CAPMix: Robust KPI Anomaly Detection for AIOps in Noisy and Dynamic Environments

Xudong Mou[1], Rui Wang[1], Tiejun Wang[1], Zexin Wu[1], Fangda Guo[2], Jie Sun[1], Shiru Chen[3], Penghao Zhang[4], Tiezi Zhang[4], Tianyu Wo[1], Hao Peng[1], Chunming Hu[1], Xudong Liu[1], Renyu Yang[1†]
[1]Beihang University [2] Chinese Academy of Sciences [3]Inspur Inc. [4] Kuaishou Technology

Abstract

Time-series anomaly detection is crucial in AIOps for maintaining large-scale service reliability. In production, streams of Key Performance Indicators (KPI) are high-dimensional, non-stationary, and affected by noise, deployment changes, and latent anomalies, making real failures hard to distinguish from benign variation. Most existing methods assume either normality (learning from "normal" history) or rely on injected anomalies for training. Yet injected patterns often misalign with real failure modes, skewing decision boundaries – aka. *Anomaly Shift*. We propose CAPMix, a controllable anomaly augmentation framework with prior-guided injection for realistic temporal behaviors. CAPMix combines label revision and dual-space mixup to enhance robustness under contaminated and mixed data. CAPMix consistently outperforms state-of-the-art methods on public AIOps and time-series benchmarks. It has been deployed in Kuaishou's large-scale production system, reducing false alarms and improving monitoring reliability. A real-world dataset is also released to enrich the research on robust KPI anomaly detection.

CCS Concepts

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

Keywords

KPI Anomaly detection; AIOps; time series; anomaly augmentation

ACM Reference Format:
Xudong Mou[1], Rui Wang[1], Tiejun Wang[1], Zexin Wu[1], Fangda Guo[2], Jie Sun[1], Shiru Chen[3], Penghao Zhang[4], Tiezi Zhang[4], Tianyu Wo[1], Hao Peng[1], Chunming Hu[1], Xudong Liu[1], Renyu Yang[1†] . 2018. CAPMix: Robust KPI Anomaly Detection for AIOps in Noisy and Dynamic Environments. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

Corresponding Author: Renyu Yang (renyuyang@buaa.edu.cn).

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Large-scale AI models have made high-performance computing clusters – usually with over 10,000 GPUs – mission-critical for services like real-time video recommendation and e-commerce. As these systems grow more complex, anomalies become more likely and more damaging, with even transient failures causing major revenue loss and disruption. To ensure reliability, AIOps platforms continuously monitor massive streams of Key Performance Indicators (KPIs), which form high-dimensional, time-varying series [1–3]. These sequences exhibit strong temporal dependencies, non-stationarity, and cross-metric correlations.

In one of our production environments, we observed a task-level failure rate of 19.3%. Interpretation of key performance indicator (KPI) signals is further confounded by measurement noise and by benign operations such as canary deployment and auto-scaling. These factors introduce transient fluctuations and irregular patterns that depart from true anomalies, causing detectors to miss temporal structure, raise excessive false alarms, and create alert fatigue. This wastes computational resources and delays incident response and development. Hence, it is highly imperative for real-world KPI Anomaly Detection (KAD) to learn robust anomaly representations under noisy, evolving, weakly distinguishable temporal patterns.

Existing KPI anomaly detection methods usually rely on two implicit assumptions:

Normality Assumption (NA). Many approaches assume that the training data is clean and mostly normal samples [2–5]. Detectors learn a compact description of normal KPI dynamics and flag deviations as anomalies. For instance, reconstruction-based methods such as KAD-Disformer [3] reconstruct the expected KPI curve and use the reconstruction error as the anomaly score. Other deviation-based frameworks, such as ART [6], also model normal behavior and flag large deviations as anomalies. However, production KPI streams are often noisy and may contain latent anomalies, violating NA and making the learned boundary overly sensitive and less robust to stochastic variations. As shown in Fig. 1, a typical NA-based detector like OC-SVM can raise frequent false alarms on benign fluctuations; with a 5-minute sampling interval, these can persist all day. This limitation motivates incorporating explicit anomaly knowledge to better guide decision boundary learning.

Anomaly assumption (AA). Some methods introduce anomaly patterns for supervision [7–10]. Outlier Exposure (OE) [7] uses auxiliary outlier datasets as negative samples. NCAD [9] and AnomalyBert [10] inject random point- or context-level corruptions to mimic anomalies. CutAddPaste [11] controllably injects five types of time-series anomalies. Overall, AA effectively learns discriminative boundaries under contaminated data but requires sufficiently rich and well-calibrated anomaly patterns. In practice, AA methods suffer from *Anomaly Shift*, where injected anomaly patterns

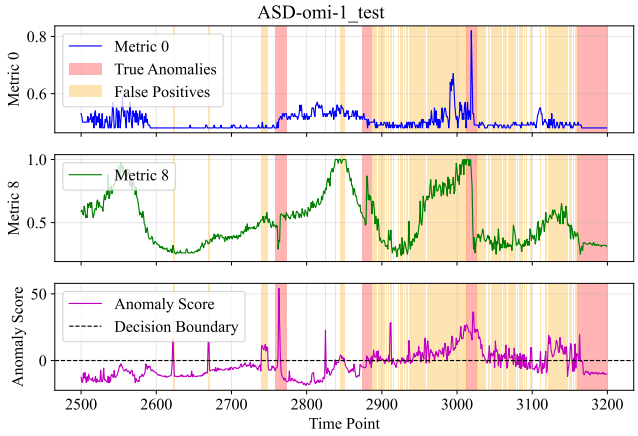


Figure 1: False alarms of a normality-assumption detector OC-SVM on a representative KPI stream. The top panels show raw KPI sequences, and the bottom panel shows the OC-SVM anomaly score. Benign system fluctuations are repeatedly flagged as anomalies (orange), obscuring true anomalies (red) and causing an excessive false-positive rate.

misalign with real failure modes. It is due to either *low distinctiveness* – injected anomalies are insufficiently differentiated from normal operational patterns, resulting in missed detections – or *high deviation* where injected anomalies deviate excessively from realistic behaviors thereby distorting the learned decision boundary. In the absence of ground-truth supervision, systematically calibrating both the injection intensity and the anomaly pattern space to adequately span the heterogeneous spectrum of industrial failure modes remains a non-trivial task. This challenge underscores the necessity for a controllable and robust anomaly generation framework that can better align synthetic anomalous patterns with real-world temporal dynamics.

We propose CAPMix, a robust, controllable anomaly augmentation framework for KPI anomaly detection in contaminated, dynamic AIOps environments. CAPMix uses prior-guided augmentation to align synthetic anomalies with real system behavior, injecting five types of realistic time-series anomalies in a targeted, controllable way. To mitigate *Anomaly Shift*, we train a Temporal Convolutional Network (TCN) with two simple insights: Label Revision (LR) soft-labels normal-biased synthetic anomalies, and Dual-Space (DS) Mixup regularizes anomalies in both raw and latent spaces to better match real failures, together promoting smoother, more reliable decision boundaries between true failures and benign fluctuations. We evaluate CAPMix on four public AIOps datasets, and four cross-domain time-series benchmarks indicate its potential to generalize from KPI streams to broader time-series anomaly detection tasks. CAPMix has been integrated and deployed as a part of KAIOps [12], the anomaly detector in Kuaishou's large-scale AI clusters. We also present case studies on critical failure modes and benign scaling-induced fluctuations, to demonstrate practical benefits in reducing alert fatigue and accelerating incident response.

