Cluster-Aware Causal Mixer for Online Anomaly Detection in Multivariate Time Series

Md Mahmuddun Nabi Murad [1] **Yasin Yilmaz** [1]

Abstract

Early and accurate detection of anomalies in time-series data is critical due to the substantial risks associated with false or missed detections. While MLP-based mixer models have shown promise in time-series analysis, they do not maintain temporal causality during data processing. Moreover, real-world multivariate time series often contain numerous channels with diverse inter-channel correlations. Spurious correlations in the reconstructed time series lead to noisy representations, resulting in inaccurate anomaly detection. In addition, anomaly scoring methods that ignore temporal continuity can mislead sequential detection. To address these challenges, we propose a cluster-aware causal mixer for multivariate time-series anomaly detection. Channels are grouped into clusters based on their correlations, and each cluster is embedded through a dedicated embedding layer. A causal mixer is introduced to integrate information while maintaining temporal causality. We further develop a sequential anomaly-scoring method that accumulates evidence over time and refines anomaly boundaries. Our proposed model operates in an online fashion, making it suitable for real-time time-series anomaly detection. Experimental evaluations across six public benchmark datasets demonstrate that the proposed approach consistently achieves superior performance.

1. Introduction

Multivariate time series data consists of sequential measurements collected from multiple sources (e.g., sensors) over time. When there is an anomaly such as sensor malfunction or malicious data manipulation, the resulting patterns often deviate from normal behavior. Accurate and timely detection of such anomalies is critical in a wide range of applications, including the identification of cyberattacks (Ten et al., 2011) or sensor failures in critical infrastructure systems, such as water distribution networks (Goh et al., 2017). Considering the open-set possibilities for anomalies and the difficulty to sufficiently label and train on anomalous samples in real-world datasets, most research in time series anomaly detection has focused on unsupervised methods (Blázquez-García et al., 2021). These approaches generally involve training models to learn the temporal (and spatial for multivariate analysis) characteristics of normal training data.

The mainstream deep learning approaches learn normal patterns through training prediction or reconstruction models, in which anomalous test instances are expected to cause statistically larger forecast or reconstruction errors. Recent methods further incorporate uncertainty estimation, assigning low uncertainty to normal data and higher uncertainty to anomalous observations (Müller et al., 2025).

In recent years, transformer-based models have gained popularity in time series analysis, with many studies adopting transformer variants as the core architecture (Chen et al., 2021; Xu et al., 2021; Wang et al., 2023; Tuli et al., 2022; Lai et al., 2023; Feng et al., 2024). Among them, NPSR (Lai et al., 2023) and SensitiveHUE (Feng et al., 2024) further enhance performance by refining anomaly scoring mechanisms. In addition, recent studies, including (Zeng et al., 2023), question the effectiveness of transformer-based models for time series forecasting, showing that simpler multilayer perceptron (MLP) based models can achieve comparable performance. Furthermore, MLP-Mixer models have shown superior performance against transformer variants in time series forecasting tasks (Murad et al., 2025; Chen et al., 2023).

While transformers enforce temporal causality via attention masking, existing MLP-Mixer models (Zhong et al., 2025b) do not have such mechanisms. This motivates us to *study whether an explicit mechanism to ensure causal mixing can help in detecting anomalies in multivariate time series.*

[1]Department of Electrical Engineering, University of South Florida, Tampa, FL, USA. Correspondence to: Md Mahmuddun Nabi Murad <mmurad@usf.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

Our contributions are threefold:

- **First**, motivated by the strong performance of MLP-Mixer models and addressing their limitations regarding causality, we propose a novel **causal mixer** module that enforces strict temporal causality, ensuring that each representation at time t depends only on past and present information, eliminating any future leakage.
- **Second**, to capture meaningful inter-channel dependencies and mitigate spurious correlations in normal data, we introduce **cluster-aware multi-embedding**, which enhances the discriminability between normal and anomalous representations.
- **Third**, we propose a **sequential anomaly scoring method** that accumulates anomaly evidence over time, refining anomaly segment boundaries and improving overall anomaly detection performance.

2. Related Work

Time series anomaly detection methods are commonly classified as unsupervised (Munir et al., 2018) or supervised (Ma et al., 2016), and further categorized into point-based (Wang et al., 2025b) or sequence-based (Doshi et al., 2022) approaches. Extensive research on time-series anomaly detection has also been conducted in statistics through change-point detection and outlier detection methods (Aminikhanghahi & Cook, 2017). Classical machine learning approaches, such as k-nearest-neighbors (kNN), support vector machine (SVM), and Isolation Forest, have also been applied for anomaly detection (Chandola et al., 2009). With the reduced need for manual feature engineering and the increasing availability of computational resources, deep learning methods have become increasingly popular for time-series anomaly detection. With the rise of deep learning, models like LSTM (Lipton et al., 2015; Cho et al., 2025) have been used to capture long-term dependencies, and hybrid approaches such as LSTM-VAE have emerged (Park et al., 2018). Autoencoder-based models learn low-dimensional representations of normal data (Zong et al., 2018; Garg et al., 2021; Audibert et al., 2020). Time series can also be represented as graphs, enabling graph-based models like GDN (Deng & Hooi, 2021), MTAD-GAT (Zhao et al., 2020), and Graph-MoE (Huang et al., 2025). Recently, transformer-based models have shown superior performance in anomaly detection (Chen et al., 2021; Xu et al., 2021; Wang et al., 2023; Tuli et al., 2022; Lai et al., 2023; Feng et al., 2024). Some models improve the anomaly detection performance by refining the anomaly scores (Lai et al., 2023; Feng et al., 2024; Yue et al., 2024).

NPSR (Lai et al., 2023) introduces a nominality score that accounts for the influence of neighboring points while SensitiveHUE (Feng et al., 2024) incorporates heteroscedastic uncertainty into the reconstruction loss. To capture spatio-temporal dependencies, SensitiveHUE employs statistical feature elimination and robust normalization based on the median and interquartile range of anomaly scores, derived from the entire test set. However, this normalization requires access to the anomaly score of the entire test set beforehand, limiting the model's applicability in online anomaly detection. Although removing the robust normalization enables real-time detection, it substantially reduces performance, highlighting a key limitation of SensitiveHUE in scenarios requiring timely anomaly detection.

Recent studies (Sarfraz et al., 2024) show that simple baselines, such as PCA_Error, can outperform or perform similarly to complex models with Transformer and Graph-based variants, suggesting a simple architecture for the reconstruction model. Furthermore, the performance of the transformer-based models is questioned in paper (Zeng et al., 2023), while MLP-mixer-based models (Murad et al., 2025; Wang et al., 2024; Chen et al., 2023; Ekambaram et al., 2024; Wang et al., 2025a; Hong et al., 2025) outperform transformer variants in time series analysis. To ensure causality in time series, transformers employ masking in attention. However, to the best of our knowledge, no existing MLP-Mixer model incorporates mechanisms to preserve temporal causality during the mixing process.

To address these limitations, we propose a novel **C**luster-aware **C**ausal **M**ixer model for **T**ime Series **A**nomaly **D**etection (CCM-TAD). CCM-TAD enforces temporal causality within MLP-Mixer layers, ensuring that each time step's representation depends only on past and present inputs. Unlike causal discovery methods, our objective is anomaly detection with causally correct temporal mixing, enhanced by cluster-aware multi-embedding to improve normal data modeling and a sequential statistical scoring method with refined anomaly boundaries.

3. Proposed Method

Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_t, \ldots\}$ represent a multivariate time series, where $\boldsymbol{x}_t \in \mathbb{R}^{1\times C}$ represents the observation at time t, and C is the number of the channels. Each instance $\boldsymbol{x}_t$ is associated with a label $y_t \in \{0, 1\}$, where $y_t = 1$ and $y_t = 0$ indicate anomalous and normal states, respectively. The goal is to predict y_t for each time step.

The training set comprises only normal instances, while the test set includes both normal and anomalous data. At each time step t, a look-back window $\boldsymbol{X}_L = \{\boldsymbol{x}_{t-L+1}, \ldots, \boldsymbol{x}_t\} \in \mathbb{R}^{L\times C}$ is input to the model to reconstruct the latest observation $\boldsymbol{x}_t$. The reconstruction loss between $\boldsymbol{x}_t$ and the reconstructed $\hat{\boldsymbol{x}}_t$ is then used to derive the anomaly score. For subsequent predictions, the look-back window moves forward by one time step, and the

process repeats iteratively. All data are normalized using min–max scaling, with statistics computed from the training set.

Note that, similar to prior reconstruction-based anomaly detection methods, we assume that the training data is predominantly clean. Although this setting is commonly referred to as unsupervised anomaly detection in the literature, contamination in the training set may degrade detection performance.

3.1. Overall Model Architecture

The reconstruction architecture of our Cluster-aware Causal Mixer model for Time Series Anomaly Detection (CCM-TAD) is depicted in Figure 1. Our model consists of two key components: cluster-aware multi-embedding and causal mixer. At the beginning of our model, we pass the input $\boldsymbol{X}_L \in \mathbb{R}^{L\times C}$ through the cluster-aware multi-embedding module (Section 3.2) to get embedded output $\boldsymbol{X}_d \in \mathbb{R}^{L\times d}$. We then apply 1D batch normalization to the output of the embedding module, resulting in $\underline{\boldsymbol{X}_d} \in \mathbb{R}^{L\times d}$. $\underline{\boldsymbol{X}_d}$ is then processed through multiple causal mixer modules. Each causal mixer module (Section 3.3) comprises a temporal mixer module and an embedding mixer module. The temporal mixer module consists of two causal linear layers connected by a non-linear GELU activation. The operations of the temporal mixer module are summarized as

$$\boldsymbol{X}_c = \mathcal{P}(\underline{\boldsymbol{X}_d}) \in \mathbb{R}^{d\times L} \tag{1}$$

$$\boldsymbol{X}_c^{'} = \mathcal{L}_2(\mathcal{G}(\mathcal{L}_1(\boldsymbol{X}_c))) \in \mathbb{R}^{d\times L} \tag{2}$$

$$\boldsymbol{X}_e = \mathcal{BN}(\mathcal{P}(\boldsymbol{X}_c^{'}) + \underline{\boldsymbol{X}_d})) \in \mathbb{R}^{L\times d}, \tag{3}$$

where $\mathcal{L}_1 : \mathbb{R}^{\cdots\times L} \to \mathbb{R}^{\cdots\times L}$, $\mathcal{L}_2 : \mathbb{R}^{\cdots\times L} \to \mathbb{R}^{\cdots\times L}$, $\mathcal{G}(.)$, $\mathcal{P}(.)$, and $\mathcal{BN}(.)$ represent the causal linear layer-1, causal linear layer-2, GELU activation, permute operation, and 1D batch normalization, respectively. After the temporal mixer, the data passes through the embedding mixer module:

$$\boldsymbol{X}_e^{'} = \mathcal{L}_2^{'}(\mathcal{G}(\mathcal{L}_1^{'}(\boldsymbol{X}_e))) \in \mathbb{R}^{L\times d} \tag{4}$$

$$\underline{\boldsymbol{X}_d^{'}} = \mathcal{BN}(\boldsymbol{X}_e^{'} + \boldsymbol{X}_e + \underline{\boldsymbol{X}_d}) \in \mathbb{R}^{L\times d} \tag{5}$$

where, $\mathcal{L}_1^{'} : \mathbb{R}^{\cdots\times d} \to \mathbb{R}^{\cdots\times(d\cdot d_f)}$ and $\mathcal{L}_2^{'} : \mathbb{R}^{\cdots\times(d\cdot d_f)} \to \mathbb{R}^{\cdots\times d}$ represent standard linear layers, d_f is a hyperparameter (expansion factor).

After the data is processed through a series of causal mixer modules, it is combined with $\underline{\boldsymbol{X}_d}$ and subsequently normalized, yielding $\boldsymbol{X}_h \in \mathbb{R}^{L\times d}$. This representation serves as the input to the head layer, which consists of a single linear layer $\mathcal{L}_h : \mathbb{R}^{\cdots\times d} \to \mathbb{R}^{\cdots\times C}$, mapping the d dimensional data to C dimensional representation. The model then outputs the last reconstructed observation $\hat{\boldsymbol{x}}_t \in \mathbb{R}^{1\times C}$, which is further processed to obtain the anomaly score using our proposed anomaly detection method described in Section 3.4.

3.2. Cluster-Aware Multi-Embedding

Our proposed cluster-aware multi-embedding combines channel clustering and multi-embedding to enhance normal representation. We first assign each channel a cluster index based on its correlation pattern computed offline from the training dataset. This cluster index for each channel is then utilized in the clustering module during training and inference to group the input channels into clusters. Then, a separate embedding layer is employed for each channel cluster. The outputs of all embedding layers are concatenated to obtain the final high-dimensional representation. This cluster-aware multi-embedding technique mitigates spurious correlations that arise when channels with heterogeneous relationships are jointly embedded, and improves the discriminability between normal and anomalous representations.

3.2.1. Channel Clustering

Real-world time series data often contains a mix of highly correlated, weakly correlated, and uncorrelated channels. A single shared embedding layer cannot distinguish these relations and may inadvertently impose artificial dependencies during the learning process. This can lead to degraded generalization performance, especially in anomaly detection tasks that require precise modeling of normal patterns. To address this, we propose a cluster-aware multi-embedding technique that utilizes spectral clustering (Von Luxburg, 2007) to cluster the channels into M clusters based on their correlations and assigns a dedicated embedding layer to each group.

