MLP-Mixer based Masked Autoencoders Are Effective, Explainable and Robust for Time Series Anomaly Detection

Ben Trovato
Institute for Clarity in Documentation
Dublin, Ireland
trovato@corporation.com

Lars Thørväld
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Wang Xiu Ying
Zhe Zuo
East China Normal University
Shanghai, China
firstname.lastname@ecnu.edu.cn

ABSTRACT

Time series anomaly detection remains one of the most active research areas in data mining due to its wide range of real-world applications. In recent years, with the flourishing of deep learning, numerous deep learning-based methods have been proposed for time series anomaly detection. However, these methods often fail to detect subsequence anomalies with prolonged durations. Additionally, deep learning-based methods lack explainability and are vulnerable to training set contamination. These drawbacks have raised doubts about the feasibility of deep learning in time series anomaly detection. This paper addresses above doubts by proposing a more suitable deep learning framework, MMA, to achieve effective, explainable, and robust time series anomaly detection. In detail, we incorporate the MLP-Mixer backbone with a masked autoencoder-based anomaly detection approach to allow for a significantly larger input window size (10 to 20 times larger than the input window sizes of current models). Benefitting from this large input window, our model can detect challenging subsequence anomalies. Meanwhile, a contrast learning module designed explicitly for anomaly detection is proposed to aid in detecting subtle anomalies that fail to be identified by residual errors. Furthermore, a dynamic anomaly filtering method is introduced to mitigate the impact of anomalies on the reconstruction of surrounding normal regions.

We evaluate the performance of MMA using trustworthy datasets and rigorous evaluation metrics. The results show that our proposed model achieves significant improvements over the state-of-the-art methods, with an average absolute improvement of 10% to 18% across all metrics on multivariate time series datasets. Besides, MMA outperforms state-of-the-art non-deep learning models for the first time on the univariate time series dataset: the UCR Archive. We also observe that MMA has a strong ability to reconstruct potential normal patterns in anomalous regions, thereby providing high explainability. In our designed explainability quantification experiment, MMA reports a substantial increase in explainability compared to other deep learning-based methods. Besides, MMA demonstrates high robustness to various types of training set contamination.

PVLDB Reference Format:
Ben Trovato, Lars Thørväld, Wang Xiu Ying, and Zhe Zuo. MLP-Mixer based Masked Autoencoders Are Effective, Explainable and Robust for Time Series Anomaly Detection. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at URL_TO_YOUR_ARTIFACTS.

1 INTRODUCTION

Time series anomaly detection refers to identifying unusual patterns that significantly deviate from the majority of observations in a sequence of data collected over time. This technique is essential in various application domains, including healthcare monitoring, financial fraud detection, spacecraft telemetry, and server center operations. Due to the cost and difficulty of labeling work in these real-world applications, time series anomaly detection is often formulated as an unsupervised task with unlabeled training data [3]. Unsupervised anomaly detection generally presumes that the training data only contains normal samples, allowing the model to capture the normal pattern of time series. The data that deviates from the learned normal patterns are then identified as anomalies. Classic unsupervised methods include discord discovery-based methods [37], graph-based methods [4], and density-based methods [2]. In recent years, with the rise of deep learning, many methods based on deep learning have claimed that neural networks can help learn long-period, complex nonlinear temporal relationships in time series, and are therefore beneficial for anomaly detection [10, 21, 42, 53, 62]. However, recent studies [26, 36, 38, 43, 44, 46, 51] have indicated that deep learning-based methods do not perform well on reputable datasets [56] when rigorous evaluation metrics are employed [14, 24, 40]. The above problem has sparked skepticism about the effectiveness of deep learning methods for time series anomaly detection [18]. Our observations align with these studies and we find the following three limitations in current deep learning methods:

Lack of the ability to detect anomalies with prolonged duration (Limitation 1): As illustrated in Figure 1, state-of-the-art deep learning methods, including reconstruction-based methods like TranAD [54], MAUT [42], and prediction-based methods such as CAD [50] and MTAD-GAT [64], tend to overfit prolonged continuous anomalies. These methods utilize the residual error as the

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

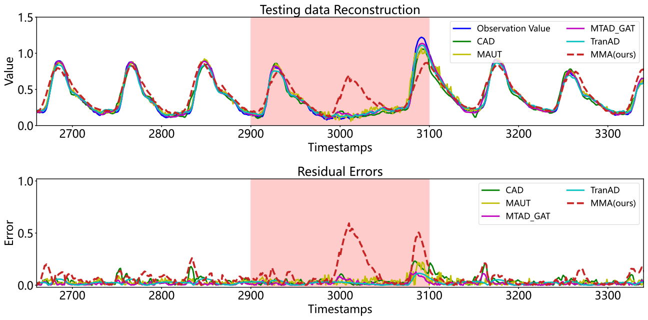


Figure 1: Detection Results on UCR 004 dataset. The anomalous region is highlighted in light red. Compared to the existing deep learning models, our model allows for a significantly larger input window size of 1728, enabling effective detection of anomalies with prolonged duration.

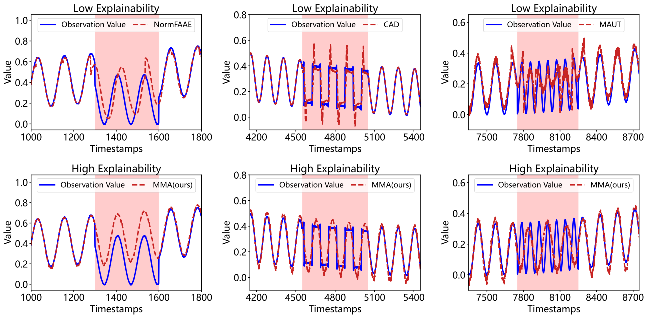


Figure 2: Comparison between our high explainability model and other low explainability models on NeurIPS-TS-Synthetic dataset. The blue line represents the observation values, while the red dashed line represents the predicted or reconstructed values. The anomalous regions are highlighted in light red.

anomaly score, failing to detect such anomalies. It is embarrassing that deep learning methods claim to model long-term temporal dependencies but fail to detect "long-term" anomalies. In our opinion, this problem stems from the inappropriate architecture design of current deep learning models, leading to a constrained input contextual window size, specifically, the input window size for reconstruction models and the prediction horizon for prediction models. According to the original papers on these methods [42, 50, 54, 64], the optimal input window size is 3 for CAD, 10 for TranAD, and 100 for MAUT and MTAD-GAT. However, as exemplified in Figure 1, when the length of anomalies significantly exceeds the input window size, existing models tend to overfit anomalies, as the anomaly values dominate the input window. Increasing the window size alone does not address this issue, as larger input windows make the prediction and reconstruction more challenging. This leads to high residual errors in normal regions, causing severe false alarms [7]. In the ideal scenario, we expect a large input window size to prevent anomaly values from dominating the input while ensuring that normal values have minimal residual error.

Lack of explainability (Limitation 2): Considering that anomaly detection models may be deployed in safety-critical domains, such as disease diagnosis and spacecraft condition monitoring, we expect the model to not only provide accurate detection results but also offer tangible explanations for why a specific region is detected as an anomaly [33]. In real applications, domain experts often explain why they identify something as an anomaly by describing how it should look if it were normal [48]. Therefore, we argue that an explainable anomaly detection model should be able to reveal the potential normal patterns in anomalous regions. As depicted in Figure 2, although both models can detect the anomalous regions through residual errors, our model with high explainability can reconstruct the potential normal patterns within the anomalous areas. In contrast, current models with low explainability can only offer a noisy reconstruction. Additionally, operators can infer the causes of the anomalies by comparing the observed anomaly values with the reconstructed normal behavior. For example, by referring to the reconstructed values, we can infer that the anomaly in Figure 2: (a) is caused by a shift in trend, (b) is due to shape distortion, and (c) results from a change in periodicity.

Lack of Robustness (Limitation 3): The robustness of anomaly detection models has been a long-standing concern for researchers [21, 23, 31, 34, 47, 62]. In real applications, models are often trained on datasets that may be polluted with unknown anomalies. When training data contains anomalies, the performance of deep learning models can severely degrade. To address this problem, many recent studies have incorporated sophisticated modules to enhance model robustness. However, these studies often lack explicit robustness evaluation [20, 21, 59, 65] or only consider adding Gaussian Noises as anomalies in the training set [1, 10, 49, 63]. Such approaches do not provide compelling evidence of model robustness, as real-world datasets are unlikely to contain only Gaussian Noises. In our experiments, various anomalies are injected into the training sets, such as Gaussian Noises, trend anomalies, seasonal anomalies, shape anomalies, uniform anomalies, and real anomalies clipped from the testing sets. Unfortunately, we find that currently available models with additional robustness designs perform poorly on these contaminated datasets.

Considering the aforementioned limitations, it appears that, as Eamonn Keogh has suggested, deep learning may not be the solution for time series anomaly detection [18]. However, our work uncovers the true potential of deep learning in time series anomaly detection. We propose MMA, an MLP-Mixer based Masked Autoencoder, to achieve effective, explainable, and robust time series anomaly detection. Our model adopts a patch-based input scheme [39], allowing for the input of contextual windows with a length of 1024, 2048, or larger (10 to 20 times larger than the input window lengths of current models). Such a long input window size prevents anomalies from dominating the input, thereby benefitting the detection of anomalies with prolonged durations. To allow for a long input window size while ensuring accurate reconstruction of normal regions, a training and detection mode resembling Masked Autoencoders [13] is utilized. During training, 50% of the input patches are randomly masked, and the model is trained to

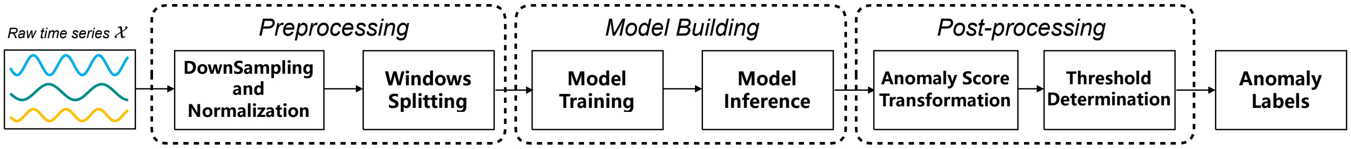


Figure 3: The Time Series Anomaly Detection Pipeline

reconstruct these masked patches. In the testing phase, the reconstruction error of the masked patches is used as an indicator for identifying anomalies. Since the neighboring unmasked values offer additional information, the reconstruction errors for normal regions can be minimized. Furthermore, we observe that some abnormal values exhibit similar amplitude with the potential normal patterns, making them hard to detect by reconstruction errors alone. We propose a novel contrastive learning module to identify these subtle anomalies by comparing the discrepancy between their embeddings and the embeddings of reconstructed normal values. In addition, anomalies disrupt the reconstruction of neighboring normal regions, resulting in false alarms in these areas. A simple yet effective dynamic anomaly filtering method is designed to solve this problem.

