DRPAD: A DYNAMIC-AWARE AND ROBUST PARADIGM FOR TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Forecasting-based methods dominate unsupervised time series anomaly detection but primarily emphasize feature extraction and prediction accuracy. In real-world applications, however, the distinctiveness of anomalies depends on additional critical factors. We identify three major challenges: (1) anomaly propagation, (2) distribution shifts, and (3) univariate anomalies—common phenomena that are often overlooked. To address these issues, we propose DRPAD (Dynamic-Aware and Robust Paradigm for Time Series Anomaly Detection), introducing three novel components: Dynamic Prediction Replacement, Segmentation-Based Normalization, and a Mean & Dimension Dual-Check Strategy. Extensive experiments on nine benchmark datasets demonstrate that DRPAD can significantly enhance the performance of a wide range of forecasting-based methods, achieving state-of-the-art results. The source code is publicly available at `https://anonymous.4open.science/r/DRPAD-BEC8/`.

1 INTRODUCTION

In the field of time series anomaly detection, prediction-based approaches have been widely adopted due to their simplicity and effectiveness. Specifically, given a historical window of a time series as $x_{1:t} \in R^{N\times t}$ and the observation at time $t+1$ as $x_{t+1} \in R^N$, where N denotes the number of dimensions. a forecasting model $f(\cdot)$ is employed to predict the next value $\hat{x}_{t+1}$. An anomaly is subsequently detected by comparing the predicted value $\hat{x}_{t+1}$ with the actual observation x_{t+1}, based on the assumption that anomalies induce larger prediction errors and thus can be identified as outliers.

This paradigm has motivated extensive research into prediction-based anomaly detection methods, which predominantly focus on extracting features from input sequences and modeling normal patterns with high accuracy Chen et al. (2021); Zhao et al. (2020); Zhang et al. (2022); Deng & Hooi (2021b). While the core idea is closely aligned with traditional time series forecasting, we argue that, in the context of anomaly detection, enhancing forecasting accuracy alone is insufficient to ensure robust detection performance. Instead, the effectiveness of these methods is influenced by several critical factors, as discussed below.

1. **Anomaly Propagation.** Historical anomalies within the input window may propagate their influence into future predictions, thereby degrading detection performance Shen et al. (2024).

2. **Distribution Shift.** In many real-world time series, changes in environment, machine operating conditions, or user behavior can lead to rapid shifts in the underlying data distribution. Such distribution shifts induce substantial variations in statistical properties and sequence patterns across different temporal segments. Consequently, prediction errors are highly sensitive to the statistical scale of the input window (e.g., mean and variance). In low-variance segments, anomalies become harder to detect, whereas in high-variance segments, normal points may be falsely flagged as anomalies. This statistical heterogeneity increases both false positives and false negatives, undermining model Kim et al. (2021); Liu et al. (2022c); Shen et al. (2024).

3. **Univariate Anomalies.** Another underexplored challenge arises from univariate anomalies—abnormal deviations that occur in only a single feature dimension while the others remain normal. Such anomalies often exhibit relatively small magnitudes and can be masked by the overall statistical characteristics of the multivariate sequence, thereby increasing the risk of missed detections.

Related Work (a) Most existing studies on anomaly contamination have primarily focused on the training phase, addressing issues such as label noise or corrupted training samples, which can impair model learning. However, in prediction-based anomaly detection methods, anomalies in the test sequence can also degrade detection performance by contaminating subsequent predictions. This phenomenon has received little explicit attention in the literature. To our knowledge, the only work that explicitly attempts to address this issue is the AFMF framework Shen et al. (2024), which introduces Progressive Adjacent Masking (PAM). PAM alleviates anomaly propagation via mean substitution, but it rests on strong assumptions—namely, that anomalies always amplify prediction errors and that mean substitution necessarily improves performance. Furthermore, its masking strategy is restricted to the tail of the input sequence, rendering it ineffective for anomalies occurring at arbitrary positions or for more complex structural anomalies. This leaves open the need for a more general and effective solution to the anomaly propagation problem.

(b) RevIN Kim et al. (2021) is a popular normalization method in time series forecasting community to solve distribution shift problems. However, if directly introducing it to anomaly detection, the inverse transformation (denormalization) of it will revive the problem of scale disparity. The AFMF framework Shen et al. (2024) addresses this issue via Local Instance Normalization (LIN), which performs normalization independently within each fixed-length window and alleviates scale inconsistency across adjacent windows. However, when a window simultaneously contains both high-magnitude and low-magnitude segments, the normalization is dominated by the larger values, thereby suppressing small-scale anomalies and causing detection performance to degrade significantly, even to the point of failure. The more introduction of related work can see Appendix F.

We propose **DRPAD**, a **D**ynamic-aware **R**obust **P**aeadigm for Time Series **A**nomaly **D**etection, explicitly designed to address the aforementioned challenges through three dedicated components. (a) ***Dynamic Prediction Replacement (DPR)***: Unlike PAM's mean substitution strategy, DPR does not rely on the assumption that "anomalies necessarily amplify errors and mean substitution necessarily improves detection." PAM often fails in the presence of periodic or structural anomalies and is further restricted to handling anomalies only at the sequence tail. In contrast, DPR leverages context-aware predictions to dynamically replace anomalies at arbitrary positions, aligning more closely with the intrinsic temporal dependencies of the data and thereby suppressing anomaly propagation more comprehensively and effectively. (b) ***Segmentation-Based Normalization via Change Point Detection(SN)*** : Under LIN's fixed-window normalization, if a window contains both high- and low-magnitude segments, the normalization scale is dominated by the larger values, effectively masking small-scale anomalies and severely compromising detection. SN addresses this limitation by applying change point detection to partition the sequence into segments with comparable statistical scales and normalizing each segment independently. This design fundamentally eliminates the "window mixing failure" scenario and ensures stable detection performance under heterogeneous distributions. (c) ***Mean & Dimension Dual-Check Strategy(MDDC)*** : To improve the detection of univariate anomalies, we develop a hybrid thresholding approach based on multidimensional sensitivity. This strategy combines global statistical indicators with per-dimension checks to better capture subtle and localized deviations. Our contributions are threefold:

- We identify and systematically analyze key limitations of prediction-based anomaly detection methods, including *anomaly propagation*, *distribution shifts*, and *univariate anomalies*, moving beyond the conventional focus on forecasting accuracy.
- We propose **DRPAD**, a novel and model-agnostic anomaly detection paradigm, which integrates three innovative components: (a) Dynamic Prediction Replacement (DPR), (b) Segmentation-Based Normalization via Change Point Detection (SN), and (c) a Mean & Dimension Dual-Check Strategy (MDDC).
- We provide a theoretical analysis of the proposed Dynamic Prediction Replacement mechanism, offering insights into its effectiveness in mitigating the influence of anomalous inputs and improving prediction stability.
- We conduct extensive experiments on ten benchmark datasets, demonstrating that DRPAD significantly improves anomaly detection performance across various backbone predictors, including CNN-, RNN-, Transformer-, MLP-, and GNN-based architectures.

2 METHOD

The overall framework of DRPAD is illustrated in Figure 1. We first introduce the three key components of DRPAD and the specific problems each is designed to address. The important notations utilized throughout this paper are summarized in Table 6 in Appendix C.

2.1 DYNAMIC PREDICTION REPLACEMENT

Algorithm. Traditional time series anomaly detection methods typically rely on historical observations for prediction. However, when the input window contains anomalous values, these outliers can propagate errors to subsequent predictions through autoregressive mechanisms. To mitigate this issue, we propose a novel method called **Dynamic Prediction Replacement (DPR)**. The core procedure is detailed in Algorithm 1. DPR comprises two main phases:

Threshold Initialization (Lines 1–6): The model first performs global prediction over the entire sequence using the base predictor. For each time step, the mean squared error (MSE) of the prediction is computed. The global anomaly threshold α is then determined based on the $r - th$ quantile of the MSE distribution.

Dynamic Replacement Prediction (Lines 7–30): Starting from $t = L + 1$, DPR dynamically updates the input window. If the current MSE exceeds α, corresponding observation is considered anomalous. If the number of consecutive anomalies does not exceed δ, the observed value is replaced by its predicted counterpart to prevent contamination of subsequent inputs. If the consecutive anomaly count exceeds δ, the input window is reset to the original observations, and the prediction is recomputed for the current step.

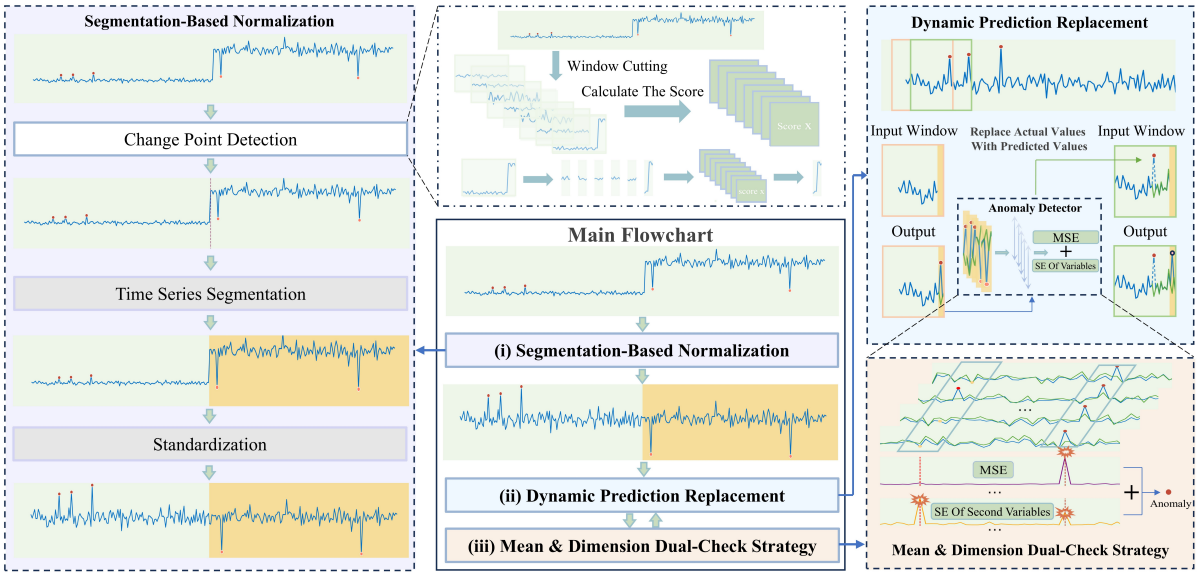

Figure 1: Overview of the DRPAD framework. It consists of three core components: (i) Segmentation-Based Normalization performs change point detection and piecewise standardization; (ii) Dynamic Prediction Replacement mitigates anomaly contamination in forecasting inputs; (iii) Mean & Dimension Dual-Check Strategy detects anomalies by thresholding either mean squared error (MSE) or the standard error of variables.

This replacement strategy effectively prevents the propagation of anomalous values while maintaining a robust and consistent input history. It ensures that only genuinely deviant observations are substituted, while the reset condition prevents long-term prediction drift caused by the accumulation of replaced values.

Theoretical Analysis We provide a theoretical analysis of the dynamic replacement strategy, with detailed mathematical proofs included in the Appendix D. This section presents the main conclusions.

We adopt a fully connected neural network as the base forecasting model. The training set is constructed using a sine function, while the test set is generated by adding Gaussian noise to the standard time series. The sine wave is selected due to its representativeness and analytical tractability. Although the analysis is based on a linear model, the Appendix D.10 demonstrates that the proposed dynamic replacement strategy is also effective in nonlinear models (e.g., fully connected networks with ReLU activation), validating its generality.

To construct the test set, we add Gaussian noise to the standard time series in order to simulate realistic noise perturbations. The noisy test sequence is defined as: $x_t = f(t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, where ε_t is Gaussian noise. To introduce anomalies, we inject a bias Δ_i at a random time i, where $\Delta_i \sim \mathcal{D}$ with mean μ_Δ and variance σ_Δ^2. The corresponding anomalous point becomes: $x_i = f(i) + \varepsilon_i + \Delta_i$.

We use a single-layer fully connected network to predict the next value based on the past L observations: $\hat{x}_t = \sum_{j=1}^{L} w_j x_{t-j} + b$. And we compare two settings:

Baseline Group: Standard Forecasting without Correction. The baseline group employs a traditional forecasting approach, in which modeling and prediction are directly performed on the entire time series without any correction for the detected anomalies. Specifically, the model takes raw observations as input, potentially contaminated by anomalies, and generates predictions for the next time step based on these inputs. Since anomalous points can cause prediction errors to accumulate, the performance of the baseline group serves as a benchmark to assess the impact of anomalous data on prediction accuracy.

Suppose at time t, the input window contains an anomalous value at time step $t - i$ (i.e., a randomly occurring anomaly at time k), modeled as $x_{t-i} = f(t-i) + \varepsilon_{t-i} + \Delta$. The predicted value at time t is:

$$\hat{x}_t = \sum_{j \neq i}^{L} w_j f(t-j) + \varepsilon_{t-j} + b + w_i f(t-i) + \varepsilon_{t-i} + \Delta,$$

where w_i is the weight associated with the anomalous input. Substituting into the prediction error expression yields:

$$e_t = \hat{x}_t - (f(t) + \varepsilon_t) = \sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t + w_i \Delta.$$

Algorithm 1 Dynamic Prediction Replacement (DPR)

Input: Observation sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$; Base prediction model $\mathbf{f}_\theta(\cdot)$; Window length L; Quantile parameter r; Number of features N; Anomaly reset threshold δ

Output: Corrected predictions $\widehat{\mathbf{X}}$, anomaly indicators $\mathbf{A}$

Phase 1: Threshold Initialization
for $t = L + 1$ to T **do**
  $\hat{x}_t \leftarrow \mathbf{f}_\theta([\mathbf{x}_{t-L}, \ldots, \mathbf{x}_{t-1}])$
  $e_t \leftarrow \frac{1}{N}\sum_{j=1}^{N}(\hat{x}_t^{(j)} - x_t^{(j)})^2$
end for
$\alpha \leftarrow \text{Quantile}(\{e_t\}, r)$
Phase 2: Dynamic Replacement Prediction
Initialize sliding window $\mathbf{H}_t \leftarrow [\mathbf{x}_L, \ldots, \mathbf{x}_{t-1}]$
Initialize anomaly counter $c \leftarrow 0$
for $t = L + 1$ to T **do**
  $\hat{x}_t \leftarrow \mathbf{f}_\theta(\mathbf{H}_t)$
  $A_t \leftarrow \mathbb{I}(|\hat{x}_t - \mathbf{x}_t| > \alpha)$
  if $A_t = 1$ **then**
    $c \leftarrow c + 1$
    if $c \leq \delta$ **then**
      $\mathbf{H}_{t+1} \leftarrow [\mathbf{H}_t[2:L], \hat{x}_t]$
    else
      Reset window: $\mathbf{H}_t \leftarrow [\mathbf{x}_{t-L}, \ldots, \mathbf{x}_{t-1}]$
      $\hat{x}_t \leftarrow \mathbf{f}_\theta(\mathbf{H}_t)$
      $A_t \leftarrow \mathbb{I}(|\hat{x}_t - \mathbf{x}_t| > \alpha)$
      if $A_t = 0$ **then**
        $c \leftarrow 0$
      end if
      $\mathbf{H}_{t+1} \leftarrow [\mathbf{H}_t[2:L], \mathbf{x}_t]$
    end if
  else
    $c \leftarrow 0$
    $\mathbf{H}_{t+1} \leftarrow [\mathbf{H}_t[2:L], \mathbf{x}_t]$
  end if
end for

The mean squared error (MSE) is defined as $\text{MSE} = \mathbb{E}[e_t^2]$. Expanding e_t^2 gives:

$$e_t^2 = \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t\right)^2 + 2\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t\right)(w_i\Delta) + (w_i\Delta)^2.$$

Taking expectation over noise and anomaly distributions, we obtain:

$$\text{MSE}_{\text{Baseline}} = \mathbb{E}\left[\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t\right)^2\right] + w_i^2\sigma_\Delta^2 + w_i^2\mu_\Delta^2 = \sigma^2\left(1 + \sum_{j=1}^{L} w_j^2\right) + w_i^2\left(\sigma_\Delta^2 + \mu_\Delta^2\right),$$

where σ^2 is the variance of noise, and σ_Δ^2, μ_Δ^2 denote the variance and mean of the anomaly magnitude Δ.