The main contributions are as follows:

- A controllable framework that incorporates industrial prior knowledge into anomaly augmentation, bridging the gap between simulated and real-world failure patterns.
- Dual-space mixup and label revision mechanisms to address the *Anomaly Shift* problem, through learning robust decision boundaries under highly contaminated training data.
- Industrial case studies to showcase improved detection accuracy and reduced false alarm rates in production environments.
- Releasing an evaluation dataset with 16 days of production KPI streams, featuring real-world noise and deployment-induced dynamics.

Source code is available at https://github.com/alsike22/CAPMix.

2 Related Work

KAD in AIOps builds on advances in time-series anomaly detection for software reliability engineering. We briefly review the most relevant work: (i) normality-assumption methods [13], (ii) anomaly-assumption/injection methods, and (iii) sample-mixing strategies for robustness. Table 1 summarizes representative approaches, including KAD methods and general time-series anomaly detectors validated on KPI datasets, and highlights the gaps our method addresses.

Normality assumption based methods. Normality-assumption (NA) methods assume most training windows are normal and learn a compact *normality* model via GANs [14, 15], autoencoders / reconstruction [4, 16, 17], one-class objectives [5, 18–20], or clustering [21, 22]. Windows that deviate from this normal manifold are flagged as anomalous. In AIOps KPI anomaly detection, NA-style detectors are popular due to scarce labels. KAD-Disformer [3], for example, uses a pre-trained disentangled Transformer to reconstruct or estimate expected KPI patterns and treats reconstruction error as the anomaly signal. ART [6] similarly models normal microservice behavior within an incident-management framework and flags large deviations as abnormal. To broaden coverage, some methods combine multiple priors or signals: [23] uses a two-stage procedure based on dataset-level representations, while COCA [24] and RoCA [25] fuse objectives to suppress features irrelevant to anomaly detection. However, NA methods can be over-sensitive to benign KPI turbulence (e.g., workload-driven oscillations), and the lack of explicit anomaly knowledge often forces models to relearn failure signatures implicitly.

Anomaly assumption based methods. Rather than modeling only normality, anomaly-assumption methods use synthetic or real abnormal samples to directly shape the decision boundary. Outlier Exposure (OE) [7] trains with auxiliary out-of-distribution data, and Deep SAD [26] (extending Deep SVDD [5]) uses labeled anomalies to separate normal and abnormal representations. In CV, CutPaste [8] simulates defects via cut-and-paste augmentation, and [27] shows that even limited OE data can be effective. For time series, NCAD [9] injects contextual and point anomalies to learn a classifier, but does not explicitly target richer structured anomalies. AnomalyBert [10] perturbs points and subsequences to approximate both point- and pattern-wise anomalies. Generative methods such as GenIAS [28] produce abnormal patches while discouraging normal-like generations. RedLamp [29] applies diverse augmentations and label refurbishment to improve robustness under false anomalies/contamination, focusing on false-positive reduction. Our CAPMix complements these works by (i) revising label confidence

Table 1: Comparison of related KAD methods.

Method	N	A[1]	Anomaly Shift[2]		Covered structured TS anomaly type(s)[3]
			Close	Far	
KAD-Disformer [3]	✓	×	-	-	×
ART [6]	✓	×	-	-	×
RoCA [25]	✓	×	-	-	×
NCAD [9]	✓	✓	×	×	G, Ctx
AnomalyBert [10]	✓	✓	×	×	G, Ctx
GenIAS [28]	✓	✓	✓	×	G, Ctx, Sh, Sea, Tr
RedLamp [29]	✓	✓	✓	×	G, Ctx, Sh, Sea, Tr
CAPMix	✓	✓	✓	✓	G, Ctx, Sh, Sea, Tr

[1] N and A indicate whether the method uses normal and abnormal samples.
[2] "Close" means addressing normal-like synthetic anomalies, and "Far" means avoiding generating excessively deviated outliers.
[3] Abbrev.: G=Global, Ctx=Contextual, Sh=Shapelet, Sea=Seasonal, Tr=Trend.

for normal-like synthetic samples and (ii) constraining overly deviated synthetic anomalies via configurable dual-space mixup, jointly mitigating anomaly shift while preserving realism.

Mixup. Mixup interpolates samples and labels to regularize models. MixUp [30] and its CV variants (CutMix [31], SnapMix [32], SmoothMix [33]) enhance robustness by creating diverse yet label-faithful mixtures; manifold mixup [34] extends this to latent spaces, motivating our dual-space design. For time series, [35] mixes phase and amplitude, and TimeMixer++ [36] mixes across scales, but these methods mainly target classification robustness rather than KPI anomaly detection with injected anomalies.

3 Our Approach

3.1 Problem Definition

We formulate the monitored KPIs as time series. 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 [37] 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 $\mathbf{S}_i$ instead of giving the binary labels directly, and a predicted label can be obtained by comparing $\mathbf{S}_i$ to a predefined threshold τ.

3.2 Overview

To effectively leverage anomaly knowledge and reduce the influence of benign fluctuations in noisy and dynamic environments, we propose CAPMix for KPI anomaly detection. After injecting structured anomalies into time-series windows, CAPMix mitigates the anomaly-shift issue in anomaly-assumption training through two key insights: (i) label revision prevents the few normal-like synthetic samples from being assigned hard anomaly labels; and (ii) dual-space mixup blends synthetic anomalies with normal samples in both input and latent spaces, preventing unrealistic anomalies from distorting the decision boundary. As shown in Fig. 2, CAPMix maps multivariate time-series windows to anomaly scores via a left-to-right pipeline: anomaly injection, label revision, dual-space mixup with TCN-based representation learning, and classification. Starting from raw time-series windows, we inject sparse but informative synthetic anomalies using the controlled base method. These samples are then processed by the Label Revision (LR) module based on Dynamic Time Warping (DTW) distance to reduce the impact of anomaly shift. Finally, the revised samples are fed into Dual-Space (DS) mixup and TCN blocks to learn robust representations in both input and latent spaces, and the classifier outputs anomaly probabilities optimized with Binary Cross-Entropy (BCE) loss.

3.3 CAPMix Pipeline

3.3.1 Base Anomaly Injection. We adopt CutAddPaste [11]—a generic time-series anomaly augmentation method—as the base injection module in CAPMix. It extends CutPaste [8] to time series and generates anomalies by cutting a patch from a source window, optionally adding a trend term, and pasting it into a destination window. While effective, naive cut-add-paste may introduce abrupt boundaries, potentially aggravating anomaly shift.

- **Cut:** sample a patch of length $r \geq \zeta$ from another arbitrary window X_j.
- **Add:** add a linear trend term V_{trend} optionally on selected dimensions.
- **Paste:** replace values at a random position in X_i with the patched segment.