Let $\boldsymbol{\Phi} \in [-1, 1]^{C\times C}$ denote the Pearson correlation matrix derived from the training data, and $\boldsymbol{\Phi}_{\text{abs}} \in [0, 1]^{C\times C}$ represent its element-wise absolute value. Channel i's correlations with other channels, given by the ith row $\boldsymbol{\phi}_i \in [0, 1]^C$ of $\boldsymbol{\Phi}_{\text{abs}}$, is used as its correlation profile. Some of the channels in the time series may remain constant throughout the series, resulting in undefined (NaN) correlation values with other channels. To address this, we replace all NaN entries in $\boldsymbol{\Phi}$ with zeros before proceeding. During the initial clustering phase, we group all channels with zero-valued correlation profiles in the Mth cluster. Let the number of remaining channels be $C^{'}$ and the corresponding absolute correlation matrix be $\boldsymbol{\Phi}_{\text{abs}}^{'} \in [0, 1]^{C^{'}\times C^{'}}$ with rows $\boldsymbol{\phi}_i^{'}$.

To group $C^{'}$ channels into $M^{'} = M - 1$ clusters based on their correlation profiles, we construct a similarity-based weighted adjacency matrix $\boldsymbol{W} \in [0, 1]^{C^{'}\times C^{'}}$, where each off-diagonal entry w_{ij} represents the cosine similarity between the correlation profiles of channels i and j and is given by,

$$w_{ij} = \begin{cases} \frac{(\phi_i^{'})^T \phi_j^{'}}{\|\phi_i^{'}\| \|\phi_j^{'}\|}, & \text{if } i \neq j \\ 0, & \text{if } i = j. \end{cases} \tag{6}$$

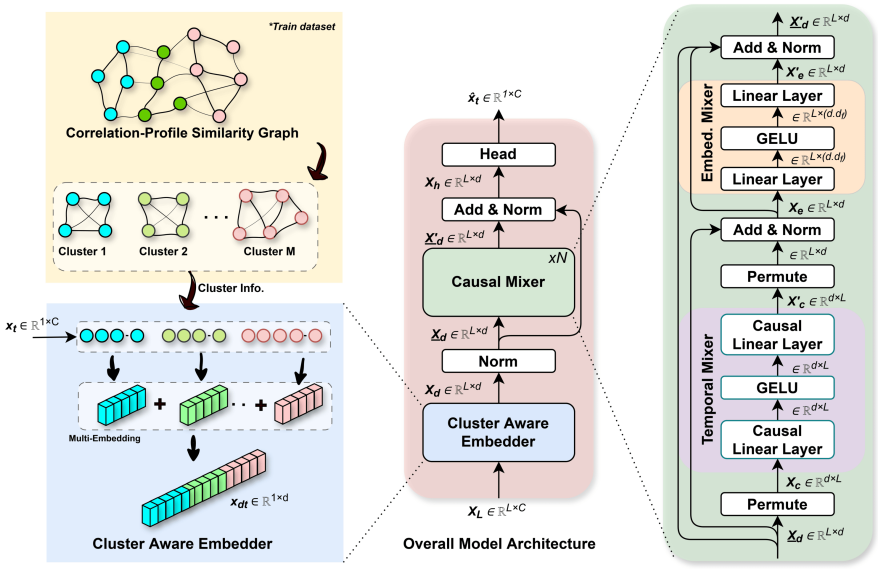


Figure 1. An architectural view of our reconstruction model. First, the multivariate input is embedded into a d-dimensional representation using a cluster-aware embedding module. Then, a causal mixer is used to extract features for reconstruction. In the correlation-profile similarity graph, each node represents a channel, and edge weights correspond to the cosine similarity between its correlation profile and those of its neighbors. Code is available at: https://github.com/Secure-and-Intelligent-Systems-Lab/CCM-TAD

Here, $\|.\|$ denotes the Euclidean norm. Next, we compute the normalized graph Laplacian matrix $\boldsymbol{L}_n$ as,

$$\boldsymbol{L}_n = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} \tag{7}$$

where, $\boldsymbol{I}$ is the identity matrix and the diagonal matrix $\boldsymbol{D}$ is the degree matrix with diagonal element $d_{ii} = \sum_j w_{ij}$. We then perform spectral embedding by extracting the first $M^{'}$ non-trivial eigenvectors of $\boldsymbol{L}_n$, corresponding to the smallest non-zero eigenvalues. Let $\boldsymbol{V} \in \mathbb{R}^{C^{'} \times M^{'}}$ contains these eigenvectors as columns. To normalize the embedded representations, we compute,

$$\boldsymbol{U} = \boldsymbol{D}^{-1/2}\boldsymbol{V} \tag{8}$$

where the ith row of $\boldsymbol{U}$, given by $\boldsymbol{u}_i = d_{ii}^{-1/2}\boldsymbol{v}_i \in \mathbb{R}^{M^{'}}$, represents the spectral embedding of channel i. Finally, K-Means clustering is applied to the rows of $\boldsymbol{U}$ to cluster the channel embeddings into $M^{'}$ clusters and get the cluster index for each channel as,

$$k^{'} = \text{K-MEANS}(\boldsymbol{U},\ M^{'}). \tag{9}$$

A pseudocode for the spectral clustering algorithm is given in Algorithm 1 in Appendix B. Also, a detailed discussion of the proposed clustering technique is given in Appendix J.

3.2.2. MULTI-EMBEDDING LAYER

After obtaining the cluster indices based on the training dataset, we construct M clusters of channels for both the training and the test datasets. We employ multiple embedders, one for each cluster, to learn channel representations in a cluster-specific manner. This approach allows the model to embed channels with different correlation profiles in separate latent subspaces, mitigating the risk of entangling unrelated features.

Let d denote the desired final embedding dimension. Since the number of channels assigned to each cluster is not uniform, we assign a different embedding dimension d_i for each cluster $i \in \{1, \ldots, M\}$, proportional to the number of channels C_i in that cluster, using the following formula,

$$d_i = \begin{cases} \left\lfloor \dfrac{C_i}{C} d \right\rfloor, & \text{for } i = 1, \ldots, M-1 \\ d - \sum_{j=1}^{M-1} d_j, & \text{for } i = M, \end{cases} \tag{10}$$

which ensures $\sum_{i=1}^{M} d_i = d$. Additionally, all d_i are lower-bounded by one due to either $d > C$ as in all experiments with benchmark datasets in Sec. 4 or a manual constraint. Each channel cluster is processed through its respective embedding layer of dimension d_i. These are subsequently concatenated to form the final unified d-dimensional embedding to feed into the next module of the model.

The use of multiple embedders not only enhances the model's capacity to learn localized representations within each cluster, it also reduces the number of trainable parameters in the embedding layer. In a conventional single linear embedding layer, the number of trainable parameters (weights) is $C \times d$. In contrast, our approach partitions the channels into M clusters, and each cluster is embedded independently using a reduced embedding dimension. Assuming uniform cluster sizes, each cluster contains $O\left(\frac{C}{M}\right)$ channels and is assigned an embedding dimension of $O\left(\frac{d}{M}\right)$. There-

fore, the number of trainable weights in each embedding layer is $O\left(\frac{C \cdot d}{M^2}\right)$. Since we employ M embedding layers, the total number of trainable parameters becomes $O\left(\frac{C \cdot d}{M}\right)$. Thus, the proposed method substantially reduces, by a factor of M, the number of trainable parameters in the embedding layer when $M > 1$.

3.3. Causal Mixer

While, in existing MLP-Mixer models for time series analysis, each data point interacts with all others within the look-back window (Murad et al., 2025; Chen et al., 2023; Wang et al., 2024), we only let a data point at time t interact with points from $\leq t$ to maintain causality, mimicking the real-world systems. We use two mixing modules that operate along the temporal dimension L and the embedding dimension d of the data space $\mathbb{R}^{L \times d}$.

Note that the causality enforced in our model is purely temporal. Our objective is not to infer inter-channel causal relationships like Granger causality. We let the embedding mixer to learn useful patterns for C channels.

3.3.1. Temporal Mixer Module

This module consists of two causal linear layers connected by GELU activation. The input to the temporal mixer module is denoted by $\boldsymbol{X}_c \in \mathbb{R}^{d \times L}$, where L is the look-back window length and d is the embedding dimension. To introduce causality in the linear layer, we perform masking on the weight $\boldsymbol{\Theta}_c \in \mathbb{R}^{L \times L}$ by an upper-triangular mask $\boldsymbol{\Gamma} \in \mathbb{R}^{L \times L}$ given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,L} \\ 0 & \gamma_{2,2} & \cdots & \gamma_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{L,L} \end{bmatrix}, \quad \gamma_{i,j} = \begin{cases} \frac{1}{j}, & \text{if } i \leq j \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

to get the final masked weight $\boldsymbol{\Theta}_u \in \mathbb{R}^{L \times L}$:

$$\boldsymbol{\Theta}_u = \boldsymbol{\Gamma} \odot \boldsymbol{\Theta}_c, \tag{12}$$

where $\odot$ refers to element-wise multiplication. Here, $1/j$ normalization accounts for the increasing number of aggregated past time steps at later positions, improving optimization stability during training, described in Section J.4 in the Appendix. The operation in the first *causal linear layer* can be written as,

$$\boldsymbol{x}_{c1}^{(k)} = \boldsymbol{x}_c^{(k)} \boldsymbol{\Theta}_u + \boldsymbol{b}; \quad k = 1, \ldots, d \tag{13}$$

where $\boldsymbol{x}_{c1}^{(k)} \in \mathbb{R}^{1 \times L}$ is the k^{th} row vector of the output $\boldsymbol{X}_{c1}$, $\boldsymbol{x}_c^{(k)} \in \mathbb{R}^{1 \times L}$ is the k^{th} row vector of the input $\boldsymbol{X}_c$, and $\boldsymbol{b} \in \mathbb{R}^{1 \times L}$ is the learnable bias. To enforce temporal causality, the causal linear layer restricts each output neuron j to depend only on input neurons $i \leq j$, effectively preventing information flow from future neurons. We have another causal linear layer in the temporal mixer module, with operations similar to the first one. Figure 15 in Appendix provides a schematic illustration of the temporal mixer.

3.3.2. Embedding Mixer

The embedding mixer employs two linear layers with a GELU activation in between. It projects the input from d to $(d \cdot d_f)$ dimensions and then back to d, following the design in (Murad et al., 2025).

3.4. Anomaly Detection Method

Anomalies in time series data often exhibit sequential dependencies, where the likelihood of a point being anomalous increases if preceding points are anomalous. Leveraging this temporal association, we propose an anomaly detection method that considers not only the anomaly evidence of the current data point, but also that of preceding points. Our approach accumulates anomaly evidence over time, and when the accumulated evidence exceeds a predefined threshold, the corresponding instance is identified as anomalous. Furthermore, by analyzing the rise and fall of the anomaly evidence, our method effectively delineates the full extent of the anomaly sequence.

Let the original training and test datasets be denoted by $\mathcal{X}_N$ and $\mathcal{X}_T$, with sizes N and T, respectively. We partition the training dataset $\mathcal{X}_N$ into two subsets: a training subset $\mathcal{X}_{N_1}$ and a validation subset $\mathcal{X}_{N_2}$, such that $N_1 + N_2 = N$. Our reconstruction model is trained on $\mathcal{X}_{N_1}$ to minimize the reconstruction loss. After training, we compute the validation reconstruction loss for each sample in $\mathcal{X}_{N_2}$. The reconstruction losses are then sorted in ascending order to form a sorted reconstruction loss series denoted by $\mathcal{E} \in \mathbb{R}^{N_2}$. Similarly, the reconstruction loss series for the test dataset is denoted by $\mathcal{G} = \{g_t\}_{t=1}^{T}$.

To evaluate the anomaly likelihood of a test point g_t, we calculate its empirical p-value (i.e., percentage of greater values) in $\mathcal{E}$ as

$$p_t = \frac{\#\{e \in \mathcal{E} : e \geq g_t\}}{\#\mathcal{E}},$$

where $\#$ denotes the number of elements in a set. The anomaly evidence β_t is then computed using a statistical significance parameter $\alpha \in [0, 1)$ as:

$$\beta_t = \log\left(\frac{\alpha}{p_t + \epsilon}\right). \tag{14}$$

where ϵ is a small value that avoids division by zero. We then accumulate the anomaly evidence β_t over time using the following formula to get the accumulated evidence series,

$$s_t = \begin{cases} \max(s_{t-1} + \beta_t, 0), & \text{if } \neg\left(\bigwedge_{i=1}^{\delta}(\beta_{t-i} < 0)\right) \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Here, s_t is initialized with $s_0 = 0$. The symbols $\bigwedge$ and $\neg$ are the logical AND and NOT, respectively. δ is a predefined window length that controls the reset condition: bring s_t to zero after observing δ consecutive negative anomaly evidences, potentially indicating the end of anomalous event. An instance is flagged as anomalous if s_t exceeds a predefined threshold h. Let t_a and t_b denote the temporal index of the starting and endpoint of an initially detected anomaly segment. We further update t_a and t_b employing backward search,

$$t_a^* = \arg\max\{t \leq t_a \mid s_t = 0\} \tag{16}$$
$$t_b^* = \arg\max\{t_a \leq t \leq t_b \mid \beta_t > 0\} \tag{17}$$

Here, t_a^* and t_b^* refer to the updated temporal index for the corresponding t_a and t_b, respectively. The updated onset time t_a^* denotes the point at which the accumulated evidence begins to rise from zero before the alarm triggered by $s_t > h$, while the updated offset time t_b^* represents the point at which the accumulated evidence starts to decline near the end of the anomaly segment. A statistical interpretation of our anomaly detection method is provided in Appendix A, while the overall scoring process is illustrated in Figure 5 in the Appendix.