Experimental Group: Dynamic Prediction Replacement (DPR). The experimental group adopts a dynamic replacement strategy, in which the detected anomalous value is substituted with the prediction value of the model, and then the modified sequence is used for future forecasting. The core idea is to mitigate the influence of anomalies on subsequent predictions, thereby enhancing overall accuracy.

In the case where the input window contains a single anomalous point x_{t-i}, we replace it with the prediction value of the model at that time step, i.e., $\hat{x}_{t-i}$. The replaced input becomes:

$$x'_{t-i} = \hat{x}_{t-i} = f(t-i) + \varepsilon_{t-i} + e_{t-i},$$

where $e_{t-i} = \hat{x}_{t-i} - (f(t-i) + \varepsilon_{t-i})$ is the historical prediction error. As proven in Appendix D.11, the expectation satisfies $\mathbb{E}[e_{t-i}] = 0$, and we denote its variance by $\text{Var}(e_{t-i}) = \sigma_e^2$.

Under this replacement, the predicted value at time t is denoted by $\hat{x}'_t$, with error:

$$e'_t = \hat{x}'_t - (f(t) + \varepsilon_t) = \sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t + w_i e_{t-i}.$$

Substituting into the MSE expression:

$$(e'_t)^2 = \underbrace{\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t\right)^2}_{A} + \underbrace{2\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t\right)(w_i e_{t-i})}_{B} + \underbrace{(w_i e_{t-i})^2}_{C}.$$

Taking expectations, we analyze the three terms separately: Term **A** and term **C** follow the same derivation as in the baseline group. Specifically, term A involves only noise terms and can be treated as independent under standard assumptions, while term C consists solely of the past error term and is unaffected by noise. Their expectations can therefore be directly computed in the same manner as before.

In contrast, term **B** involves the interaction between the noise term ε_{t-j} and the past error term e_{t-i}, which are not strictly independent due to overlapping time indices (see Appendix D.3.3). This dependence complicates the expectation computation and requires a more refined analysis. By carefully expanding and evaluating the cross-terms, we obtain the following expression for the mean squared error under the DPR strategy:

$$\mathrm{MSE}_{\mathrm{DPR}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2\right) + w_i^2 \sigma_e^2 + 2 w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

The difference in mean squared error between the control and experimental groups is:

$$\mathrm{MSE}_{\text{Baseline}} - \mathrm{MSE}_{\text{DPR}} = w_i^2 \left(\sigma_\Delta^2 + \mu_\Delta^2 - \sigma_e^2\right) - 2 w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

Thus, DPR improves prediction performance when the second-order moment of anomaly deviation satisfies:

$$\mathbb{E}[\Delta^2] = \sigma_\Delta^2 + \mu_\Delta^2 > \sigma_e^2 + 2\sigma^2 \left(\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} - 1\right). \tag{1}$$

where $\sigma\Delta^2 + \mu\Delta^2$ denotes the second-order moment of the anomaly signal. To rigorously assess the practical reliability of the inequality, we conducted a comprehensive numerical simulation study on time series data satisfying the Lipschitz smoothness condition to provide robust empirical evidence. Specifically, for each sequence of length $n + L$, we constructed a lagged feature matrix $X \in \mathbb{R}^{n \times L}$ and target vector $y \in \mathbb{R}^n$, fitting a ridge regression model to obtain weights $w \in \mathbb{R}^L$.

To ensure robustness, we performed a grid search over sample sizes $n \in \{200, 500, 1000, 5000\}$ and lag windows $L \in \{10, 20, 50, 100\}$, yielding 16 configurations, each evaluated through 100 independent experiments with distinct random seeds. The heatmap demonstrates that the inequality was satisfied with a probability of 99.98% ± 0.35% across 1600 experiments, thereby substantiating the reliability of the proposed method. Detailed experimental settings are provided in Appendix D.5.

While these simulations establish strong empirical evidence, the lack of a closed-form characterization limits deeper theoretical understanding. The presence of the regression weight w_i in the denominator, which depends on data-driven estimates, renders a closed-form analytical guarantee for equation 1 intractable. To complement these findings with analytical intuition and enable tractable analysis of the upper bound on Z, we consider a simplified but representative data-generating process. Specifically, we substitute a sine function for the underlying signal, i.e., let $x_t = \sin(t)$, which preserves the structure of the derivation and leads to the same inequality condition while enabling tractable analysis.

Under this specialization, we use the following assumptions. When the weight reaches the local optimal value, the partial derivative of the loss function for each weight w_j can be considered to be zero, that is $\frac{\partial \mathcal{L}}{\partial w_j} = 0$, $\forall j = 1, 2, \ldots, L$, and derive the following equation:

$$\sum_{i=1}^{L} w_i \cos(i - j) = \cos(j),\ \forall j = 1, 2, \ldots, L.$$

Solving this (derivation in Appendix D.12), for a sine time series input, the optimal weights are:$w_j = \frac{2}{L}\cos(j)$. Using this weight formula, we compute an upper bound of $\sigma_e^2 + 2\sigma^2 \left(\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} - 1\right)$ at the 95% confidence level. We thus conclude that, under 95% confidence, DPR reduces prediction error when:

$$\mathbb{E}[\Delta^2] = \sigma_\Delta^2 + \mu_\Delta^2 > \left(\frac{4.312}{L} + 1\right)\sigma^2$$

The detailed mathematical derivations can be found in Appendix D.

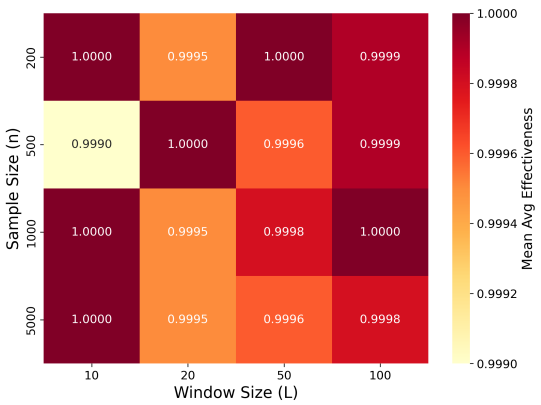

Figure 2: Heatmap of Mean Average Effectiveness Across Sample and Window Sizes. Each cell represents the average effectiveness probability from 100 independent experiments. The color gradient, from light yellow (lower effectiveness) to dark red (higher effectiveness). Most configurations achieve probabilities near or at 1.0000.

2.2 Segmentation-based Normalization via Change Point Detection

Scale variation in time series is fundamentally caused by *distributional shift* Kim et al. (2021), which reflect dynamic changes in statistical properties across different local windows. Consequently, the prediction error at a given time point depends not only on the presence of anomalies but also on the statistical characteristics of the input window. Without ensuring comparable statistical properties across windows, prediction errors cannot serve as reliable indicators for anomaly detection.

To mitigate detection bias introduced by global normalization, we propose a **segment-wise normalization method based on change point detection (SN)**. Specifically, the time series is first segmented into statistically independent intervals using change point detection, each segment is independently normalized, and the full sequence is then reconstructed for downstream anomaly detection.

The process begins with the detection of coarse change points using the Pruned Exact Linear Time (PELT) algorithm Killick et al. (2012). In real-world applications involving large-scale datasets, directly applying PELT with fine granularity across the entire sequence can incur substantial computational costs—our empirical analysis shows that its time complexity reaches the order of $O(n^2)$. To balance detection accuracy and computational efficiency, we adopt a sliding window-based refinement strategy. For each preliminary change point detected by PELT, denoted as $\mathcal{C}_1 = \{c_1, c_2, \ldots, c_m\}$, we perform localized discrepancy analysis within the neighborhood region $[c_i - R, c_i + R]$ for each c_i, using a two-window difference function (see Truong et al. (2020)) to identify the most significant local change points. As proven in the appendix E, this optimization strategy reduces the complexity from $O(n^2)$ to $O(n)$, making it more suitable for large-scale time series.

By partitioning the sequence at adjacent refined change points, a set of contiguous subsequences $\{\mathbf{S}_j\}$ is obtained, where each segment $\mathbf{S}_j \in \mathbb{R}^{T_j \times N}$ (T_j is the time step length of the temporal segment) represents a multivariate block to be normalized independently. Specifically, each $\mathbf{S}_j$ corresponds to a continuous segment of the original sequence, defined as $\mathbf{S}_j = [\mathbf{x}_{\tau_1}, \ldots, \mathbf{x}_{\tau_1+T_j-1}]$, where $\mathbf{x}_t \in \mathbb{R}^N$ denotes a multivariate observation at time t. Each segment $\mathbf{S}_j$ is then independently normalized:

$$\widetilde{\mathbf{S}}_j = (\mathbf{S}_j - \mu_j)/\sigma_j, \quad \mu_j = \tfrac{1}{T_j}\sum_k \mathbf{S}_{j,k}, \ \sigma_j = \sqrt{\tfrac{1}{T_j}\sum_k (\mathbf{S}_{j,k} - \mu_j)^2}, \tag{2}$$

where μ_j and σ_j are the mean and variance of each segment. Finally, segments are concatenated $\widetilde{\mathbf{S}} = [\widetilde{\mathbf{S}}_1, \ldots, \widetilde{\mathbf{S}}_k]$ for downstream anomaly scoring. This pipeline—offloading change-point estimation to established libraries—ensures efficiency while focusing our contribution on the subsequent distribution-adaptive normalization.

Due to space limitations, a detailed visualization of anomaly detection results using segmentation-based normalization on real-world sequences is provided in Appendix B.

2.3 Mean & Dimension Dual-Check Strategy

To mitigate the limitations of dimension-view evaluations, we introduce **Mean & Dimension Dual-Check (MDDC)** strategy, combining global error evaluation with dimension-wise assessment for comprehensive anomaly detection.

Given ground-truth $X_t \in \mathbb{R}^d$ and prediction $\hat{x}_t$, the global error is defined as

$$\mathcal{E}_t^{\text{global}} = \frac{1}{d}\sum_{i=1}^{d}(X_t^{(i)} - \hat{x}_t^{(i)})^2,$$

with threshold $\tau^{\text{global}} = Q_p(\{\mathcal{E}_t^{\text{global}}\}_{t\in\mathcal{D}_{\text{val}}})$. To capture dimension-specific anomalies, a simple yet effective **Dimension-wise Alarm (DA)** module is employed, applying adaptive thresholds to each dimension.

The squared error is defined as $\mathcal{E}_t^{(i)} = (X_t^{(i)} - \hat{x}_t^{(i)})^2$. For each dimension i, we compute the expected error $\mu^{(i)} = \mathbb{E}[\mathcal{E}^{(i)}]$ and standard deviation $\sigma^{(i)} = \sqrt{\text{Var}[\mathcal{E}^{(i)}]}$, both estimated over the validation set. The adaptive threshold is given by $\tau_t^{(i)} = \mu^{(i)} + \varphi \cdot \sigma^{(i)})$.

An anomaly is flagged if at least one dimension satisfies $\mathcal{E}_t^{(i)} > \tau_t^{(i)}$. The final decision rule is:

$$\text{Anomaly}(t) = \mathbb{I}(\mathcal{E}_t^{\text{global}} > \tau^{\text{global}}) \vee \mathbb{I}\left(\sum_{i=1}^{d}\mathbb{I}(\mathcal{E}_t^{(i)} > \tau_t^{(i)}) \geq 1\right), \tag{3}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, and the symbol $\vee$ represents the logical OR, meaning that an anomaly is flagged if either the global deviation or at least one dimension-wise deviation exceeds its threshold. This dual-check mechanism ensures sensitivity to both global and localized deviations. In addition, this study incorporates the Lopsided Forecasting module (LF) proposed in AFMF Shen et al. (2024) as part of the DRPAD implementation. The module processes discrete and continuous variables separately. Both types are used as input, and only continuous variables are retained in the output.

3 Experiment

3.1 Dataset

We evaluate DRPAD on ten real-world time series anomaly detection benchmarks, including SMD Su et al. (2019) , PSM Abdulaal et al. (2021), MSL Hundman et al. (2018a) , SMAP Hundman et al. (2018a), SWaT Mathur & Tippenhauer (2016), WADI Ahmed et al. (2017), MBA Moody & Mark (2001), NAB Ahmad et al. (2017), and MSDS Nedelkoski et al. (2020). Each dataset is divided into training and testing subsets. Within the training subset, 80% of the data is used for training and 20% for validation. Anomalies are annotated exclusively in the testset. Detailed descriptions of each dataset are provided in the Appendix.

3.2 Baselines

To comprehensively evaluate the performance of DRPAD, we selected a range of state-of-the-art baselines representing various technical paradigms. These include a density estimation-based approach (DAGMM Zong et al. (2018)), reconstruction-based methods (CAE-M Zhang et al. (2021a), MEMTO Song et al. (2023)), and prediction-based detectors (GDN Deng & Hooi (2021a), GTA Chen et al. (2021)).

For DRPAD, we incorporated six representative time series forecasting models from different architectural families as base predictors, including RTNet Shen et al. (2022) (CNN-based), DeepAR Zhou et al. (2023b) (RNN-based), Autoformer Wu et al. (2021) and FEDformer (Transformer-based) Zhou et al. (2022), DLinear Zeng et al. (2023) (MLP-based), and GTA Chen et al. (2021) (GNN-based). Among them, GTA is a prediction-based anomaly detection method, while the others are pure forecasting models.

3.3 Settings

Anomaly scores at each timestamp are computed using MSE , defined as $MSE = \frac{1}{N}\sum_{n=1}^{N}(\hat{x}_t^n - x_t^n)^2$, and a point is flagged as anomalous if its score exceeds a threshold δ. Following Xu et al. (2021), δ is set by assuming the top $r\%$ of the test data are anomalies.

Unlike methods that apply post-processing techniques such as anomaly range adjustment strategy Shen et al. (2020); Xu et al. (2018), we adopt a strict **point-wise evaluation** protocol for three reasons: (1) **Practical relevance** — accurately identifying the onset of failures is crucial in industrial diagnostics, while range adjustment may obscure early indications; (2) **Model fidelity** — anomaly range adjustment can inflate performance and obscure the true detection ability of model; (3) **Comparative fairness** — evaluation without auxiliary enhancements ensures a fair comparison across methods.

All methods use the same data preprocessing pipeline as AFMF Shen et al. (2024). Hyperparameters follow the original settings (see Appendix G). DRPAD is trained using the AdamW optimizer with a OneCycle learning rate scheduler. Results are averaged over five independent runs. The batch size is set to 128 for all models. The initial learning rate is 1×10^{-4}.