In the experiment part, given the existing flawed benchmarks [55, 56], ill-posed evaluation metrics [15, 24, 46], and non-standard data processing procedures, we make the following efforts to ensure a trustworthy, fair, and transparent evaluation: **1. Trustworthy Dataset:** We adopt reliable datasets from multiple domains, including UCR [56], ASD [32], Satellite (collected by us), and NeurIPS-TS-Synthetic [26]. **2. Rigorous Evaluation Metrics:** We select recently proposed rigorous evaluation metrics specifically designed for time series anomaly detection, including Range-AUC-PR [40], VUS-PR [40], F1-Affiliation [14], and F1-(PA%K) [24]. **3. A Unified Implementation Pipeline:** All methods are implemented using a unified implementation pipeline, which includes consistent preprocessing and post-processing procedures.

Compared to 12 state-of-the-art deep learning methods and 4 non-deep learning methods, our model achieves an average improvement of 18.51% in AUC-PR, 14.52% in Range-AUC-PR, 14.70% in VUS-PR, 14.93% in F1-Raw, 10.12% in F1-(PA%K) and 2.12% in F1-Affiliation on multivariate datasets. Additionally, our model surpasses state-of-the-art non-deep learning methods on the UCR dataset for the first time. ***(Address Limitation 1)***. Furthermore, we observe that our model exhibits a strong capability to reconstruct potential normal patterns in anomalous regions, thereby providing high explainability ***(Address Limitation 2)***. Besides offering visual evidence of explainability, we develop a post-hoc explainability analysis method to quantify the explainability of various anomaly detection models. Regarding the validation of robustness, we demonstrate that our model maintains stable performance despite various types of training set contamination without requiring any additional robustness design ***(Address Limitation 3)***.

In conclusion, we make the following contributions:

- **Exploration of new anomaly detection schemas.** To the best of our knowledge, our model is the first deep learning-based anomaly detection approach that accommodates such a large input window size. Benefitting from this large input window, our model demonstrates significant performance improvements.
- **Adoption of a novel model architecture.** We choose MLP-Mixer as the backbone instead of the commonly used LSTM or Transformer models and prove that linear models outperform other backbones. In addition, we propose a contrastive learning approach to help detect subtle anomalies. Moreover, a dynamic anomaly filtering method is devised to reduce false alarms in normal regions.
- **Reliable Evaluation.** We employ trustworthy datasets, rigorous evaluation metrics, and a unified implementation pipeline for performance assessment.
- **Rigorous Explainability and Robustness Verification.** We design a quantitative measure for model explainability and test the robustness of models on training sets containing various types of pollution.

2 PRELIMINARIES

2.1 Problem Formulation

A time series is denoted as $\mathcal{X} = \{x_1, ..., x_T\}$, where T represents the number of observations and each observation $x_t \in \mathbb{R}^C$. C represents the number of channels in $\mathcal{X}$. When $C = 1$, $\mathcal{X}$ is a univariate time series, and when $C > 1$, $\mathcal{X}$ is a multivariate time series. In the unsupervised setting, a training time series $\mathcal{X}^{Train} \in \mathbb{R}^{C\times T_1}$ without any label indicating anomaly is given. The task is to compute an anomaly score $AS\left(x_t\right) \in \mathbb{R}$ for each observation x_t in the testing time series $\mathcal{X}^{Test} \in \mathbb{R}^{C\times T_2}$. A higher anomaly score $AS\left(x_t\right)$ indicates that the observation x_t is more likely to be an anomaly. With a predefined threshold θ_a, an observation x_t is assigned a label based on $\hat{y}_t = \mathbb{I}\left(AS\left(x_t\right) \geq \theta_a\right)$, where $\mathbb{I}(\cdot)$ is the indicator function, 1 denotes an anomaly and 0 denotes normal.

2.2 The Time Series Anomaly Detection Pipeline

As depicted in Figure 3, time series anomaly detection primarily involves three main processes: preprocessing, model building, and post-processing [12]. Down-sampling is generally used to reduce the number of observations in the original sequence, thereby decreasing the training time of models. Window splitting divides raw, long time series into several windows of the input size allowed by the model. Anomaly score transformation contains a series of operations that convert the obtained residual errors into anomaly scores. Threshold determination is to find the threshold for identifying anomalies.

Recent works indicate that both down-sampling and normalization in preprocessing [17], as well as anomaly score transformation and threshold determination in post-processing [12], significantly

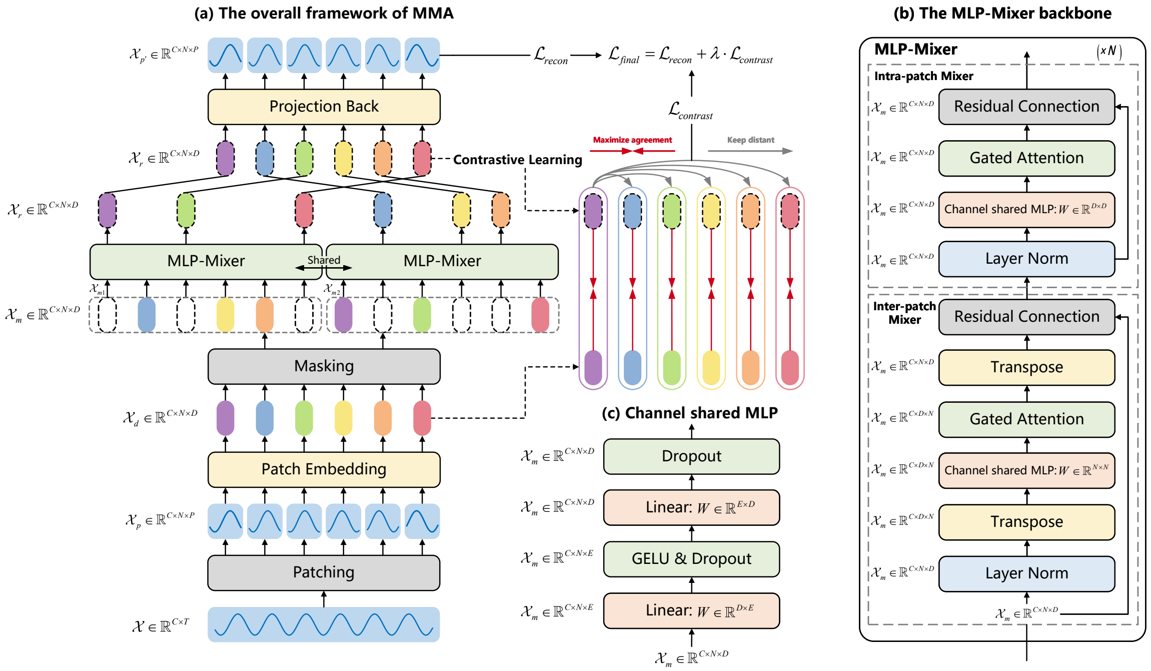


Figure 4: The workflow of MMA framework

impact the final performance of models. Consequently, it is challenging to discern whether performance improvements originate from the model or these various processing procedures. To ensure a fair evaluation, all models in our study are implemented using the following unified pipeline:

- **Downsampling and Normalization.** All models are trained and tested on datasets with the same sampling rates. All datasets are processed with the MinMax normalization, as described in [54].
- **Anomaly Score Transformation.** The residual error r_t^c in the c-th channel is denoted as the difference between the ground truth value and the reconstructed or predicted value:

$$r_t^c = \left| x_t^c - \hat{x}_t^c \right| \tag{1}$$

To mitigate the impact of variations in the residual errors across channels, we subtract the channel-wise mean training residual error from the test errors before computing the root-mean-square across channels as the anomaly scores:

$$r_t^c = \left| r_t^c - \text{mean}\left(r_{train}^c\right) \right|; \; AS\left(x_t\right) = \sqrt{\sum_{c=0}^{C} \left(r_t^c\right)^2 / C} \tag{2}$$

Lastly, a moving average is applied to the anomaly scores, which amplifies the anomaly score even when multiple channels respond to an anomaly at slightly different times.
- **Threshold Determination.** The AUC-PR, Range-AUC-PR, and VUS-PR metrics are independent of the threshold. For F1-Raw, F1-Affiliation, and F1-(PA%K), we employ a commonly used method in evaluation papers [12, 24, 27, 28, 55], which involves searching across all possible thresholds to find the one that yields the optimal result.

3 METHODOLOGY

The overview of MMA is depicted in Figure 4 (a), and the details are described in the following sections.

3.1 Patching and Patch Embedding

We consider each channel in the original input window $\mathcal{X} \in \mathbb{R}^{C \times T}$ as an independent univariate time series $\mathcal{X}^c \in \mathbb{R}^{1 \times T}, c = 1, 2, ..., C$. Each univariate time series $\mathcal{X}^c$ is subsequently divided into non-overlapping patches with patch length P. After that, the $\mathcal{X} \in \mathbb{R}^{C \times T}$ is reshaped into $\mathcal{X}_p \in \mathbb{R}^{C \times N \times P}$, where N denotes the number of patches ($N = T/P$). The patches $\mathcal{X}_p \in \mathbb{R}^{C \times N \times P}$ are then mapped to embeddings $\mathcal{X}_d \in \mathbb{R}^{C \times N \times D}$ via a shared trainable linear projection layer $W_{proj} \in \mathbb{R}^{P \times D}$.