Table 1. The statistics of the six publicly available datasets.

Data.	Entities	Channels	Train	Test	Anom. Rate
SWaT	1	51	495000	449919	12.14%
WADI	1	123	1209601	172801	5.75%
PSM	1	25	132481	87841	27.76%
SMD	28	38	708405	708420	4.16%
MSL	27	55	58317	73729	10.48%
SMAP	55	25	140825	444035	12.83%

4. Experiments

Datasets: We evaluate the effectiveness of our model using six publicly available datasets. These datasets are: SWaT (Secure Water Treatment) (Goh et al., 2017), WADI (Water Distribution) (Ahmed et al., 2017), PSM (Pooled Server Metrics) (Abdulaal et al., 2021), SMD (Server Machine Dataset) (Su et al., 2019), SMAP (Soil Moisture Active Passive) (Entekhabi et al., 2010), and MSL (Mars Science Laboratory) (Hundman et al., 2018). The statistics of the datasets are shown in Table 1. The details of the datasets are discussed in Section C in the Appendix.

Baselines: We compare our model against 23 baselines, including autoencoder-based, recurrent-based, generative, graph-based, transformer-based, MLP-mixer-based, and classical outlier detection approaches, reporting the best F1 scores in Table 2.

For multi-entity datasets, prior works adopt different evaluation protocols: (1) training a single model across all entities, (2) training separate models per entity and averaging entity-wise F1 scores, or (3) training per-entity models and aggregating predictions before computing F1. To ensure a fair and comprehensive comparison, we evaluate our model across all three protocols and report the results for each.

We additionally report SensitiveHUE under two settings: Offline and Online. The offline variant follows the original implementation, which uses robust normalization based on statistics computed over the entire test set, making it unsuitable for online inference. For fair comparison with our method, which performs online inference, we remove this post-hoc normalization to obtain the online SensitiveHUE variant. Detailed descriptions of all baselines and result sources are provided in Appendix E.

Evaluation Metrics We use the best-F1 score as the primary evaluation metric, following prior works (Lai et al., 2023; Feng et al., 2024). The best-F1 score is obtained through a threshold-sweeping approach commonly adopted in the literature. Although some studies employ the point-adjusted F1 score (Song et al., 2023; Huang et al., 2025), we do not use it, as it disproportionately rewards detection performance for long-duration anomalies (Doshi et al., 2022; Lai et al., 2023; Kim et al., 2022; Garg et al., 2021). We additionally report PR_AUC, which is particularly informative for imbalanced anomaly detection tasks (Saito & Rehmsmeier, 2015). Results on additional metrics, including VUS_PR and VUS_ROC (Paparrizos et al., 2022), are provided in the Appendix H.

4.1. Results

Table 2 presents the best-F1 scores across six publicly available datasets. Our model consistently outperforms existing methods on nearly all datasets. It improves the best-F1 scores by 8.71% and 8.65% for WADI and PSM, respectively. For the SWAT dataset, although SensitiveHUE (Offline) slightly outperforms our model, our method surpasses SensitiveHUE (Online), which is a more appropriate baseline given that our model detects anomalies online. We also outperform all baselines across multi-entity datasets under each evaluation protocol. For MSL, the improvements are $+23.3\%(0.300 \rightarrow 0.370)$, $+11.3\%(0.551 \rightarrow 0.613)$, and $+41.2\%(0.452 \rightarrow 0.638)$ across the three protocols, respectively. For SMAP, the improvements are $+32.3\%(0.294 \rightarrow 0.389)$, $+4.8\%(0.505 \rightarrow 0.529)$, and $+5.4\%(0.523 \rightarrow 0.551)$ across the three protocols, respectively. Furthermore, the improvements in the best-F1 score for the SMD dataset are also consistent. We observe that training a model for each entity yields better results than training a single model for all entities, as entity-entity variations are considerable.

In addition, Table 3 presents the PR_AUC results across the datasets. Our model consistently outperforms the baselines

Table 2. Performance comparison using the best-F1 score. For multi-entity datasets, marked with * (MSL, SMD, SMAP), methods annotated with a superscript ^{*i} are evaluated under protocol-i. The definitions of the evaluation protocols are provided in Section 4 (Baselines). Best (Bold) and second-best (Underlined) results are selected within each protocol for multi-entity datasets. Best-F1 score is defined in Appendix F.

	WADI	PSM	SWAT	MSL*	SMD*	SMAP*
Anom. Trans.^{*1}	0.108	0.434	0.220	0.191	0.080	0.227
TimesNet^{*1}	0.322	0.436	0.260	0.261	<u>0.264</u>	0.250
xLSTMAD^{*1}	0.568	0.651	0.823	0.265	0.245	0.279
CATCH^{*1}	-	0.116	0.087	0.143	0.237	0.061
D3R^{*1}	0.129	0.442	0.459	0.240	-	0.227
PatchAD^{*1}	0.528	0.493	0.777	0.249	-	0.247
SimAD^{*1}	0.640	0.521	0.820	<u>0.300</u>	-	<u>0.294</u>
TimesNet^{*2}	-	-	-	0.369	0.477	0.354
xLSTMAD^{*2}	-	-	-	0.441	0.521	0.451
DAGMM^{*2}	0.121	0.483	0.750	0.199	0.238	0.333
LSTM-VAE^{*2}	0.227	0.455	0.776	0.212	0.435	0.235
MSCRED^{*2}	0.046	0.556	0.757	0.250	0.382	0.170
OmniAnom.^{*2}	0.223	0.452	0.782	0.207	0.474	0.227
MAD-GAN^{*2}	0.370	0.471	0.770	0.267	0.220	0.175
MTAD-GAT^{*2}	0.437	0.571	0.784	0.275	0.400	0.296
USAD^{*2}	0.233	0.479	0.792	0.211	0.426	0.228
UAE^{*2}	0.354	0.427	0.453	0.451	0.435	0.390
GDN^{*2}	0.570	0.552	0.810	0.217	0.529	0.252
TranAD^{*2}	0.415	0.649	0.669	0.251	0.310	0.247
NPSR^{*2}	0.642	0.648	0.839	<u>0.551</u>	0.535	<u>0.505</u>
SAT^{*2}	0.635	0.653	0.842	0.503	0.506	0.452
OracleAD^{*2}	-	<u>0.659</u>	0.765	-	0.430	-
PCA-Error^{*2}	-	-	0.833	-	<u>0.572</u>	-
Sen.H.(Off)^{*3}	<u>0.700</u>	0.517	**0.911**	<u>0.452</u>	<u>0.398</u>	0.415
Sen.H.(On)^{*3}	0.678	0.507	0.874	0.418	0.346	<u>0.523</u>
CAROTS	0.143	0.603	0.791	-	-	-
CAROTS+	0.391	0.534	0.789	-	-	-
CCM-TAD^{*1}	**0.761**	**0.716**	<u>0.883</u>	**0.370**	**0.300**	**0.389**
^{*2}	**-**	**-**	**-**	**0.613**	**0.612**	**0.529**
^{*3}	**-**	**-**	**-**	**0.638**	**0.602**	**0.551**

on almost all datasets by a considerable margin. Although CCM-TAD does not achieve the best PR_AUC on the PSM dataset, its performance remains competitive.

4.2. Ablation Study

In the ablation studies, hyperparameters are tuned for each experiment separately. Further details of hyperparameters are provided in Appendix G. Additional ablation studies are presented in Appendix J.

4.2.1. Anomaly Detection Method

To evaluate the effectiveness of our proposed sequential anomaly detection method, we compare its performance against the point-based anomaly detection version of our method. In point-based anomaly detection, we employ the reconstruction loss (MSE) as the anomaly score and compare it to a threshold h to detect anomalies. As shown in Table 4, our method without boundary correction (Equations 16 and 17) already outperforms the point-based detection across most datasets. The performance is further improved by boundary correction. One reason for this improvement is that, unlike the point-based approach, which may classify an anomalous instance as normal even when adjacent anomalies are correctly detected, our method accumulates anomaly evidence over time. As a result, the likelihood of missing anomalous instances within a contiguous sequence is substantially reduced, as illustrated in Figure 2.

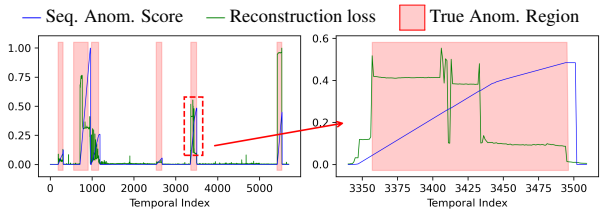


Figure 2. Visualization of point-based anomaly score (Recons. loss) and our proposed sequential anomaly score. The proposed sequential anomaly score produces smooth, temporally coherent responses across anomaly regions, thereby improving detection performance.

4.2.2. Modular Architecture

In Table 5, we present 10 cases, showing different configurations of our model, to demonstrate the impact of different modules on the model's performance. It is observed that the channel clustering and the causal temporal mixer modules are complementary to each other. Case-10 corresponds to the full configuration of our model. By comparing Case-5 and Case-10, it is observed that clustering improves the best F1 score on both SWAT and PSM datasets significantly. We also evaluate the performance of our model with a regular temporal mixer (non-causal), our proposed causal temporal mixer, and without a temporal mixer. It is observed that the performance with the non-causal temporal mixer (Case-8) and without any temporal mixer (Case-6) have similar performances. We can conclude that the non-causal temporal mixer does not effectively capture the temporal information for anomaly detection. However, the proposed causal temporal mixer (Case-10) significantly improves F1 score, compared to Case-6 and Case-8.

4.2.3. Effect of Clustering on Representation Separability

We quantify how clustering affects the geometry of the learned representations using the nearest-neighbor separation ratio NN-sep $= \frac{\mathbb{E}[d(a \to n)]}{\mathbb{E}[d(n \to n)]}$, where $d(a \to n)$ is the distance from each anomalous sample to its nearest normal neighbor and $d(n \to n)$ is the nearest-neighbor distance among normals. Here *nearest* refers to Euclidean proximity in the representation space $\mathbb{R}^C$. Higher NN-sep values

Table 3. Performance comparison using PR AUC. Multi-entity datasets annotated with superscript^{*i} correspond to results under evaluation protocol-i. Best and second-best results are shown in bold and underline, respectively, within each protocol for multi-entity datasets. Some methods provide results only on a subset of datasets; therefore, some entries are unavailable.

	WADI	PSM	SWAT	MSL^{*1}	SMD^{*1}	SMAP^{*1}	MSL^{*2}	SMD^{*2}	SMAP^{*2}
TimesNet	0.250	0.368	0.155	<u>0.183</u>	<u>0.183</u>	0.113	0.247	0.431	0.208
xLSTMAD	0.550	**0.598**	0.805	0.178	0.136	<u>0.132</u>	<u>0.312</u>	<u>0.457</u>	<u>0.353</u>
CATCH	-	0.434	0.166	0.167	0.172	0.131	-	-	-
CAROTS	0.056	<u>0.595</u>	0.764	-	-	-	-	-	-
CAROTS+	0.260	0.535	0.760	-	-	-	-	-	-
Sen.H.(On)	<u>0.612</u>	0.460	<u>0.852</u>	-	-	-	0.283	0.407	0.346
CCM-TAD	**0.720**	0.558	**0.911**	**0.283**	**0.311**	**0.336**	**0.610**	**0.587**	**0.552**

Table 4. Contribution from the proposed sequential anomaly detection method (Sec. 3.4). "Seq. w/o BC" denotes our sequential anomaly scoring method without boundary correction (Eqs. 16 and 17), while "Seq. w/ BC" corresponds to the method with boundary correction.

	SWAT		PSM		WADI		MSL		SMD	
	F1	PR_AUC	F1	PR_AUC	F1	PR_AUC	F1	PR_AUC	F1	PR_AUC
Point-based	<u>0.882</u>	0.869	0.657	<u>0.536</u>	<u>0.739</u>	0.660	0.557	0.339	0.562	0.509
Seq. w/o BC	0.879	<u>0.888</u>	<u>0.659</u>	0.525	<u>0.739</u>	<u>0.678</u>	<u>0.626</u>	<u>0.498</u>	<u>0.595</u>	<u>0.530</u>
Seq. w/ BC	**0.883**	**0.911**	**0.716**	**0.558**	**0.761**	**0.720**	**0.638**	**0.610**	**0.602**	**0.587**

Table 5. Ablation study evaluating the impact of channel clustering, temporal mixer, and embedding mixer modules. The symbols × and ✓ denote the presence and absence of each module. The symbols ♢ and ♠ indicate non-causal and causal temporal mixers, respectively. Reported values are the best F1 scores obtained. Case 10 represents the complete model configuration.

Case	Clustering	Temporal Mx	Embed. Mx	SWAT	PSM
1	×	×	✓	0.866	0.670
2	×	♢	×	0.873	0.687
3	×	♢	✓	0.871	0.673
4	×	♠	×	0.861	0.653
5	×	♠	✓	0.865	0.703
6	✓	×	✓	0.874	0.657
7	✓	♢	×	0.872	0.661
8	✓	♢	✓	0.871	0.652
9	✓	♠	×	0.875	0.681
10	✓	♠	✓	**0.883**	**0.716**

Table 6. Nearest-neighbor separation (NN-sep) across datasets. Imp(%) denotes the relative improvement of the proposed clustering approach compared to the model without clustering.

Dataset	w/o Clustering	w/ Clustering	Imp.(%)
SWAT	65.63	**91.30**	39.11
WADI	12.01	**12.47**	3.83
PSM	66.74	**66.78**	0.06

indicate a larger normal–anomaly margin. As shown in Table 6, clustering substantially enlarges this margin (91.30) on SWAT, compared to the embedding without clustering (65.63), indicating a +39% improvement over without clustering. For WADI, this improvement is 3.83%, while it is 0.06% for PSM. These results demonstrate that cluster-aware embedding enhances the discrimination between normal and anomalous representations.

