomalies generated by our model on the UCR dataset are somewhat mixed with normal samples, most of them fall outside the range of normal samples. Meanwhile, some pseudo-anomalies are scattered near the original anomalies indicating that CutAddPaste can help anomaly detectors generalize to and perform well on unseen distributions of anomalies, especially on AIOps.

4.6 Hyperparameters Analysis

We conducted a sensitivity analysis to study the impact of hyperparameters including the degree of the trend term (ρ) and the minimum length of the patch (ζ), as shown in Fig. 7. The y-axis represents the RPA F1 score. Overall, our model is more sensitive to ρ and ζ. For the analysis of ρ, we primarily considered the AIOps dataset, which mainly contains point-wise anomalies and is therefore strongly influenced by the proportion of trend items. As shown in Fig. 7(a), it is evident that within a certain range, increasing the degree of the trend term enhances the model's sensitivity to point anomalies, leading to improved anomaly detection performance. However, beyond a certain point, a larger ρ can harm the performance, likely because it causes the model to focus excessively on the generated extreme points. We observe that $\rho = 1$ performs the best. Fig. 7(b) shows the results of varying ζ in a range between 2 and 20. The graph indicates that the model performs best on the SWaT dataset when the minimum length ζ is set to 10. This emphasizes the importance of maintaining a balance between the length of the paste (destination) sample X_i and the cut (source) sample X_j to achieve optimal anomaly detection performance. Analysis of more hyperparameters can be found in Appendix D.



(a) ρ (AIOps) (b) ζ (SWaT)

Figure 7: Two sensitivity analysis experiments on AIOps and SWaT datasets. The left is the degree of the trend term ρ and the right is the minimum length of the patch ζ.

5 CONCLUSION

We propose CutAddPaste, an anomaly-knowledge-based approach dedicated to time series anomaly detection. It provides a simple yet powerful augmentation, which is the first that achieves exposing the models to all five kinds of TS anomalies including correlation ones in multivariate data, to our best knowledge. By incorporating domain knowledge, CutAddPaste enables the model to understand and detect diverse anomalies and improve overall performance. Our approach demonstrates superior capabilities on four real-world datasets. It underscores the effectiveness of exposing the model to multiple types of anomalies, even in a coarse approximation. We envision CutAddPaste augmentation could serve as a foundation of future powerful models for time series anomaly detection. Additionally, we advocate for fair evaluation metrics like RPA in TSAD, which are crucial for fostering consensus and laying the groundwork for future model innovation. Without it, the impressive results achieved could remain an illusion.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China (2023YFB3307503), in part by the Fundamental Research Funds for the Central Universities, and in part by the Kuaishou Research Program. We would also like to thank Dr. Tao Lin from Westlake University for the kind suggestions.

REFERENCES

[1] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[2] 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.

[3] 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.

[4] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2018.

[5] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021.

[6] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019.

[7] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *IJCAI*, 2022.

[8] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*, 2021.

[9] Gen Li and Jason J Jung. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, 91:93–102, 2023.

[10] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018.

[11] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018.

[12] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

[13] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[14] Xuan Xia, Xizhou Pan, Nan Li, Xing He, Lin Ma, Xiaoguang Zhang, and Ning Ding. Gan-based anomaly detection: A review. *Neurocomputing*, 493:497–535, 2022.

[15] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018.

[16] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. *ICLR*, 2021.

[17] Xudong Mou, Rui Wang, Tiejun Wang, Jie Sun, Bo Li, Tianyu Wo, and Xudong Liu. Deep autoencoding one-class time series anomaly detection. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[18] 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.

[19] 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.

[20] Alexander Lavin and Subutai Ahmad. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 38–44. IEEE, 2015.

[21] Yahoo. S5 - a labeled anomaly detection dataset, version 1.0 (16m). https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70, 2015. Accessed: 2023-04-14.

[22] 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.

[23] AIOps Challenge. The 1st match for aiops. https://github.com/NetManAIOps/KPI-Anomaly-Detection, 2018. Accessed: 2023-04-14.

[24] Aditya P Mathur and Nils Ole 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)*, pages 31–36. IEEE, 2016.

[25] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*, pages 25–28, 2017.

[26] 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.

[27] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *ACM SIGKDD*, pages 635–645, 2022.

[28] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588. Citeseer, 1999.

[29] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39, 2012.

[30] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *ICML*, pages 2712–2721. PMLR, 2016.

[31] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *ACM SIGKDD*, pages 3009–3017, 2019.