To ensure a fair comparison under point-wise evaluation metrics, we unify threshold selection across all methods by fixing δ at the top $r\%$ of test anomaly scores, avoiding biases introduced by range-based tuning. Unless stated

otherwise, the best results are highlighted in **bold** and the second-best results are underlined. The sensitivity of the dimension-wise anomaly detection threshold φ is set with 6. The values of the anomaly detection threshold r and the maximum allowed consecutive anomalies δ are summarized in Table 10. Full hyperparameters are provided in Appendix G. Evaluation metrics include Precision ($P = \frac{TP}{TP+FP}$), Recall ($R = \frac{TP}{TP+FN}$), and F1-score ($F1 = \frac{2\times P\times R}{P+R}$).

Table 1: Performance comparison of different methods across various datasets

Categorization	Baselines	SMD			MSL			PSM		
		P	R	F1	P	R	F1	P	R	F1
Density Estimation	DAGMM Zong et al. (2018)	12.34%	0.31%	0.60%	26.47%	2.90%	5.23%	67.37%	0.04%	6.91%
Reconstruction	MEMTO Song et al. (2023)	12.21%	1.76%	3.07%	11.00%	1.44%	2.55%	29.64%	1.81%	3.41%
	CAE-M Zhang et al. (2021a)	9.46%	0.50%	0.95%	5.88%	0.65%	1.16%	27.76%	1.50%	2.85%
	uFedHy-DisMTSADD Hao et al. (2025)	15.63%	1.88%	3.36%	27.82%	3.95%	6.92%	29.78%	3.50%	6.26%
Prediction	GDN Deng & Hooi (2021a)	16.25%	1.04%	1.95%	29.41%	3.23%	5.81%	34.53%	3.06%	5.62%
	GTA Chen et al. (2021)	16.90%	2.03%	3.63%	47.83%	6.81%	11.93%	71.66%	3.84%	7.29%
	FEDformer Zhou et al. (2022)	32.43%	3.89%	6.95%	32.65%	4.65%	8.14%	55.30%	2.96%	5.62%
	FEDformer_w_AFMF Shen et al. (2024)	30.80%	3.60%	6.45%	13.71%	1.34%	2.44%	53.37%	2.27%	4.35%
	FEDformer_w_DRPAD(our)	41.39%	12.77%	**19.50%**	26.71%	14.57%	**18.84%**	46.83%	22.14%	**29.66%**
Categorization	Baselines	SMAP			MSDS			NAB		
		P	R	F1	P	R	F1	P	R	F1
Density Estimation	DAGMM Zong et al. (2018)	6.32%	0.43%	0.80%	1.93%	1.50%	1.69%	38.10%	33.33%	35.56%
Reconstruction	MEMTO Song et al. (2023)	16.97%	2.64%	4.57%	2.51%	3.85%	3.04%	25.00%	7.06%	11.02%
	CAE-M Zhang et al. (2021a)	8.68%	0.91%	1.65%	3.24%	2.50%	2.82%	16.01%	15.21%	15.60%
	uFedHy-DisMTSADD Hao et al. (2025)	7.76%	2.90%	4.22%	7.83%	3.64%	4.97%	2.59%	9.52%	4.07%
Prediction	GDN Deng & Hooi (2021a)	8.05%	0.94%	1.69%	1.93%	1.50%	1.69%	38.10%	33.33%	35.56%
	GTA Chen et al. (2021)	15.79%	1.85%	3.31%	18.43%	13.55%	15.62%	47.37%	37.50%	41.86%
	FEDformer Zhou et al. (2022)	13.21%	1.54%	2.77%	31.74%	23.33%	26.89%	46.31%	36.67%	40.93%
	FEDformer_w_AFMF Shen et al. (2024)	15.02%	1.52%	2.76%	51.61%	34.74%	41.42%	25.59%	22.00%	23.65%
	FEDformer_w_DRPAD(our)	17.36%	6.87%	**8.87%**	50.61%	49.06%	**49.75%**	59.00%	39.17%	**46.82%**
Categorization	Baselines	MBA			WADI			SWaT		
		P	R	F1	P	R	F1	P	R	F1
Density Estimation	DAGMM Zong et al. (2018)	100.00%	5.92%	11.18%	1.97%	2.35%	2.14%	74.04%	3.05%	5.86%
Reconstruction	MEMTO Song et al. (2023)	68.14%	2.99%	5.73%	4.27%	40.54%	7.72%	18.54%	2.46%	4.34%
	CAE-M Zhang et al. (2021a)	33.85%	2.00%	3.79%	6.27%	7.55%	6.85%	74.49%	3.07%	5.90%
	uFedHy-DisMTSADD Hao et al. (2025)	37.87%	6.21%	10.67%	8.31%	7.41%	7.83%	27.83%	3.95%	6.92%
Prediction	GDN Deng & Hooi (2021a)	93.46%	5.92%	11.13%	4.27%	0.54%	2.72%	27.80%	3.99%	6.98%
	GTA Chen et al. (2021)	97.63%	5.80%	10.94%	34.84%	3.02%	5.55%	92.16%	3.79%	7.28%
	FEDformer Zhou et al. (2022)	92.23%	5.47%	10.34%	25.97%	2.25%	4.14%	62.88%	5.18%	9.56%
	FEDformer_w_AFMF Shen et al. (2024)	98.32%	3.76%	7.24%	8.41%	0.65%	1.21%	28.07%	0.74%	1.44%
	FEDformer_w_DRPAD(our)	81.53%	11.46%	**20.09%**	36.65%	48.31%	**12.98%**	22.64%	6.48%	**10.04%**

3.4 Main Experimental Results

We conduct a comprehensive evaluation of the proposed **DRPAD** framework on nine publicly available datasets, comparing its performance against several representative baseline methods. As shown in Table 1, the FEDformer model augmented with DRPAD (*FEDformer_w_DRPAD*) consistently achieves the highest F1-scores across all nine datasets, indicating substantial improvements over the baselines. On average, our framework yields an F1-score improvement of approximately 91.32% compared to the best-performing baseline method for each dataset.

Specifically, compared with the anomaly detection framework AFMF, which is also based on prediction methods, after combining FEDformer (*FEDformer_w_AFMF*), our method still performs well on all datasets, with an average F1 score improvement of 393.66%. These results highlight the robustness and effectiveness of DRPAD in diverse scenarios.

Furthermore, to evaluate the generalizability and performance benefits of the DRPAD framework across different forecasting architectures, we integrate it into six widely used time series forecasting models. As shown in Table 2, all models demonstrate performance improvements across the majority of datasets after being augmented with DRPAD. For instance, in terms of F1-score, the average improvement across all models and datasets is 561.89% . This enhancement is observed in 49 out of 54 model-dataset combinations (approximately 90.7%), underscoring the broad applicability of DRPAD. Nevertheless, a few exceptions are noted. On the SWaT dataset, four models—DeepAR, GTA, RTNet, and FEDformer—exhibit slight declines in F1-score. This may be due to the relatively minor distributional shifts and the lower prevalence of single-dimensional anomalies within the SWaT dataset. Additionally, DRPAD significantly improves recall across several models, suggesting enhanced sensitivity to subtle or hard-to-detect anomaly patterns.

Importantly, DRPAD achieves these performance gains without any modification to the underlying model architectures, affirming its potential as a model-agnostic plug-in module for enhancing anomaly detection in existing systems.

In addition, we conduct an ablation study within the FEDformer backbone, as presented in Table 3. PAM and LIN, originally proposed in the AFMF framework Shen et al. (2024), are functionally replaced in DRPAD by our DPR (Dynamic Prediction Replacement) and SN (Segment-wise Normalization) modules. The ablation results demonstrate that they are less useful than DRPAD components. Besides, The full DRPAD configuration achieves the highest F1 score on 7 of 9 datasets, demonstrating the effectiveness of combining all three components.

To ensure a comprehensive assessment, our framework is further evaluated under the *advanced adjustment strategy* proposed in Kim et al. (2022), where a predicted anomalous segment is considered correct if at least 20% of its timestamps overlap with the ground truth (see Appendix A.2 for details).

Table 2: Performance comparison of models with and without DRPAD framework across multiple datasets

Model	MBA			MSDS			MSL		
	P	R	F1	P	R	F1	P	R	F1
Autoformer-wo-DRPAD	82.37%	4.89%	9.23%	39.19%	28.80%	33.20%	32.77%	4.67%	8.17%
Autoformer-w-DRPAD	67.33%	9.28%	**16.08%** ↑74.21%	54.13%	41.15%	**46.76%** ↑40.84%	23.11%	5.73%	**9.06%** ↑10.89%
DLinear-wo-DRPAD	99.34%	5.90%	11.14%	59.13%	43.46%	50.10%	39.17%	5.58%	9.77%
DLinear-w-DRPAD	98.45%	7.44%	**13.83%** ↑24.15%	47.75%	50.60%	**49.13%** ↓1.94%	29.22%	14.00%	**18.89%** ↑93.35%
DeepAR-wo-DRPAD	93.29%	5.54%	10.46%	28.72%	21.11%	24.33%	42.98%	6.12%	10.72%
DeepAR-w-DRPAD	58.73%	27.48%	**35.80%** ↑242.26%	49.06%	48.80%	**48.88%** ↑100.90%	32.33%	12.15%	**17.60%** ↑64.18%
GTA-wo-DRPAD	97.63%	5.80%	10.94%	18.43%	13.55%	15.62%	47.83%	6.81%	11.93%
GTA-w-DRPAD	96.28%	6.83%	**12.74%** ↑16.45%	47.85%	38.29%	**41.95%** ↑168.50%	30.11%	13.07%	**17.95%** ↑50.46%
RTNet-wo-DRPAD	96.58%	2.87%	5.57%	43.60%	32.05%	36.95%	37.67%	5.36%	9.39%
RTNet-w-DRPAD	80.36%	12.97%	**22.33%** ↑300.90%	45.64%	59.44%	**51.57%** ↑39.57%	30.97%	16.43%	**21.36%** ↑127.48%
FEDformer-wo-DRPAD	92.24%	5.48%	10.34%	56.86%	41.79%	48.18%	32.66%	4.65%	8.14%
FEDformer-w-DRPAD	81.53%	11.46%	**20.09%** ↑94.29%	50.61%	49.06%	**49.75%** ↑3.26%	26.71%	14.57%	**18.84%** ↑131.45%

Model	NAB			PSM			SMAP		
	P	R	F1	P	R	F1	P	R	F1
Autoformer-wo-DRPAD	49.47%	39.17%	43.72%	65.11%	3.49%	6.63%	10.28%	1.21%	2.16%
Autoformer-w-DRPAD	52.86%	40.83%	**46.02%** ↑5.26%	42.37%	13.24%	**20.17%** ↑204.22%	8.93%	6.12%	**6.62%** ↑206.48%
DLinear-wo-DRPAD	46.32%	36.67%	40.93%	58.32%	3.13%	5.93%	10.10%	1.18%	2.12%
DLinear-w-DRPAD	52.56%	38.33%	**44.27%** ↑8.16%	42.37%	13.24%	**20.17%** ↑240.14%	8.43%	2.59%	**3.97%** ↑87.26%
DeepAR-wo-DRPAD	48.42%	38.33%	42.79%	71.66%	3.84%	7.29%	11.75%	1.38%	2.47%
DeepAR-w-DRPAD	53.34%	41.67%	**46.75%** ↑9.25%	31.87%	25.43%	**28.29%** ↑288.07%	7.26%	3.38%	**4.59%** ↑85.83%
GTA-wo-DRPAD	47.37%	37.50%	41.86%	67.38%	3.61%	6.86%	15.79%	1.85%	3.31%
GTA-w-DRPAD	48.42%	38.33%	**42.78%** ↑2.20%	50.93%	6.92%	**12.16%** ↑77.26%	9.42%	4.65%	**6.22%** ↑87.92%
RTNet-wo-DRPAD	50.53%	40.00%	44.65%	65.62%	3.52%	6.68%	13.15%	1.54%	2.76%
RTNet-w-DRPAD	68.21%	35.83%	**46.97%** ↑5.20%	42.67%	13.94%	**21.02%** ↑214.67%	8.15%	2.73%	**4.08%** ↑47.83%
FEDformer-wo-DRPAD	46.32%	36.67%	40.93%	55.30%	2.96%	5.63%	13.21%	1.55%	2.77%
FEDformer-w-DRPAD	59.00%	39.17%	**46.82%** ↑14.39%	46.83%	22.14%	**29.66%** ↑426.82%	12.78%	6.87%	**8.87%** ↑220.22%

Model	SMD			SWaT			WADI		
	P	R	F1	P	R	F1	P	R	F1
Autoformer-wo-DRPAD	37.98%	4.57%	8.15%	70.37%	2.90%	5.56%	2.31%	0.20%	0.37%
Autoformer-w-DRPAD	20.94%	19.81%	**13.02%** ↑59.75%	15.94%	21.52%	**13.77%** ↑147.66%	6.65%	32.76%	**10.96%** ↑2862.16%
DLinear-wo-DRPAD	41.38%	4.97%	8.88%	15.33%	0.63%	1.21%	4.24%	0.37%	0.68%
DLinear-w-DRPAD	39.09%	12.33%	**18.75%** ↑111.15%	16.60%	11.05%	**13.27%** ↑996.69%	7.21%	41.75%	**12.30%** ↑1708.82%
DeepAR-wo-DRPAD	20.81%	2.50%	4.47%	82.60%	3.40%	6.53%	0.47%	0.04%	0.07%
DeepAR-w-DRPAD	40.38%	11.96%	**18.44%** ↑312.53%	22.07%	3.20%	**5.58%** ↓14.55%	7.41%	38.72%	**12.30%** ↑17471.43%
GTA-wo-DRPAD	16.90%	2.03%	3.63%	92.16%	3.79%	7.28%	34.84%	3.02%	5.55%
GTA-w-DRPAD	41.91%	11.62%	**18.19%** ↑401.10%	32.57%	2.63%	**4.83%** ↓33.65%	9.33%	16.54%	**11.91%** ↑114.59%
RTNet-wo-DRPAD	35.37%	4.25%	7.59%	88.47%	3.64%	6.99%	3.77%	0.33%	0.60%
RTNet-w-DRPAD	40.15%	13.24%	**19.92%** ↑162.45%	16.32%	4.12%	**6.57%** ↓6.01%	7.60%	26.07%	**11.76%** ↑1860.00%
FEDformer-wo-DRPAD	31.53%	3.79%	6.77%	64.65%	5.32%	9.83%	25.97%	2.25%	4.14%
FEDformer-w-DRPAD	43.13%	12.56%	**19.45%** ↑187.30%	26.32%	4.65%	**7.90%** ↓19.63%	8.43%	26.40%	**12.75%** ↑207.97%

4 CONCLUSION

In this paper, we propose **DRPAD**, a dynamic-aware and robust paradigm for time series anomaly detection, specifically designed to address three fundamental challenges: anomaly propagation, distribution shifts, and univariate anomalies. To this end, DRPAD integrates three complementary components—**Dynamic Prediction Replacement (DPR)**, **Segmentation-Based Normalization (SN)**, and a **Mean & Dimension Dual-Check (MDDC)** strategy—into a unified, model-agnostic framework that can be seamlessly combined with a variety of forecasting-based methods. We provide theoretical analysis showing that DPR reduces prediction errors by suppressing the impact of anomalous inputs, though this analysis is currently grounded on synthetic sine-based data for analytical tractability. Extensive experiments on ten real-world benchmarks demonstrate that DRPAD consistently improves performance across diverse model architectures. We believe DRPAD provides a principled and extensible foundation for advancing anomaly detection in complex time series scenarios.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our study does not involve human subjects, personal data, or sensitive attributes, and all datasets used are publicly available benchmark datasets that have been widely adopted in prior research. We followed standard practices for data processing as described in Appendix G, and no proprietary or confidential data were used. The proposed methodology is intended solely for academic research on anomaly detection in time series data and does not directly target potentially harmful applications. We are not aware of any conflicts of interest, funding biases, or legal compliance issues arising from this work.