Table 7. Best F1 scores for various clustering variants. The number of clusters M is optimized for each variant. Variants (i), (ii), (iii) are explained in Section 4.2.4 and Appendix J.1.

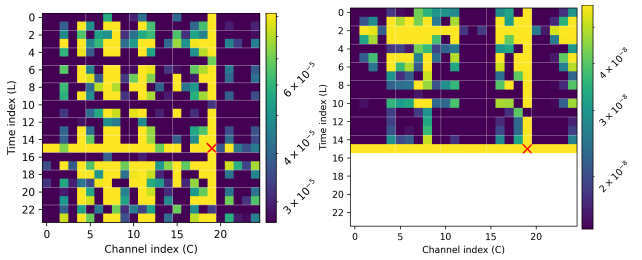
Variant	Proposed	(i)	(ii)	(iii)
Dataset	$M(F1)$	$M(F1)$	$M(F1)$	$M(F1)$
PSM	4 (**0.716**)	3 (0.688)	5 (0.674)	6 (0.673)
SWAT	6 (**0.883**)	4 (0.873)	4 (0.879)	6 (0.881)

4.2.4. Variants of Channel Clustering

To assess the robustness of our spectral clustering approach, we compare it with three alternative methods: (i) spectral clustering based on cosine similarity between channels, (ii) spectral clustering using the absolute correlation matrix, and (iii) K-Means clustering applied directly to the correlation profiles. Table 7 shows the performance comparison of these variants with optimized number of clusters M. Detailed descriptions for these variants are provided in Appendix J.1. Our approach consistently outperforms these baselines, demonstrating the effectiveness of spectral clustering based on the full correlation profile.

4.2.5. Verifying Temporal Causality in the Model via Gradient Analysis

To confirm that the proposed model maintains temporal causality, we conduct an experiment using two variants of the model: one employing a causal linear layer in the temporal mixer and the other using a standard (non-causal) linear layer. We analyze the input-gradient dependencies of the reconstruction loss. For the input $\mathbf{X}_L \in \mathbb{R}^{L\times C}$, we



(a) Non-causal *(b)* Causal

Figure 3. Gradient-based temporal dependency analysis for causal and non-causal temporal mixers. The heatmaps show the magnitude of the input gradients of the reconstruction loss evaluated at time index $t = 15$ and channel $c = 19$ with respect to all input time–channel pairs of $\mathbf{X}_L$. The red "$\times$" denotes the loss evaluation point.

evaluate the reconstruction loss at time index $t = 15$ and channel $c = 19$, and compute the absolute input gradients $\left|\partial \ell / \partial x_\tau^j\right|$ over all time–channel pairs (τ, j).

Figure 3 compares the resulting gradient maps for the models with causal and non-causal temporal mixers. The model with the non-causal temporal mixer exhibits non-zero gradients for future time steps ($\tau > 15$), indicating that the reconstruction at time t depends on future inputs. In contrast, the model with the causal temporal mixer completely suppresses gradients for all $\tau > 15$, confirming that the loss at time t is influenced only by present and past observations.

5. Conclusion

In this work, we present CCM-TAD, a unified time-series anomaly-detection framework. Our approach combines multiple complementary novelties that jointly address fundamental limitations of existing methods. First, we introduce temporal causality into the MLP-Mixer architecture, using a principled causal masking design. Extensive ablations and gradient-based analyses confirm that causal temporal mixing significantly helps anomaly detection and completely eliminates future information leakage. We also propose a correlation-profile-based channel clustering strategy and incorporate it into a cluster-aware multi-embedding module. This effectively mitigates spurious inter-channel correlations in the normal data, improves the separability of normal–anomalous representations, and consistently outperforms alternative clustering approaches. We further propose a sequential anomaly scoring method with boundary refinement that accumulates statistically grounded anomaly evidence over time. The proposed scoring mechanism captures temporal continuity, refines anomaly boundaries, and enables online anomaly detection. Together, these contributions form a coherent and effective anomaly detection framework that is causal, robust, and suitable for real-time deployment. Extensive experiments on six public benchmarks demonstrate consistent improvements over the baselines across both single-entity and multi-entity datasets.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning, specifically online anomaly detection in time series. There are many potential societal consequences of our work; we do not feel that any must be specifically highlighted beyond encouraging responsible deployment with appropriate oversight in high-stakes monitoring settings.

References

Abdulaal, A., Liu, Z., and Lancewicki, T. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2485–2494, 2021.

Ahmed, C. M., Palleti, V. R., and Mathur, A. P. 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*, pp. 25–28, 2017.

Aminikhanghahi, S. and Cook, D. J. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

Blázquez-García, A., Conde, A., Mori, U., and Lozano, J. A. A review on outlier/anomaly detection in time series data. *ACM computing surveys (CSUR)*, 54(3):1–33, 2021.

Casella, G. and Berger, R. *Statistical inference*. Chapman and Hall/CRC, 2024.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3): 1–58, 2009.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

Chen, Z., Chen, D., Zhang, X., Yuan, Z., and Cheng, X. Learning graph structures with transformer for multivariate time-series anomaly detection in iot. *IEEE Internet of Things Journal*, 9(12):9179–9189, 2021.

Cho, D., Han, J., Kang, K., Kim, M., Ryu, H., and Jung, N. Structured temporal causality for interpretable multivariate time series anomaly detection. *arXiv preprint arXiv:2510.16511*, 2025.

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

Doshi, K., Abudalou, S., and Yilmaz, Y. Reward once, penalize once: Rectifying time series anomaly detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.

Ekambaram, V., Jati, A., Dayama, P., Mukherjee, S., Nguyen, N., Gifford, W. M., Reddy, C., and Kalagnanam, J. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181, 2024.

Entekhabi, D., Njoku, E. G., O'neill, P. E., Kellogg, K. H., Crow, W. T., Edelstein, W. N., Entin, J. K., Goodman, S. D., Jackson, T. J., Johnson, J., et al. The soil moisture active passive (smap) mission. *Proceedings of the IEEE*, 98(5):704–716, 2010.

Faber, K., Pietron, M., Zurek, D., and Corizzo, R. xlstmad: A powerful xlstm-based method for anomaly detection. In *2025 IEEE International Conference on Data Mining (ICDM)*, pp. 247–256. IEEE, 2025.

Feng, Y., Zhang, W., Fu, Y., Jiang, W., Zhu, J., and Ren, W. 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*, pp. 782–793, 2024.

Garg, A., Zhang, W., Samaran, J., Savitha, R., and Foo, C.-S. An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2508–2517, 2021.

Goh, J., Adepu, S., Junejo, K. N., and Mathur, A. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*, pp. 88–99. Springer, 2017.

Han, X., Absar, S., Zhang, L., and Yuan, S. Root cause analysis of anomalies in multivariate time series through granger causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hong, Y.-C., Xiao, B., and Chen, Y. Tskanmixer: Kolmogorov-arnold networks with mlp-mixer model for time series forecasting. *arXiv preprint arXiv:2502.18410*, 2025.

Huang, X., Chen, W., Hu, B., and Mao, Z. Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17476–17484, 2025.

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

Kim, H., Mok, J., Lee, D., Lew, J., Kim, S., and Yoon, S. Causality-aware contrastive learning for robust multivariate time-series anomaly detection. *arXiv preprint arXiv:2506.03964*, 2025.

Kim, S., Choi, K., Choi, H.-S., Lee, B., and Yoon, S. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7194–7201, 2022.

Lai, C.-Y. A., Sun, F.-K., Gao, Z., Lang, J. H., and Boning, D. Nominality score conditioned time series anomaly detection by point/sequential reconstruction. *Advances in Neural Information Processing Systems*, 36:76637–76655, 2023.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., and Ng, S.-K. Madgan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks*, pp. 703–716. Springer, 2019.

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Ma, J., Sun, L., Wang, H., Zhang, Y., and Aickelin, U. Supervised anomaly detection in uncertain pseudoperiodic data streams. *ACM Transactions on Internet Technology (TOIT)*, 16(1):1–20, 2016.

Müller, M., Ernis, G., and Mock, M. Anomaly detection in multivariate time series using uncertainty estimation. In *Informed Machine Learning*, pp. 313–339. Springer, 2025.

Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access*, 7:1991–2005, 2018.

Murad, M. M. N., Aktukmak, M., and Yilmaz, Y. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19581–19588, 2025.

Paparrizos, J., Boniol, P., Palpanas, T., Tsay, R. S., Elmore, A., and Franklin, M. J. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11): 2774–2787, 2022.

Park, D., Hoshi, Y., and Kemp, C. C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Saito, T. and Rehmsmeier, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10 (3):e0118432, 2015.

Sarfraz, M. S., Chen, M.-Y., Layer, L., Peng, K., and Koulakis, M. Position: quo vadis, unsupervised time series anomaly detection? *arXiv preprint arXiv:2405.02678*, 2024.

Shahapure, K. R. and Nicholas, C. Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*, pp. 747–748. IEEE, 2020.

Song, J., Kim, K., Oh, J., and Cho, S. Memto: Memory-guided transformer for multivariate time series anomaly detection. *Advances in Neural Information Processing Systems*, 36:57947–57963, 2023.

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

Ten, C.-W., Hong, J., and Liu, C.-C. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, 2(4):865–873, 2011.

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

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

Wang, C., Zhuang, Z., Qi, Q., Wang, J., Wang, X., Sun, H., and Liao, J. Drift doesn't matter: dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection. *Advances in Neural Information Processing Systems*, 36:10758–10774, 2023.

Wang, Q., Zhu, Y., Sun, Z., Li, D., and Ma, Y. A multi-scale patch mixer network for time series anomaly detection. *Engineering Applications of Artificial Intelligence*, 140: 109687, 2025a.

Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J. Y., and Zhou, J. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

Wang, Y., Cheng, H., Xiong, J., Wen, Q., Jia, H., Song, R., Zhang, L., Zhu, Z., and Liu, Y. Noise-resilient point-wise anomaly detection in time series using weak segment labels. *arXiv preprint arXiv:2501.11959*, 2025b.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Wu, X., Qiu, X., Li, Z., Wang, Y., Hu, J., Guo, C., Xiong, H., and Yang, B. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

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

Yue, W., Ying, X., Guo, R., Chen, D., Shi, J., Xing, B., Zhu, Y., and Chen, T. Sub-adjacent transformer: Improving time series anomaly detection with reconstruction error from sub-adjacent neighborhoods. *arXiv preprint arXiv:2404.18948*, 2024.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1409–1416, 2019.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, pp. 841–850. IEEE, 2020.

Zhong, Z., Yu, Z., Xi, X., Xu, Y., Cao, W., Yang, Y., Yang, K., and You, J. Simad: A simple dissimilarity-based approach for time-series anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2025a.

Zhong, Z., Yu, Z., Yang, Y., Wang, W., Yang, K., and Chen, C. P. Patchad: A lightweight patch-based mlp-mixer for time series anomaly detection. *IEEE Transactions on Big Data*, 2025b.

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

A. Statistical Interpretation of our Anomaly Detection Method

Here, we illustrate how our method operates in Fig. 5 and analyze how the anomaly score s_t (Eq. (15)) is expected to behave statistically under normal and anomalous data. Since s_t is a running sum of anomaly evidence $\beta_t = \log(\alpha/(p_t + \epsilon))$ over time, we analyze the probability distribution and expectation of β_t under normal and anomalous data, i.e., $f_0(\beta)$ and $f_1(\beta)$, $\mathbb{E}_0[\beta]$ and $\mathbb{E}_1[\beta]$, respectively.

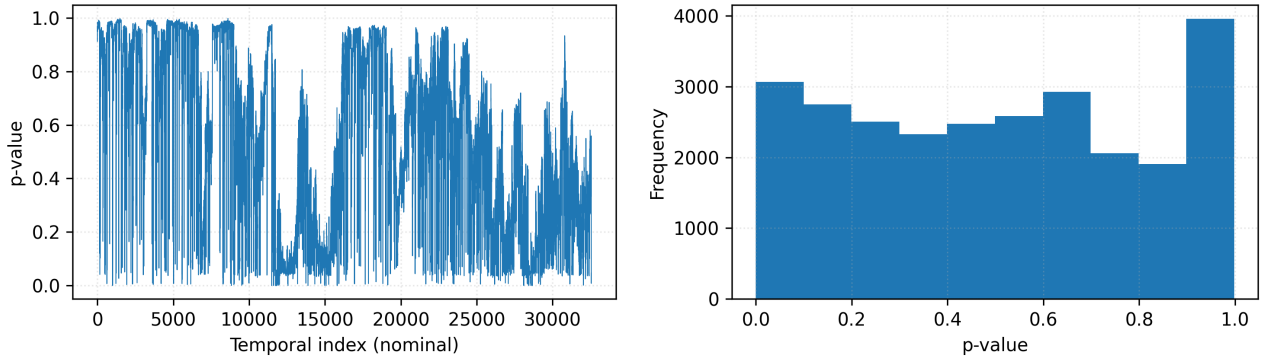


Figure 4. Temporal evolution of p-values on the normal WADI test data (left) and the corresponding empirical distribution shown as a histogram (right).