[32] Yue Lu, Renjie Wu, Abdullah Mueen, Maria A Zuluaga, and Eamonn Keogh. Matrix profile xxiv: scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1173–1182, 2022.

[33] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2021.

[34] 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. *IJCAI*, 2021.

[35] Aadyot Bhatnagar, Paul Kassianik, Chenghao Liu, Tian Lan, Wenzhuo Yang, Rowan Cassius, Doyen Sahoo, Devansh Arpit, Sri Subramanian, Gerald Woo, et al. Merlion: A machine learning library for time series. *arXiv preprint arXiv:2109.09265*, 2021.

A METHODOLOGY DETAILS

The pseudo-code for CutAddPaste in Pytorch style is provided in Algorithm 1.

Algorithm 1 CutAddPaste's main training algorithm.

```
Input: a set of subsequence samples $\{\mathbf{X}_i\}_{i=1}^{B}$, labels $\{y_i\}_{i=1}^{B}$
Parameter: batch size $B$, window size $t$, structure of $f, g$, constant $\rho, \zeta, v, e$
Output: network $f, g$
X' = X.copy()
for all $i \in \{1, \ldots, B\}$ do
    r = max(int($\zeta$), int(random.random() × t))
    cut_position = int(random.uniform(0, $t - r$))
    paste_position = int(random.uniform(0, $t - r$))
    $j$ = int(random.uniform(0, $B$))
    cut_data = $X_j$[cut_position, cut_position+r]
    trend = $\{1, 2, \ldots, r\}$
    if cut_data.shape[1]>1 then
        dim_all = $\{0, \ldots, cut\_data.shape[1] - 1\}$
        dim = random.choice(dim_all, size=e)
    else
        dim = $\{0\}$
    end if
    for all $item \in dim$ do
        factor = random.random() ×$\rho$
        add = random.choices([-1, 1])×factor×trend
        cut_data[:, item] += add
    end for
    $\mathbf{X}'_i$[paste_position, paste_position+r] = cut_data
end for
anomaly_num = int($v \times B$))
X' = random.sample(X', anomaly_num)
$y'$ = torch.ones(anomaly_num)
X = cat((X, X'), axis=0)
$y$ = cat(($y$, $y'$))
for sampled batch $\{\mathbf{X}_i\}_{i=1}^{N}$ do
    for all $i \in \{1, \ldots, N\}$ do
        $q_i = g(f(\mathbf{X}_i))$
        $p_i = q_i[1]$
    end for
    $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)]$
    update networks $f, g$ to minimize $\mathcal{L}$
end for
return network $f, g$
```

B HYPERPARAMETERS DETAILS

CutAddPaste is implemented in PyTorch, and some important parameter values used in the model are listed in Table 4. *kernel_size* and *stride* are the convolutional kernel size and stride of the first Conv1D layer, respectively. *final_out_channels* is the number of channels generated by the last Conv1D layer. *dropout* is the probability of an element being zeroed in a Dropout layer. *window_size* is the size of the time window, which is described as the length of time series t in the text, and *time_step* is the step while sliding, denoted as δ. *lr* is the learning rate. *batch_size* is the number of samples once given to the program for training our model. *num_epoch* represents the number of times that the entire sub-dataset is trained. ρ is the degree of the trend term and ζ is the minimum length of the patch. e is the number of dimensions where $m_i \neq 0$ in the trend term. v is the augmentation ratio.

Table 4: The values of hyperparameters used in CutAddPaste.

	AIOps	UCR	SWaT	WADI
kernel_size	4	8	8	4
stride	1	1	1	1
final_out_channels	32	64	32	32
dropout	0.45	0.45	0.45	0.45
window_size	32	64	32	32
time_step	32	16	16	16
lr	0.0001	0.0003	0.0003	0.0003
batch_size	512	512	512	512
num_epoch	300	300	100	50
ρ	1	0.01	0.01	0.1
ζ	9	12	10	16
v	0.6	1	1	1
e	–	–	5	15

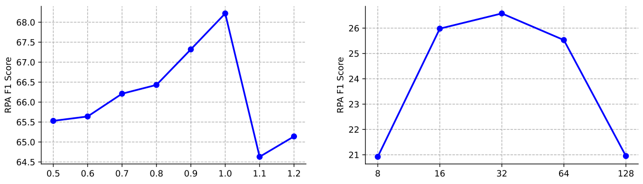
C EXPERIMENT RESULTS UNDER OTHER METRICS

To compare multiple evaluation metrics for TSAD experimentally, we provide the AD performance of the baseline and our technique in terms of PW F1-score and Affiliation F1-score in Table 5. Given the same dataset, the F1 scores achieved by each approach based on different criteria vary greatly. From this chaotic and confusing situation, the following conclusions can be drawn.