REPRODUCIBILITY STATEMENT

We have made significant efforts to ensure reproducibility of our work. The detailed algorithmic components of DRPAD, including Dynamic Prediction Replacement (DPR), Segment-wise Normalization (SN), and the Mean & Dimension Dual-Check (MDDC), are formally defined in Section 2. Complete mathematical derivations are provided in Appendix D, and proofs of complexity reduction are given in Appendix E. Experimental settings, including datasets, preprocessing steps, and baseline configurations, are described in Section 3 and Appendix G.All datasets employed are publicly available, and the source code is publicly available at `https://anonymous.4open.science/r/DRPAD-BEC8/`.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

No large language models (LLMs) were employed in this work.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. 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.

Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

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

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. 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.

Md Abul Bashar and Richi Nayak. Tanogan: Time series anomaly detection with generative adversarial networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1778–1785. IEEE, 2020. ISBN 978-1-7281-2547-3.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6989–6997, 2023.

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

Ian Cleland, Manhyung Han, Chris Nugent, Hosung Lee, Sally McClean, Shuai Zhang, and Sungyoung Lee. Evaluation of prompted annotation of activity data recorded from a smart phone. *Sensors*, 14(9):15861–15879, 2014.

Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. *arXiv preprint arXiv:2202.07857*, 2022.

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

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

Frédéric Desobry, Manuel Davy, and Christian Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.

Kyle D Feuz, Diane J Cook, Cody Rosasco, Kayela Robertson, and Maureen Schmitter-Edgecombe. Automated detection of activity transitions for prompting. *IEEE transactions on human-machine systems*, 45(5):575–585, 2014.

Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 Ieee International Conference on Big Data (Big Data)*, pp. 33–43. IEEE, 2020. ISBN 1-7281-6251-3.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Haixuan Guo, Shuhan Yuan, and Xintao Wu. Logbert: Log anomaly detection via bert. In *2021 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Junfeng Hao, Peng Chen, Juan Chen, and Xi Li. Effectively detecting and diagnosing distributed multivariate time series anomalies via unsupervised federated hypernetwork. *Information Processing & Management*, 62(4): 104107, 2025.

Siyuan Huang and Yepeng Liu. Fl-net: A multi-scale cross-decomposition network with frequency external attention for long-term time series forecasting. *Knowledge-Based Systems*, 288:111473, 2024.

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

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

Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7194–7201, 2022.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Michael P Knapp. Sines and cosines of angles in arithmetic progression. *Mathematics magazine*, 82(5):371–372, 2009.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104. ACM, 2018. ISBN 978-1-4503-5657-2.

Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In Igor V. Tetko, Věra Kůrková, Pavel Karpov, and Fabian Theis (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, volume 11730, pp. 703–716. Springer International Publishing, 2019. ISBN 978-3-030-30489-8 978-3-030-30490-4.

Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3220–3230. ACM, 2021. ISBN 978-1-4503-8332-5.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022a.

Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. Multivariate time-series forecasting with temporal polynomial graph neural networks. *Advances in neural information processing systems*, 35:19414–19426, 2022b.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35:9881–9893, 2022c.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The twelfth international conference on learning representations*, pp. 1–43, 2024.

Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pp. 31–36. IEEE, 2016.

George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50, 2001.

Sasho Nedelkoski, Jasmin Bogatinovski, Ajay Kumar Mandapati, Soeren Becker, Jorge Cardoso, and Odej Kao. Multi-source distributed system data for ai-powered analytics. In *Service-Oriented and Cloud Computing: 8th IFIP WG 2.14 European Conference, ESOCC 2020, Heraklion, Crete, Greece, September 28–30, 2020, Proceedings 8*, pp. 161–176. Springer, 2020.

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

Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):1–27, 2010.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018a.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pp. 4393–4402. PMLR, 2018b. ISBN 2640-3498.

Yunus Saatçi, Ryan D Turner, and Carl E Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 927–934, 2010.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Li Shen, Yuning Wei, and Yangzhu Wang. Respecting time series properties makes deep time series forecasting perfect. *arXiv preprint arXiv:2207.10941*, 2022.

Li Shen, Yuning Wei, Yangzhu Wang, and Hongguang Li. Afmf: Time series anomaly detection framework with modified forecasting. *Knowledge-Based Systems*, 296:111912, 2024.

Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in neural information processing systems*, 33:13016–13026, 2020.

Yunfei Shi, Bin Wang, Yanwei Yu, Xianfeng Tang, Chao Huang, and Junyu Dong. Robust anomaly detection for multivariate time series through temporal gcns and attention-based vae. *Knowledge-Based Systems*, 275:110725, 2023.

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

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in knowledge discovery and data mining: 6th Pacific-Asia conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 proceedings 6*, pp. 535–548. Springer, 2002.

Luan Tran, Min Y. Mun, and Cyrus Shahabi. Real-time distance-based outlier detection in data streams. *Proceedings of the VLDB Endowment*, 14(2):141–153, 2020.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

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

Yijie Wang, Hao Long, Linjiang Zheng, and Jiaxing Shang. Graphformer: Adaptive graph correlation transformer for multivariate long sequence time series forecasting. *Knowledge-Based Systems*, 285:111321, 2024.

Zhiwei Wang, Zhengzhang Chen, Jingchao Ni, Hui Liu, Haifeng Chen, and Jiliang Tang. Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 3726–3734, 2021.

Li Wei and Eamonn Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 748–753, 2006.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Haowen Xu, Yang Feng, Jie Chen, Zhaogang Wang, Honglin Qiao, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, and Dan Pei. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pp. 187–196. ACM Press, 2018. ISBN 978-1-4503-5639-8.

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

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022.

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

Shengming Zhang, Yanchi Liu, Xuchao Zhang, Wei Cheng, Haifeng Chen, and Hui Xiong. Cat: Beyond efficient transformer for content-aware anomaly detection in event sequences. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4541–4550, 2022.

Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. Unsupervised deep anomaly detection for multi-sensor time-series signals. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):2118–2132, 2021a.

Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. Unsupervised deep anomaly detection for multi-sensor time-series signals. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):2118–2132, 2021b.

Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, pp. 841–850. IEEE, 2020.

Binggui Zhou, Yunxuan Dong, Guanghua Yang, Fen Hou, Zheng Hu, Suxiu Xu, and Shaodan Ma. A graph-attention based spatial-temporal learning framework for tourism demand forecasting. *Knowledge-Based Systems*, 263: 110275, 2023a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023b.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023c.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

APPENDIX OVERVIEW

This appendix provides supplementary materials that support the main text, organized as follows:

- **A More Experimental Results** Includes additional ablation studies (Table 3) and evaluations under the advanced adjustment strategy Kim et al. (2022). These results further validate the contributions of each DRPAD component and provide robustness checks under relaxed evaluation criteria.

- **B Visual Evidence of Segment-wise Normalization** Presents qualitative visualization (Figure 3) comparing global normalization and our proposed segment-wise normalization (SN) on real-world datasets, highlighting how SN effectively mitigates scale disparities.

- **C Notation Summary** Summarizes the mathematical symbols used throughout the paper for ease of reference.

- **D Detailed Mathematical Proof** Provides the formal derivation and theoretical analysis underpinning the Dynamic Prediction Replacement (DPR) mechanism.

- **E Proof of Complexity Reduction in the SN Module** This section provides a comprehensive complexity analysis of the SN module, detailing the problem definition, theoretical complexity reduction of the PELT algorithm, and empirical validation through runtime experiments and model fitting.

- **F Related Works** We review key literature on time series anomaly detection, forecasting, and change point detection. Unsupervised methods are categorized into forecasting-based, reconstruction-based, density estimation, and clustering-based approaches. We also compare forecasting models and change point detection techniques. Our work builds on the AFMF framework, introducing Segment-wise Normalization (SN) and Dynamic Prediction Replacement, which overcome limitations of existing normalization strategies like Local Instance Normalization (LIN) and Progressive Adjacent Masking (PAM) to enhance anomaly detection performance.

- **G Baselines and Datasets** Describes in detail the benchmark datasets and baseline methods used in this study, with numerical dataset statistics summarized in Table 10.

A More Experimental Results

A.1 Ablation Study

We conduct an ablation study within the FEDformer backbone in Table 3, where "+X"/"-X" indicates the inclusion or removal of component X. PAM (Progressive Adjacent Masking) and LIN (Local Instance Normalization) are components originally proposed in the AFMF framework Shen et al. (2024). In DRPAD, they are functionally replaced by our DPR (Dynamic Prediction Replacement) and SN (Segment-wise Normalization) components, while the Mean & Dimension Dual-Check (MDDC) strategy serves as an auxiliary detection module additionally proposed to further enhance the overall performance.

The full DRPAD configuration achieves the highest F1 score on 7 of 9 datasets, demonstrating the effectiveness of combining all three components. Adding DPR (e.g., DRPAD vs. DRPAD-DPR) substantially improves recall by mitigating anomaly contamination and stabilizing normal pattern learning. SN generally enhances precision (e.g., DRPAD-SN vs. DRPAD), though minor recall drops may occur. DA consistently boosts recall, with small precision trade-offs in some cases.Compared to LIN and PAM from AFMF, our SN and DPR modules achieve better performance in their respective roles.

In summary, each DRPAD component contributes independently, and their combination yields a strong synergistic effect on F1 performance.

Table 3: Ablation results of DRPAD on nine datasets. We report Precision (P), Recall (R), and F1 score for each configuration

Framework	MBA			MSDS			MSL		
	P	R	F1	P	R	F1	P	R	F1
DRPAD	80.36	12.30	**21.34**	51.61	44.44	**47.76**	32.47	15.55	**21.03**
DRPAD-SN	85.23	10.82	19.20	18.77	23.50	20.87	32.47	15.55	**21.03**
DRPAD-SN+LIN	98.03	5.82	10.99	22.19	38.46	28.15	17.37	2.06	3.68
DRPAD-DPR	93.42	5.55	10.47	43.63	32.91	37.52	28.21	4.02	7.03
DRPAD-DPR+PAM	88.00	2.58	5.01	43.47	32.69	37.32	32.57	1.83	3.46
DRPAD-MDDC	80.36	12.30	**21.34**	51.39	43.38	47.05	32.47	15.55	**21.03**
DRPAD-DPR-MDDC	93.42	5.55	10.47	42.15	30.98	35.71	28.21	4.02	7.03
DRPAD-SN-MDDC	85.23	10.82	19.20	16.57	23.29	19.36	32.47	15.55	**21.03**
DRPAD-SN-DPR-MDDC	91.45	5.43	10.25	26.74	19.66	22.66	33.09	4.71	8.25
Framework	NAB			PSM			SMAP		
	P	R	F1	P	R	F1	P	R	F1
DRPAD	64.29	37.50	**47.37**	51.06	29.17	**37.13**	13.47	8.23	**10.22**
DRPAD-SN	64.29	37.50	**47.37**	59.47	10.58	17.96	12.96	7.95	9.86
DRPAD-SN+LIN	47.37	37.50	41.86	54.09	2.90	5.51	15.04	1.41	2.57
DRPAD-DPR	47.37	37.50	41.86	53.94	10.53	17.61	13.45	1.58	2.82
DRPAD-DPR+PAM	31.25	25.00	27.78	54.01	10.89	18.13	16.75	1.17	2.19
DRPAD-MDDC	52.94	37.50	43.90	58.86	6.17	11.17	13.47	8.23	**10.22**
DRPAD-DPR-MDDC	47.37	37.50	41.86	76.36	4.09	7.77	13.45	1.58	2.82
DRPAD-SN-MDDC	64.29	37.50	**47.37**	69.55	4.45	8.37	12.96	7.95	9.86
DRPAD-SN-DPR-MDDC	42.11	33.33	37.21	57.15	3.06	5.82	13.67	1.60	2.87
Framework	SMD			SWaT			WADI		
	P	R	F1	P	R	F1	P	R	F1
DRPAD	42.94	12.20	19.00	19.24	5.50	8.55	8.04	47.39	**13.74**
DRPAD-SN	17.07	35.41	**23.03**	13.48	7.49	9.63	5.86	37.18	10.13
DRPAD-SN+LIN	7.21	4.37	5.44	11.99	4.11	6.12	5.44	31.48	9.27
DRPAD-DPR	45.39	9.86	16.20	39.35	11.27	**17.53**	11.90	13.60	12.69
DRPAD-DPR+PAM	45.28	9.74	16.04	39.36	11.27	**17.53**	9.84	19.29	13.03
DRPAD-MDDC	59.01	8.37	14.66	67.13	2.31	4.46	7.17	8.68	7.85
DRPAD-DPR-MDDC	58.10	6.98	12.47	61.15	5.03	9.30	15.74	1.36	2.51
DRPAD-SN-MDDC	33.54	4.36	7.72	39.98	1.38	2.66	11.63	10.51	11.05
DRPAD-SN-DPR-MDDC	32.66	3.93	7.01	64.65	5.32	9.83	25.81	2.24	4.11

Table 4: Performance comparison of different methods across various datasets in $\eta = 20\%$

Categorization	Baselines	SMD			MSL			PSM		
		P	R	F1	P	R	F1	P	R	F1
Density Estimation	DAGMM Zong et al. (2018)	14.34%	1.91%	3.37%	29.41%	3.23%	5.81%	68.97%	4.27%	8.04%
Reconstruction	MEMTO Song et al. (2023)	16.08%	2.65%	4.54%	11.09%	1.51%	2.65%	31.58%	1.92%	3.62%
	CAE-M Zhang et al. (2021a)	19.20%	2.47%	4.38%	35.29%	3.87%	6.98%	33.14%	1.85%	3.50%
Prediction	GDN Deng & Hooi (2021a)	16.45%	2.09%	3.70%	29.41%	3.23%	5.81%	38.35%	4.14%	7.47%
	GTA Chen et al. (2021)	27.02%	3.70%	6.50%	57.75%	10.20%	17.34%	74.62%	7.42%	13.02%
	FEDformer Zhou et al. (2022)	43.01%	6.13%	10.73%	44.11%	7.57%	12.92%	63.25%	4.12%	7.74%
	FEDformer_w_AFMF Shen et al. (2024)	41.74%	5.81%	10.20%	13.71%	1.34%	2.44%	65.90%	3.83%	7.25%
	FEDformer_w_DRPAD(our)	56.55%	23.56%	**33.23%**	44.85%	32.48%	**37.66%**	60.18%	41.55%	**47.40%**
Categorization	Baselines	SMAP			MSDS			NAB		
		P	R	F1	P	R	F1	P	R	F1
Density Estimation	DAGMM Zong et al. (2018)	9.41%	1.11%	1.98%	4.83%	3.85%	4.28%	56.66%	70.83%	62.96%
Reconstruction	MEMTO Song et al. (2023)	17.77%	3.26%	5.51%	2.79%	4.49%	3.44%	33.33%	7.14%	11.76%
	CAE-M Zhang et al. (2021a)	9.37%	0.93%	1.69%	4.83%	3.85%	4.28%	56.67%	70.83%	62.96%
Prediction	GDN Deng & Hooi (2021a)	11.02%	1.31%	2.34%	35.61%	32.26%	33.86%	64.86%	100.00%	**78.69%**
	GTA Chen et al. (2021)	22.48%	2.92%	5.16%	25.74%	20.81%	23.02%	62.96%	70.83%	66.67%
	FEDformer Zhou et al. (2022)	28.88%	4.13%	7.23%	54.67%	61.03%	57.64%	62.51%	70.83%	66.41%
	FEDformer_w_AFMF Shen et al. (2024)	15.02%	1.52%	2.76%	71.58%	81.67%	**76.04%**	47.04%	59.00%	52.12%
	FEDformer_w_DRPAD(our)	27.42%	17.96%	**21.50%**	59.29%	70.17%	64.07%	74.89%	82.50%	77.76%
Categorization	Baselines	MBA			WADI			SWaT		
		P	R	F1	P	R	F1	P	R	F1
Density Estimation	DAGMM Zong et al. (2018)	100.00%	5.92%	11.18%	10.30%	6.84%	8.22%	74.96%	3.20%	6.14%
Reconstruction	MEMTO Song et al. (2023)	71.43%	3.11%	5.95%	4.30%	40.84%	7.77%	23.21%	3.04%	5.38%
	CAE-M Zhang et al. (2021a)	99.35%	5.88%	11.11%	16.70%	7.80%	10.63%	74.52%	3.18%	6.11%
Prediction	GDN Deng & Hooi (2021a)	100.00%	5.95%	11.23%	25.30%	1.84%	3.43%	29.83%	4.22%	7.39%
	GTA Chen et al. (2021)	97.79%	6.05%	11.39%	34.84%	3.02%	5.55%	93.45%	4.59%	8.75%
	FEDformer Zhou et al. (2022)	92.24%	5.48%	10.34%	25.97%	2.25%	4.14%	67.30%	6.30%	11.52%
	FEDformer_w_AFMF Shen et al. (2024)	98.32%	3.76%	7.24%	8.41%	0.65%	1.21%	55.59%	5.02%	9.21%
	FEDformer_w_DRPAD(our)	88.96%	20.91%	**33.85%**	12.67%	86.45%	**22.07%**	35.70%	12.19%	**18.14%**

A.2 Advanced Adjustment Strategy

To ensure a comprehensive assessment, our work additionally evaluates our method under the *advanced adjustment strategy* proposed in Kim et al. (2022), employing a threshold parameter $\eta = 20\%$. Under this relaxed criterion, an anomalous segment is considered detected if at least 20% of its constituent points are identified. This approach stands in contrast to our primary evaluation protocol, which adopts a stricter *point-wise detection framework* without post-processing adjustments. Therefore, this appendix provides the relaxed results of the two experiments from the main results section, obtained under advanced adjustment strategies, as shown in Tables 4 and 5.