Formally, 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, 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:

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

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., $\{x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}\}$, 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 $X_a[t_1 :$

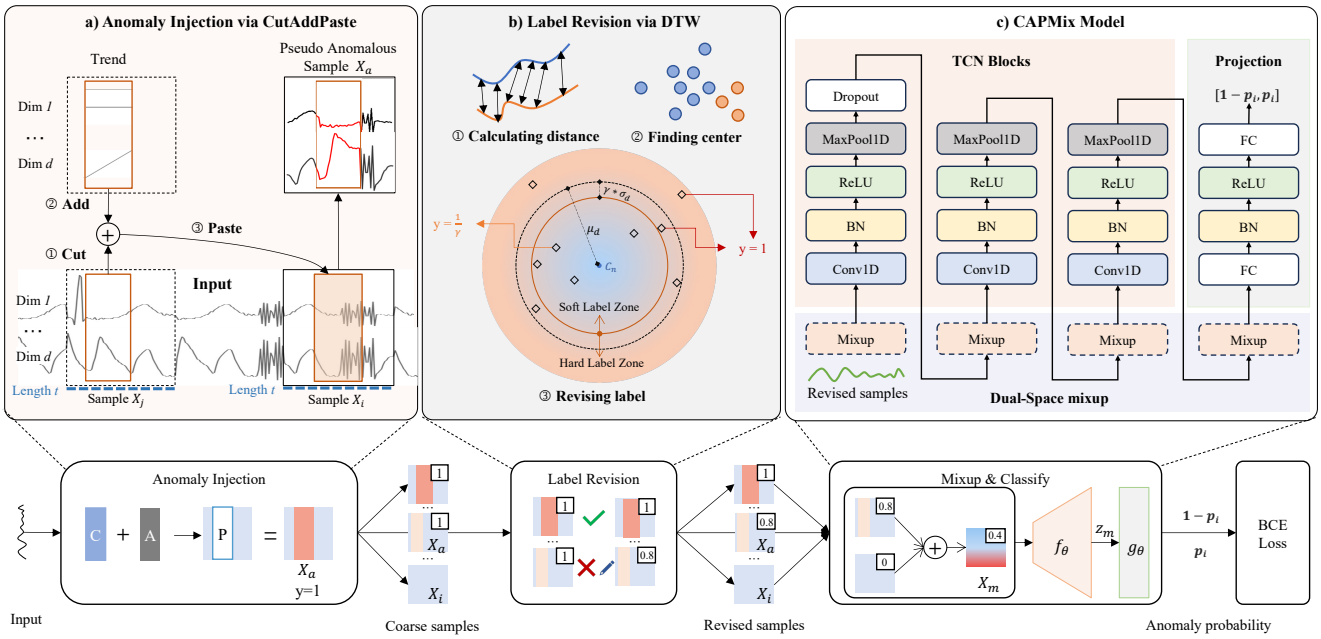


Figure 2: Overview of CAPMix. (a) CutAddPaste generates diverse synthetic anomalies by injecting patches into normal sequences. (b) Label Revision adjusts pseudo labels using DTW distance to the normal center, mitigating normal-like generations. (c) Dual-Space Mixup performs mixup in both input and latent spaces to learn robust, consistent decision boundaries.

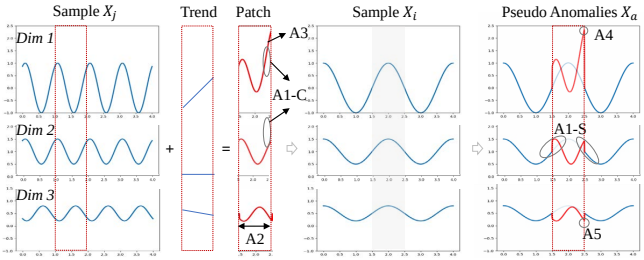


Figure 3: Example of generating an anomalous sample over sine waves. The original $Dim2 = 0.5 \cdot Dim1 + 1$. There are pattern-wise anomalies in terms of shapelet (A1-S: shape, A1-C: correlation), seasonality (A2), and trend (A3). Point-wise anomalies include global (A4) and contextual (A5) ones.

$t_2]$ 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.

3.3.2 Label Revision via DTW. KPI time series naturally exhibit benign fluctuations such as periodic oscillations, workload-driven jitter, and transient spikes. When we inject synthetic anomalies, such "normal-like" oscillations can cause an anomaly-shift effect: some injected samples may not present salient abnormal signatures, yet are still assigned one-hot anomaly labels. Such coarse supervision may blur the decision boundary and encourage over-sensitive detectors that misinterpret normal KPI turbulence as incidents.

To mitigate this label uncertainty, we propose a DTW-based label revision mechanism. For each generated sample, we compute its DTW distance to a normality center and assign a soft anomaly confidence in $[0, 1]$ rather than a hard one-hot label, so that samples close to normal KPI dynamics receive weaker anomaly supervision.

We first estimate the normality center C_n of the training data, assuming that the majority is normal. Given a collection of normal sequences $\mathcal{D} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, the center is computed as the mean sequence over all time steps:

$$C_n = \frac{1}{N} \sum_{i=1}^{N} X_i. \tag{2}$$

For each generated sample X_a, we compute its DTW distance $d(X_a, C_n)$ to C_n. Denote the mean and standard deviation of distances from the original training samples to C_n as μ_d and σ_d. We

then define a soft-label region controlled by the threshold $\mu_d + \gamma\sigma_d$:

$$d(X_a, C_n) \leq \mu_d + \gamma\sigma_d, \tag{3}$$

where $\gamma \geq 0$ adjusts the tolerance to normal KPI oscillations. Samples within this region are treated as ambiguous and assigned a soft anomaly label $y_r = 1/\gamma$. Samples outside the region are considered confidently abnormal and assigned a hard label $y_r = 1$. Then the original label set is refined as $\mathcal{Y}_r$. Formally, each label y_r in $\mathcal{Y}_r$ is:

$$y_r = \begin{cases} 1, & \text{if } d\left(X_{\text{syn},C_n}\right) > \mu_d + \gamma\sigma_d \\ \frac{1}{\gamma}, & \text{otherwise.} \end{cases} \tag{4}$$

Although one may assign continuous soft labels using a distance ratio (e.g., $d/(\mu_d+\gamma\sigma_d)$), this introduces extra computation and may yield unstable supervision due to scale sensitivity. We therefore use the controlled label $1/\gamma$ to balance robustness and training clarity.

3.3.3 Dual-Space Mixup Integrated TCN. Beyond mitigating under-distinctive synthetic anomalies via label revision, we also observe the opposite failure mode in AIOps KPI augmentation: injected anomalies may become *over-distinctive* and thus unrealistic. For instance, they may exhibit abrupt level shifts or extreme spikes that rarely occur in production KPIs. Training on such samples can bias the detector toward exaggerated signatures and reduce its sensitivity to subtle but operationally critical KPI degradations. To address this, we introduce a Dual-Space Mixup strategy within a TCN backbone, which regularizes the learned boundary in both the input space and the latent feature space.

The proposed CAPMix model employs a three-block TCN to convert a t-length sequence X_i into a fixed-size representation z_i. It is implemented by three temporal convolutional blocks $f^{(l)}(\cdot)$, where $l \in [1, 2, 3]$. Each block contains a Conv1D layer, a Batch Normalization (BN) layer, a ReLU activation function, and a MaxPool1D layer; the first block additionally includes Dropout. We perform mixup at configurable depths. For layer l, we mix intermediate features and their corresponding labels as:

$$\begin{aligned} X_{Mix}^{(l)} &= \lambda \cdot f^{(l)}(X_i) + (1-\lambda) \cdot f^{(l)}(X_j), \\ y_{Mix}^{(l)} &= \lambda \cdot y_i^{(l)} + (1-\lambda) \cdot y_j^{(l)}, \end{aligned} \tag{5}$$

where $y^{(l)}$ denotes the label associated with the l-th mixup block. In particular, setting $l = 0$ in Eq. (5) yields the standard input-space mixup, i.e., directly mixing the raw inputs X and labels y. This layer-wise design enables switching mixup on/off at different levels, making the framework extensible to different KPI characteristics and anomaly types.

The representation z_i is further fed into a learnable nonlinear projector $g_\theta : C \mapsto Q$ to obtain the projection q_i. The projector is an MLP with one hidden layer using BN and ReLU, mapping encoder features into a 2-dimensional projection space. The training objective is the binary cross-entropy loss, detailed in Section 3.4.