3.2 Masking

Since masking the latent vectors is equivalent to masking the original patches [60], the masking module is applied directly on $\mathcal{X}_d \in \mathbb{R}^{C \times N \times D}$ to generate two masking views. Firstly, we randomly select 50% of the patch embeddings in each channel and mask them with zeros to generate the first masking view $\mathcal{X}_{m1}$. Then, we mask the patches that were not masked in the previous step, while restoring the previously masked patches to their original values, to generate the second masking view $\mathcal{X}_{m2}$. Let $\odot$ denote the element-wise product and $\mathcal{M}$ represent the masking matrix, where 0 denotes positions without a mask, and 1 denotes

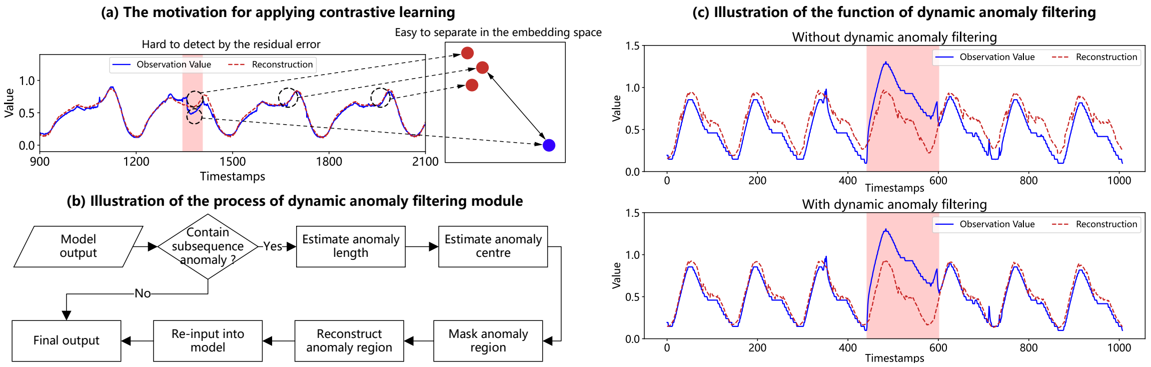


Figure 5: The explanations of the submodules. The data in (a) is from ASD 10, while the data in (c) is from Satellite 2.

the masked positions. The masking process can be described as follows:

$$\begin{aligned} \mathcal{X}_{m1} &= \mathcal{X}_d \odot \mathcal{M} \\ \mathcal{X}_{m2} &= \mathcal{X}_d \odot (1 - \mathcal{M}) \end{aligned} \tag{3}$$

3.3 MLP-Mixer backbone

The two masking views $\mathcal{X}_{m1}$ and $\mathcal{X}_{m2}$ are then input to the MLP-Mixer backbone shown in Figure 4 (b) to recover the masked embeddings. The MLP-Mixer backbone is modified from [11, 30]. We retain the inter-patch mixer and intra-patch mixer modules while removing the inter-channel mixer module to avoid over-smoothing across channels. The **Inter-patch Mixer block** employs a channel shared MLP with weights $W \in \mathbb{R}^{N\times N}$ to capture the correlation between different patches, thereby facilitating the learning of global temporal dependencies among patches. The **Intra-patch Mixer** block takes a channel shared MLP with weights $W \in \mathbb{R}^{D\times D}$ to mix the hidden features of time steps within a patch, thereby enabling the learning of local temporal dependencies in a patch.

The structure of each channel shared MLP is depicted in Figure 4 (c), which consists of two fully connected layers, a Gelu nonlinear layer, and two Dropout layers. A Gated Attention layer is added after each MLP component to filter out noise in the time series. The Gated Attention acts like a simple gating function that probabilistically upscales dominant features and downscales unimportant features based on its feature values. The computation process of the Gated Attention is as follows:

$$\begin{aligned} \mathcal{W} &= \mathrm{softmax}\left(\mathrm{W}\mathcal{X}_m\right), \mathrm{W} \in \mathbb{R}^{D\times D} \\ \mathcal{X}'_m &= \mathcal{W} \odot \mathcal{X}_m, \mathcal{W} \in \mathbb{R}^{C\times N\times D} \end{aligned} \tag{4}$$

where $\mathcal{W}$ represents the learned attention weights, and the output is the element-wise product of the attention weights and the original embeddings.

3.4 Contrastive Learning

As shown in Figure 5 (a), some anomalies in the time series have similar amplitudes to normal patterns but exhibit different shapes. Therefore, it isn't easy to detect them solely based on residuals. To address this issue, we design a contrastive learning module that clusters the embeddings of time series patches with similar shapes in the latent vector space. Consequently, these subtle anomalies can be detected by comparing the embedding of the observed values with the embedding of the reconstructed values. However, constructing proper positive and negative samples is a challenging task for time series contrastive learning [60].

In our work, as shown in Figure 4 (a), we consider the embedding obtained directly from the raw patch and the embedding reconstructed by the MLP-Mixer as positive samples, while the embedding between different patches as negative samples. We denote the embeddings recovered from the masking views $\mathcal{X}_{m1}$ and $\mathcal{X}_{m2}$ as $\mathcal{X}_{r1}$ and $\mathcal{X}_{r2}$ respectively. The embedding $\mathcal{X}_{r1}$ and $\mathcal{X}_{r2}$ are then merged in chronological order to form the final reconstructed embedding $\mathcal{X}_r$. Let $\mathcal{X}_d$ denotes the embedding directly obtained from the raw input. The embedding of the i-th patch in the c-th channel is represented as $\mathcal{X}_d^{(c,i)}$. Then, the contrastive loss for $\mathcal{X}_d^{(c,i)}$ can be written as:

$$\mathcal{L}_{contrast}^{(c,i)} = -\log \frac{\exp(\mathcal{X}_d^{(c,i)} \circ \mathcal{X}_r^{(c,i)})}{\sum_{i'\in I} \mathbb{I}_{[i'\neq i]} \exp(\mathcal{X}_d^{(c,i)} \circ \mathcal{X}_r^{(c,i')})} \tag{5}$$

where the cosine similarity $\circ$ is used as the distance metric, $\mathbb{I}$ is the indicator function and I represents the set of patches within the channel c. Then, the overall contrastive loss can be written as:

$$\mathcal{L}_{contrast} = \frac{1}{N * C} \sum_{i=1}^{N} \sum_{c=1}^{C} \left(\mathcal{L}_{contrast}^{(c,i)}\right) \tag{6}$$

Our proposed contrastive learning module has the following advantages:

- It avoids using inappropriate positive samples constructed through traditional data augmentation methods [22, 51].
- It pushes the MLP-Mixer module to reconstruct the masked areas' embeddings accurately.
- It enables the patch embeddings to be aware of the shape information of the patches, which is beneficial for downstream reconstruction tasks and can also serve as a criterion for detecting anomalies.

3.5 Training and Anomaly Scoring

The recovered embeddings $\mathcal{X}_r$ in Figure 4 (a) are subsequently mapped back to the time series patches $\mathcal{X}_{p'}$. The projection back layer is a channel shared MLP with weights $W \in \mathbb{R}^{D\times P}$. The reconstruction loss is the Mean Squared Error (MSE) between the original patches and the reconstructed patches:

$$\mathcal{L}_{recon} = \frac{1}{T * C}\left(\mathcal{X}_p - \mathcal{X}_{p'}\right)^2 \tag{7}$$

The overall training loss is the weighted sum of the reconstruction loss and the contrastive loss:

$$\mathcal{L}_{final} = \mathcal{L}_{recon} + \lambda \cdot \mathcal{L}_{contrast} \tag{8}$$

During the inference stage, the anomaly score is composed of two components: (1) the residual error between the original samples and the reconstructed samples; and (2) the cosine distance between the original embeddings and the recovered embeddings. Considering the i-th patch in the c-th channel, the anomaly score is written as follows:

$$AS\left(\mathcal{X}_p^{(c,i)}\right) = \left|\mathcal{X}_p^{(c,i)} - \mathcal{X}_{p'}^{(c,i)}\right| + \left(1 - \mathcal{X}_d^{(c,i)} \circ \mathcal{X}_r^{(c,i)}\right) \tag{9}$$

It is worth noting that the residual error $\left|\mathcal{X}_p^{(c,i)} - \mathcal{X}_{p'}^{(c,i)}\right| \in \mathbb{R}^P$, while $\left(\mathcal{X}_d^{(c,i)} \circ \mathcal{X}_r^{(c,i)}\right) \in \mathbb{R}$. To ensure dimensional consistency, we consider the points within a patch to share the same cosine distance and repeat the cosine distance value for P times, where P represents patch length. In addition, we propagate the input window forward 5 times and take the average of the anomaly scores to reduce the uncertainty caused by random masking.

3.6 Dynamic Anomaly Filtering

Figure 5 (c) shows that the anomaly region has a negative impact on the reconstruction of surrounding normal regions, resulting in false alarms for the neighboring areas. To solve this problem, we add a dynamic anomaly filtering module during the inference stage. The overall workflow of this module is illustrated in Figure 5 (b). For a given input window, we first calculate the anomaly score of this window using the process described in the previous sections and then employ the 3σ principle to determine whether the window contains subsequence anomalies or not:

$$y = \mathbb{I}\left(AS\left(\mathcal{X}^c\right) \geq \mu_{train} + 3\sigma_{train}\right) \tag{10}$$

where μ_{train} and σ_{train} represent the mean value and standard deviation of the anomaly scores calculated from the training set windows. If $y = 1$, we assume the window contains anomalies and prepare for further processing.

We continue to apply the 3σ principle in the window above to search for patches that may contain anomalies. Let I_a represents the set of patches in window $\mathcal{X}^c$ that potentially contains anomalies, and I_p represents the set of their positions in this window. The estimated anomalous region p_a can be expressed as follows:

$$\begin{aligned} p_c &= \text{median}\left(I_p\right) \\ p_a &= \left[p_c - \frac{P}{2}\left|I_a\right|, p_c + \frac{P}{2}\left|I_a\right|\right] \end{aligned} \tag{11}$$

where p_c denotes the estimated center of the anomalous region, and $|I_a|$ represents the number of patches in the set I_a.

After obtaining the anomalous region p_a, we mask this region and utilize the proposed model MMA to recover the masked parts. Our model can recover the normal behaviors of the anomalous region, thus avoiding any negative impact on the reconstruction of neighboring normal areas. Next, we input the window without the anomalous portions back into the model and calculate the anomaly score for this window. As shown in Figure 5 (c), when the dynamic anomaly filtering module is incorporated, our model performs well in reconstructing the normal areas surrounding the anomalies.

4 EXPERIMENTS AND ANALYSIS

In this section, we first introduce the benchmark datasets, baseline methods, and details of the models' implementation. Then, we conduct experiments to answer the following research questions:

- **RQ1. Effectiveness.** How does MMA perform on time series anomaly detection datasets compared to other state-of-the-art approaches?
- **RQ2. Explainability.** How accurately does MMA recover the potential normal patterns for anomalous regions?
- **RQ3. Robustness.** Can MMA maintain stable performance in the presence of various kinds of training set contaminations?
- **RQ4. Ablation.** How much does each component in MMA contribute to the overall performance?
- **RQ5. Visualization.** Can MMA provide detection results and interpretable outcomes that align with human institutions?
- **RQ6. Insights.** What insights can be gained from our method?