Under the relatively lenient high-level detection adjustment strategy, as shown in Table 4, our framework combined with *FedFormer* achieves the best F1 score on 7 out of 9 datasets, with an average improvement of 125.96% over the best-performing baseline. Meanwhile, as presented in Table 5, when comparing the same model with and without the DRPAD framework, enabling DRPAD leads to an average relative improvement of 1084.28% in F1 score. These results demonstrate that the DRPAD framework can significantly enhance model performance under both detection strategies in most cases.

Table 5: Performance comparison of models with and without DRPAD framework across multiple datasets in $\eta = 20\%$

Model	MBA			MSDS			MSL		
	P	R	F1	P	R	F1	P	R	F1
Autoformer_wo_DRPAD	82.37%	4.89%	9.23%	53.32%	53.42%	53.32%	43.20%	7.28%	12.46%
Autoformer_w_DRPAD	74.60%	14.23%	**23.15%**$_{\uparrow 150.67\%}$	63.59%	60.90%	**62.21%**$_{\uparrow 16.67\%}$	30.37%	8.74%	**13.34%**$_{\uparrow 6.97\%}$
DLinear_wo_DRPAD	99.34%	5.90%	11.14%	75.61%	93.12%	**83.45%**	39.82%	5.73%	10.02%
DLinear_w_DRPAD	98.45%	7.44%	**13.83%**$_{\uparrow 24.21\%}$	54.06%	65.17%	59.10%$_{\downarrow 29.23\%}$	47.95%	31.34%	**37.82%**$_{\uparrow 277.02\%}$
DeepAR_wo_DRPAD	93.56%	5.79%	10.90%	44.96%	42.95%	43.92%	46.55%	7.07%	12.28%
DeepAR_w_DRPAD	76.24%	69.79%	**71.73%**$_{\uparrow 558.12\%}$	57.36%	68.21%	**62.24%**$_{\uparrow 41.79\%}$	48.62%	24.15%	**32.15%**$_{\uparrow 161.90\%}$
FEDformer_wo_DRPAD	92.24%	5.48%	10.34%	54.67%	61.03%	57.64%	44.11%	7.57%	12.92%
FEDformer_w_DRPAD(our)	88.96%	20.91%	**33.85%**$_{\uparrow 227.49\%}$	59.29%	70.17%	**64.07%**$_{\uparrow 11.18\%}$	44.85%	32.48%	**37.66%**$_{\uparrow 191.45\%}$
GTA_wo_DRPAD	97.79%	6.05%	11.39%	25.74%	20.81%	23.02%	57.75%	10.20%	17.34%
GTA_w_DRPAD	96.28%	6.83%	**12.74%**$_{\uparrow 11.91\%}$	59.04%	60.90%	**59.47%**$_{\uparrow 158.49\%}$	42.37%	23.85%	**30.03%**$_{\uparrow 73.22\%}$
RTNet_wo_DRPAD	96.58%	2.87%	5.57%	65.25%	78.55%	**71.24%**	43.70%	6.89%	11.90%
RTNet_w_DRPAD	88.93%	25.66%	**39.71%**$_{\uparrow 613.15\%}$	56.50%	92.01%	69.95%$_{\downarrow 1.83\%}$	44.83%	29.72%	**35.60%**$_{\uparrow 198.99\%}$

Model	NAB			PSM			SMAP		
	P	R	F1	P	R	F1	P	R	F1
Autoformer_wo_DRPAD	66.87%	82.50%	73.68%	62.72%	4.08%	7.65%	13.97%	1.72%	3.07%
Autoformer_w_DRPAD	71.86%	94.17%	**81.28%**$_{\uparrow 10.30\%}$	50.44%	81.00%	**62.17%**$_{\uparrow 712.82\%}$	17.43%	13.20%	**13.64%**$_{\uparrow 344.35\%}$
DLinear_wo_DRPAD	65.56%	82.50%	72.85%	67.00%	3.80%	7.19%	13.94%	1.71%	3.05%
DLinear_w_DRPAD	68.52%	76.67%	**72.14%**$_{\downarrow 0.96\%}$	55.82%	22.75%	**32.33%**$_{\uparrow 349.55\%}$	12.46%	4.01%	**6.06%**$_{\uparrow 98.55\%}$
DeepAR_wo_DRPAD	64.92%	76.67%	70.18%	61.60%	3.58%	6.77%	17.35%	2.20%	3.91%
DeepAR_w_DRPAD	73.24%	100.00%	**84.53%**$_{\uparrow 20.44\%}$	49.60%	55.49%	**52.18%**$_{\uparrow 671.61\%}$	14.12%	7.13%	**9.42%**$_{\uparrow 141.07\%}$
FEDformer_wo_DRPAD	62.51%	70.83%	66.41%	63.25%	4.12%	7.74%	28.88%	4.13%	7.23%
FEDformer_w_DRPAD(our)	74.89%	82.50%	**77.76%**$_{\uparrow 17.09\%}$	60.18%	41.55%	**47.40%**$_{\uparrow 512.98\%}$	27.42%	17.96%	**21.50%**$_{\uparrow 197.37\%}$
GTA_wo_DRPAD	62.96%	70.83%	66.67%	74.62%	7.42%	13.02%	22.48%	2.92%	5.16%
GTA_w_DRPAD	64.97%	76.67%	**70.15%**$_{\uparrow 5.22\%}$	63.15%	11.49%	**19.44%**$_{\uparrow 49.42\%}$	18.88%	10.40%	**13.41%**$_{\uparrow 159.70\%}$
RTNet_wo_DRPAD	64.42%	70.83%	67.47%	74.31%	5.06%	9.47%	17.09%	2.11%	3.75%
RTNet_w_DRPAD	80.95%	70.83%	**75.56%**$_{\uparrow 12.00\%}$	55.12%	23.03%	**32.49%**$_{\uparrow 242.86\%}$	15.94%	5.93%	**8.62%**$_{\uparrow 129.93\%}$

Model	SMD			SWaT			WADI		
	P	R	F1	P	R	F1	P	R	F1
Autoformer_wo_DRPAD	47.80%	6.83%	11.95%	80.42%	5.03%	9.47%	2.31%	0.20%	0.37%
Autoformer_w_DRPAD	28.96%	33.21%	**21.54%**$_{\uparrow 80.20\%}$	27.29%	50.01%	**30.04%**$_{\uparrow 217.30\%}$	14.89%	78.18%	**24.84%**$_{\uparrow 6632.57\%}$
DLinear_wo_DRPAD	51.52%	7.49%	13.08%	15.33%	0.63%	1.21%	4.24%	0.37%	0.68%
DLinear_w_DRPAD	55.25%	23.74%	**33.20%**$_{\uparrow 153.92\%}$	21.75%	15.42%	**18.05%**$_{\uparrow 1388.41\%}$	15.70%	100.00%	**27.14%**$_{\uparrow 3914.15\%}$
DeepAR_wo_DRPAD	30.78%	4.23%	7.44%	84.69%	3.80%	7.28%	0.47%	0.04%	0.07%
DeepAR_w_DRPAD	54.43%	21.12%	**30.40%**$_{\uparrow 308.15\%}$	30.64%	5.11%	**8.75%**$_{\uparrow 20.18\%}$	14.62%	79.75%	**24.52%**$_{\uparrow 33118.46\%}$
FEDformer_wo_DRPAD	43.01%	6.13%	10.73%	67.30%	6.30%	11.52%	25.97%	2.25%	4.14%
FEDformer_w_DRPAD(our)	56.55%	23.56%	**33.23%**$_{\uparrow 209.84\%}$	35.70%	12.19%	**18.14%**$_{\uparrow 57.43\%}$	12.67%	86.45%	**22.07%**$_{\uparrow 433.11\%}$
GTA_wo_DRPAD	27.02%	3.70%	6.50%	93.45%	4.59%	8.75%	34.84%	3.02%	5.55%
GTA_w_DRPAD	58.10%	22.35%	**32.27%**$_{\uparrow 395.92\%}$	47.55%	5.05%	**9.06%**$_{\uparrow 3.62\%}$	17.29%	34.22%	**22.95%**$_{\uparrow 313.47\%}$
RTNet_wo_DRPAD	45.65%	6.53%	11.42%	90.13%	4.43%	8.44%	3.77%	0.33%	0.60%
RTNet_w_DRPAD	55.60%	24.73%	**34.23%**$_{\uparrow 199.70\%}$	23.03%	6.33%	**9.92%**$_{\uparrow 17.51\%}$	18.26%	71.06%	**29.02%**$_{\uparrow 4727.82\%}$

B VISUAL EVIDENCE OF SEGMENT-WISE NORMALIZATION ON REAL-WORLD DATA

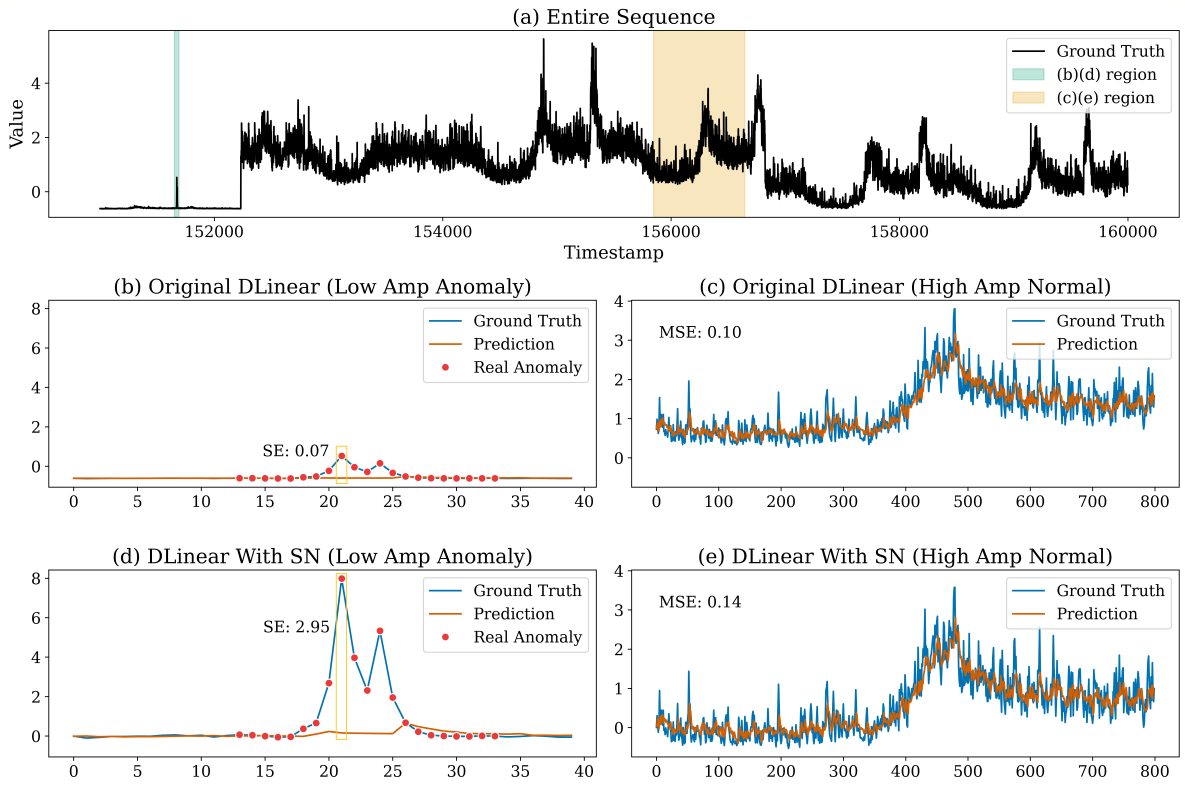

Figure 3: Visualization of Anomaly Detection Performance under Global vs. Segment-wise Normalization. This figure compares the performance of our proposed segment-wise normalization (SN Model) with conventional global normalization (Original DLinear) in time series anomaly detection. The top panel displays the entire sequence with ground truth values, highlighting low-amplitude (left) and high-amplitude (right) regions. The bottom panels illustrate the squared error (SE) for a single dimension in these regions. In the Original DLinear model, the SE of anomalies in the low-amplitude region (b) is overshadowed by the higher MSE of normal values in the high-amplitude region (c), resulting in undetected anomalies. However, with segment-wise normalization (DLinear with SN), the SE in the low-amplitude anomalous region (d) exceeds the MSE in the high-amplitude normal region (e), enabling effective detection. Metrics shown include SE for anomalies and MSE for the segments.

As shown in Figure 3, under the global normalization scheme, statistical properties such as standard deviation are dominated by segments with large fluctuations or extreme outliers. As a result, anomalies occurring in segments with relatively low variance may produce only small standardized errors and thus be overlooked. For instance, in the low-amplitude region, the anomaly under the Original DLinear yields a low SE of only **0.07**, even lower than the MSE of normal fluctuations in the high-amplitude region, which is **0.10**. Consequently, the anomaly in the low-amplitude region is missed.

By contrast, our SN Model applies change point detection to partition the sequence into statistically consistent segments and performs normalization within each segment independently. This allows local anomalies to be evaluated under fairer statistical scales. In the low-amplitude region, the anomaly becomes much more distinguishable under SN normalization, with SE increasing to **2.95**, exceeding the MSE in the high-amplitude region of **0.14**, enabling effective detection.