Let $\hat{\mathcal{D}_{syn}}$ denote the revised synthetic anomaly distribution after applying DTW-based soft labeling and DS-Mixup. Intuitively, these two components jointly reduce the distribution gap between synthetic and real anomalies, i.e.,

$$Dist(\mathcal{D}_{real}, \hat{\mathcal{D}_{syn}}) < Dist(\mathcal{D}_{real}, \mathcal{D}_{syn}), \tag{6}$$

which we further validate via visualization in our experiments.

Algorithm 1 CAPMix Pipeline

Input: Subsequence set $\mathcal{D} = \{\mathbf{X}_i\}_{i=1}^N$, initial labels $\mathcal{Y} = \{y_i\}_{i=1}^N$. Parameters: $\rho, \zeta, e, \gamma, v$.
Output: Optimized networks f, g.

Stage 1: Adaptive Data Augmentation
for each $\mathbf{X}_i \in \mathcal{D}$ **do**
  $r \leftarrow \max(\zeta, \text{randint}(1, t))$, $pos_c, pos_p \leftarrow \text{randint}(1, t-r)$;
  Extract $\mathbf{X}_{sub} \leftarrow \mathbf{X}_j[pos_c : pos_c{+}r]$ from random $j \in [1, N]$;
  Select e dimensions $\mathcal{D}_{sub} \subset \{1, \ldots, d\}$;
  $\mathbf{X}_{sub}[:, \mathcal{D}_{sub}] \leftarrow \mathbf{X}_{sub}[:, \mathcal{D}_{sub}] + \text{Trend}(\rho)$ ▷ Inject drift on e dimensions;
  Generate augmented $\mathbf{X}_i'$ by pasting $\mathbf{X}_{sub}$ into $\mathbf{X}_i$ at pos_p;
Stage 2: Label Revision & Refinement
$\mathcal{D}' \leftarrow \text{RandomSample}(\{\mathbf{X}_i'\}, v \cdot N)$, set initial labels $\mathcal{Y}' = \{1\}^N$;
Compute normal center C_n = Eq.(2) using $\mathcal{D}$;
Revise labels $y_i' \in \{1, 1/\gamma\}$ for $\mathbf{X}_i' \in \mathcal{D}'$ via Eq. (4);
$\mathcal{D}_{total} \leftarrow \mathcal{D} \cup \mathcal{D}'$, $\mathcal{Y}_{total} \leftarrow \mathcal{Y} \cup \mathcal{Y}'$;
Stage 3: Joint Optimization with DS-Mixup
while not converged **do**
  Sample a mini-batch $\{\mathbf{X}_b, y_b\} \subset \{\mathcal{D}_{total}, \mathcal{Y}_{total}\}$;
  Forward pass with DS-Mixup via Eq. (5);
  Compute $\mathcal{L}$ via Eq. (7) and update f, g;
return f, g

3.3.4 Putting LR and DS-Mixup Together. Label Revision (LR) via DTW addresses the *low-distinctiveness* side of anomaly shift: due to benign KPI oscillations (e.g., periodicity and workload-driven jitter), some injected samples may remain close to normal dynamics, and hard anomaly labels can be overconfident. LR uses DTW distance to the normality center to calibrate such samples with soft labels, reducing misleading supervision. DS-Mixup complements LR by addressing the *over-distinctiveness* side: it constrains synthetic anomalies that deviate excessively from plausible KPI behaviors. Input-space mixup generates hybrid KPI segments that preserve realistic local dynamics, while latent-space mixup interpolates intermediate representations and soft labels to encourage smooth transitions in feature space. Together, these operations promote smoother decision boundaries and improve generalization across diverse KPI fluctuations and anomaly patterns.

We will validate the contribution of LR and DS-Mixup through ablation experiments in Section 4.3.

3.4 Model Training Objective

The loss is obtained by calculating the projection q_i and the label y_i. Inspired by hypersphere classification (HSC) [7, 27], we use the binary 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 binary cross-entropy loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(p_i) + (1-y_i) \cdot log(1-p_i)], \tag{7}$$

Table 2: Datasets description.

Dataset	#Ent	#Var	Domain	Points	Tra-Ano
AIOps [38]	29	1	Cloud KPIs	203,316	3.86%
ASD [39]	12	19	Application	154,171	0%
SMD [39, 40]	28	38	Server	608,342	0%
Exathlon [41]	8	19	Spark	141,074	0%
UCR [42]	250	1	Various	1,383,502	0%
SWaT [43]	1	51	Waterworks	63,441	0%
WADI [44]	1	127	Waterworks	61,993	0%
ESA [45]	1	6	Satellite	24,547,200	0.6%

where y_i is the label in $\mathcal{Y}_r$. For the training set with labels, $\mathcal{Y}_r$ 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 shown in Table 3, CAPMix performs well on UCR, SWaT, and WADI, all of which are generic training sets without anomalies. The pseudo-code of CAPMix in Pytorch style is provided in Algorithm 1.

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:

$$x_i = \begin{cases} anomaly, & S_i > \tau \\ normal, & S_i \leq \tau \end{cases}, \tag{8}$$

where τ is a predefined threshold.

4 Experiments

We systematically evaluate CAPMix on multiple real-world KPI anomaly detection datasets to answer the following research questions.

- RQ1: What is the overall performance of CAPMix?
- RQ2: How do the key components of CAPMix contribute to performance and alleviate anomaly shift?
- RQ3: How robust is CAPMix under varying contamination/noise ratios?
- RQ4: How do the key hyperparameters of CAPMix affect its performance?

The code is available at https://github.com/alsike22/CAPMix.

4.1 Experiment Setup

4.1.1 Datasets. The anomaly detection evaluation is conducted over KPI metrics datasets including AIOps [38], ASD [39], SMD [39, 40], and Exathlon [41]. On the other hand, time-series datasets from other domains are also included to verify the generalizability and scalability of CAPMix: UCR [42], SWaT [43], WADI [44], and ESA [45] datasets.

Table 2 summarizes the datasets and our preprocessing/splitting protocol. We segment each time series into sliding windows of length t with step δ. Here, Ent denotes the number of subsets/entities, Var the number of variables, Points the total number of data points, and Tra-Ano the proportion of anomalies in the training set. AIOps shows non-trivial training contamination (Tra-Ano > 0), aligning with the unsupervised setting.

4.1.2 Metrics. We evaluate anomaly detection using Revised Point-Adjusted (RPA) F1 [56], which avoids the underestimation of point-wise (PW) metrics and the overly optimistic behavior of point-adjusted (PA) metrics [57]. Following common practice, we report the *best* F1 score (Best RPA-F1) by sweeping the decision threshold over anomaly scores, which reflects a model/method's intrinsic ability to separate normal and anomalous patterns. In practical deployment, anomaly scores can also be converted to alarms via unsupervised thresholding methods (e.g., POT). For datasets with multiple subsets (AIOps and UCR), we report a weighted average over subsets, $\mathrm{F1}_{\mathrm{entire}} = \sum_{i=1}^{M} \frac{e_i}{E} \mathrm{F1}_i$, where M is the number of subsets, E is the total number of anomalies, and e_i is the anomaly count in subset i.

4.1.3 Baselines. We compare CAPMix with representative baselines, covering traditional methods, deep detectors built on the normality assumption, and anomaly-assumption/injection approaches.

Traditional AD baselines. We include One-Class SVM (OC-SVM) [46], Isolation Forest (IF) [47], Robust Random Cut Forest (RRCF) [48], Spectral Residual (SR) [49], and DAMP [50]. For the large ESA dataset, these baselines are adapted to handle long windows.

Normality-assumption baselines. We evaluate deep detectors under single assumptions—reconstruction, one-class classification, or prediction—including LSTM-ED [4], Deep SVDD [5], Anomaly Transformer [53], MixMamba [54], MTSCAD [55], Sensitive-HUE (SHUE) [16], CATCH [17], and MOC [20]. We also include fused/multi-assumption baselines: RoCA [24], AOC [51], and TCC [23, 52].