The empirical p-value p_t converges to the actual p-value as the validation set size N_2 grows. The p-value of independent and identically distributed (iid) samples are known to be uniformly distributed in $[0, 1]$ regardless of the continuous probability distribution of samples (Casella & Berger, 2024). Although the objective of reconstruction model is to capture all learnable pattern and leave only iid noise as reconstruction loss, in general, for time series data, reconstruction loss, and in turn its p-value, also constitute time series with temporal dependency, as shown in Fig. 4. Despite the temporal correlation, the p-value of reconstruction loss approximately takes uniformly random values in $[0, 1]$, as illustrated in the histogram in Fig. 4, with possibly correlated durations around some value (e.g., initially values mostly close to 1 in Fig. 4). Note that the quality of this approximation depends on the dataset, as well as the performance of reconstruction model.

Approximating the distribution of empirical p-value p_t with $\mathcal{U}(0, 1)$, we write the cumulative distribution function (cdf) of β_t as

$$\begin{aligned} F_0(\beta) = \mathbb{P}(\beta_t \leq \beta) &= \mathbb{P}\left(\log \frac{\alpha}{p_t + \epsilon} \leq \beta\right) \\ &= \mathbb{P}\left(p_t \geq \frac{\alpha}{e^{\beta}} - \epsilon\right) \\ &\approx \begin{cases} 1 - \frac{\alpha}{e^{\beta}} + \epsilon, & \text{if } \beta \in [\log \frac{\alpha}{1+\epsilon}, \log \frac{\alpha}{\epsilon}] \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Taking the derivative of cdf we obtain the probability density function (pdf) of β_t as follows:

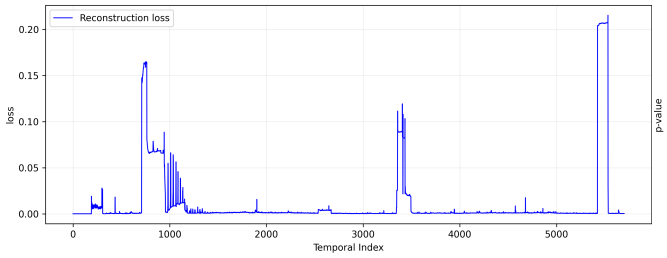
$$f_0(\beta) = \frac{\partial F_0(\beta)}{\partial \beta} \approx \begin{cases} \alpha e^{-\beta}, & \text{if } \beta \in [\log \frac{\alpha}{1+\epsilon}, \log \frac{\alpha}{\epsilon}] \\ 0, & \text{otherwise.} \end{cases}$$

Thus, we can approximate the expectation under normal data as

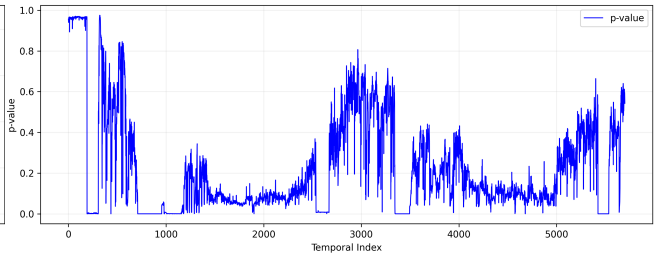
$$\mathbb{E}_0[\beta] = \int_{\log \frac{\alpha}{1+\epsilon}}^{\log \frac{\alpha}{\epsilon}} \beta \alpha e^{-\beta} \mathrm{d}\beta = \left(\log \frac{\alpha}{1+\epsilon} + 1\right)(1+\epsilon) - \underbrace{\left(\underbrace{\log \frac{\alpha}{\epsilon}}_{>0} + 1\right)}_{<0} \epsilon,$$

which is < 0 for $\frac{\alpha}{1+\epsilon} < e^{-1}$, i.e., $\alpha < \frac{1+\epsilon}{e}$. Hence, when data is normal and from the same distribution as training and validation, the anomaly evidence β_t is expected to be negative for $\alpha < e^{-1} \approx 0.36788$ and in turn the running anomaly

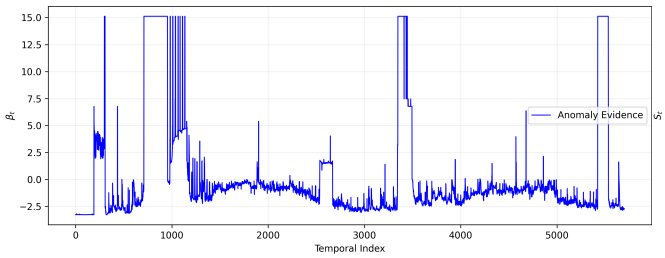
score s_t is expected to hover around zero. On the other hand, when data becomes anomalous, reconstruction loss is expected to grow such that p_t becomes smaller than α, making β_t positive and s_t grow.



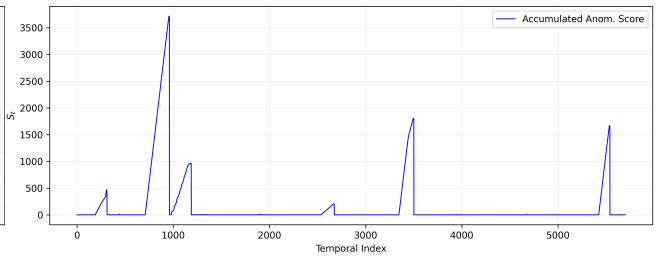
(a) Reconstruction loss



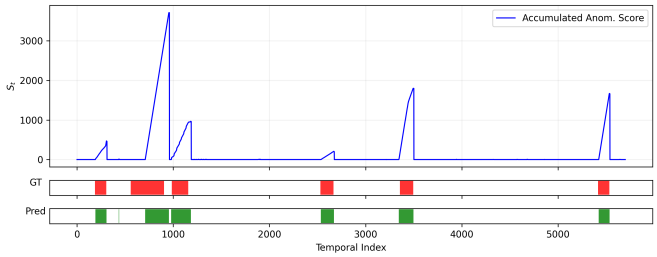
(b) p-value



(c) Anomaly evidence β_t



(d) Accumulated Anom. evidence s_t



(e) Final Prediction (GT: Ground Truth, Pred: Prediction)

Figure 5. Overview of the proposed anomaly detection scoring process. The reconstruction loss is converted to a p-value, which is then used to generate anomaly evidence β_t and accumulated over time to produce the final anomaly score s_t and predictions. Anomaly detection method is detailed in Section 3.4 in the main paper.

B. Spectral Channel Clustering Procedure

A pseudocode for the spectral channel clustering algorithm is given in Algorithm 1.

C. Datasets

We evaluate the effectiveness of our model using six publicly available datasets: SWaT, WADI, PSM, SMD, SMAP, and MSL.

- The SWaT (Secure Water Treatment) (Goh et al., 2017) dataset contains time-series data from 51 sensors and actuators of a water treatment system. It spans 11 days, with the first 7 days representing normal operation and the remaining 4 days including attack scenarios.
- WADI (Water Distribution) (Ahmed et al., 2017) dataset includes data from 123 sensors and actuators over 16 days. The first 14 days correspond to normal operation, and the last 2 days involve attack scenarios.
- PSM (Pooled Server Metrics) (Abdulaal et al., 2021) data is collected from multiple application servers at eBay. This dataset comprises 25 dimensions, each representing a different server.

- SMD (Server Machine Dataset) (Su et al., 2019) is a multi-entity dataset containing telemetry data from 28 server machines. Each machine provides data from 38 sensors, and the duration of the dataset is five weeks.

- SMAP (Soil Moisture Active Passive) and MSL (Mars Science Laboratory) (Entekhabi et al., 2010; Hundman et al., 2018) are multi-entity datasets provided by NASA. These consist of real spacecraft telemetry data, including recorded anomalies. SMAP contains 55 entities with 25 dimensions, while MSL includes 27 entities with 55 dimensions.

The details of the datasets are shown in Table 1 of the main paper.

D. Data processing

Following the approach of SensitiveHUE (Feng et al., 2024), we downsampled the SWAT and WADI datasets by a factor of 5 to ensure fair comparisons. Data normalization was performed using min-max scaling, with parameters computed from the training set. To prevent extreme values from affecting the model, we clipped the normalized test data to the range $[-4, +4]$, as recommended by NPSR (Lai et al., 2023). We adopted the preprocessing steps below following (Lai et al., 2023; Feng et al., 2024):

- For the SWAT dataset, the P201 and LIT401 channels exhibited inconsistencies between the training and test sets; hence, we set their values to zero (Feng et al., 2024). To reduce noise, we applied a moving average filter to the entire dataset.

- For the WADI dataset (2017 version), we ignored columns with excessive missing values and forward-filled a small number of remaining NaNs. One channel was also set to zero based on the same reference. Additionally, we excluded the first 21,600 data points from the training set to account for system initialization (Lai et al., 2023).

Algorithm 1 PROPOSED SPECTRAL CLUSTERING BASED ON CORRELATION PROFILE

function SPECTRALCLUSTERING($\mathcal{X}_N \in \mathbb{R}^{N\times C}, M < C \in \mathbb{N}$)
  Input: Training dataset $\mathcal{X}_N$, Number of clusters M
  Output: Cluster index for C channels $k \in \{1, 2, \ldots, M\}^C$
  $\boldsymbol{\Phi} \leftarrow$ CORRCOEFFMATRIX($\mathcal{X}_N$) ▷ $\in [-1, 1]^{C\times C}$
  $\boldsymbol{\Phi} \leftarrow$ FILLNAN($\boldsymbol{\Phi}, 0$)
  $\boldsymbol{\Phi}_{\text{abs}} \leftarrow |\boldsymbol{\Phi}|$ ▷ Element-wise absolute value
  $f_d \leftarrow \{i \mid \boldsymbol{\Phi}_{\text{abs}}[i, :] = \mathbf{0}\}$ ▷ Channel index with zero correlation profile vector
  $f_c \leftarrow \{1, 2, \ldots, C\} \setminus f_d$ ▷ Remaining channel index f_c, and $C^{'} = |f_c|$)
  $\boldsymbol{\Phi}^{'}_{\text{abs}} \leftarrow \boldsymbol{\Phi}_{\text{abs}}[f_c, f_c]$ ▷ $\in [0, 1]^{C^{'}\times C^{'}}$
  $\boldsymbol{W} \leftarrow$ COSINESIMILARITY($\boldsymbol{\Phi}^{'}_{\text{abs}}$) ▷ Using Eq. (6). $\boldsymbol{W} \in [0, 1]^{C^{'}\times C^{'}}$
  $\boldsymbol{D}_{ii} \leftarrow \sum_j \boldsymbol{W}_{ij}$
  $\boldsymbol{L} \leftarrow \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$
  if $f_d = \emptyset$ **then**
    $\boldsymbol{V} \leftarrow 2^{nd}$ to $(M{+}1)^{th}$ smallest eigenvectors of $\boldsymbol{L}$ ▷ $\in \mathbb{R}^{C^{'}\times M}$
    $\boldsymbol{U} \leftarrow \boldsymbol{D}^{-1/2}\boldsymbol{V}$ ▷ $\in \mathbb{R}^{C^{'}\times M}$
    $k^{'} \leftarrow$ K-MEANS($\boldsymbol{U}, M$)
  else
    $\boldsymbol{V} \leftarrow 2^{nd}$ to M^{th} smallest eigenvectors of $\boldsymbol{L}$ ▷ $\in \mathbb{R}^{C^{'}\times (M-1)}$
    $\boldsymbol{U} \leftarrow \boldsymbol{D}^{-1/2}\boldsymbol{V}$ ▷ $\in \mathbb{R}^{C^{'}\times (M-1)}$
    $k^{'} \leftarrow$ K-MEANS($\boldsymbol{U}, M{-}1$)
  end if
  $k \leftarrow$ zeros(C)
  $k[f_c] \leftarrow k^{'}, \quad k[f_d] \leftarrow M$
  return k
end function

- For the PSM dataset, we applied a moving average for noise reduction, and forward-filled all missing values (Lai et al., 2023).

E. Baselines and Result Sources

We employ 23 baselines, including autoencoder-based, recurrent-based, generative, graph-based, transformer-based, MLP-mixer-based, and classical outlier detection approaches.

Specifically, DAGMM (Zong et al., 2018), UAE (Garg et al., 2021), and USAD (Audibert et al., 2020) are autoencoder-based methods, while LSTM-VAE (Park et al., 2018) adopts a variational autoencoder framework. MSCRED (Zhang et al., 2019), OmniAnomaly (Su et al., 2019), and MAD-GAN (Li et al., 2019) represent CNN-based, recurrent-based, and GAN-based approaches, respectively. GDN (Deng & Hooi, 2021) and MTAD-GAT (Zhao et al., 2020) are graph-based models. We further include transformer-based methods such as Anomaly Transformer (Xu et al., 2021), D3R (Wang et al., 2023), TranAD (Tuli et al., 2022), SAT (Yue et al., 2024), SimAD (Zhong et al., 2025a), NPSR (Lai et al., 2023), and SensitiveHUE (Feng et al., 2024). In addition, PatchAD (Zhong et al., 2025b) is an MLP-mixer-based model, while OracleAD (Cho et al., 2025) is an LSTM-based approach that explicitly models causality. Furthermore, we include PCA-Error (Sarfraz et al., 2024), a classical linear reconstruction-based baseline for anomaly detection. We have also included CAROTS (Kim et al., 2025), a causality-aware contrastive learning method; CATCH (Wu et al., 2025), a frequency-domain patching-based reconstruction method; TimesNet (Wu et al., 2022), a 2D temporal variation modeling backbone; and xLSTMAD (Faber et al., 2025), an encoder-decoder xLSTM-based anomaly detection method.

To obtain the reported **best F1 scores**, we primarily refer to the original publications of each baseline. When the original paper does not report best-F1 scores, we adopt the corresponding results from subsequent peer-reviewed works. Since not all baselines evaluate on all datasets, some entries in Table 2 are unavailable.