Note that the error depicted in the left plots represents the squared error (SE) for a single dimension. The MSE shown on the right side refers to the mean squared error averaged across the entire high-amplitude region for that single dimension.

C NOTATION SUMMARY

Table 6: Notation Summary

Symbol	Description
x_t^n	Time series observation of the n-th dimension at time step t
H_t	Input window at time t
A_t	Anomaly indicator at time step t
c	Continuous anomaly count counter
r	Percentile threshold for anomaly detection
N	Number of data features
δ	Maximum allowed consecutive anomalies
ε_t	Gaussian noise, $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$
σ^2	Variance of the Gaussian noise
Δ	Anomalous deviation, $\Delta \sim \mathcal{D}$
$\mathcal{D}$	Distribution of Δ with mean μ_Δ and variance σ_Δ^2
μ_Δ	Mean of the anomalous deviation
σ_Δ^2	Variance of the anomalous deviation
L	Length of the input window for prediction
$\hat{x}_t$	Model prediction at time step t
w_j	Weight corresponding to the j-th lagged input
b	Bias term of the prediction model
i	Index of the anomaly within the input window
e_{t-i}	Prediction error at time $t-i$
σ_e^2	Variance of historical prediction errors
$\varepsilon_{\max}$	Upper bound of Gaussian noise
$\mathbb{I}(\cdot)$	Indicator function
$\mathbf{S}_j$	Temporal segments segmented based on change points
μ_j	Variance of each segment.
σ_j	Mean and variance of each segment.
φ	Sensitivity of the dimension-wise anomaly detection threshold
η	Threshold parameter for advanced adjustment strategy

D DETAILED MATHEMATICAL PROOF

This paper proposes a dynamic replacement strategy: when an anomaly is detected, the model's prediction is used to replace the true value for subsequent forecasting. To verify the effectiveness of this strategy, this section provides a step-by-step mathematical proof. The essence of the dynamic replacement strategy is to enhance forecasting robustness by iteratively correcting the reliability of the input sequence. We use a linear model as the theoretical tool due to its transparency for analyzing anomaly propagation mechanisms. The strategy can be directly extended to nonlinear models (see Appendix D.10). Specifically, we derive general conclusions by considering the case where the input window contains only a single anomaly.

We assume a single anomaly in the input window. Suppose the anomaly introduces a fixed deviation Δ compared to the true value. We first analyze the case where Δ is a deterministic value, and then generalize to the case where Δ follows an arbitrary distribution. Based on this, we prove that under certain conditions, the dynamic replacement strategy can effectively reduce the impact of anomalies on the prediction results, thereby improving forecasting accuracy. The detailed proof is as follows:

D.1 DATA GENERATION MODEL

To simulate the normal patterns of time series data in a general manner, we assume an arbitrary underlying function $f(t)$ that satisfies the Lipschitz continuity condition, ensuring the sequence is sufficiently smooth. Specifically, $f(t)$ is Lipschitz continuous if there exists a constant $K > 0$ such that for all t_1, t_2,

$$|f(t_1) - f(t_2)| \leq K|t_1 - t_2|.$$

This condition guarantees bounded variation and prevents abrupt changes in the normal data patterns.

We construct the training set standard time series using this function:

$$x_t = f(t).$$

The test set standard time series is constructed by superimposing Gaussian noise on the function:

$$x_t = f(t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2),$$

where ε_t is Gaussian noise with mean 0 and variance σ^2. To introduce anomalies, we add a fixed deviation Δ at a random time k, generating an anomalous data point as:

$$x_k = f(k) + \varepsilon_k + \Delta.$$

Based on this setup, we use a single-layer fully connected neural network as the prediction model, with the input being the past L time steps and the output being the next time step's prediction:

$$\hat{x}_t = \sum_{j=1}^{L} w_j x_{t-j} + b,$$

where w_j denotes the weight corresponding to x_{t-j}, that is, $[f(t-1), f(t-2), \ldots, f(t-L)]$ correspond to $[w_1, w_2, \ldots, w_L]$. We assume the model has been trained sufficiently on clean data so that the weights w_j and bias b have converged to optimal values, allowing the model to accurately predict the underlying signal without noise or anomalies:

$$\sum_{j=1}^{L} w_j f(t-j) + b \approx f(t).$$

This assumption ensures that the network can accurately fit the normal time series in the absence of anomalies, laying the foundation for the subsequent analysis of anomaly impact and the effectiveness of the dynamic replacement strategy.

In this study, we design a control group and an experimental group to evaluate the effectiveness of the dynamic replacement strategy.

The **control group** uses the traditional forecasting method, i.e., modeling and predicting directly on the entire time series without correcting the detected anomalies. The input to the model may thus contain anomalies, and predictions are made based on these inputs. The results of the control group help measure the degradation of predictive performance due to the presence of anomalies.

The **experimental group** uses the dynamic replacement strategy, where detected anomalies are replaced by the model's predicted values, and the modified sequence is then used for subsequent predictions. The core idea is to weaken the influence of anomalies on future forecasts and improve overall prediction accuracy. The MSE results of the experimental group can evaluate the strategy's effectiveness in mitigating anomaly interference.

By comparing the control and experimental groups, we can quantify the advantages of the dynamic replacement strategy under different anomaly types and distribution conditions, and further analyze its applicability and limitations.

D.2 Error Analysis of Control Group (Without Replacing Anomalies)

Control Group (No Replacement):

Suppose at time t, the input window contains an anomaly at time step $t-i$ (random moment k), where

$$x_{t-i} = f(t-i) + \varepsilon_{t-i} + \Delta.$$

Then the predicted value is:

$$\hat{x}_t = \underbrace{\sum_{j \neq i}^{L} w_j \left(f(t-j) + \varepsilon_{t-j}\right) + b}_{\text{normal prediction terms}} + w_i \left(f(t-i) + \varepsilon_{t-i} + \Delta\right).$$

Simplifying:

$$\hat{x}_t = \sum_{j=1}^{L} w_j f(t-j) + b + \sum_{j=1}^{L} w_j \varepsilon_{t-j} + w_i \Delta.$$

Given the model assumption:

$$\sum_{j=1}^{L} w_j f(t-j) + b \approx f(t).$$

the prediction error is:

$$e_t = \hat{x}_t - \left(f(t) + \varepsilon_t\right) = \underbrace{\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t}_{\text{noise error term}} + w_i \Delta.$$

The mean squared error (MSE) is defined as:

$$\text{MSE} = \mathbb{E}[e_t^2].$$

Substituting e_t:

$$e_t^2 = \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t + w_i \Delta \right)^2.$$

Expanding the square:

$$e_t^2 = \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right)^2 + 2 \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right) (w_i \Delta) + (w_i \Delta)^2,$$

where:ε_{t-j} and ε_t are Gaussian noises with $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ and are assumed to be independent. w_j and w_i are constants (model weights).

Since the expectation operator $\mathbb{E}[\cdot]$ is linear: - The second term's expectation is:

$$\mathbb{E}\left[2 \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right) (w_i \Delta) \right] = 2 w_i \Delta \cdot \mathbb{E}\left[\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right] = 0.$$

- The third term's expectation is:

$$\mathbb{E}\left[(w_i \Delta)^2 \right] = w_i^2 \Delta^2.$$

For the first term:

$$\mathbb{E}\left[\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right)^2 \right] = \text{Var}\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right).$$

For a linear combination $X = \sum_k a_k Y_k$, the variance is:

$$\text{Var}\left(\sum_k a_k Y_k \right) = \sum_k a_k^2 \text{Var}(Y_k) + 2 \sum_{k<l} a_k a_l \text{Cov}(Y_k, Y_l).$$

Since the noises are independent:

$$\text{Cov}(\varepsilon_{t-j}, \varepsilon_t) = 0.$$

Thus:

$$\text{Var}\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right) = \sum_{j=1}^{L} w_j^2 \sigma^2 + \sigma^2.$$

Substituting into the MSE expression:

$$\text{MSE}_{\text{control}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2 \right) + w_i^2 \Delta^2. \tag{4}$$

When Δ follows an arbitrary distribution $\mathcal{D}$:

$$\Delta_k \sim \mathcal{D}.$$

with mean μ_Δ and variance σ_Δ^2. In practical time series anomaly detection, the second moment of anomalies often exceeds that of Gaussian noise:

$$\mathbb{E}[\Delta_k^2] = \sigma_\Delta^2 + \mu_\Delta^2 > \sigma^2.$$

The computation for MSE remains consistent, except that:

$$\mathbb{E}\left[(w_i \Delta)^2 \right] = w_i^2 (\sigma_\Delta^2 + \mu_\Delta^2).$$

Thus, the MSE of the control group under an arbitrary distribution is:

$$\text{MSE}_{\text{control, arbitrary distribution}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2 \right) + w_i^2 (\sigma_\Delta^2 + \mu_\Delta^2). \tag{5}$$

D.3 Error Analysis for Experimental Group (Dynamic Prediction Replacement)

The experimental group replaces outliers x_{t-i} with historical predictions $\hat{x}_{t-i}$. The replacement value is defined as:

$$x'_{t-i} = \hat{x}_{t-i} = f(t-i) + \varepsilon_{t-i} + e_{t-i},$$

where $e_{t-i} \triangleq \hat{x}_{t-i} - [f(t-i) + \varepsilon_{t-i}]$ represents the historical prediction error. From Appendix D.11, we have established that $\mathbb{E}[e_t] = 0$ for any time t, and let $\mathrm{Var}(e_{t-i}) = \sigma_e^2$.

Prediction Error Derivation

Following similar derivation logic as the control group, the prediction becomes:

$$\hat{x}'_t = \underbrace{\sum_{j=1}^{L} w_j f(t-j) + b}_{\text{Normal prediction term}} + w_i e_{t-i}.$$

The prediction error is then:

$$\begin{aligned} e'_t &= \hat{x}'_t - [f(t) + \varepsilon_t] \\ &= \left[\sum_{j=1}^{L} w_j f(t-j) + b + w_i e_{t-i} \right] - [f(t) + \varepsilon_t] \\ &= \underbrace{\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t}_{\text{Noise error term}} + w_i e_{t-i}. \end{aligned} \tag{6}$$

MSE Decomposition

The mean squared error (MSE) is given by $\mathrm{MSE} = \mathbb{E}[{e'_t}^2]$. Expanding $(e'_t)^2$:

$$(e'_t)^2 = \underbrace{\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right)^2}_{A} + \underbrace{2 \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right) (w_i e_{t-i})}_{B} + \underbrace{(w_i e_{t-i})^2}_{C}. \tag{7}$$

D.3.1 Term A Analysis

$$\begin{aligned} \mathbb{E}[A] &= \mathbb{E}\left[\left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} - \varepsilon_t \right)^2 \right] \\ &= \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2 \right). \end{aligned}$$

This matches the control group's noise error variance derivation.

D.3.2 Term C Analysis

$$\mathbb{E}[C] = w_i^2 \mathrm{Var}(e_{t-i}) = w_i^2 \sigma_e^2.$$

D.3.3 Term B Analysis

Figure 4 illustrates the temporal structure of the input sequence used for autoregressive prediction, highlighting the influence of dynamic anomaly replacement on prediction error. The lower two timelines depict how an anomalous input x_{t-i} (marked in orange) is involved in both the prediction of x_t and the historical prediction of x_{t-i} itself. The top timeline decomposes the weight allocation into two regions: the first i terms (affected by the anomaly through e_{t-i}), and the remaining $L - i$ terms, which may share overlapping noise components due to common history. This overlap results in cross-terms such as $\mathbb{E}[e_{t-i}\varepsilon_{t-j}]$ in the error expansion, breaking independence and introducing additional variance. Such dependency explains the emergence of the term $2w_i\sigma^2(\sum_{k=1}^{L-i} w_{i+k}w_k - w_i)$ in the MSE derivation.

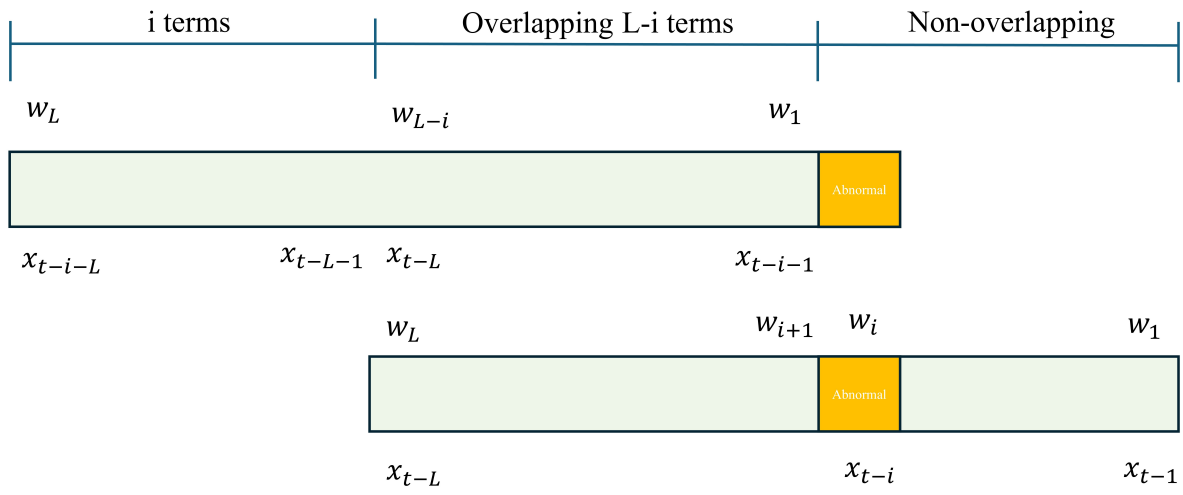

Figure 4: Illustration of Temporal Input Structure and Cross-Term Interference in Dynamic Replacement.

So the cross-term expectation in Equation (7) requires careful analysis:

$$\begin{aligned}\mathbb{E}[B] &= 2w_i\mathbb{E}\left[\left(\sum_{j=1}^{L} w_j\varepsilon_{t-j} - \varepsilon_t\right) e_{t-i}\right] \\ &= 2w_i\left(\sum_{j=1}^{L} w_j\mathbb{E}[\varepsilon_{t-j}e_{t-i}] - \underbrace{\mathbb{E}[\varepsilon_t e_{t-i}]}_{0}\right),\end{aligned}$$

where $\mathbb{E}[\varepsilon_t e_{t-i}] = 0$ due to temporal independence.

Following the same decomposition as in Equation (6), the historical prediction error e_{t-i} is given by:

$$e_{t-i} = \sum_{k=1}^{L} w_k\varepsilon_{t-i-k} - \varepsilon_{t-i}. \tag{8}$$

This decomposition comes from the model's training on normal data where $\sum_{k=1}^{L} w_k f(t-i-k) + b \approx f(t-i)$.