Anomaly-assumption baselines. We consider NCAD [9], AnomalyBert [10],RedLamp [29], and CutAddPaste [11]. For NCAD, we use its supervised setting on AIOps as it yields better performance. Following prior work [9, 24, 51], Conv1D is adopted to migrate Deep SVDD-style objectives to time series. For TCC, we pre-train representations with TS-TCC [52] and fine-tune using Deep SVDD principles.

4.2 Main Results (RQ1)

We report the AD performance of the baselines and CAPMix in Table 3, in terms of F1 score. From top to bottom, the table is organized into three groups: traditional, normality assumption-based, and anomaly assumption-based methods, from which several key observations can be made.

First, on the KPI anomaly detection (KAD) benchmarks (AIOps, SMD, ASD, and Exathlon), CAPMix consistently achieves strong RPA F1 scores and remains competitive across datasets with diverse KPI dynamics. In contrast, traditional and normality-assumption-based methods (e.g., OC-SVM, Deep SVDD, and AOC) show larger performance fluctuations, suggesting sensitivity to dataset-specific noise patterns and anomaly characteristics. Second, on the remaining four datasets, the results indicate that CAPMix also serves as a general-purpose time-series anomaly detector. In particular, multivariate settings remain challenging for most baselines, while CAPMix provides consistent gains and achieves the best overall performance on SWaT/WADI and a strong result on ESA.

Table 3: Main results[1].

Method	KPI benchmarks				Other time-series domain				Average
	AIOps	SMD	ASD	Exathlon	UCR	SWaT	WADI	ESA	
OC-SVM [46]	23.65	11.37	24.05	31.52	16.42	0.02	0.03	29.79	17.11
IF [47]	3.86±0.07	6.07±0.32	16.82±0.72	44.05±1.65	4.51±0.44	12.37±2.39	0.88±0.13	0.09±0.00	11.08
RRCF [48]	2.89±0.06	4.24±0.15	10.74±0.67	15.46±1.16	5.91±0.67	0.99±0.05	0.95±0.09	6.43±1.72	5.95
SR[2] [49]	8.52	–	–	–	22.00	-	-	-	15.26
DAMP [50]	0.18	4.29	4.87	16.87	32.26	0.70	0.18	0.09	7.43
LSTM-ED [4]	14.05±0.52	31.08±0.43	24.09±1.11	51.64±0.28	18.12±0.62	4.44±2.22	3.86±0.52	54.80±2.92	25.26
Deep SVDD [5]	22.36±8.05	46.93±4.02	33.83±5.16	56.05±1.83	47.54±2.33	13.57±10.35	6.59±2.10	4.00±0.72	28.86
AOC [51]	39.82±4.95	**60.85±0.29**	35.11±1.07	61.59±0.44	21.39±0.80	27.59±0.00	0.36±0.00	55.90±2.64	37.83
TCC [23, 52]	3.28±1.36	5.54±1.22	9.83±0.83	59.54±19.35	0.50±0.09	16.71±8.80	2.09±4.03	25.30±21.00	15.35
AnoTrans [53]	0.58±0.51	17.49±6.79	5.65±6.01	23.03±12.16	6.38±2.56	33.07±1.75	1.68±0.02	1.13±0.20	11.13
MixMamba [54]	16.72±0.13	27.72±0.27	31.54±0.81	44.30±0.53	12.88±0.54	2.42±0.07	3.68±0.68	25.20±5.47	20.56
MTSCAD [55]	44.85±1.24	51.70±0.70	27.22±0.73	63.34±0.37	10.04±0.92	19.70±0.40	5.01±2.50	51.64±5.82	34.19
RoCA [25]	50.36±5.12	10.04±4.72	18.88±10.23	45.26±9.71	55.49±2.47	30.31±3.63	14.68±6.68	31.41±22.64	32.05
SHUE [16]	11.01±2.47	22.89±3.48	12.25±1.51	52.5±1.96	10.22±1.12	30.14±3.67	**35.57±0.84**	16.36±4.93	23.87
CATCH [17]	25.63±0.91	57.51±1.07	25.89±0.15	56.00±0.06	0.06±0.00	2.72±0.01	0.03±0.00	16.46±0.00	23.04
MOC [20]	53.66±3.76	38.58±10.09	36.77±5.01	41.61±5.23	7.18±2.40	27.19±2.14	14.14±8.46	3.07±3.51	27.78
NCAD [9]	41.06±3.32	16.49±3.07	32.14±7.64	24.55±2.94	22.24±2.99	7.54±2.48	6.84±2.53	22.03±1.07	21.61
AnomalyBert [10]	23.99±1.22	19.54±7.15	23.75±6.85	45.49±8.53	2.77±0.12	13.36±2.21	11.79±1.56	6.26±5.50	18.37
RedLamp [29]	23.21±1.26	54.72±0.97	**69.59±1.33**	63.4±0.61	1.41±0.08	27.6±0.03	4.29±0.00	45.59±12.93	36.23
CutAddPaste [11]	77.44±0.86	22.19±11.84	38.20±2.59	23.80±6.89	69.98±1.44	43.43±4.86	26.55±5.66	18.56±2.70	40.02
CAPMix	**80.45±0.43**	27.42±12.50	61.30±1.96	**78.89±1.61**	**71.89±1.89**	**47.04±2.70**	34.08±6.67	**84.46±5.50**	**60.69**

[1] Average Best RPA F1-score (%) with standard deviation for baselines and our method on datasets over 10 runs. The best results are in bold, and the suboptimal ones are underlined.

[2] SR is a univariate AIOps solution, yet it cannot be applied to multivariate time series anomaly detection.

Across baselines, classic methods such as DAMP and SR, can be competitive on some univariate cases, whereas fused-normality approaches including RoCA and AOC are generally stronger than single-assumption deep models, highlighting the value of combining complementary normality signals. Finally, among anomaly-assumption-based methods, CutAddPaste and RedLamp are strong baselines, and CAPMix further mitigates anomaly shift, yielding the best average performance across all eight datasets.

4.3 Ablation Study (RQ2)

4.3.1 Component Ablation. We study four variants, ranging from a vanilla TCN to the full CAPMix. Table 4 reports RPA scores, where the Average column measures the relative drop with respect to the full model, revealing the contribution of each component.

- **NoAug.** TCN trained on original data with BCE loss.
- **CAP.** Adds CutAddPaste-based anomaly injection.
- **CAP-*lr*.** Further introduces label revision by assigning soft labels $(1/\gamma)$ to near-normal synthetic anomalies.
- **CAP-*mix*.** Applies dual-space mixup on injected samples without label revision.