4.1 Datasets and Baselines

Benchmark Datasets. The quality of benchmark datasets is an important issue in time series anomaly detection. Through a comprehensive review of recent dataset evaluation papers [16, 41, 45, 52, 55, 56], and a meticulous examination of the visualized datasets, we choose a set of relatively high-quality datasets from diverse domains as benchmarks. These datasets include the univariate time series anomaly detection dataset: the UCR Time Series Anomaly Archive [56], and the multivariate time series anomaly detection datasets: ASD [32] and NeurIPS-TS-Synthetic (Synthetic) [26]. Moreover, we provide a real-world dataset collected from a satellite. The statistics of the datasets are summarized in Table 1.

The UCR Time Series Anomaly Archive (UCR) comprises datasets from various domains, including healthcare, sports, industry, robotics, etc. The ASD dataset is collected from internet server machines. The NeurIPS-TS-Synthetic dataset utilizes the sinusoidal wave as the base shapelet and injects multiple predefined anomalies into it. These datasets are less affected by issues such as unrealistic anomaly density, position bias, distribution shift, and mislabeled ground truth, as mentioned in [55, 56]. Therefore, they are more suitable for evaluation. To support our claim, we provide visualizations of these datasets [1].

Baseline Methods. We compare the proposed MMA with 16 state-of-the-art baselines [2], including 12 deep learning-based models and 4 non-deep learning models.

Deep learning-based models. We classify the deep learning-based baselines into four categories: (1) The reconstruction-based methods (e.g., NSPR [25], NormFAAE [59], MAUT [42], TranAD

[1] https://drive.google.com/drive/folders/1ZmOJ-lAN0FfgDr6unwsU2LLubdQovv1x?usp=sharing

[2] We do not include popular but flawed baselines, including AnomalyTransformer and DCdetector. Details are shown at https://github.com/thuml/Anomaly-Transformer/issues/65 and https://github.com/DAMO-DI-ML/KDD2023-DCdetector/issues/35

Table 1: Datasets used in this study before preprocessing.

Datasets	Entities	Dims	Train	Test	Anomaly
UCR	250	1	2238349	6143541	0.6%
ASD	12	19	102331	51840	4.61%
Satellite	3	9	11862	3793	7.22%
Synthetic	1	5	10000	10000	13.06%

[54], FGANomaly [10] and USAD [1]) build models to understand the normal behavior of the time series and detect anomalies according to the reconstruction errors; (2) The prediction-based models (e.g., CAD [50], GDN [9] and MTAD-GAT [64]) learn to predict the future values of the time series and identify anomalies based on the prediction errors; (3) The imputation-based methods (e.g., DiffAD [57] and ImDiffusion [7]) learn to impute the missing values based on observed values and find anomalies using the imputation errors; (4) The density-based methods (e.g., MTGFLOW [66]) evaluate the density of the time series and treat anomalies as regions with low density.

Non-deep learning models. Matrix Profile [58] identify subsequences with large distances to their nearest neighbors as anomalies (called time series discords). DAMP [36] is an improved version of the original matrix profile designed for online detection and ultra-fast arriving time streams. SAND [5] is a cluster-based model that detects anomalies based on the distance to a model representing normal behaviors. Series2Graph [4] aims to detect subsequence anomalies based on a graph representation of a low-dimensionality embedding of raw time series.

4.2 Implementation Details.

There are five hyperparameters to be determined for MMA, including the input window size T, the patch length P, the hidden dimension D, the MLP-Mixer layers, and the weight λ in $\mathcal{L}_{final}$. In all experiments, we set the MLP-Mixer layers to 3, the hidden dimension D to 64, and the weight λ to 0.005. For the ASD, Satellite, and NeurIPS-TS-Synthetic datasets, we set the input window size T to 1024 and the patch length P to 16. Since the UCR dataset contains entities with different sampling frequencies, we first apply the Fast Fourier Transform to identify the primary period of each entity. Then, we set the input window size to 8 times the primary period and the patch length to the period divided by 8. We use the Adam optimizer with a learning rate of 0.001. We train our model for 200 epochs on the UCR dataset and 100 epochs on other datasets.

Regarding the baselines, we re-implement all the deep learning-based models based on their official open-source codes and integrate them into our unified pipeline. All the non-deep learning models are implemented based on the TSB-UAD library [41]. The hyperparameters of the baseline models are set according to the information provided in their original papers.

4.3 RQ1. Effectiveness

Evaluation Metrics. Since there is currently no universally accepted fair and rigorous evaluation metric for time series anomaly detection, we adopt some commonly used metrics and recently proposed metrics specifically designed for time series anomaly detection. In light of the severe flaws in the widely used point adjustment strategy [14, 15, 24, 55], all our results are reported **without point adjustment.** Due to the significant class distribution skew in the anomaly detection datasets, we opt to use AUC-PR (Area Under the Precision-Recall Curve) instead of AUC-ROC (Area Under the Receiver Operating Characteristic Curve) [8, 29, 40]. Here is a brief introduction to the evaluation metrics:

The F1-Raw and AUC-PR are the most widely used evaluation metrics that calculate the point-wise F1 value and the area under the precision-recall curve. The R-AUC-PR [40] is recently proposed to address the inconsistent labeling problem in the evaluation of time series anomaly detection. It adds a buffer region at the boundary of anomalies, thereby giving some credit to the high anomaly score in the vicinity of the anomaly boundary. VUS-PR [40] further solves the issue of buffer length determination in R-AUC-PR, enhancing the flexibility and robustness of R-AUC-PR usage. F1-(PA%K) [24] optimizes original point-wise evaluation by considering the balance between conventional F1-Raw measurements and the ill-posed point-adjustment strategy. To mitigate the dependency on the selection of the K value, we adopt the approach suggested in [24], where the F1-(PA%K) value is calculated as the area under the K-F1 curve. F1-Affiliation [14] tackles the problems of unawareness of temporal adjacency and unawareness of anomaly durations in current evaluation methods by identifying the local affiliation of predicted anomalies to their closest ground truths.

Results Analysis. The comparison results against SOTA models are presented in Table 2 and Table 3. Since DAMP, SAND, Series2Graph, and Matrix Profile only accept univariate time series as input, we do not evaluate them on the multivariate time series datasets. Due to the large size of the UCR dataset and the high cost of training and testing, we only consider the top 8 performing deep learning methods in Table 2 for evaluation on the UCR dataset. Summary from the tables, we have the following observations:

First, several studies indicate that an individual anomaly detection model typically performs well only on a particular domain of datasets [45, 52]. This declaration aligns with our experimental results, such as CAD demonstrating excellent performance on the ASD dataset, MTAD-GAT performing well on the NeurIPS-TS-Synthetic dataset, and FGANomaly excelling on the Satellite dataset. However, our model achieves the best results across all datasets, thereby validating the efficacy of our proposed methodology in handling time series with diverse patterns.

Second, our proposed method outperforms existing deep learning models by a significant margin. On the multivariate time series datasets, we report an average absolute improvement of 18.51% in AUC-PR, 14.52% in R-AUC-PR, 14.70% in VUS-PR, 10.12% in F1-(PA%K) and 2.12% in F1-Aff. The UCR Time Series Anomaly Archive comprises subsequence anomalies that are challenging to detect. Existing deep learning models generally perform poorly on this dataset [36, 41, 43]. Nevertheless, our work is the first framework to rely solely on deep learning models and achieve superior performance on the UCR dataset compared to state-of-the-art non-deep learning methods.

Third, some recently proposed deep learning models with intricate architectures, such as DiffAD, ImDiffusion, and MTGFLOW,

Table 2: Overall results on multivariate datasets. All results are in %, the best ones are in Bold, and the second ones are underlined. It is worth noting that ImDiffusion takes a voting anomaly score calculating strategy, thus failing to evaluate using the f1-affiliation score. The bottom right section of the table presents the average values across the three datasets. Improvement denotes the difference between the value of our model and the highest baseline value.

Method	ASD						Satellite					
	AUC-PR	R-AUC-PR	VUS-PR	F1-Raw	F1-(PA%K)	F1-Aff	AUC-PR	R-AUC-PR	VUS-PR	F1-Raw	F1-(PA%K)	F1-Aff
NSPR	42.83	36.65	36.20	50.68	54.90	78.08	68.68	68.12	67.85	73.26	85.75	89.77
CAD	44.26	46.16	45.37	45.71	57.91	85.81	73.23	72.31	71.95	74.77	90.25	98.52
DiffAD	6.17	10.07	10.01	11.36	20.83	71.69	17.94	27.77	26.11	27.72	48.50	80.92
ImDiffusion	19.41	22.10	21.74	19.28	29.23	N/A	13.66	19.26	19.39	22.12	13.48	N/A
NormFAAE	28.58	27.99	27.80	36.20	47.27	81.91	60.00	59.01	58.44	62.15	80.39	97.50
MTGFLOW	5.79	8.65	8.49	12.38	17.45	69.55	20.92	32.81	32.82	28.68	42.67	80.47
MAUT	39.52	40.00	39.58	45.33	56.63	83.21	55.40	57.65	57.05	64.59	81.93	89.03
TranAD	36.42	38.78	38.63	41.95	55.25	83.46	81.32	74.65	75.40	78.30	91.86	**99.55**
FGANomaly	35.98	35.41	35.36	40.29	52.44	82.34	86.11	74.50	76.42	81.18	94.24	98.62
GDN	31.07	28.24	28.31	38.15	51.61	79.92	62.97	61.02	61.20	70.30	86.37	92.17
USAD	33.21	27.47	27.17	36.91	45.96	77.58	69.26	67.24	66.24	74.32	85.47	88.94
MTAD-GAT	35.99	36.63	36.43	40.81	54.74	81.89	54.16	54.54	53.92	64.98	81.04	86.53
MMA(ours)	**58.08**	**58.29**	**57.67**	**58.23**	**69.59**	**89.71**	**92.46**	**87.51**	**87.72**	**87.66**	**96.89**	99.50
Improvement	+13.82	+12.13	+12.3	+7.55	+11.68	+3.9	+6.35	+12.86	+11.3	+6.48	+2.65	-0.05