As illustrated in Figure 4, by substituting Equation (8) into the expectation, we obtain:

$$\sum_{j=1}^{L} w_j \mathbb{E}[\varepsilon_{t-j} e_{t-i}] = \sum_{j=1}^{L} w_j \mathbb{E}\left[\varepsilon_{t-j}\left(\sum_{k=1}^{L} w_k \varepsilon_{t-i-k} - \varepsilon_{t-i}\right)\right]$$

(Substituting the expression for e_{t-i} from Equation equation 8)

$$= \sum_{j=1}^{L}\sum_{k=1}^{L} w_j w_k \mathbb{E}[\varepsilon_{t-j}\varepsilon_{t-i-k}] - \sum_{j=1}^{L} w_j \mathbb{E}[\varepsilon_{t-j}\varepsilon_{t-i}]$$

(Distributing the expectation and weights)

$$= \sum_{k=1}^{L} w_k \left(\sum_{j=1}^{L} w_j \mathbb{E}[\varepsilon_{t-j}\varepsilon_{t-i-k}]\right) - \sum_{j=1}^{L} w_j \mathbb{E}[\varepsilon_{t-j}\varepsilon_{t-i}]$$

(Reordering summation operations)

$$= \sum_{k=1}^{L} w_k \left(\sigma^2 \sum_{j=1}^{L} w_j \delta_{j,i+k}\right) - \sigma^2 \sum_{j=1}^{L} w_j \delta_{j,i}$$

(Applying i.i.d. noise property: $\mathbb{E}[\varepsilon_a \varepsilon_b] = \sigma^2 \delta_{a,b}$)

$$= \sigma^2 \sum_{k=1}^{L} w_k w_{i+k} \mathbb{I}(i+k \le L) - \sigma^2 w_i$$

(Evaluating Kronecker delta $\delta_{j,i+k}$)

$$= \sigma^2 \left(\sum_{k=1}^{L-i} w_k w_{i+k}\right) - \sigma^2 w_i$$

(Truncating sum since $w_{i+k} = 0$ for $i + k > L$)

$$= \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right), \tag{9}$$

where we use the following mathematical constructs:

- **Kronecker delta**: $\delta_{a,b} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$
- **Indicator function**: $\mathbb{I}(P) = \begin{cases} 1 & \text{if proposition } P \text{ is true} \\ 0 & \text{otherwise} \end{cases}$
- **Boundary condition**: $w_m = 0$ for all $m > L$

The key insight comes from the temporal alignment condition:

$$\mathbb{E}[\varepsilon_{t-j}\varepsilon_{t-i-k}] = \sigma^2 \delta_{j,i+k} = \begin{cases} \sigma^2, & \text{if } t - j = t - i - k \ (j = i + k), \\ 0, & \text{otherwise.} \end{cases}$$

This derivation explicitly shows how the temporal correlations between:

- Current window's noise terms (ε_{t-j})
- Historical prediction error components (ε_{t-i-k})

thus, we generate the weight coupling terms in the final expression:

$$\mathbb{E}[B] = 2 w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

D.4 Final MSE Expression

Combining all components:

$$\text{MSE}_{\text{exp}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2\right) + w_i^2 \sigma_e^2 + 2w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

D.4.1 Analytical Expression for the MSE Difference Between Two Groups

Summarizing:

$$\text{MSE}_{\text{ctrl}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2\right) + w_i^2 \Delta^2.$$

$$\text{MSE}_{\text{exp}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2\right) + w_i^2 \sigma_e^2 + 2w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

Thus, the difference is:

$$\text{MSE}_{\text{ctrl}} - \text{MSE}_{\text{exp}} = w_i^2 (\Delta^2 - \sigma_e^2) - 2w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

When Δ is extended to a random variable with mean μ_Δ and variance σ_Δ^2, the difference becomes:

$$\text{MSE}_{\text{ctrl}} - \text{MSE}_{\text{exp}} = w_i^2 (\sigma_\Delta^2 + \mu_\Delta^2 - \sigma_e^2) - 2w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i\right).$$

The experimental group outperforms the control group when:

$$\sigma_\Delta^2 + \mu_\Delta^2 > \sigma_e^2 + 2\sigma^2 \left(\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} - 1\right).$$

where $\sigma_e^2 = \sigma^2 \left(\sum_{i=1}^{L} w_i^2 + 1\right)$. Combining terms, the inequality becomes:

$$\sigma_\Delta^2 + \mu_\Delta^2 > \sigma^2 \left(\sum_{i=1}^{L} w_i^2 + 1\right) + 2\sigma^2 \left[\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} - 1\right], \tag{10}$$

where $\sigma_\Delta^2 + \mu_\Delta^2$ represents the second moment of the anomaly signal. The presence of the regression weight w_i in the denominator, which depends on data-driven estimates, renders analytical derivation of a closed-form guarantee for equation 10 intractable. To address this, we conducted an extensive numerical simulation study to empirically evaluate the probability that the inequality holds, thereby assessing the practical robustness of the method.

D.5 Numerical Simulation

To validate the inequality equation 10, we conducted simulations on time series data satisfying the Lipschitz smoothness condition, which ensures bounded gradients. We generated sequences using a random walk process smoothed with a Gaussian filter (sigma = 2.0):

$$x_t = \sum_{s=1}^{t} \eta_s, \quad \eta_s \sim \mathcal{N}(0, 0.3^2),$$

followed by convolution with a Gaussian kernel to enforce smoothness and the Lipschitz condition while capturing temporal dependencies and stochastic fluctuations. All sequences were normalized to the unit interval $[0, 1]$, with μ and σ^2 representing the mean and variance of the normalized sequence.

For each sequence of length $n + L$, we constructed a lagged feature matrix $X \in \mathbb{R}^{n \times L}$ and target vector $y \in \mathbb{R}^n$, fitting a ridge regression (with L2 regularization) to obtain weights $w \in \mathbb{R}^L$. The regularization parameter λ

was adaptively selected based on the condition number of $X^\top X$, ranging from 10^{-6} to 10^{-3} times the average eigenvalue to ensure numerical stability. The noise variance σ^2 was estimated from the residuals. The anomaly second moment $\mu_\Delta^2 + \sigma_\Delta^2$ was approximated using the derived form $\mu^2 + 8.575\mu\sigma + 20.014\sigma^2$, and the right-hand side threshold was computed as $\sigma^2 \left(\sum_{i=1}^{L} w_i^2 + 1\right) + 2\sigma^2 \left(Q_i - 1\right)$, where $Q_i = \frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i}$. The inequality was evaluated for each valid Q_i (where $|w_i| > 10^{-30}$ to avoid division-by-zero errors), yielding the effectiveness probability as the proportion of indices i for which the inequality holds.

To ensure robustness, we performed a grid search over sample sizes $n \in \{200, 500, 1000, 5000\}$ and lag windows $L \in \{10, 20, 50, 100\}$, resulting in 16 configurations. Each configuration was tested with 100 independent experiments using distinct random seeds. As shown in the figure, across all 1600 experiments, the overall mean effectiveness probability was 0.9998 ± 0.0035, indicating that the inequality holds with approximately 99.98% probability and low variability. These results provide strong empirical support for the method's reliability on Lipschitz-smooth time series in finite-sample settings.

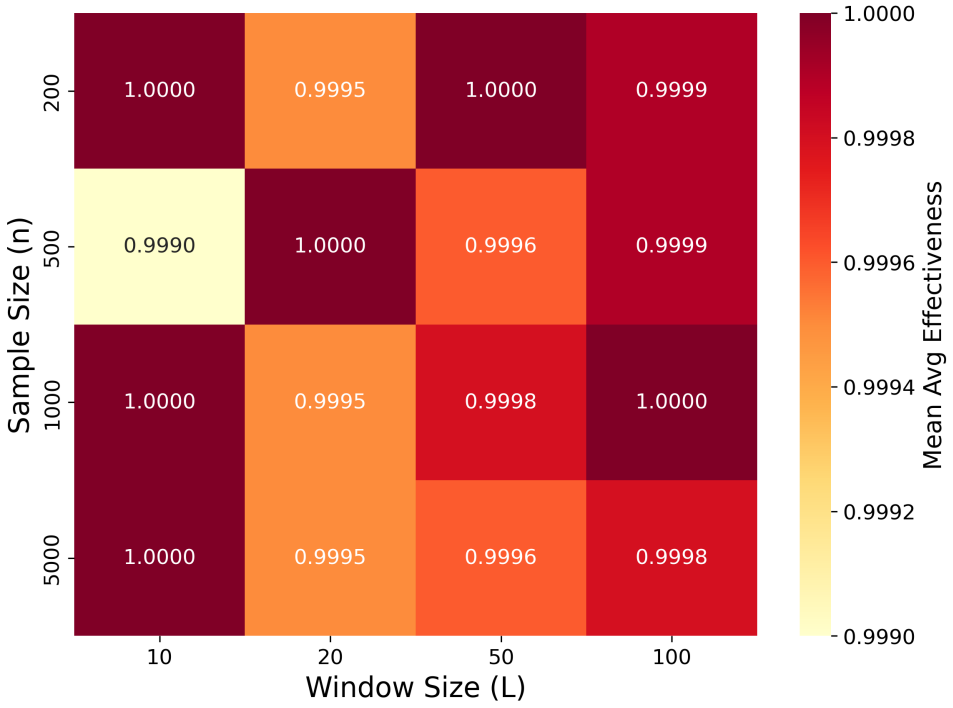

Figure 5: Heatmap of Mean Average Effectiveness Across Sample and Window Sizes. Each cell represents the average effectiveness probability from 100 independent experiments, with values ranging from 0.9990 to 1.0000. The color gradient, from light yellow (lower effectiveness) to dark red (higher effectiveness), highlights the robustness of the inequality, with most configurations achieving probabilities near or at 1.0000, indicating near-certain satisfaction across the tested parameter space. This visual representation complements the numerical findings, reinforcing the method's reliability for Lipschitz-smooth time series under varying data conditions.

D.5.1 Supplement: Analytical Derivation of the Anomaly Second Moment

In the context of anomaly detection, let the original time series random variable X have mean $E[X] = \mu$ and variance $\mathrm{Var}[X] = \sigma^2$. Anomalies are introduced by injecting a bias δ at random positions, ensuring detectability under the 3-sigma rule. Specifically, the value at an anomaly point is $A = X + \delta$, where δ follows a truncated normal distribution $N(\mu_\delta = 4\sigma, \tau^2 = \sigma^2)$ with $\delta \geq 3\sigma$, and X and δ are assumed independent. Our goal is to compute the second moment $E[A^2] = \mathrm{Var}[A] + [E[A]]^2$.

Mean of the Anomaly: $E[A]$

Since $A = X + \delta$ and X and δ are independent, the mean is:

$$E[A] = E[X] + E[\delta] = \mu + E[\delta].$$

For $\delta \sim N(\mu_\delta = 4\sigma, \tau^2 = \sigma^2)$ truncated at $\delta \geq 3\sigma$, the conditional expectation of a truncated normal distribution is:

$$E[\delta \mid \delta \geq 3\sigma] = \mu_\delta + \tau \cdot \frac{\phi\left(\frac{a - \mu_\delta}{\tau}\right)}{1 - \Phi\left(\frac{a - \mu_\delta}{\tau}\right)},$$

where $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$ is the standard normal probability density function, $\Phi(z) = P(Z \leq z)$ is the cumulative distribution function, and the truncation point is $a = 3\sigma$. Let the standardized variable be:

$$z = \frac{a - \mu_\delta}{\tau} = \frac{3\sigma - 4\sigma}{\sigma} = -1.$$

Using standard normal tables, $\phi(-1) \approx 0.2419707$ and $\Phi(-1) \approx 0.1586553$, so $1 - \Phi(-1) \approx 0.8413447$. Thus:

$$E[\delta \mid \delta \geq 3\sigma] = 4\sigma + \sigma \cdot \frac{0.2419707}{0.8413447} \approx 4.2877\sigma.$$

Hence:

$$E[A] = \mu + 4.2877\sigma.$$

VARIANCE OF THE ANOMALY: $\mathrm{Var}[A]$

Since X and δ are independent, the variance is:

$$\mathrm{Var}[A] = \mathrm{Var}[X] + \mathrm{Var}[\delta] = \sigma^2 + \mathrm{Var}[\delta].$$

The variance of the truncated normal distribution is:

$$\mathrm{Var}[\delta \mid \delta \geq 3\sigma] = \tau^2 \left[1 + \frac{\frac{a-\mu_\delta}{\tau} \phi\left(\frac{a-\mu_\delta}{\tau}\right)}{1 - \Phi\left(\frac{a-\mu_\delta}{\tau}\right)} - \left(\frac{\phi\left(\frac{a-\mu_\delta}{\tau}\right)}{1 - \Phi\left(\frac{a-\mu_\delta}{\tau}\right)} \right)^2 \right].$$

Substituting $\tau = \sigma$, $a = 3\sigma$, $\mu_\delta = 4\sigma$, and $\frac{a-\mu_\delta}{\tau} = -1$, with $\phi(-1) \approx 0.2419707$ and $1 - \Phi(-1) \approx 0.8413447$, we compute:

$$\frac{\phi(-1)}{1 - \Phi(-1)} \approx \frac{0.2419707}{0.8413447} \approx 0.2876821,$$

$$\frac{a - \mu_\delta}{\tau} \cdot \frac{\phi\left(\frac{a-\mu_\delta}{\tau}\right)}{1 - \Phi\left(\frac{a-\mu_\delta}{\tau}\right)} = (-1) \cdot 0.2876821 \approx -0.2876821,$$

$$\left(\frac{\phi(-1)}{1 - \Phi(-1)} \right)^2 \approx (0.2876821)^2 \approx 0.0827608.$$

Thus:

$$\mathrm{Var}[\delta] = \sigma^2 \left[1 - 0.2876821 - 0.0827608 \right] \approx 0.6296\sigma^2.$$

Therefore:

$$\mathrm{Var}[A] = \sigma^2 + 0.6296\sigma^2 \approx 1.6296\sigma^2.$$

SECOND MOMENT: $E[A^2]$

The second moment is given by:

$$E[A^2] = \mathrm{Var}[A] + [E[A]]^2.$$

Substituting $E[A] = \mu + 4.2877\sigma$ and $\mathrm{Var}[A] \approx 1.6296\sigma^2$, we obtain:

$$E[A^2] = 1.6296\sigma^2 + (\mu + 4.2877\sigma)^2 = \mu^2 + 8.5754\mu\sigma + 20.0142\sigma^2.$$

In simulations, we used the approximated coefficients (8.575 and 20.014), which are consistent with the analytical result within numerical rounding.

D.6 UPPER BOUND ANALYSIS OF Z UNDER THE SINUSOIDAL MODEL

We aim to derive an upper bound for the right-hand side of the key inequality:

$$\sigma_\Delta^2 + \mu_\Delta^2 > \sigma^2 \left(\sum_{i=1}^{L} w_i^2 + 1 \right) + 2\sigma^2 \left[\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} - 1 \right],$$

We define:

$$Z = \sigma_e^2 + 2\sigma^2 \left(\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} - 1 \right).$$

While the numerical simulations provide robust empirical evidence that the inequality holds with high probability across a range of practical settings, offering confidence in the method's applicability to general Lipschitz-smooth time series, deriving a closed-form analytical guarantee remains challenging due to the data-dependent nature of the regression weights. To gain deeper theoretical insights and enable further tractable analysis of the upper bound on Z, we now consider a simplified yet representative data generation model. Specifically, we adopt a sinusoidal function to model the underlying time series, which captures periodic behaviors commonly observed in real-world signals while allowing explicit computation of the weights and bounds. This specialization facilitates the derivation of analytical expressions without loss of generality for the core principles, bridging the empirical findings to precise theoretical results.

Data Generation Model

To simulate normal time series patterns, we substitute a sine function for the arbitrary underlying function $f(t)$ when constructing the standard training time series, defined as:

$$x_t = \sin(t),$$

which preserves the structure of the derivation and leads to the same inequality condition, while enabling tractable analysis.

The test set standard time series is constructed by superimposing Gaussian noise on the sine function:

$$x_t = \sin(t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

where ε_t is Gaussian noise with mean 0 and variance σ^2. To introduce anomalies, we add a fixed deviation Δ at a random time k, generating an anomalous data point as:

$$x_k = \sin(k) + \varepsilon_k + \Delta.$$

We begin by analyzing the upper bound of the variance term Z defined as:

$$Z = \sigma_e^2 + 2\sigma^2 \left[\underbrace{\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i}}_{Q} - 1 \right]. \tag{11}$$

where σ_e^2 represents the error variance and Q is a correlation term between weight vectors.