First, on the UCR dataset, which contains only one short-term anomalous sequence on each time series, most of the mentioned methods obtain low PW F1 scores, even less than 10%. This phenomenon indicates that PW might be too strict and underestimate the detection capability. It checks whether the prediction is correct point by point, while the borders of anomalous segments are typically not so apparent, making it unsuitable for assessing the model over short pattern-wise anomalies. Second, even RAS and AAS could achieve higher affiliation F1 scores than many other baselines, and most of the methods attain F1 scores in a narrow range, e.g., all methods get 70% to 72% on the WADI dataset, showing that affiliation metrics tend to overestimate the performance of models and poorly differentiate them. Last, the proposed CutAddPaste outperforms all baselines on most datasets under both the tight (PW) and loose (affiliation) metrics, demonstrating the effectiveness and robustness of anomaly assumptions-based AD methods.

Table 5: Average PW F1-score(%) and Affiliation F1-score(%) with standard deviation for baselines and our method on AIOps, UCR, SWaT, and WADI datasets over 10 runs. The best results are in bold.

Methods	**PW F1**				**Affiliation F1**			
	AIOps	**UCR**	**SWaT**	**WADI**	**AIOps**	**UCR**	**SWaT**	**WADI**
OC-SVM	19.32	0.13	30.57	12.48	79.97	48.85	76.58	**71.92**
IF	20.43±0.85	0.09±0.01	74.14±0.40	**24.20±1.16**	78.85±0.13	36.83±1.00	72.88±0.98	71.06±0.16
RRCF	10.20±0.48	0.06±0.01	21.80±0.05	11.13±0.21	74.12±0.26	45.00±1.22	70.26±0.46	71.05±0.62
SR	8.66	0.11	–	–	76.18	60.54	–	–
DAMP	4.77	0.20	22.15	12.01	48.90	66.71	69.54	70.50
RAS	5.46±0.80	1.56±0.23	0.98±0.62	1.46±0.69	68.36±0.11	59.81±2.55	70.30±0.60	71.25±0.58
AAS	23.41	1.34	76.78	12.13	72.47	56.97	72.19	49.81
LSTM-ED	19.92±3.24	0.12±0.01	77.24±0.00	10.92±0.00	82.52±0.31	61.02±0.92	74.86±0.63	70.50±0.00
Deep SVDD	17.50±6.11	3.63±0.19	48.65±18.64	14.09±2.59	72.56±2.25	72.21±2.15	72.77±1.93	70.93±0.59
COCA	40.14±4.81	4.20±0.13	62.43±17.62	15.60±4.49	81.99±1.02	75.19±1.19	72.00±0.40	71.47±0.96
AOC	28.47±2.34	1.62±0.06	77.18±0.01	11.09±0.00	80.19±1.33	58.31±1.62	71.42±0.93	70.60±0.00
TCC	6.99±0.54	0.11±0.03	21.85±0.00	11.28±0.28	68.03±0.07	43.26±0.99	69.53±0.33	70.79±0.39
Anomaly Transformer	4.72±0.26	0.03±0.01	21.64±0.00	10.92±0.00	68.21±0.36	52.18±1.34	70.17±1.00	70.50±0.00
NCAD	41.61±2.75	2.13±0.29	25.77±8.53	11.40±0.66	84.13±1.24	60.13±2.91	69.27±0.00	70.80±0.65
CutAddPaste	**62.43±1.42**	**7.69±0.23**	**77.92±1.05**	20.90±8.43	**87.87±0.65**	**82.85±1.49**	**76.30±2.28**	71.45±1.80



(a) v (UCR)

(b) t (WADI)

Figure 8: Sensitivity analysis experiments on UCR and WADI datasets. The left is the augmentation ratio v and the right is the window size t.

D OTHER HYPERPARAMETERS ANALYSIS

We also provide a sensitivity analysis over augmentation ratio v and window size t on UCR and WADI datasets. Fig. 8(a) shows the results of varying v in a range between 0.5 and 1.2. The graph indicates that the model performs best on the UCR dataset when the augmentation ratio v is set to 1. This emphasizes the importance of maintaining a balance between the numbers of normal and abnormal samples to achieve optimal anomaly detection performance. Fig. 8(b) shows that CutAddPaste achieves its best on WADI when the window size is 32, and too large or small windows will limit the performance.