Method	Synthetic						Average					
	AUC-PR	R-AUC-PR	VUS-PR	F1-Raw	F1-(PA%K)	F1-Aff	AUC-PR	R-AUC-PR	VUS-PR	F1-Raw	F1-(PA%K)	F1-Aff
NSPR	17.93	29.17	29.17	31.84	47.28	69.52	43.15	44.65	44.41	51.93	62.64	79.12
CAD	66.21	74.73	74.59	69.55	83.77	92.23	61.23	64.40	63.97	63.34	77.31	92.19
DiffAD	15.15	26.59	26.81	24.21	43.13	67.93	13.09	21.48	20.98	21.10	37.49	73.51
ImDiffusion	31.53	44.29	43.60	23.26	23.91	N/A	21.53	28.55	28.24	21.55	22.21	N/A
NormFAAE	39.48	45.21	44.40	37.71	50.75	77.96	42.69	44.07	43.55	45.35	59.47	85.79
MTGFLOW	11.08	21.02	20.93	24.18	34.92	70.41	12.60	20.83	20.75	21.75	31.68	73.48
MAUT	67.63	67.90	66.15	63.28	78.77	87.40	54.18	55.18	54.26	57.73	72.44	86.55
TranAD	13.07	23.81	22.99	23.12	36.09	69.69	43.60	45.75	45.67	47.79	61.07	84.23
FGANomaly	18.30	27.80	27.90	28.77	44.94	69.09	46.80	45.90	46.56	50.08	63.87	83.35
GDN	15.03	22.18	23.07	23.56	39.38	74.99	36.36	37.15	37.53	44.00	59.12	82.36
USAD	14.42	20.94	21.16	23.11	33.79	69.78	38.96	38.55	38.19	44.78	55.07	78.77
MTAD-GAT	76.56	76.62	76.76	77.39	88.53	93.65	55.57	55.93	55.70	61.06	74.77	87.36
MMA(ours)	**88.67**	**90.96**	**90.62**	**88.93**	**95.82**	**93.72**	**79.74**	**78.92**	**78.67**	**78.27**	**87.43**	**94.31**
Improvement	+12.11	+14.34	+13.86	+11.54	+7.29	+0.07	+18.51	+14.52	+14.70	+14.93	+10.12	+2.12

Table 3: Average results on the UCR Time Series Anomaly Archive. Improve(deep) represents the improvement compared to the SOTA deep learning models, while Improve (non-deep) represents the improvement compared to the SOTA non-deep learning models.

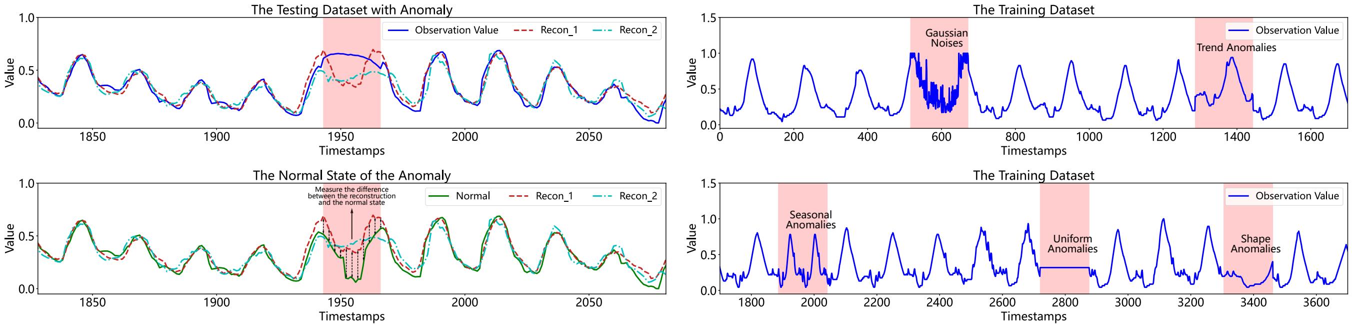
Method	UCR					
	AUC-PR	R-AUC-PR	VUS-PR	F1-Raw	F1-(PA%K)	F1-Aff
NSPR	8.37	8.28	8.27	12.07	16.75	72.41
CAD	23.90	23.25	23.03	30.03	33.88	79.37
NormFAAE	3.12	3.76	3.66	6.39	11.59	70.26
MAUT	23.84	23.94	23.65	30.20	35.45	80.34
TranAD	7.78	7.94	7.89	11.23	12.81	72.85
FGANomaly	8.65	8.15	8.10	12.79	13.34	73.28
USAD	8.33	8.37	8.34	12.16	15.21	73.96
MTAD-GAT	26.00	26.18	25.66	38.04	36.99	78.05
DAMP	18.99	25.09	24.18	29.31	29.95	84.88
SAND	34.72	34.23	33.78	39.43	47.54	84.07
Series2Graph	4.36	8.18	7.87	8.70	11.94	78.42
Matrix Profile	18.50	25.37	24.13	28.00	31.49	80.58
MMA(ours)	**36.54**	**34.92**	**34.50**	**42.53**	**49.38**	**85.41**
Improve(deep)	+10.54	+8.74	+8.84	+4.49	+12.39	+5.07
Improve(non-deep)	+1.82	+0.69	+0.72	+3.10	+1.84	+0.53

perform poorly on reliable datasets when rigorous evaluation methods are employed. This causes the "Creating the Illusion of Progress" problem as suggested in [56]. Additionally, we find that the F1-Affiliation metric tends to overestimate the performance of models. Even when models perform poorly on other metrics, F1-Affiliation consistently assigns high scores to them. Future researchers should be cautious in using the F1-Affiliation metric.

4.4 RQ2. Explainability

Experiment Setting. As shown in Figure 6 (a), we substitute the original normal region with a subsequence anomaly (the anomaly is clipped from anomalies present in the testing set itself). Subsequently, we compare the disparity between the reconstructed and original normal values. A smaller disparity indicates a stronger ability of the model to restore the normal state within the anomalous portion. In Figure 6 (a), the explainability of Reon_1 is better than that of Recon_2, as it accurately reconstructs the "peak-valley-peak" pattern in the normal state. We consider the following three metrics to measure the distance between reconstructed values and original normal values:

- **Euclidean distance.** The Euclidean distance is the most commonly used similarity measurement for time series. It



(a) Illustration of the explainability measurement

(b) Illustration of the simulated anomalies injected in the robustness verification

Figure 6: Illusion of the experimental settings for explainability and robustness. The data in (a) is from UCR 006, while the data in (b) is from Satellite 1.

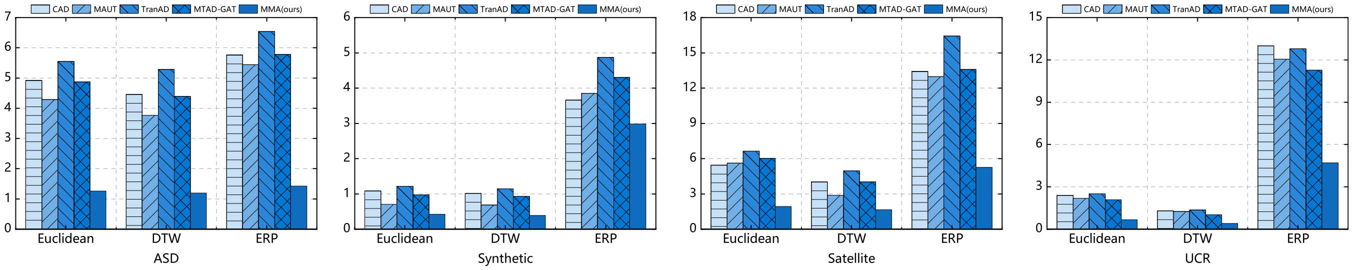


Figure 7: The explainability evaluation results on top 5 performing deep learning methods.

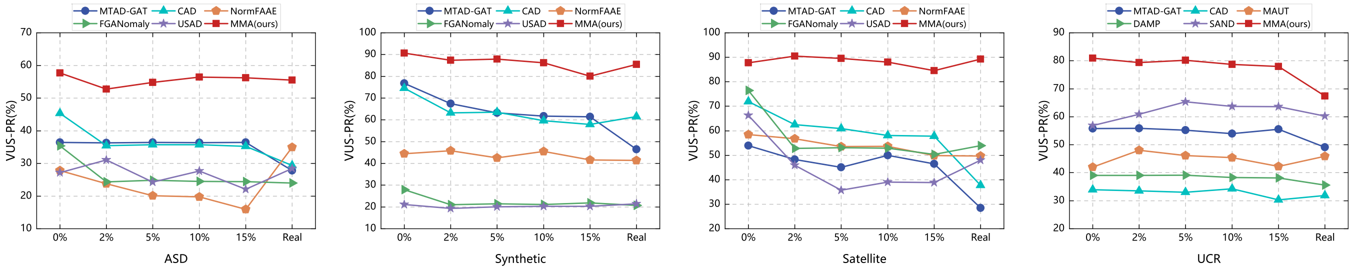


Figure 8: The robustness evaluation results. Due to the extensive size of the UCR dataset, evaluating it under all contamination scenarios would incur substantial time costs. Therefore, we only assess the model's performance on entities 006, 025, 048, 141, 145, 160, and 173.

is calculated by summing the squared differences at corresponding time points of two time series. However, it is sensitive to temporal shifts in time series and has difficulty measuring the shape similarity between time series.

- **Dynamic Time Warping (DTW).** The core idea of Dynamic Time Warping (DTW) is to find the optimal alignment between time series to minimize their cumulative distance [19]. DTW is robust to scaling and shifting of time series and can measure the shape similarity.
- **Edit Distance with Real Penalty (ERP).** ERP [6] calculates the minimum number of basic operations required to transform one time series into another. It can mitigate the impact of shifting and extreme values in time series on similarity measurement.

Results Analysis. The explainability evaluation results are presented in Figure 7. Across all four datasets, the reconstruction disparity of our proposed model is significantly smaller than that of other models. Our model's reconstruction error is below 2 for the Euclidean and DTW metrics and below 6 for the ERP metric on all datasets. These results demonstrate our model's strong capability of restoring the normal patterns within the anomalous regions. By comparing the observed anomalous values with the reconstructed normal patterns, operators can confirm why a certain region is detected as an anomaly.

It is worth noting that MAUT tries to memorize the normal patterns in the training set by incorporating a memory network, resulting in better explainability than other baseline models. However, the discrepancy between MAUT's reconstruction and the

Table 4: Performance of MMA and its variants in terms of VUS-PR and F1-(PA%K).