SensitiveHUE reports results only on the SWAT, WADI, MSL, and SMD datasets. For completeness, we evaluate SensitiveHUE (Offline) on the PSM and SMAP datasets, and SensitiveHUE (Online) on all datasets using the publicly available code. In addition, TimesNet reports point-adjusted F1, which has known flaws, and xLSTMAD does not report per-dataset F1; we therefore reproduce these baselines using their publicly available code for all datasets. A detailed summary of the result sources for all baselines is provided in Table 8.

Table 8. Details of the baselines and their corresponding result sources. The listed sources primarily refer to the reported best-F1 results. For additional metrics (e.g., PR AUC, VUS PR, and VUS ROC), the corresponding values, when available, are obtained from the same sources as the best-F1 scores.

Baselines	Venue	Result Source	Baselines	Venue	Result Source
DAGMM (Zong et al., 2018)	ICLR-2018	NPSR	TranAD (Tuli et al., 2022)	VLDB-2022	NPSR
LSTM-VAE (Park et al., 2018)	IEEE-2018	NPSR	NPSR (Lai et al., 2023)	NeurIPS-2023	NPSR
MSCRED (Zhang et al., 2019)	AAAI-2019	NPSR	SAT (Yue et al., 2024)	IJCAI-2024	SAT
OmniAnomaly (Su et al., 2019)	KDD-2019	NPSR	OracleAD (Cho et al., 2025)	NeurIPS-2025	OracleAD
MAD-GAN (Li et al., 2019)	ICANN-2019	NPSR	PCA-Error (Sarfraz et al., 2024)	ICML-2025	Position
MTAD-GAT (Zhao et al., 2020)	ICDM-2020	NPSR	PatchAD (Zhong et al., 2025b)	IEEE-2025	PatchAD
USAD (Audibert et al., 2020)	KDD-2020	NPSR	SimAD (Zhong et al., 2025a)	IEEE-2025	SimAD
UAE (Garg et al., 2021)	IEEE-2020	NPSR	Sens.HUE(Offline) (Feng et al., 2024)	KDD-2024	SensitiveHUE
GDN (Deng & Hooi, 2021)	AAAI-2021	NPSR	Sens.HUE(Online) (Feng et al., 2024)	KDD-2024	Evaluated by us
Anomaly Trans. (Xu et al., 2021)	ICLR-2022	NPSR	CAROTS (Kim et al., 2025)	ICML-2025	CAROTS
TimesNet (Wu et al., 2022)	ICLR-2023	Evaluated by us	CATCH (Wu et al., 2025)	ICLR-2025	CATCH
D3R (Wang et al., 2023)	NeurIPS-2023	SimAD	xLSTMAD (Faber et al., 2025)	IEEE-2025	Evaluated by us

F. Definition of Best-F1 Score

Our anomaly detection framework depends on the statistical significance α and the threshold h (Section 3.4). So, the predicted label $\hat{y}_t \in \{0, 1\}$ is a function of α and h. True Positive (TP), False Positive (FP), and False Negative (FN) are defined as follows:

$$\mathrm{TP} = \{\, t \mid \hat{y}_t = 1,\ y_t = 1 \,\} \tag{18}$$

$$\mathrm{FP} = \{\, t \mid \hat{y}_t = 1,\ y_t = 0 \,\} \tag{19}$$

$$\mathrm{FN} = \{\, t \mid \hat{y}_t = 0,\ y_t = 1 \,\} \tag{20}$$

where $\hat{y}_t$ and y_t denote the predicted and ground-truth labels at time step t, respectively. The Precision (P), Recall (R), and F1 score are derived by

$$\mathrm{P} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FP}} \tag{21}$$

$$\mathrm{R} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FN}} \tag{22}$$

$$\mathrm{F1} = \frac{2PR}{P + R}. \tag{23}$$

The Best-F1 score is obtained by selecting α and h using the threshold sweeping method described in (Feng et al., 2024; Lai et al., 2023) to maximize the F1 score:

$$\text{Best-F1} = \max_{\alpha,\, h} \mathrm{F1}\big(\hat{\boldsymbol{y}}(\alpha, h), \boldsymbol{y}\big), \tag{24}$$

where $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ are the set of predicted and ground truth labels.

Table 9. Search range of the hyperparameters

	SWAT	WADI	PSM	SMAP	MSL	SMD
Number of clusters (M)	$[1, 8]$	$[1, 10]$	$[1, 7]$	$[1, 10]$	$[1, 10]$	$[1, 10]$
Initial learning rate	$[10^{-5}, 10^{-2}]$					
Look-back window length (L)	24					
α (Equ. 14)	$[0, 0.001]$	$[0, 0.047]$	$[0, 0.603]$	[0, 1)		
Embedding mixer exp. factor (d_f)	$\{1, 3, 5\}$					
Embedding dimension (d)	$\{128, 256\}$					
Train epoch	30					
Batch size	512					

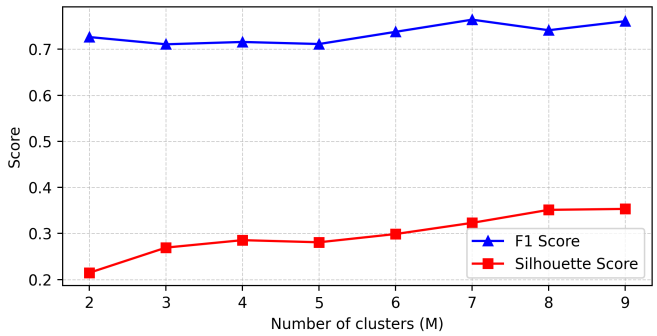
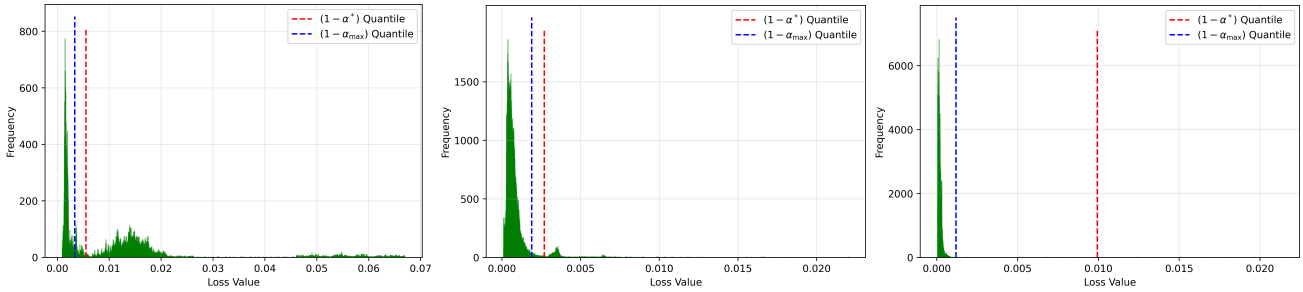


Figure 6. Performance of the model on the WADI dataset with varying M. A higher silhouette score indicates better clustering quality. Cluster numbers greater than 9 are ignored, as they result in at least one cluster containing a single channel.

G. Training and Hyperparameter Search

We allocate 20% of the training dataset for validation purposes. Experiments with multi-entity datasets are performed in three-protocols: (i) training a single model across all entities, (ii) training separate models per entity and averaging entity-wise F1 scores, and (iii) training per-entity models and aggregating predictions before computing F1. Unless otherwise specified, results for multi-entity datasets correspond to protocol (iii), following (Feng et al., 2024). Hyperparameter optimization is performed using the Optuna framework. The ranges of the hyperparameters searched during hyperparameter optimization are listed in Table 9. For all experiments, we use the Adam optimizer to optimize the model. We also use MSE as the reconstruction loss to optimize the reconstruction model during training. We set the look-back window length to 24 and δ to 5 for all experiments.



(a) PSM *(b)* WADI *(c)* SWAT

Figure 7. Histograms of the validation loss distributions for different datasets. Here, α^* denotes the optimal value that yields the highest F1 score, while $\alpha_{\max}$ represents the maximum value of the search range for α. The value of $\alpha_{\max}$ for the corresponding datasets are mentioned in Table 9.

To determine the optimal number of clusters M, we first compute ϕ_{abs} from the training data, where each row represents the correlation profile of a channel. Channels are then clustered based on these profiles. To narrow the search space for M, we use the silhouette score as a measure of clustering quality, where a higher score indicates better separation (Shahapure & Nicholas, 2020). Since directly optimizing M over a wide range of M (e.g., $M = 2$ to 50 for a dataset with 100 channels) is computationally expensive, the silhouette score is used to preselect a smaller, promising range. As shown in Figure 6, the $F1$ score peaks within the same range of M (from 6 to 9) that yields higher silhouette scores, confirming the effectiveness of silhouette-guided selection. The model is subsequently fine-tuned using Optuna across this range to determine the final M.

Our model is implemented in PyTorch 2.5.1 with CUDA 11.8, using Python 3.10.12. All experiments are conducted on a system equipped with an NVIDIA GeForce RTX 4090 GPU (24 GB), an AMD Ryzen 9 7950X 16-core processor, and 128 GB of RAM.

G.1. Selection of the significance parameter α

The parameter α controls the statistical significance of anomaly evidence by defining a $(1-\alpha)$-percentile over the validation reconstruction loss distribution. For datasets with sufficiently large validation sets (SWaT, PSM, and WADI), we empirically observe that the validation reconstruction loss can be well characterized by a mixture of Gaussian components, where the dominant component with mean closest to zero. Based on this observation, we employ a principled heuristic to restrict the search range of α. Specifically, we identify the Gaussian component whose mean is nearest to zero and compute an upper bound in the loss domain as $\mu + \sigma$ of that component. This value is then mapped to its empirical quantile to obtain $(1 - \alpha_{\max})$, which defines the upper limit $\alpha_{\max}$. The optimal significance parameter α^* is subsequently searched within the reduced interval $[0, \alpha_{\max}]$, substantially narrowing the search space. As illustrated in Figure 7, the selected α^* consistently lies well below $\alpha_{\max}$, indicating that this bound serves as a conservative and effective range reduction rather than a hard threshold.

In contrast, for datasets with limited validation samples due to entity-wise training (MSL, SMAP, and SMD), reliable estimation of the validation loss distribution is not feasible. Therefore, for these datasets, α is searched over the full range $[0, 1)$ to avoid bias introduced by unstable distributional estimates.

G.2. Training vs. Inference Behavior

During training, Batch Normalization relies on batch-level statistics. However, during inference, the model operates in evaluation mode using fixed running statistics. As anomaly detection is performed only at inference time, predictions and anomaly scores at time t do not depend on future observations, ensuring temporal causal behavior in detection.

H. Additional Results

In addition to the best F1 score, we present our model's performance using VUS_ROC and VUS_PR, which are shown in Table 10. The VUS metrics were introduced to mitigate excessive penalization caused by minor boundary misalignment between detected anomaly regions and ground-truth annotations, which are often unavoidable in time-series data (Paparrizos

Table 10. Performance comparison using additional metrics (VUS_PR, VUS_ROC) across six benchmark datasets. For multi-entity datasets, baselines annotated with superscript^{*i} correspond to results under evaluation protocol-i.

Dataset	Metrics	TimesNet^{*1}	xLSTMAD^{*1}	CATCH^{*1}	CCM-TAD^{*1}	Sen.H.(On)^{*2}	CCM-TAD^{*2}
MSL	VUS_PR	0.211	0.180	**0.256**	0.249	0.291	**0.581**
	VUS_ROC	0.703	0.646	**0.735**	0.660	0.673	**0.777**
SMD	VUS_PR	0.220	0.108	0.159	**0.263**	0.349	**0.492**
	VUS_ROC	0.795	0.620	**0.797**	0.525	**0.801**	0.750
SMAP	VUS_PR	0.120	0.145	0.155	**0.326**	0.335	**0.533**
	VUS_ROC	0.471	0.527	0.543	**0.569**	0.700	**0.740**

Dataset	Metrics	TimesNet	xLSTMAD	CATCH	Sen.H.(On)	CCM-TAD
SWAT	VUS_PR	0.223	0.469	0.241	0.729	**0.779**
	VUS_ROC	0.541	0.625	0.462	**0.828**	0.758
PSM	VUS_PR	0.428	0.484	0.436	0.454	**0.485**
	VUS_ROC	0.617	0.621	0.639	**0.676**	0.545
WADI	VUS_PR	0.379	0.550	-	0.597	**0.776**
	VUS_ROC	0.813	**0.861**	-	0.857	0.822

Table 11. Performance comparison between CCM-TAD and CATCH using Aff_F1 and R_F1 metrics.

Metrics	Models	PSM	SWAT	MSL^{*1}	SMD^{*1}	SMAP^{*1}
R_F1	CCM-TAD	0.428	**0.681**	**0.292**	**0.293**	**0.347**
	CATCH	**0.498**	0.148	0.185	0.158	0.173
Aff_F1	CCM-TAD	0.694	**0.781**	**0.756**	0.695	**0.703**
	CATCH	**0.859**	0.755	0.740	**0.847**	0.699

et al., 2022). The results show that our model consistently achieves strong performance in VUS_PR across most datasets, outperforming the baselines included in the table in the majority of cases. Although the improvements in VUS_ROC are comparatively less consistent, VUS_PR is generally more informative than VUS_ROC for highly imbalanced anomaly detection tasks (Saito & Rehmsmeier, 2015). Additionally, in Table 11, we compare our model against CATCH using Aff_F1 and R_F1. It shows that our model also outperforms CATCH in these two metrics in most cases.