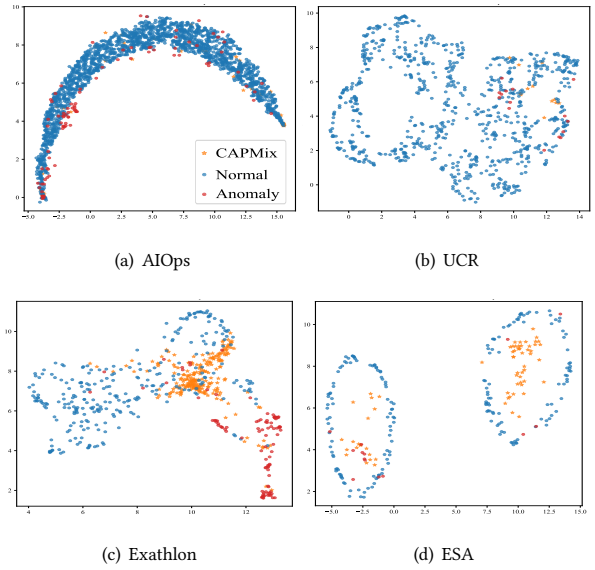
Removing augmentation (*NoAug*) yields the weakest results on all datasets, suggesting that learning from limited normal data alone is insufficient for robust anomaly detection in noisy settings; moving to *CAP* consistently improves performance, with larger gains on KPI benchmarks such as AIOps and ASD, indicating that prior-guided anomaly injection is more effective than relying on raw noisy observations because it encourages the model to attend to structured deviation patterns rather than incidental fluctuations. Adding label revision (*CAP-lr*) further improves over CAP, especially on KPI datasets, implying that calibrated supervision under noise stabilizes representation learning and reduces the impact of contaminated or ambiguous samples. Dual-space mixup (*CAP-mix*) provides substantial benefits across domains, with particularly clear gains on challenging datasets such as ESA; compared with CAP and CAP-*lr*, DS improves generalization, suggesting that smoothing decision boundaries in both input and representation spaces enhances robustness to distribution shifts and noise. Finally, the full model achieves the best results on nearly all datasets, confirming that the three components are complementary: AJ and LR mainly strengthen robustness under noisy supervision, while DS further improves cross-domain generalization, yielding a more stable and reliable anomaly detection model.

Overall, CAPMix attains the best performance across datasets, reflecting the complementarity of label revision and DS-mixup:

Table 4: Ablation results of average Best F1-score (%)) with standard deviation over 10 runs.

Variant	Components[1]			KPI benchmarks				Other time-series domain				Average
	C	L	M	AIOps	SMD	ASD	Exathlon	UCR	SWaT	WADI	ESA	
NoAug	×	×	×	74.25±2.60	15.58±11.74	15.56±6.50	23.80±6.89	59.08±2.42	21.64±8.42	7.90±6.44	15.33±7.84	29.14 (↓ 31.55)
CAP	✓	×	×	77.44±0.86	22.19±11.84	38.20±2.59	73.73±4.72	69.98±1.44	43.43±4.86	26.55±5.67	18.56±2.70	46.26 (↓ 14.43)
CAP-*lr*	✓	✓	×	80.13±0.53	24.06±10.74	42.29±3.40	73.73±4.72	69.98±1.44	43.43±4.86	26.55±5.67	18.56±2.70	47.34 (↓ 13.35)
CAP-*mix*	✓	×	✓	78.15±0.86	23.77±11.62	45.55±3.66	78.18±2.90	68.65±2.45	47.04±2.70	34.08±6.67	84.46±6.67	57.49 (↓ 3.20)
Total	✓	✓	✓	80.45±0.43	27.42±12.50	61.30±1.96	78.89±1.61	71.89±1.89	47.04±2.70	34.08±6.67	84.46±6.67	60.69

[1] C refers to the process of anomaly injection via CutAddPaste. L refers to label revision. M refers to dual-space mixup.



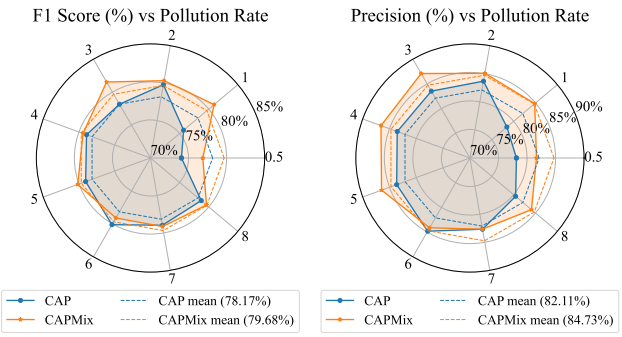
(a) AIOps (b) UCR

(c) Exathlon (d) ESA

Figure 4: UMAP visualization of samples generated by CAPMix (orange), along with normal (blue) and real anomalous (red) test samples. CAPMix aligns better with real anomalies while largely avoiding overlap with normal regions.

calibrated supervision improves anomaly plausibility, while dual-space interpolation increases sample diversity and generalization.

4.3.2 Alleviation of Anomaly Shift. To further analyze whether CAPMix can alleviate anomaly shift, we visualize UMAP embeddings of synthetic samples in Fig. 4. The orange stars denote synthetic anomalies generated by CAPMix, and the blue/red dots correspond to normal/real anomalous test samples. We observe that CAPMix produces pseudo-anomalies that align better with the real abnormal distribution while largely avoiding overlap with normal regions, which helps reduce the risk of learning a biased decision boundary. Especially in Fig. 4(c), the orange points not only cover anomalous samples close to the normal cluster but also effectively inject anomalies in the lower-right area far away from the normal distribution, suggesting strong ability against anomaly shift.



(a) F1-score (b) Precision

Figure 5: Robustness of CAPMix vs. CutAddPaste (CAP) under increasing training-set contamination (0.5×–8×), evaluated by F1 and Precision.

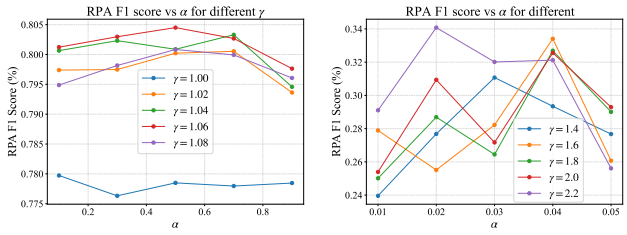
4.4 Robustness Analysis (RQ3)

To further evaluate robustness under realistic conditions, we conduct a robustness study on the AIOps dataset, where training data naturally contains anomalous samples. We simulate different levels of data contamination by progressively injecting additional anomaly samples into the training set, thereby controlling the severity of training-set pollution.

As Fig.5 shows, both CAPMix and the original CAP baseline demonstrate a certain degree of robustness under mild contamination. However, a clear performance gap emerges as the pollution becomes more severe. While CAP maintains relatively stable performance at lower corruption levels, its performance gradually degrades as the contamination increases. In contrast, CAPMix consistently preserves higher F1-scores and precision across all settings, with the advantage becoming more pronounced at high pollution levels (e.g., 8×).

This trend suggests that CAPMix is more resilient to heavily contaminated or drifted training distributions. The improvement can be attributed to two factors: (i) label revision mitigates the impact of noisy or misleading supervision, and (ii) dual-space mixup smooths the decision boundary by interpolating both input and representation spaces, reducing overfitting to corrupted samples.

Overall, these results indicate that CAPMix not only inherits the robustness of CAP under moderate noise, but also provides



(a) α and γ on AIOps (b) α and γ on WADI

Figure 6: Effectiveness of different γ values (five lines) and α settings on the RPA F1 score. Each line represents one γ; points along each line correspond to different α values.

additional stability under severe contamination, making it more suitable for real-world deployments where training data is often imperfect and continuously evolving.

4.5 Hyperparameter Analysis (RQ4)

We conduct a sensitivity analysis on two representative datasets, AIOps and WADI, to evaluate the impact of the label revision range (γ) and the mixup ratio (α). These datasets exhibit distinct characteristics in terms of dimensionality and temporal complexity, enabling us to examine how the two components behave under different data regimes. For each dataset, we vary γ across multiple levels and sweep α within a practical range; the results are shown in Fig. 6. Other hyperparameters, such as the trend degree ρ and minimum patch length ζ, follow the settings in our previous work.