Method	ASD		Synthetic		Satellite		UCR		Average (Δ)	
	VUS-PR	F1-(PA%K)	VUS-PR	F1-(PA%K)	VUS-PR	F1-(PA%K)	VUS-PR	F1-(PA%K)	VUS-PR	F1-(PA%K)
MMA	**57.67**	**69.59**	**90.62**	**95.82**	**87.72**	96.89	34.50	**49.38**	**67.62**	**77.92**
MMA_GRU	41.38	57.57	84.63	94.25	67.10	87.12	34.07	46.39	56.80 (-10.82)	71.33 (-6.59)
MMA_Transformer	34.65	53.99	67.03	80.47	65.25	87.53	24.11	32.28	47.76 (-19.86)	63.57 (-14.35)
MMA w/o CL	42.38	61.93	77.01	88.01	85.38	**97.05**	30.39	41.87	58.79 (-8.83)	72.22 (-5.70)
MMA w/o DAF	54.65	65.50	88.99	94.05	80.96	93.73	**36.07**	40.70	65.17 (-2.45)	73.50 (-4.42)
MMA w/o CL&DAF	42.39	59.01	73.89	84.47	81.57	96.99	27.92	39.67	56.44 (-11.48)	70.04 (-7.88)

normal values is still much larger than that of our model. This is because the embedding of anomalous values significantly differs from the normal values, making it difficult for the memory network to determine which possible normal pattern the anomalous region belongs to. In contrast, our model utilizes a masked-autoencoder based detection approach and can effectively infer the normal patterns within the anomalous regions with supplementary information on surrounding normal patterns.

4.5 RQ3. Robustness

Experiment Setting. We pollute the training dataset with various kinds of simulated anomalies presented in Figure 6 (b), including Gaussian noise, trend anomalies, seasonal anomalies, uniform anomalies, and shape anomalies. We gradually increase the ratio of injected simulated from 5% to 20% to observe the performance of the models under different contamination levels. Real anomalies extracted from the testing set are also introduced to evaluate the models' performance under the presence of real anomaly contamination.

In the comparative experiments conducted on the multivariate time series datasets ASD, Synthetic, and Satellite, three models with additional robustness designs, including NormFAAE, FGANomaly, and USAD, along with 2 top performing models, MTAD-GAT and CAD, are chosen for comparison. For the univariate time series UCR dataset, since NormFAAE, FGANomaly, and USAD fail to detect anomalies within it, only the top 6 performing models are included.

Results Analysis. It can be seen in Figure 8 that most of the baseline models experience a sharp decline in performance when simulated anomalies are injected. However, as the proportion of injected anomalies gradually increases, the performance of the baseline models becomes stable. In addition, the baseline models generally experience the most significant drop in performance when real anomalies are added. Despite incorporating specialized robustness designs, NormFAAE, FGANomaly, and USAD still face severe performance losses when trained on contaminated datasets. In comparison, our proposed model shows minimal performance degradation when simulated anomalies are introduced. As the proportion of simulated anomalies increases from 2% to 15%, our model only experiences a decrease in VUS-PR within 10%. Moreover, our model maintains stable performance even in the presence of real anomalies. In the case of real anomaly pollution, our model's decrease in VUS-PR remains within 5% (except for the UCR dataset).

4.6 RQ4. Ablation

To study the effectiveness of each component in our proposed model, we exclude every major component to observe how it affects the performance in terms of the VUS-PR scores and F1-(PA%K) scores for each dataset. We first consider the MMA with different backbones. Then, we investigate the importance of our proposed modules. The five variant models are designed as follows:

- *MMA_GRU:* We replace the MLP-Mixer backbone in MMA with Gate Recurrent Units (GRU) and utilize GRU to model the temporal relationships between patches. The overall structure of MMA_GRU resembles the encoding part of SegRNN [35].
- *MMA_Transformer:* We replace the MLP-Mixer backbone in MMA with a Transformer backbone, utilizing the attention mechanism to model temporal relationships between patches. The overall structure of MMA_Transformer resembles that of PatchTST [39].
- *MMA w/o CL:* We remove the Contrastive Learning (CL) module in MMA and the anomaly score calculation is solely based on the residual error.
- *MMA w/o DAF:* We remove the Dynamic Anomaly Filtering (DAF) module, disregarding the impact of anomalous regions on the reconstruction of neighboring normal regions.
- *MMA w/o CL&DAF:* We remove both the Contrastive Learning (CL) module and the Dynamic Anomaly Filtering (DAF) module.

The results of the ablation study are summarized in Table 4. Regarding the backbones, the original model with MLP-Mixer performs the best, followed by the model utilizing GRU, while the model using Transformer performs the worst. This finding highlights the effectiveness of MLP-Mixer in modeling temporal dependencies within time series, indicating that researchers should pay more attention to linear models in the time series anomaly detection field. Although the patch design helps address long-term dependency issues in GRU, it persists when the window size is large. The self-attention mechanism in Transformer is permutation invariant, causing inevitable temporal information loss [61]. It is found that MMA_Transformer is only capable of reconstructing the trends in the time series but fails to reconstruct the fine-grained patterns on the Synthetic dataset and UCR dataset.

Regarding the submodules, MMA achieves optimal performance when all submodules are incorporated. Removing any individual submodule results in a decline in model performance, with

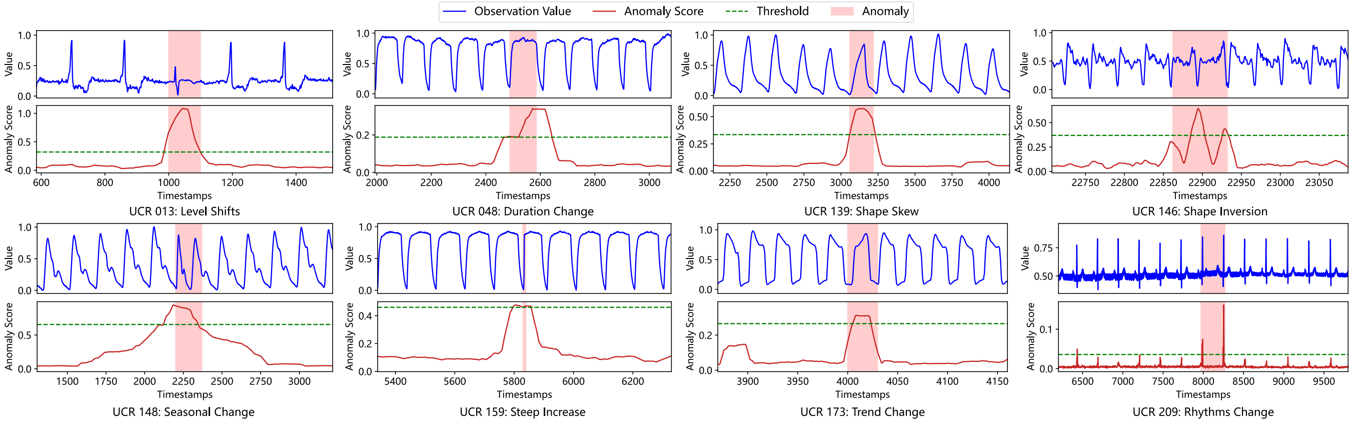
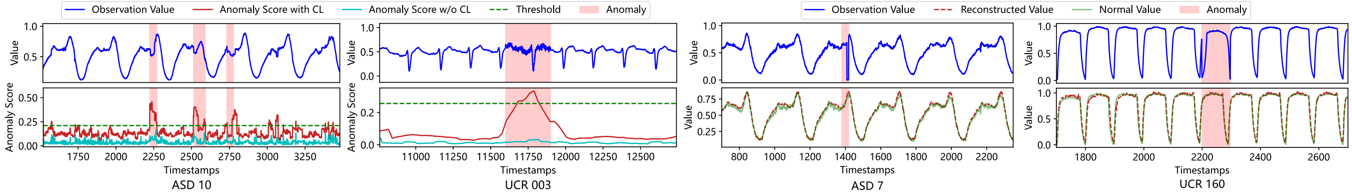


Figure 9: Visualization of the various types of anomalies detected by MMA.



(a) Visualization of contrastive learning for detecting subtle anomalies

(b) Visualization of the values reconstructed by MMA

Figure 10: Visualization of the effect of contrastive learning and the model explainability.

the most severe decrease observed when both submodules are removed. The removal of the contrastive learning module causes an average performance loss of 8.83% in VUS-PR and 5.70% in F1-(PA%K). The contrastive learning module generally increases anomaly scores in anomalous regions, making it easier to distinguish between normal and abnormal time points. Additionally, it helps detect subtle anomalies that are difficult to identify solely based on residual errors. Removing the dynamic anomaly filtering module leads to an average performance drop of 2.45% in VUS-PR and 4.42% in F1-(PA%K), indicating that mitigating the influence of anomalous regions on the reconstruction of surrounding normal parts can help reduce false alarms.

4.7 RQ5. Visualization

In Figure 9, we showcase the capability of MMA in detecting various types of anomalies, regardless of their duration and shape. Some of these anomalies are easily overlooked by other models, such as the duration change in UCR 048 and the abrupt slope variation in UCR 159. Some anomalies are challenging even for human experts to identify, such as the rhythmic changes in UCR 209, where a short beat should follow a long beat. However, our model consistently assigns high anomaly scores to these anomalous regions, demonstrating the effectiveness of our proposed method.

Figure 10 (a) demonstrates the function of contrastive learning in aiding the detection of subtle anomalies. As these anomalies exhibit similar amplitude to potential normal values, "Anomaly Score w/o CL" fails to detect relying solely on the residual errors. However, "Anomaly Score with CL" can detect them by differentiating them in the embedding space. Figure 10 (b) shows that MMA generates reconstruction values in the anomalous region very close to the corresponding normal values. By comparing the observation values with the reconstructed values, we can infer that the anomaly in ASD 7 is caused by transient value loss and the anomaly in UCR 160 is caused by shape deformation. This provides further evidence and insights into why these regions are identified as anomalies.

4.8 RQ6. Insights

Influence of the Input Windows Size. The impact of the input window size on the model's performance is illustrated in Figure 11. As shown in the figure, increasing the input window significantly enhances the model's performance. This observation highlights the importance of considering longer contextual information for anomaly detection, particularly for anomalies with longer durations. We believe the large input window design is a promising direction for future deep learning-based anomaly detection methods, as only with large windows can deep learning models fully leverage their ability to capture long-term dependencies in time series.