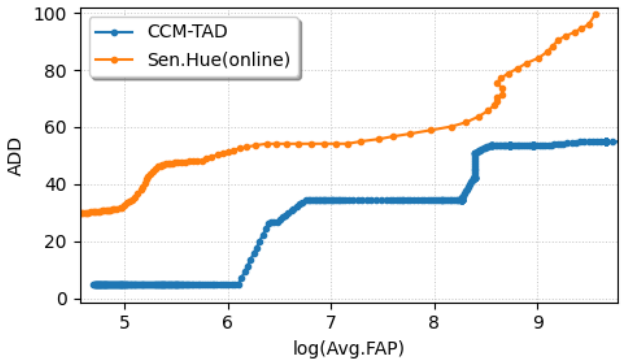


Figure 8. ADD-FAP Trade-off comparison for WADI dataset.

Figure 8 shows the trade-off between average detection delay (ADD) and the logarithm of the average false-alarm period (log(Avg.FAP)), which is commonly used to evaluate the performance sequential change/anomaly detection methods. False positive rate (i.e., false alarm probability) corresponds to the inverse of FAP. Our method (CCM-TAD) attains lower ADD than SensitiveHUE(online) at comparable false alarm periods, indicating faster detection for the same false-alarm period. Overall, the CCM-TAD curve lies consistently below the SensitiveHUE curve, demonstrating a more favorable ADD-FAP trade-off for online anomaly detection.

I. Computational Cost and Robustness

Table 12 reports the robustness comparison between our model and SensitiveHUE (Online) across four metrics, with results averaged over five random seeds. Our model achieves superior performance on most metrics across the datasets. While SensitiveHUE occasionally attains higher VUS-ROC scores, this metric is less informative under severe class imbalance. In anomaly detection settings, VUS-PR is more appropriate, as it better reflects precision–recall trade-offs and penalizes false positives, consistent with prior studies favoring PR-based metrics over ROC-based metrics for imbalanced data (Saito & Rehmsmeier, 2015).

Table 13 reports the computational cost in terms of GFLOPs per batch, number of trainable parameters, and average inference time. Compared to SensitiveHUE (Online), our model consistently requires fewer GFLOPs across all datasets. In terms of model size, CCM-TAD uses fewer parameters on SWAT, PSM, WADI, and SMAP, while requiring more parameters on MSL and SMD. Although CCM-TAD incurs higher inference latency across datasets compared to SensitiveHUE (Online), the absolute inference times remain low (on the order of milliseconds per batch). Overall, these results indicate that CCM-TAD improves robustness and detection performance with a moderate and controlled computational overhead.

Table 12. Robustness Comparison

Dataset	Metrics	CCM-TAD (ours)	Sen.HUE(Online)	Dataset	Metrics	CCM-TAD (ours)	Sen.HUE(Online)
SWAT	F1	0.877 ± 0.004	0.850 ± 0.033	MSL	F1	0.633 ± 0.023	0.421 ± 0.005
	PR_AUC	0.907 ± 0.004	0.832 ± 0.022		PR_AUC	0.592 ± 0.024	0.280 ± 0.006
	VUS_ROC	0.749 ± 0.011	0.819 ± 0.014		VUS_ROC	0.782 ± 0.015	0.670 ± 0.005
	VUS_PR	0.770 ± 0.009	0.711 ± 0.025		VUS_PR	0.564 ± 0.024	0.291 ± 0.006
PSM	F1	0.688 ± 0.022	0.502 ± 0.003	SMD	F1	0.507 ± 0.068	0.336 ± 0.019
	PR_AUC	0.558 ± 0.023	0.457 ± 0.004		PR_AUC	0.561 ± 0.030	0.404 ± 0.006
	VUS_ROC	0.546 ± 0.007	0.668 ± 0.005		VUS_ROC	0.744 ± 0.010	0.799 ± 0.004
	VUS_PR	0.487 ± 0.013	0.449 ± 0.003		VUS_PR	0.460 ± 0.032	0.347 ± 0.005
WADI	F1	0.744 ± 0.015	0.679 ± 0.006	SMAP	F1	0.558 ± 0.030	0.532 ± 0.014
	PR_AUC	0.698 ± 0.024	0.617 ± 0.004		PR_AUC	0.571 ± 0.020	0.346 ± 0.003
	VUS_ROC	0.813 ± 0.017	0.864 ± 0.011		VUS_ROC	0.743 ± 0.007	0.699 ± 0.006
	VUS_PR	0.756 ± 0.016	0.598 ± 0.010		VUS_PR	0.551 ± 0.016	0.334 ± 0.002

Table 13. Comparison of computational costs in terms of GFLOPs per batch, number of trainable parameters, and average inference time per batch. For multi-entity datasets, the reported values are averaged over all entities. All experiments are performed on the same hardware described in Section G.

	SWAT			PSM			WADI		
	GFLOPs	#Param.	Inf. time	GFLOPs	#Param.	Inf. time	GFLOPs	#Param.	Inf. time
Sen.HUE(Online)	3.73	155k	0.0008	13.41	551k	0.0008	28.19	1152k	0.0015
CCM-TAD	1.05	43k	0.0015	3.55	144k	0.002	10.68	435k	0.0021

	MSL			SMD			SMAP		
	GFLOPs	#Param.	Inf. time	GFLOPs	#Param.	Inf. time	GFLOPs	#Param.	Inf. time
Sen.HUE(Online)	14.02	291k	0.0014	13.7	284k	0.0013	26.34	1077k	0.0013
CCM-TAD	10.6	432k	0.0028	8.98	366k	0.0022	10.47	426k	0.0021

J. Additional Ablation Studies

J.1. Channel Clustering Variants

To evaluate the effectiveness of our proposed spectral clustering method based on correlation profiles, we conduct a comprehensive ablation study comparing three alternative channel clustering strategies. These strategies isolate the influence of different similarity metrics and clustering paradigms on the structural organization of features and their downstream impact on anomaly detection performance.

(1) Channel Similarity-Based Spectral Clustering: In this variant, we construct the similarity-based adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{C\times C}$ using the cosine similarity between channel time series. The similarity score $w_{ij} \in [0, 1]$ is computed as,

$$w_{ij} = \begin{cases} \max\left(0, \frac{\langle \mathcal{X}_{N(:,i)}, \mathcal{X}_{N(:,j)} \rangle}{\|\mathcal{X}_{N(:,i)}\| \|\mathcal{X}_{N(:,j)}\|}\right), & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{25}$$

where, $\mathcal{X}_{N(:,i)} \in \mathbb{R}^{N\times 1}$ is the i^{th} channel of the training data $\mathcal{X}_N \in \mathbb{R}^{N\times C}$. N and C are the length and the number of channels of the training data, respectively. We then apply a similar spectral clustering approach described in Section 3.2.1 to the resulting similarity weight matrix.

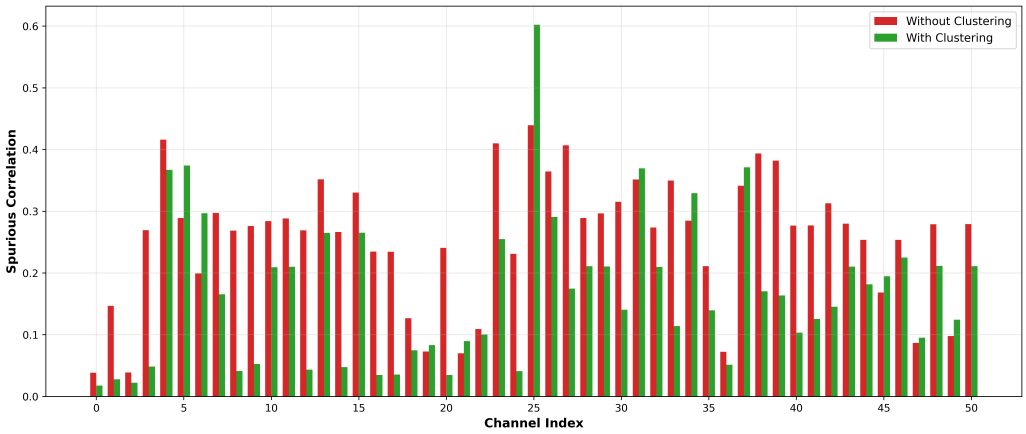
(2) Absolute Correlation-Based Spectral Clustering: In this configuration, the adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{C\times C}$ is directly derived from the absolute Pearson correlation matrix $\boldsymbol{\Phi}_{\text{abs}} \in \mathbb{R}^{C\times C}$, where $w_{ij} = \phi_{ij}$. Spectral clustering is then performed on this graph to determine channel groupings. This method captures the pairwise correlation without considering the full correlation profile.

(3) K-Means Clustering on Correlation Matrix: Here, instead of forming a graph and applying spectral clustering, we directly cluster the rows of $\boldsymbol{\Phi}_{\text{abs}}$ using the K-Means algorithm. Each row ϕ_i can be interpreted as the correlation profile of the ith channel, representing its relationship with all other channels. This method evaluates the efficacy of clustering in the original correlation profile space, independent of the properties of the spectral graph.

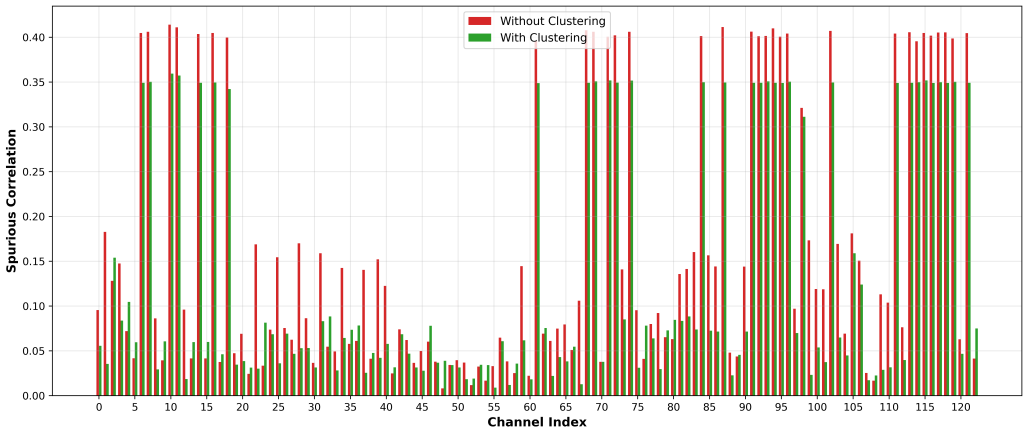
The anomaly detection results of all these approaches are shown in Table 7 of the main paper. Our proposed approach, which utilizes spectral clustering on the full correlation profiles, consistently outperforms these baselines, highlighting the importance of preserving inter-channel relationships during clustering.

Table 14. Spurious correlation values obtained with clustering (SC^W) and without clustering ($SC^{W/O}$) for the PSM dataset. The column $\Delta SC(\%)$ denotes the percentage difference, computed as $\Delta SC(\%) = 100 \times (SC^{W/O} - SC^W)/SC^{W/O}$.

Channel	SC^W	$SC^{W/O}$	$\Delta SC(\%)$
1	0.132	0.189	30.2
2	0.177	0.253	30
3	0.260	0.247	-5.3
4	0.225	0.285	21.1
5	0.081	0.137	40.9
6	0.098	0.151	35.1
7	0.142	0.174	18.4
8	0.076	0.166	54.2
9	0.086	0.115	25.2
10	0.085	0.124	31.5
11	0.080	0.117	31.6
12	0.092	0.162	43.2
13	0.120	0.436	72.5
14	0.126	0.557	77.4
15	0.127	0.180	29.4
16	0.155	0.173	10.4
17	0.087	0.118	26.3
18	0.085	0.143	40.6
19	0.122	0.145	15.9
20	0.079	0.070	-12.9
21	0.148	0.386	61.7
22	0.123	0.187	34.2
23	0.185	0.124	-49.2
24	0.277	0.290	4.5
25	0.099	0.147	32.7
Avg.			27.98



(a) SWAT



(b) WADI

Figure 9. Spurious correlation reduction using our clustering approach

J.2. Effect of Channel Clustering on Reducing Spurious Correlation in Prediction

Our proposed model estimates channel correlations for normal data more accurately than the non-clustered approach. To evaluate the accuracy of the estimated correlations, we use the spurious correlation (SC) metric. The spurious correlation for channel-i (SC_i) is defined as,

$$\mathrm{SC}_i = \frac{1}{C} \sum_{j=1}^{C} \left| \phi_{ij} - \hat{\phi}_{ij} \right| \tag{26}$$

where ϕ_{ij} and $\hat{\phi}_{ij}$ denote the true and estimated correlations between channels i and j, respectively, computed from the normal test data. Table 14 presents the detailed results for the PSM dataset, showing that our clustering approach reduces the average spurious correlation among channels by 27.98%. Similarly, the reductions for the SWAT and WADI datasets are 32.74% and 14.22%, respectively. Figure 9 further illustrates the spurious correlation reduction across the SWAT and WADI datasets.

J.3. Sensitivity Analysis for the Hyperparameters

Sensitivity Analysis on δ: The parameter δ controls the decay of the accumulated anomaly evidence s_t by requiring δ consecutive negative anomaly evidences β_t before resetting s_t to zero. Figure 10 shows that the anomaly detection performance remains largely unchanged across different δ settings. This robustness is attributed to the proposed anomaly boundary refinement mechanism Equation (17), which determines the optimal anomaly endpoint of an anomaly sequence.