On AIOps, the performance curves under different γ are clearly separated, indicating that the revision range is the dominant factor. In contrast, the model is relatively insensitive to α, suggesting that mixup provides limited benefit when anomalies are simple and locally distinguishable. On WADI, the trend differs: performance varies more noticeably with α, highlighting the importance of dual-space mixup in capturing complex inter-variable dependencies. Meanwhile, γ still provides complementary gains by constraining near-normal synthetic anomalies and stabilizing supervision.

Overall, these results indicate that the effectiveness of the two components depends on data characteristics. The revision range γ primarily calibrates supervision for ambiguous or normal-like synthetic samples, while the mixup ratio α improves diversity and robustness, particularly in settings with complex temporal dependencies. The two components, therefore, play complementary roles across different data regimes.

5 Case Study in Kuaishou

CAPMix is integrated with KAIOps [12], the production-grade AIOps framework for Kuaishou's AI clusters that accommodate large-scale model training jobs and real-time recommendation services. Figure 7 showcases the end-to-end workflow that encompasses both offline and online phases. In particular, a feedback loop is formed by alert validation results, such as operator-confirmed false positives and true incidents, which are periodically aggregated and used to refine the label revision stage. In this section, we provide a qualitative case analysis and a controlled evaluation dataset to demonstrate robustness under real operational fluctuations.

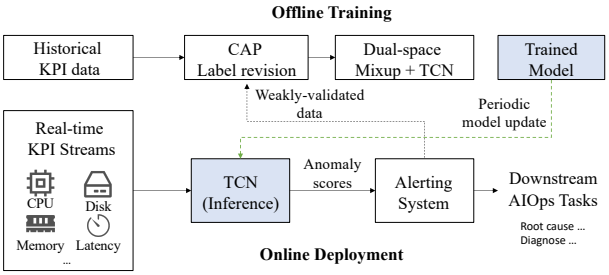


Figure 7: Deployment overview of CAPMix in KAIOps.

5.1 Applying CAPMix in Large-scale AI Clusters

5.1.1 KPI Streams in AI Clusters. We collected KPI streams generated by production-grade workloads, together with resource utilization logs, from industrial-scale AI computing clusters.

The measurements are inherently noisy, and the system dynamics exhibit temporal drift. Since most events correspond to benign operational behaviors, the setting is highly contaminated, i.e., normal events often exhibit anomaly-like patterns. Therefore, we conduct chaos engineering by intentionally introducing failures to validate the infrastructure's resilience.

We injected anomalies by executing representative fault workloads directly on the target nodes. This approach preserves natural system dynamics and yields KPI traces that reflect realistic fault behaviors. Because faults are injected at the node level, anomalies often affect only a subset of metrics while others remain normal, resulting in cross-metric inconsistency and partial observability. Combined with the prevalence of benign fluctuations, this creates a challenging evaluation scenario with noisy, ambiguous, and weakly structured anomalies.

To ensure reproducibility, we publicly released a dataset covering a continuous period of 16 days with a temporal resolution of one minute. A statistical overview of the dataset is provided in Table 5. The dataset consists of node-level, multivariate KPI traces from a representative and heterogeneous subset of seven nodes. These KPIs include, for example, CPU utilization, memory consumption, disk input/output (I/O) metrics, and network traffic measurements. In addition, the dataset contains cluster-level observability indicators, such as the number of pending tasks, API request latency, and the number of container restart events.

5.1.2 CAPMix's Performance. We evaluate CAPMix's performance in tackling real-world KPI data streams. The streams encompass both genuine anomalous events and regular operational events (such as metric scraping and log rotation) that frequently induce transient perturbations in the observed metrics, which are close to failure patterns.

As shown in Fig. 8, the first two rows depict raw KPI signals from two nodes with annotated intervals. Dark-shaded regions indicate anomalous periods, such as those induced by memory leakage, whereas light-shaded regions correspond to benign events. Notably, certain extreme spikes (e.g., negative CPU values) arise from measurement noise or monitoring artifacts, rather than actual system events. The last three rows present a comparison of anomaly

Table 5: Statistical overview of the AIClusterKPI dataset.

Dataset Configuration	
Total Data	20,452 points
Training Set	17,571 points
Test Set	3,029 points / 598 events
Test Anomaly	1,117 points / 145 events
Node Signals	CPU, Memory, IO, Network
Cluster Signals	API Latency, Pending Tasks, Pod Restarts
Label Granularity	Binary (0/1), Fault Type, Node ID
Typical Faults	CPU_high, Mem_leak, IO_burst
CAPMix Performance as Baseline	
Detection Accuracy	RPA Precision: 0.80 / F1-score: 0.79
Alert Reliability	PR-AUC: 0.87 / NAB Score: 0.76

scores produced by CAPMix against scores by a representative point-wise baseline, OC-SVM, and CutAddPaste.

OC-SVM exhibits frequent oscillations in its anomaly scores across both anomalous and non-anomalous regions, leading to spurious peaks during benign events. CutAddPaste baseline partially mitigates this phenomenon but still produces undesirably high scores in several non-anomalous intervals. In contrast, CAPMix produces consistently low scores in the presence of transient noise-like fluctuations while producing clear, temporally sustained responses to genuine anomalous events. Around time point 250, a scheduled log rotation triggers a sharp, but operationally benign, spike in disk I/O. OC-SVM responds with pronounced score fluctuations, whereas CAPMix remains comparatively stable, indicating enhanced robustness to such benign perturbations. A similar pattern is observed in the benign intervals preceding indices 150 and 200, where both OC-SVM and CutAddPaste assign elevated anomaly scores. Likewise, near index 400, metric scraping again introduces irregular peaks, resulting in a high anomaly score for CutAddPaste. CAPMix maintains a lower and more stable response.

We report raw outputs without post-processing, highlighting the intrinsic behavior of each method. Overall, CAPMix demonstrates improved robustness to operational noise and more reliable alignment with sustained anomaly intervals, indicating its potential to reduce false alarms in real-world deployments.

5.2 Engineering Lessons and Experiences

We summarize several lessons from deploying CAPMix in Kuaishou.

Robust boundary learning is more effective than clean-data assumptions. Production KPI streams inevitably contain noise and transient fluctuations, under which methods relying on clean training data tend to overreact and trigger false alarms. We observe that learning robust decision boundaries under contamination is more effective in practice. By combining CAP-based augmentation with Dual-Space Mixup, CAPMix focuses on structural failure patterns while remaining insensitive to benign fluctuations, which is consistent with the reduction in false positives observed above.

Labels should be treated as uncertain supervision signals. Industrial anomaly labels are sparse, delayed, and often unreliable, and pseudo-anomalies may overlap with real failures. Treating such labels as ground truth can degrade performance. The Label Revision (LR) mechanism addresses this issue by refining supervision based on representation consistency, improving stability under noisy and evolving environments.

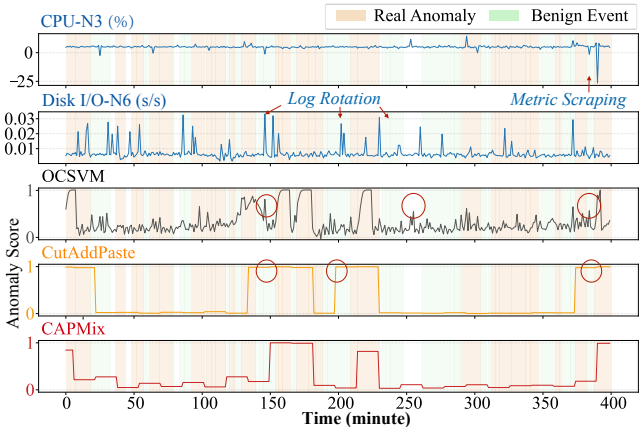


Figure 8: Detection performance on real-world data. CAPMix successfully suppresses false alarms during benign events.