Learned Patch Embeddings. We calculate the pairwise similarity between all patch embeddings within a window and get a 64×64 heat map, where a cell in the i-th column and j-th row represents the cosine similarity between the i-th and j-th patch embeddings. The heat maps in Figure 12 exhibit a periodicity that aligns with the curves, and the similarity between patch embeddings is highest when the patches are spaced at integer multiples

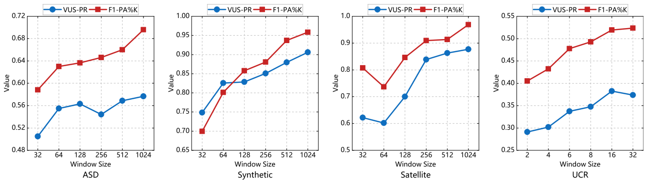
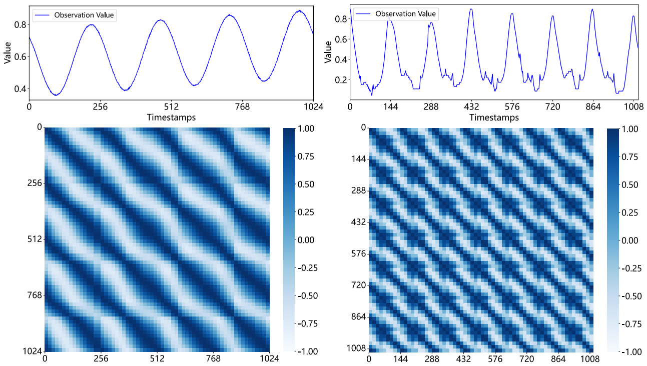


Figure 11: Influence of window size on the performance of MMA. Due to the distinct window sizes employed for different entities in the UCR dataset, the horizontal axis of the UCR dataset represents multiples of the patch length.



(a) The curve with a period of approximately 256 and its corresponding similarity map of embeddings among different patches.

(b) The curve with a period of approximately 144 and its corresponding similarity map of embeddings among different patches.

Figure 12: Cosine similarity of embeddings among different patches. The horizontal and vertical coordinates represent absolute positions within the time series, indicating that the coordinate values on both axes are obtained by multiplying the patch index by the patch length (x16).

of the period. This demonstrates that our contrastive learning module enables the patch embeddings to capture the shape information of the patches.

Efficiency. We compute the FLOPs, parameters, and inference time for all deep learning models on the NeurIPS-TS-Synthetic dataset. To ensure a fair comparison, all models are tested using a batch size of 64, the window size that yields optimal performance, and a stride size equal to the window size. All the experiments are conducted on the same NVIDIA RTX 3090 GPU.

The results are summarized in Table 5. Given that MMA employs a significantly larger input window, its computational cost and parameter load surpass those of other models. Nevertheless, a large input window also allows processing more data points in a single forward pass, giving MMA a clear advantage in overall inference time. MMA can process 10,000 time points in just 0.101 seconds, second only to NormFAAE, MAUT, and USAD. MMA is an imputation-based method and shows significant efficiency improvements compared to other imputation-based models such as DiffAD and Imdiffusion. MMA using the MLP-Mixer backbone is also found to be more lightweight than MMA using GRU or Transformer backbones. This is because the MLP-Mixer architecture does not require recurrent or attention mechanisms, reducing the overall number of parameters. Furthermore, we observe that increasing the window size significantly amplifies the computational load, as MMA (1024) with a window size of 1024 has 14 times the FLOPs compared to MMA (128) with a window size of 128. Therefore, our future research will focus on increasing the window size while maintaining a low computational load.

Table 5: Efficiency Comparison Results on the NeurIPS-TS-Synthetic Dataset.

Method	FLOPs (unit: M)	Params (unit: K)	Inference Time (unit: s)	Method	FLOPs (unit: M)	Params (unit: K)	Inference Time (unit: s)
NSPR	0.97	21.30	0.218	MAUT	4.20	59.28	0.009
CAD	0.29	290.40	0.333	TranAD	0.007	1.09	0.391
DiffAD	9700	38830	22.224	FGANomaly	87.50	83.97	0.297
ImDiffusion	106.66	312.83	7.852	GDN	0.024	4.74	0.497
NormFAAE	19.63	151.68	0.009	USAD	0.032	19.66	0.012
MTGFLOW	4.82	36.66	0.245	MTAD-GAT	23.18	274.90	0.654
MMA(GRU)	490.05	760.46	0.120	MMA(Transformer)	285.41	449.30	0.093
MMA(128)	14.84	166.84	0.159	MMA(1024)	207.49	325.45	0.101

5 CONCLUSION

This paper presents a novel algorithm named MMA that combines the MLP-Mixer backbone with Masked Autoencoders to allow a significantly larger input window for time series anomaly detection. Benefitting from longer contextual information, MMA can detect anomalies with longer durations. In addition, MMA can reconstruct the possible normal patterns in the anomalous regions, providing high explainability. Furthermore, MMA is robust to various kinds of training set pollutions.

In future research, we will focus on the lightweight design of our model, aiming to increase the input window size while maintaining low computational costs. Additionally, we will also focus on developing a unified model for time series anomaly detection. For example, the UCR dataset contains 250 entities, and training a separate model for each entity is time-consuming. We hope to develop a unified model that can handle multiple entities simultaneously.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61871388 and 72071208).

REFERENCES

[1] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. 2020. USAD:UnSupervised Anomaly Detection on Multivariate Time Series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event CA USA, 3395–3404. https://doi.org/10.1145/3394486.3403392

[2] Bjorn Barz, Erik Rodner, Yanira Guanche Garcia, and Joachim Denzler. 2019. Detecting Regions of Maximal Divergence for Spatio-Temporal Anomaly Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 5 (May 2019), 1088–1101. https://doi.org/10.1109/TPAMI.2018.2823766

[3] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. 2022. A Review on Outlier/Anomaly Detection in Time Series Data. *Comput. Surveys* 54, 3 (April 2022), 1–33. https://doi.org/10.1145/3444690

[4] Paul Boniol and Themis Palpanas. 2020. Series2Graph: : Graph-based Subsequence Anomaly Detection for Time Series. *Proceedings of the VLDB Endowment* 13, 12 (Aug. 2020), 1821–1834. https://doi.org/10.14778/3407790.3407792

[5] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J. Franklin. 2021. SAND: Streaming Subsequence Anomaly Detection. *Proceedings of the VLDB Endowment* 14, 10 (June 2021), 1717–1729. https://doi.org/10.14778/3467861.3467863

[6] Lei Chen and Raymond Ng. 2004. On the marriage of Lp-norms and edit distance. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30* (Toronto, Canada) *(VLDB '04)*. VLDB Endowment, 792–803.

[7] Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2023. ImDiffusion: Imputed Diffusion Models for Multivariate Time Series

Anomaly Detection. *Proceedings of the VLDB Endowment* 17, 3 (Nov. 2023), 359–372. https://doi.org/10.14778/3632093.3632101

[8] Jesse Davis and Mark Goadrich. 2006. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*. ACM Press, Pittsburgh, Pennsylvania, 233–240. https://doi.org/10.1145/1143844.1143874

[9] Ailin Deng and Bryan Hooi. 2021. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 5 (May 2021), 4027–4035. https://doi.org/10.1609/aaai.v35i5.16523

[10] Bowen Du, Xuanxuan Sun, Junchen Ye, Ke Cheng, Jingyuan Wang, and Leilei Sun. 2021. GAN-Based Anomaly Detection for Multivariate Time Series Using Polluted Training Set. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3128667

[11] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. TSMixer: Lightweight MLP-Mixer Model for Multivariate Time Series Forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Long Beach CA USA, 459–469. https://doi.org/10.1145/3580305.3599533

[12] Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. 2022. An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (June 2022), 2508–2517. https://doi.org/10.1109/TNNLS.2021.3105827

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 15979–15988. https://doi.org/10.1109/CVPR52688.2022.01553

[14] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. 2022. Local Evaluation of Time Series Anomaly Detection Algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Washington DC USA, 635–645. https://doi.org/10.1145/3534678.3539339

[15] Won-Seok Hwang, Jeong-Han Yun, Jonguk Kim, and Byung Gil Min. 2022. Do You Know Existing Accuracy Metrics Overrate Time-Series Anomaly Detections?. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. ACM, Virtual Event, 403–412. https://doi.org/10.1145/3477314.3507024

[16] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. 2021. Exathlon: A Benchmark for Explainable Anomaly Detection over Time Series. *Proceedings of the VLDB Endowment* 14, 11 (July 2021), 2613–2626. https://doi.org/10.14778/3476249.3476307

[17] Sevvandi Kandanaarachchi, Mario A. Muñoz, Rob J. Hyndman, and Kate Smith-Miles. 2020. On Normalization and Algorithm Selection for Unsupervised Outlier Detection. *Data Mining and Knowledge Discovery* 34, 2 (March 2020), 309–354. https://doi.org/10.1007/s10618-019-00661-z

[18] Eamonn Keogh. 2021. Irrational Exuberance: Why we should not believe 95% of papers on Time Series Anomaly Detection. Retrieved May 21, 2024 from https://kdd-milets.github.io/milets2021/slides/Irrational%20Exuberance_Eammon_Keogh.pdf

[19] Eamonn J. Keogh and Michael J. Pazzani. 2000. Scaling up dynamic time warping for datamining applications. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Boston, Massachusetts, USA) *(KDD '00)*. Association for Computing Machinery, New York, NY, USA, 285–289. https://doi.org/10.1145/347090.347153

[20] Tung Kieu, Bin Yang, Chenjuan Guo, Razvan-Gabriel Cirstea, Yan Zhao, Yale Song, and Christian S. Jensen. 2022. Anomaly Detection in Time Series with Robust Variational Quasi-Recurrent Autoencoders. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, Kuala Lumpur, Malaysia, 1342–1354. https://doi.org/10.1109/ICDE53745.2022.00105

[21] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S. Jensen, Yan Zhao, Feiteng Huang, and Kai Zheng. 2022. Robust and Explainable Autoencoders for Unsupervised Time Series Outlier Detection. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, Kuala Lumpur, Malaysia, 3038–3050. https://doi.org/10.1109/ICDE53745.2022.00273

[22] HyunGi Kim, Siwon Kim, Seonwoo Min, and Byunghan Lee. 2023. Contrastive Time-Series Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering* (2023), 1–14. https://doi.org/10.1109/TKDE.2023.3335317

[23] Minkyung Kim, Jongmin Yu, Junsik Kim, Tae-Hyun Oh, and Jun Kyun Choi. 2023. An Iterative Method for Unsupervised Robust Anomaly Detection Under Data Contamination. *IEEE Transactions on Neural Networks and Learning Systems* (2023), 1–13. https://doi.org/10.1109/TNNLS.2023.3267028

[24] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. 2022. Towards a Rigorous Evaluation of Time-Series Anomaly Detection. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 7 (June 2022), 7194–7201. https://doi.org/10.1609/aaai.v36i7.20680

[25] Chih-Yu Lai, Fan-Keng Sun, and Zhengqi Gao. 2023. Nominality Score Conditioned Time Series Anomaly Detection by Point/Sequential Reconstruction. In *37th Conference on Neural Information Processing Systems*.

[26] Kwei-Herng Lai, Daochen Zha, Yue Zhao, Guanchu Wang, Junjie Xu, and Xia Hu. 2021. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. (2021).