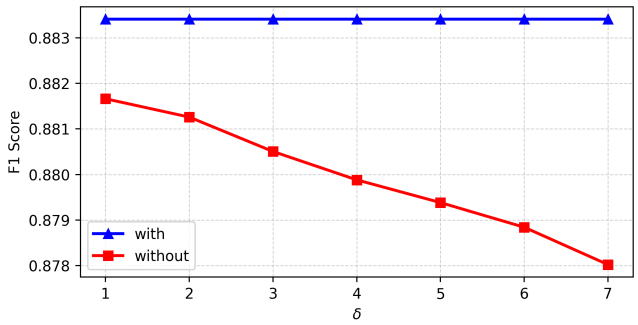


Figure 10. Performance of our proposed anomaly detection method for the SWAT dataset with varying δ is shown. "With" denotes the method incorporating the optimal end-point detection formula Equation (17), while "Without" indicates its exclusion. The results demonstrate that performance remains consistent across different δ values when the optimal end-point detection is applied.

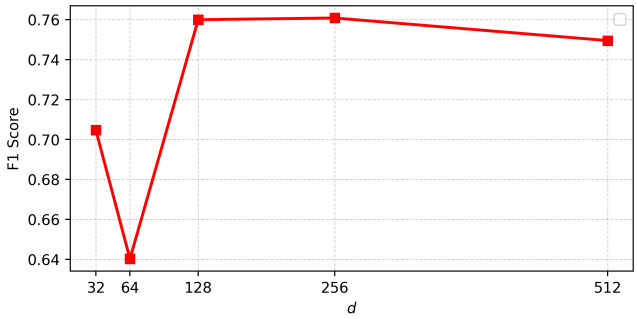
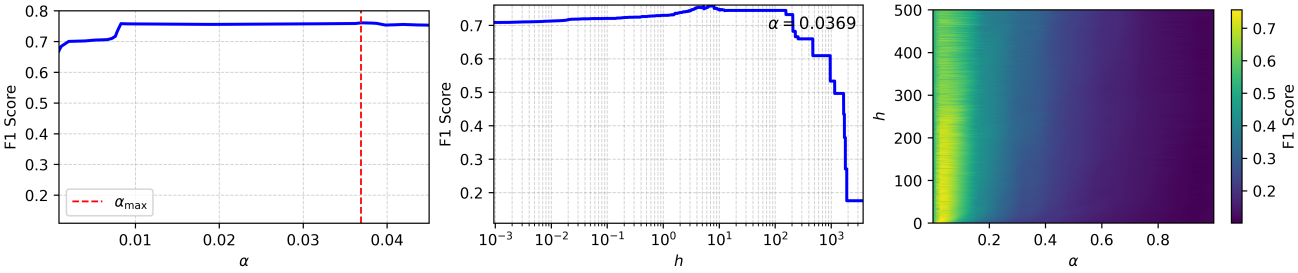


Figure 11. Performance of our model for the WADI dataset with varying d is shown.



(a) Sensitivity on α *(b)* Sensitivity on h *(c)* Joint sensitivity on α and h

Figure 12. F1 score sensitivity on α and h for the WADI dataset.

Sensitivity Analysis on d: Figure Figure 11 illustrates the model's performance on the WADI dataset for different values of $d \in \{32, 64, 128, 256, 512\}$. The F1 score varies with d but shows no clear linear trend. Therefore, we determine the optimal d using Optuna in our experiments.

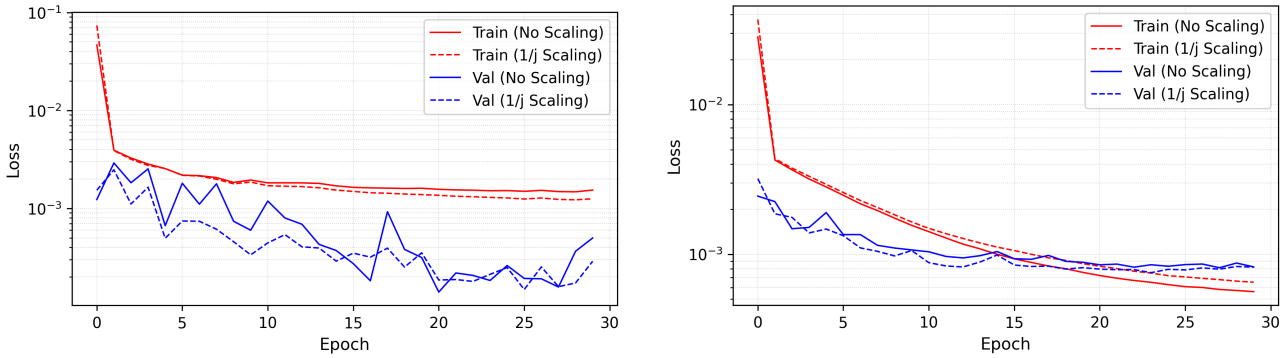
Sensitivity Analysis for α and h: The parameter α controls the statistical significance of anomaly evidence. Reconstruction error g_t greater than the $(1-\alpha)$-percentile in validation set result in a positive anomaly evidence β_t (Eq. (14)). Tuning this hyper-parameter allows the detector to adapt to dataset-specific nominal behavior. Figure 12a shows the sensitivity of the F1 score on α for the WADI dataset, where smaller values of α achieve superior performance, indicating close alignment between validation and test normal losses. It also shows that the F1 score remains stable and consistently high within the relevant search range $[0, \alpha_{\max}]$, confirming that the method is robust within its operating region. Figure 12b analyzes the effect of the decision threshold h for a fixed α, showing stable performance up to $h \approx 10^2$, beyond which the F1 score degrades. Figure 12c presents the joint sensitivity on α and h, revealing a broad region of robust performance.

J.4. Effect of Causal Mask's $1/j$ Scaling

Figure 13 shows the effect of $1/j$ scaling in the causal temporal mask on the SWAT and WADI datasets. Here, "No scaling" refers to unit masking. While both unscaled and scaled masked variants exhibit similar convergence in training loss, the unscaled masked variant leads to noticeable oscillations and spikes in validation loss on the SWAT dataset. The WADI dataset also exhibits more oscillations in the validation loss with the unscaled masked variant. Overall, these results suggest that incorporating $1/j$ scaling moderates the optimization dynamics.

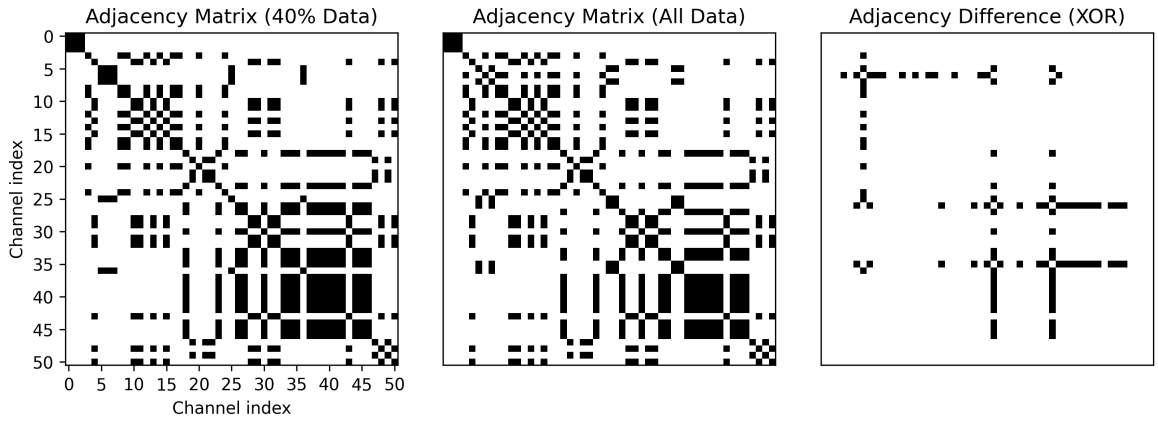
J.5. Impact of Distributional Shifts in Inter-Channel Correlation

Time-series data can exhibit distributional shifts in inter-channel correlations. To examine their impact, we consider two clustering settings: one using the first 40% of the training data to derive channel clusters, and another using the full training

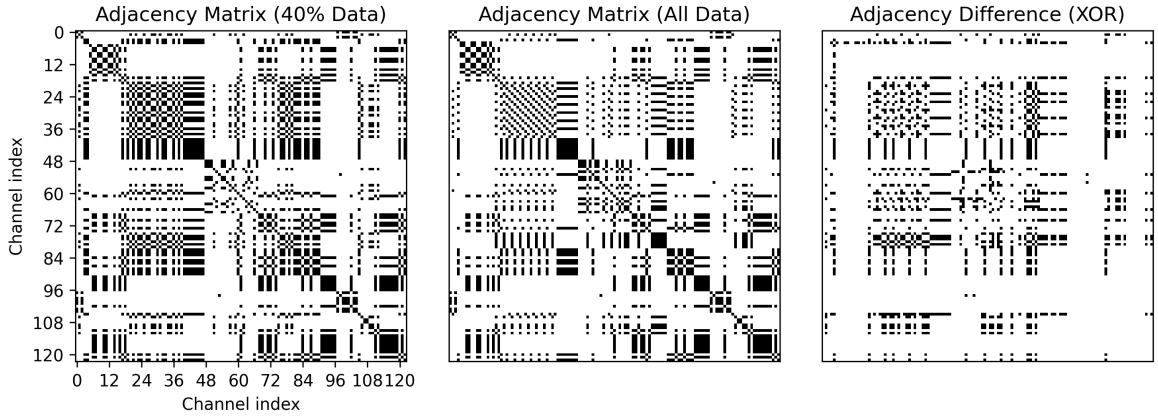


(a) SWAT *(b)* WADI

Figure 13. Training and validation loss with and without $1/j$ scaling in the causal mask. "No Scaling" refers to unit masking ($\gamma_{ij} = 1; i \leq j$).



(a) SWAT



(b) WADI

Figure 14. Adjacency matrices presenting the graph structure of the channels, derived from different portions of the train dataset. Black and white entries denote the presence (1) and absence (0) of edges. The right panel shows the element-wise XOR between the two adjacency matrices, highlighting structural differences induced by data availability.

set. The number of clusters is kept fixed, and the model is trained for both settings. As shown in Figure 14, the adjacency difference (XOR) highlights changes in the induced graph structure caused by correlation shifts. Despite these changes, performance remains largely consistent. The best-F1 scores on SWAT are 0.878 and 0.883, while on WADI they are 0.762

and 0.761 for the two settings, respectively. This suggests robustness to moderate correlation shifts, while more substantial shifts may require re-estimation of the cluster numbers.

K. Motivation for the Causal Mixer

In time series reconstruction, preserving temporal causality is critical. Without it, the representation at any time may be contaminated by future anomalies or vice versa. This can cause information leakage and distort the anomaly score. A conventional temporal mixer mixes information across the sequence without maintaining temporal order, which conflicts with the requirements of anomaly detection. We therefore mask the temporal mixer's weights to maintain temporal causality during information mixing. This directional information flow ensures effective model learning. Note that this causality constraint applies specifically to the information mixing process inside the temporal mixer, rather than to the last reconstructed point of the input window, as no future information is accessible at the last point of the input window.

L. Online Applicability of our Anomaly Detection Method

Our proposed sequential anomaly detection method is applicable to detecting anomalies in real time. Unlike several prior methods, we do not apply any normalization to the *anomaly score* that requires access to future test data. For example, SensitiveHUE normalizes anomaly scores using the median and interquartile range computed from the *entire test set*, which necessitates observing all test samples beforehand and therefore limits its applicability to offline settings. In this paper, we denote their original implementation as "SensitiveHUE(Offline)".

In our method, the anomaly score s_t at time t depends solely on information available up to that time: (i) the reconstruction error at t, (ii) past anomaly evidences $\beta_{<t}$, (iii) past anomaly scores $s_{<t}$, and (iv) statistics derived from the validation set, which are obtained prior to deployment. No future test observations are required at any stage. Therefore, it operates in real-time.

Furthermore, the inference latency of our reconstruction model is low, as reported in Table 13, enabling timely processing of incoming measurements. These properties make the proposed approach suitable for practical online anomaly monitoring.

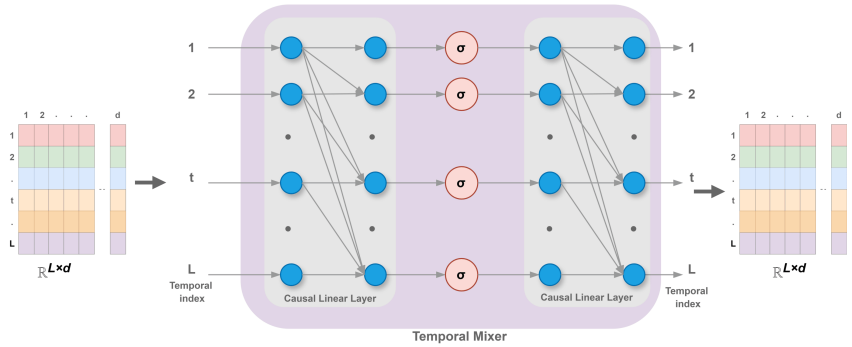


Figure 15. Neural visualization of Temporal mixer module.

M. Limitations

Causality in time series analysis is a broad research area. This work focuses exclusively on *temporal causality*. We do not aim to discover Granger causality, inter-channel causality, or the specific channels responsible for an anomaly (Han et al., 2025). In the temporal mixer, each neuron corresponding to a temporal index t is allowed to interact only with neurons corresponding to indices $\leq t$ from the previous layer, illustrated in Figure 15. This enforces directional information flow through weight masking, similar to the attention mask in transformers, but it does not model explicit causal graphs among variables. Moreover, since the model is built for multivariate time series anomaly detection, and cluster-aware multi-embedding by nature requires multiple channels, evaluating it on univariate datasets falls outside the intended scope of this work.