Computational efficiency is a primary constraint in deployment. In SLA-critical monitoring systems, inference cost directly affects usability at scale. We observe that heavyweight architectures can introduce significant latency under peak workloads. By adopting a lightweight TCN backbone with efficient dual-space operations, CAPMix achieves low-latency inference, enabling practical deployment in large-scale AI clusters.

6 Conclusions

This paper addresses the challenge of robust KPI anomaly detection in noisy and non-stationary AIOps environments. By formalizing *Anomaly Shift*, we highlight the critical gap between synthetic training patterns and evolving real-world failures. To bridge this gap, we propose CAPMix, a controllable augmentation framework that integrates prior-guided injection, DTW-based label refinement, and Dual-Space Mixup into a lightweight TCN architecture. Extensive evaluations across eight public datasets demonstrate that CAPMix significantly outperforms state-of-the-art baselines in Best RPA-F1 score and maintains exceptional resilience under data contamination. Beyond academic metrics, CAPMix has been successfully integrated into Kuaishou's production AI clusters, where it effectively mitigated alert fatigue and reclaimed operational trust. By releasing the AIClusterKPI dataset and our codebase, we aim to provide the community with a more realistic dataset for advancing robust and deployment-ready anomaly detection.

7 Mandatory Data Availability Statement

AIClusterKPI is a dataset derived from industrial production settings, containing 16 days of multi-dimensional metrics with real dynamic system changes. Sensitive metadata is de-identified, and metrics are normalized; the code is publicly available on GitHub (https://github.com/alsike22/CAPMix) and the data is archived on Zenodo (https://doi.org/10.5281/zenodo.19926114). All datasets used in our main experiments (in Table 3) are public benchmarks.

Acknowledgment

We would very much like to thank anonymous reviewers for their valuable comments. This work is supported in part by National Key R&D Program of China (Grant No. 2024YFB4505901), in part by NSFC (Grant No. 62402024, 62302485), in part by the Fundamental Research Funds for the Central Universities, in part by the Key Research Project of Chinese Academy of Sciences (No. RCJJ-145-24-21), and, last but not the least, by Kuaishou Research Fund. For any correspondence, please refer to the project lead and coordinator Prof. Renyu Yang (renyuyang@buaa.edu.cn).

References

[1] Nengwen Zhao, Junjie Chen, Zhaoyang Yu, Honglin Wang, Jiesong Li, Bin Qiu, Hongyu Xu, Wenchi Zhang, Kaixin Sui, and Dan Pei. Identifying bad software changes via multimodal anomaly detection for online service systems. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 527–539, 2021.

[2] Yongqian Sun, Daguo Cheng, Tiankai Yang, Yuhe Ji, Shenglin Zhang, Man Zhu, Xiao Xiong, Qiliang Fan, Minghan Liang, Dan Pei, et al. Efficient and robust kpi outlier detection for large-scale datacenters. *IEEE Transactions on Computers*, 72(10):2858–2871, 2023.

[3] Zhaoyang Yu, Changhua Pei, Xin Wang, Minghua Ma, Chetan Bansal, Saravan Rajmohan, Qingwei Lin, Dongmei Zhang, Xidao Wen, Jianhui Li, et al. Pre-trained kpi anomaly detection model through disentangled transformer. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6190–6201, 2024.

[4] 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.

[5] 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.

[6] Yongqian Sun, Binpeng Shi, Mingyu Mao, Minghua Ma, Sibo Xia, Shenglin Zhang, and Dan Pei. Art: A unified unsupervised framework for incident management in microservice systems. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 1183–1194, 2024.

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

[8] 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.

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

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

[11] Rui Wang, Xudong Mou, Renyu Yang, Kai Gao, Pin Liu, Chongwei Liu, Tianyu Wo, and Xudong Liu. Cutaddpaste: Time series anomaly detection by exploiting abnormal knowledge. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3176–3187, 2024.

[12] Zeying Wang, Junhong Liu, Penghao Zhang, Xiaoyang Sun, Xu Wang, Tianyu Wo, Chunming Hu, Chengru Song, Jin Ouyang, and Renyu Yang. Kaiops: A platform solution of end-to-end multi-modal aiops for ai training at scale. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 3192–3203. IEEE, 2025.

[13] 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.

[14] 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.

[15] 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.

[16] Yuye Feng, Wei Zhang, Yao Fu, Weihao Jiang, Jiang Zhu, and Wenqi Ren. Sensitivehue: Multivariate time series anomaly detection by enhancing the sensitivity to normal patterns. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 782–793, 2024.

[17] Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *The Thirteenth International Conference on Learning Representations*.

[18] Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[19] Yan Qiao, Kui Wu, and Peng Jin. Efficient anomaly detection for high-dimensional sensing data with one-class support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):404–417, 2021.

[20] Tiejun Wang, Rui Wang, Xudong Mou, Xu Li, Tianyu Wo, Shiru Chen, Xudong Liu, and Renyu Yang. Moc: Mamba-based multi-scale one-class time-series anomaly detection. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1106–1110. IEEE, 2026.

[21] 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.

[22] Qihang Zhou, Shibo He, Haoyu Liu, Jiming Chen, and Wenchao Meng. Label-free multivariate time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3166–3179, 2024.

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

[24] 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.

[25] Xudong Mou, Rui Wang, Bo Li, Tianyu Wo, Jie Sun, Hui Wang, and Xudong Liu. Roca: Robust contrastive one-class time series anomaly detection with contaminated data. *arXiv preprint arXiv:2503.18385*, 2025.

[26] 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.

[27] 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.

[28] Zahra Zamanzadeh Darban, Qizhou Wang, Geoffrey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. Genias: Generator for instantiating anomalies in time series. *arXiv preprint arXiv:2502.08262*, 2025.

[29] Kohei Obata, Yasuko Matsubara, and Yasushi Sakurai. Robust and explainable detector of time series anomaly via augmenting multiclass pseudo-anomalies. 2025.

[30] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[31] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[32] Shaoli Huang, Xinchao Wang, and Dacheng Tao. Snapmix: Semantically proportional mixing for augmenting fine-grained data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1628–1636, 2021.

[33] Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Do-Guk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness. *Advances in Neural Information Processing Systems*, 34:30153–30168, 2021.

[34] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019.

[35] Berken Utku Demirel and Christian Holz. Finding order in chaos: A novel data augmentation method for time series in contrastive learning. *Advances in Neural Information Processing Systems*, 36:30750–30783, 2023.

[36] Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. *arXiv preprint arXiv:2410.16032*, 2024.

[37] 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.

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

[39] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 3220–3230, 2021.

[40] 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.

[41] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. Exathlon: A benchmark for explainable anomaly detection over time series. *arXiv preprint arXiv:2010.05073*, 2020.

[42] 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.

[43] 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.

[44] 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.

[45] Krzysztof Kotowski, Christoph Haskamp, Jacek Andrzejewski, Bogdan Ruszczak, Jakub Nalepa, Daniel Lakey, Peter Collins, Aybike Kolmas, Mauro Bartesaghi, Jose Martinez-Heras, et al. European space agency benchmark for anomaly detection in satellite telemetry. *arXiv preprint arXiv:2406.17826*, 2024.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] Khaled Alkilane, Yihang He, and Der-Horng Lee. Mixmamba: Time series modeling with adaptive expertise. *Information Fusion*, 112:102589, 2024.

[55] Haotian Si, Changhua Pei, Zhihan Li, Yadong Zhao, Jingjing Li, Haiming Zhang, Zulong Diao, Jianhui Li, Gaogang Xie, and Dan Pei. Beyond sharing: Conflict-aware multivariate time series anomaly detection. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1635–1645, 2023.

[56] 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.

[57] 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.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009