[27] Dongwen Li, Shenglin Zhang, Yongqian Sun, Yang Guo, Zeyu Che, Shiqi Chen, Zhenyu Zhong, Minghan Liang, Minyi Shao, Mingjie Li, Shuyang Liu, Yuzhi Zhang, and Dan Pei. 2023. An Empirical Analysis of Anomaly Detection Methods for Multivariate Time Series. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, Florence, Italy, 57–68. https://doi.org/10.1109/ISSRE59848.2023.00014

[28] Gen Li and Jason J. Jung. 2023. Deep Learning for Anomaly Detection in Multivariate Time Series: Approaches, Applications, and Challenges. *Information Fusion* 91 (March 2023), 93–102. https://doi.org/10.1016/j.inffus.2022.10.008

[29] Longyuan Li, Junchi Yan, Qingsong Wen, Yaohui Jin, and Xiaokang Yang. 2022. Learning Robust Deep State Space for Unsupervised Anomaly Detection in Contaminated Time-Series. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–1. https://doi.org/10.1109/TKDE.2022.3171562

[30] Muyang Li, Xiangyu Zhao, Chuan Lyu, Minghao Zhao, Runze Wu, and Ruocheng Guo. 2022. MLP4Rec: A Pure MLP Architecture for Sequential Recommendations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 2138–2144. https://doi.org/10.24963/ijcai.2022/297

[31] Wenkai Li, Cheng Feng, Ting Chen, and Jun Zhu. 2022. Robust Learning of Deep Time Series Anomaly Detection Models with Contaminated Training Data. arXiv:2208.01841

[32] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate Time Series Anomaly Detection and Interpretation Using Hierarchical Inter-Metric and Temporal Embedding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event Singapore, 3220–3230. https://doi.org/10.1145/3447548.3467075

[33] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. 2024. A Survey on Explainable Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (Jan. 2024), 1–54. https://doi.org/10.1145/3609333

[34] Haoran Liang, Lei Song, Jianxing Wang, Lili Guo, Xuzhi Li, and Ji Liang. 2021. Robust Unsupervised Anomaly Detection via Multi-Time Scale DCGANs with Forgetting Mechanism for Industrial Multivariate Time Series. *Neurocomputing* 423 (Jan. 2021), 444–462. https://doi.org/10.1016/j.neucom.2020.10.084

[35] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. 2023. SegRNN: Segment Recurrent Neural Network for Long-Term Time Series Forecasting. arXiv:2308.11200

[36] Yue Lu, Renjie Wu, Abdullah Mueen, Maria A. Zuluaga, and Eamonn Keogh. 2022. Matrix Profile XXIV: Scaling Time Series Anomaly Detection to Trillions of Datapoints and Ultra-fast Arriving Data Streams. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Washington DC USA, 1173–1182. https://doi.org/10.1145/3534678.3539271

[37] Takaaki Nakamura, Makoto Imamura, Ryan Mercer, and Eamonn Keogh. 2020. MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, Sorrento, Italy, 1190–1195. https://doi.org/10.1109/ICDM50108.2020.00147

[38] Takaaki Nakamura, Ryan Mercer, Makoto Imamura, and Eamonn Keogh. 2023. MERLIN++: Parameter-Free Discovery of Time Series Anomalies. *Data Mining and Knowledge Discovery* 37, 2 (March 2023), 670–709. https://doi.org/10.1007/s10618-022-00876-7

[39] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A TIME SERIES IS WORTH 64 WORDS: LONG-TERM FORECASTING WITH TRANSFORMERS. *ICLR* (2023).

[40] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. 2022. Volume under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Proceedings of the VLDB Endowment* 15, 11 (July 2022), 2774–2787. https://doi.org/10.14778/3551793.3551830

[41] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S. Tsay, Themis Palpanas, and Michael J. Franklin. 2022. TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly DetectionTSB-UAD. *Proceedings of the VLDB Endowment* 15, 8 (April 2022), 1697–1711. https://doi.org/10.14778/3529337.3529354

[42] Shuxin Qin, Yongcan Luo, and Gaofeng Tao. 2023. Memory-Augmented U-Transformer For Multivariate Time Series Anomaly Detection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Rhodes Island, Greece, 1–5. https://doi.org/10.1109/ICASSP49357.2023.10096179

[43] Ferdinand Rewicki, Joachim Denzler, and Julia Niebling. 2023. Is It Worth It? Comparing Six Deep and Classical Methods for Unsupervised Anomaly Detection in Time Series. *Applied Sciences* 13, 3 (Jan. 2023), 1778. https://doi.org/10.3390/app13031778

[44] M. Saquib Sarfraz, Mei-Yen Chen, Lukas Layer, Kunyu Peng, and Marios Koulakis. 2024. Position: Quo Vadis, Unsupervised Time Series Anomaly Detection? arXiv:2405.02678 [cs]

[45] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly Detection in Time Series: A Comprehensive Evaluation. *Proceedings of the VLDB Endowment* 15, 9 (May 2022), 1779–1797. https://doi.org/10.14778/3538598.3538602

[46] Mohamed El Amine Sehili and Zonghua Zhang. 2023. Multivariate Time Series Anomaly Detection: Fancy Algorithms and Flawed Evaluation Methodology. arXiv:2308.13068 [cs, stat]

[47] Zuogang Shang, Zhibin Zhao, Ruqiang Yan, and Xuefeng Chen. 2023. Core Loss: Mining Core Samples Efficiently for Robust Machine Anomaly Detection against Data Pollution. *Mechanical Systems and Signal Processing* 189 (April 2023), 110046. https://doi.org/10.1016/j.ymssp.2022.110046

[48] Shashi Shekhar, Vagelis Papalexakis, Jing Gao, Zhe Jiang, and Matteo Riondato. 2024. PUPAE: Intuitive and Actionable Explanations for Time Series Anomalies. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, Philadelphia, PA. https://doi.org/10.1137/1.9781611978032

[49] Yunfei Shi, Bin Wang, Yanwei Yu, Xianfeng Tang, Chao Huang, and Junyu Dong. 2023. Robust Anomaly Detection for Multivariate Time Series through Temporal GCNs and Attention-Based VAE. *Knowledge-Based Systems* 275 (Sept. 2023), 110725. https://doi.org/10.1016/j.knosys.2023.110725

[50] Haotian Si, Changhua Pei, Zhihan Li, Yadong Zhao, Jingjing Li, Haiming Zhang, Zulong Diao, Jianhui Li, Gaogang Xie, and Dan Pei. 2023. Beyond Sharing: Conflict-Aware Multivariate Time Series Anomaly Detection. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1635–1645. https://doi.org/10.1145/3611643.3613896

[51] Yuting Sun, Guansong Pang, Guanhua Ye, Tong Chen, Xia Hu, and Hongzhi Yin. 2023. Unraveling the "Anomaly" in Time Series Anomaly Detection: A Self-supervised Tri-domain Solution. arXiv:2311.11235 [cs]

[52] Emmanouil Sylligardos, Paul Boniol, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2023. Choose Wisely: An Extensive Evaluation of Model Selection for Anomaly Detection in Time Series. *Proceedings of the VLDB Endowment* 16, 11 (July 2023), 3418–3432. https://doi.org/10.14778/3611479.3611536

[53] Haicheng Tao, Jiawei Miao, Lin Zhao, Zhenyu Zhang, Shuming Feng, Shu Wang, and Jie Cao. 2023. HAN-CAD: Hierarchical Attention Network for Context Anomaly Detection in Multivariate Time Series. *World Wide Web* (May 2023). https://doi.org/10.1007/s11280-023-01171-1

[54] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. 2022. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proceedings of the VLDB Endowment* 15, 6 (Feb. 2022), 1201–1214. https://doi.org/10.14778/3514061.3514067

[55] Dennis Wagner, Tobias Michels, Florian C F Schulz, Arjun Nair, Maja Rudolph, and Marius Kloft. 2023. TimeSeAD: Benchmarking Deep Multivariate Time-Series Anomaly Detection. (2023).

[56] Renjie Wu and Eamonn Keogh. 2021. Current Time Series Anomaly Detection Benchmarks Are Flawed and Are Creating the Illusion of Progress. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3112126

[57] Chunjing Xiao, Zehua Gou, Wenxin Tai, Kunpeng Zhang, and Fan Zhou. 2023. Imputation-Based Time-Series Anomaly Detection with Conditional Weight-Incremental Diffusion Models. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Long Beach CA USA, 2742–2751. https://doi.org/10.1145/3580305.3599391

[58] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, Barcelona, Spain, 1317–1322. https://doi.org/10.1109/ICDM.2016.0179

[59] Jiahao Yu, Xin Gao, Baofeng Li, Feng Zhai, Jiansheng Lu, Bing Xue, Shiyuan Fu, and Chun Xiao. 2024. A Filter-Augmented Auto-Encoder with Learnable Normalization for Robust Multivariate Time Series Anomaly Detection. *Neural Networks* 170 (Feb. 2024), 478–493. https://doi.org/10.1016/j.neunet.2023.11.047

[60] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. TS2Vec: Towards Universal Representation of Time Series. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 8 (June 2022), 8980–8987. https://doi.org/10.1609/aaai.v36i8.20881

[61] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 9 (June 2023), 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317

[62] Yuxin Zhang and Yiqiang Chen. 2023. Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 35, 2 (2023).

[63] Zhijie Zhang, Wenzhong Li, Wangxiang Ding, Linming Zhang, Qingning Lu, Peng Hu, Tong Gui, and Sanglu Lu. 2023. STAD-GAN: Unsupervised Anomaly Detection on Multivariate Time Series with Self-training Generative Adversarial Networks. *ACM Transactions on Knowledge Discovery from Data* 17, 5 (Oct. 2023), 1–18. https://doi.org/10.1145/3572780

[64] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate Time-Series Anomaly Detection via Graph Attention Network. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, Sorrento, Italy, 841–850. https://doi.org/10.1109/ICDM50108.2020.00093

[65] Guoxiang Zhong, Fagui Liu, Jun Jiang, Bin Wang, and C.L. Philip Chen. 2024. Refining One-Class Representation: A Unified Transformer for Unsupervised Time-Series Anomaly Detection. *Information Sciences* 656 (Jan. 2024), 119914. https://doi.org/10.1016/j.ins.2023.119914

[66] Qihang Zhou, Jiming Chen, Haoyu Liu, Shibo He, and Wenchao Meng. 2023. Detecting Multivariate Time Series Anomalies with Zero Known Label. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 4 (June 2023), 4963–4971. https://doi.org/10.1609/aaai.v37i4.25623