D.6.1 Upper Bound of σ_e^2

From Appendix D.12, we have the expression for the error variance in Equation (11):

$$\sigma_e^2 = \sigma^2 \left(\sum_{j=1}^{L} w_j^2 + 1 \right).$$

The weight coefficients w_j are given by the cosine weighting function:

$$w_j = \frac{2}{L} \cos(j).$$

The squared weights therefore satisfy:

$$w_j^2 = \frac{4}{L^2} \cos^2(j).$$

Since $\cos^2(j) \leq 1$ for all j, we can bound the sum of squared weights:

$$\sum_{j=1}^{L} w_j^2 \leq L \cdot \frac{4}{L^2} = \frac{4}{L}.$$

Substituting this into the error variance expression yields:

$$\sigma_e^2 \leq \sigma^2 \left(\frac{4}{L} + 1 \right).$$

This establishes $\sigma^2 \left(\frac{4}{L} + 1 \right)$ as an upper bound for σ_e^2.

D.6.2 Upper Bound of D

To proceed, we analyze the term D in Equation (11) more carefully,

$$D = \frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i} = \frac{2}{L} \frac{\sum_{k=1}^{L-i} \cos(i+k)\cos(k)}{\cos(i)}.$$

Our goal is to find the maximum possible value $D_{\max}$ at a given confidence level p (e.g., 95%), such that $P(D \leq D_{\max}) \geq p$.

Step 1: Simplification Using Trigonometric Identities

Let S denote the summation in the numerator:

$$S = \sum_{k=1}^{L-i} \cos(i+k)\cos(k).$$

Using the product-to-sum identity:

$$\cos A \cos B = \frac{1}{2}[\cos(A+B) + \cos(A-B)].$$

We set $A = i + k$ and $B = k$ to obtain:

$$\cos(i+k)\cos(k) = \frac{1}{2}[\cos(i+2k) + \cos(i)].$$

The summation then decomposes into two parts:

$$S = \frac{1}{2} \sum_{k=1}^{L-i} \cos(i+2k) + \frac{1}{2}\cos(i)(L-i).$$

Substituting back into D:

$$D = \frac{2}{L} \cdot \frac{S}{\cos(i)} = \frac{1}{L} \left[\frac{\sum_{k=1}^{L-i} \cos(i+2k)}{\cos(i)} + (L-i) \right].$$

The remaining summation can be evaluated using the trigonometric sum formulaKnapp (2009):

$$\sum_{k=1}^{N} \cos(\theta + \alpha k) = \frac{\sin\left(\frac{N\alpha}{2}\right) \cos\left(\theta + \frac{(N+1)\alpha}{2}\right)}{\sin\left(\frac{\alpha}{2}\right)}.$$

With $\theta = i$ and $\alpha = 2$, we get:

$$\sum_{k=1}^{L-i} \cos(i+2k) = \frac{\sin(L-i)\cos(L+1)}{\sin(1)}.$$

Thus, D simplifies to:

$$D = \frac{\sin(L-i)\cos(L+1)}{L\sin(1)\cos(i)} + \frac{L-i}{L}.$$

STEP 2: ANALYSIS OF THE DISTRIBUTION OF $\cos i$

Since i is an integer, the values of $\cos i$ are distributed within the interval $[-1, 1]$. To compute the statistical properties of A, we need to characterize the distribution pattern of $\cos i$.

By the Equidistribution Theorem, when i is uniformly distributed across the integers, the expression $\cos i = \cos(i \mod 2\pi)$ implies that $i \mod 2\pi$ becomes asymptotically uniformly distributed in $[0, 2\pi)$ as i ranges over large integer values. This allows us to approximate the distribution of $i \mod 2\pi$ as uniform over $[0, 2\pi)$. Consequently, the cumulative distribution function (CDF) of $\cos i$ can be derived as:

$$P(\cos i \leq c) = 1 - \frac{1}{\pi} \arccos c, \quad c \in [-1, 1]. \tag{12}$$

The derivation of Equation (12) follows from the symmetry of the cosine function. For any $c \in [-1, 1]$, the inequality $\cos\theta \leq c$ holds when θ lies in the union of intervals $[\arccos c, 2\pi - \arccos c]$. The probability measure of this set is given by the ratio of its length to 2π:

$$P(\cos\theta \leq c) = \frac{(2\pi - \arccos c) - \arccos c}{2\pi} = 1 - \frac{1}{\pi} \arccos c.$$

In our problem formulation, the condition D requires $\cos(i) > 0$ (as negative values would be meaningless in this context). This restriction allows us to focus on the positive half of the cosine distribution. By exploiting the symmetry of the cosine function about zero, we can equivalently analyze the distribution of $|\cos i|$, which simplifies our calculations. The probability that $|\cos i|$ exceeds a threshold c is:

$$P(|\cos i| \geq c) = 2 \cdot P(\cos i \geq c) = \frac{2}{\pi} \arccos c.$$

To establish a lower bound with confidence level p, we require:

$$P(|\cos i| \geq c) \geq p \quad \Rightarrow \quad \frac{2}{\pi} \arccos c \geq p.$$

Solving for c and noting that the arccosine function is monotonically decreasing, we obtain:

$$\arccos c \geq \frac{\pi}{2} p \quad \Rightarrow \quad c \leq \cos\left(\frac{\pi}{2} p\right).$$

Thus, the lower bound for $|\cos i|$ at confidence level p is:

$$c_p = \cos\left(\frac{\pi}{2} p\right).$$

For a 95% confidence level ($p = 0.95$), we compute:

$$c_{0.95} = \cos\left(\frac{\pi}{2} \times 0.95\right) \approx 0.0785.$$

This result indicates that with 95% confidence, $|\cos i|$ will be greater than or equal to approximately 0.0785. Only 5% of cases may fall outside this range, which we consider exceptional.

STEP 3: ESTIMATING THE UPPER BOUND OF D

We begin with the following approximation of the term D:

$$D \approx \frac{\sin(L - i)\cos(L + 1)}{\sin(1) \cdot L \cdot \cos(i)} + \frac{L - i}{L}. \tag{13}$$

To estimate the upper bound of D, we leverage the well-known trigonometric inequalities:

$$|\cos(\theta)| \leq 1, \quad |\sin(\theta)| \leq 1.$$

Thus, the numerator in the first term is bounded as:

$$|\sin(L - i)\cos(L + 1)| \leq 1. \tag{14}$$

Next, consider the valid range of i, which satisfies:

$$1 \leq i \leq L-1 \quad \Rightarrow \quad \frac{L-i}{L} < 1.$$

Combining this with inequalities equation 13 and equation 14, we obtain:

$$D \lesssim \frac{1}{\sin(1) \cdot L \cdot \cos(i)} + 1.$$

To find the worst-case (i.e., maximal) upper bound for D, we consider the scenario where $\cos(i)$ attains its minimum value in absolute magnitude. For a given confidence level p, we assume:

$$|\cos(i)| \geq c_p,$$

for some constant c_p, leading to the refined upper bound:

$$D \leq \frac{1}{\sin(1) \cdot L \cdot c_p} + 1.$$

Assuming that the cosine bound c_p is derived from quantiles of the standard normal distribution such that:

$$c_p = \cos\left(\frac{\pi}{2}p\right),$$

we arrive at:

$$D \leq \frac{1}{\sin(1) \cdot L \cdot \cos\left(\frac{\pi}{2}p\right)} + 1.$$

In the case where the confidence level $p = 0.95$, we substitute $\sin(1) \approx 0.841$, $\cos\left(\frac{\pi}{2} \cdot 0.95\right) \approx 0.0785$, yielding:

$$D \leq \frac{1}{0.841 \cdot 0.0785 \cdot L} + 1 \approx \frac{1}{15.14 \cdot L} + 1. \tag{15}$$

FINAL UPPER BOUND OF Z

Recall the expression of the error term Z, which involves the estimated error variance σ_e^2, the noise variance σ^2, and a weighted cross-correlation component:

$$Z = \sigma_e^2 + 2\sigma^2 [\underbrace{\frac{\sum_{k=1}^{L-i} w_{i+k} w_k}{w_i}}_{D} - 1].$$

We substitute the upper bounds of σ_e^2 and D derived previously. If the upper bound of σ_e^2 is given by:

$$\sigma_e^2 \leq \sigma^2 \left(\frac{4}{L} + 1\right),$$

and from Eq. equation 15, the upper bound of $D-1$ is:

$$D - 1 \leq \frac{1}{15.14 \cdot L},$$

then the upper bound of Z becomes:

$$Z \leq \sigma^2 \left(\frac{4}{L} + 1\right) + 2\sigma^2 \cdot \left(\frac{1}{15.14 \cdot L}\right).$$

Combining the terms yields:

$$Z \leq \sigma^2 \left(\frac{4 + \frac{2}{15.14}}{L} + 1\right) \approx \sigma^2 \left(\frac{4.132}{L} + 1\right).$$

D.7 CONCLUSION

At 95% confidence level, the dynamic replacement strategy will effectively reduce prediction error and improve detection performance when the second moment of anomaly deviation satisfies:

$$\mathbb{E}[\Delta^2] = \sigma_\Delta^2 + \mu_\Delta^2 > \left(\frac{4.312}{L} + 1\right) \sigma^2. \tag{16}$$

This establishes a quantitative threshold for anomaly detection effectiveness based on window length L and noise variance σ^2.

D.8 Special Case: No Gaussian Noise in the Test Set

Control Group (No Replacement of Anomalous Value) Assume the input window contains an anomaly $x_{t-i} = \sin(t-i) + \Delta$, then the predicted value is:

$$\hat{x}_t = \sum_{j=1}^{L} w_j \sin(t-j) + b = \sin(t) + w_i \Delta$$

where $\sum_{j=1}^{L} w_j \sin(t-j) + b = \sin(t)$ is the normal prediction term and $w_i\Delta$ is the contribution of the anomaly.

The prediction error is:

$$e_t = \hat{x}_t - \sin(t) = w_i \Delta$$

The mean squared error (MSE) is:

$$\text{MSE}_{\text{Control}} = (w_i \Delta)^2$$

Experimental Group (Dynamic Replacement of Anomalous Value) Replace the anomalous input $x_{t-i} = \sin(t-i) + \Delta$ with the predicted value $\hat{x}_{t-i} = \sin(t-i)$, so that the input window is free of anomalies. Then the predicted value becomes:

$$\hat{x}'_t = \sum_{j=1}^{L} w_j \sin(t-j) + b = \sin(t)$$

The prediction error is:

$$e'_t = \hat{x}'_t - \sin(t) = 0$$

The MSE is:

$$\text{MSE}_{\text{Experimental}} = 0$$

Since the test set contains no noise, the experimental group's MSE is strictly zero, while the control group's MSE is $(w_i\Delta)^2$. Therefore:

$$\text{MSE}_{\text{Experimental}} = 0 < \text{MSE}_{\text{Control}} = (w_i \Delta)^2$$

This inequality strictly holds, indicating that the dynamic replacement strategy is effective in this special case.

D.9 Dynamic Prediction Replacement Experiments

This experiment aims to evaluate the effectiveness of the dynamic prediction replacement (DPR) strategy in handling time series anomalies.

Data Generation

Two types of synthetic time series with anomalies are generated:

- **Sequential Anomalies Dataset:** Based on a sine wave with added random noise. Several contiguous anomaly segments are inserted at random locations, each consisting of 6 to 16 consecutive points. Anomalies are generated by injecting large random perturbations (standard deviation = 0.8).
- **Point Anomalies Dataset:** Also based on a sine wave. Anomalous points are scattered randomly, making up 5% of the total data. Anomalies are generated by adding large noise perturbations (standard deviation = 0.9).

Both datasets contain 1200 time steps, with a sliding window size of 40.

Model Architecture

A simple single-layer fully connected network is used for prediction:

$$f(X) = W \cdot X + b$$

where X is the input window of length 40, and W, b denote the weight matrix and bias term. The model is trained using Mean Squared Error (MSE) loss and the Adam optimizer for 50 epochs. Training is conducted on noise-added but anomaly-free data to simulate realistic deployment scenarios.

Dynamic Prediction Replacement Algorithm

The DPR algorithm operates as follows:

1. For each time step t, predict the value at t using observations from window $[t - w, t - 1]$.
2. Compute the squared error between the predicted and observed value.
3. If the error exceeds a predefined threshold (set as the 95th percentile of the baseline error distribution), flag it as an anomaly.
4. Replace the detected anomalous value with the prediction for use in subsequent forecasts.

Experimental Results

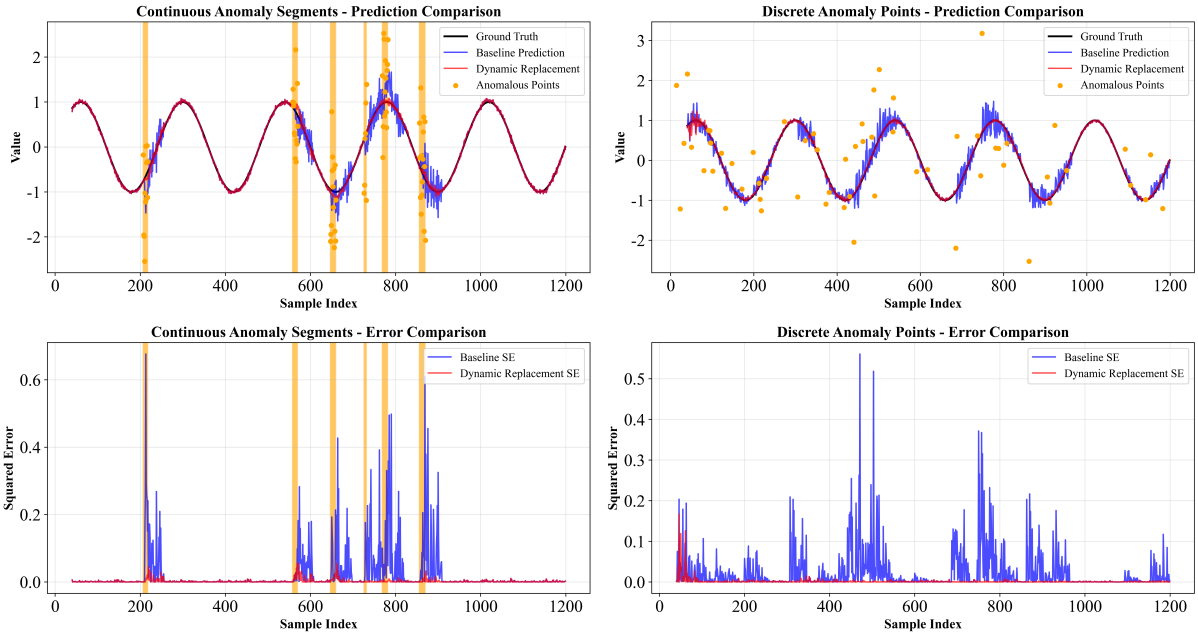

Figure 6: Comparison of prediction and squared error between the baseline and Dynamic Prediction Replacement methods under two scenarios: sequential anomaly segments (left) and scattered anomaly points (right). DPR consistently reduces the influence of anomalies on both prediction and error.

We compare two groups:

- **Baseline Group:** Forecasting directly on data with anomalies, without replacement.
- **DPR Group:** Forecasting after applying the dynamic prediction replacement strategy.

Figure 6 provides a visual comparison of predictions and errors for both scenarios. It highlights how the DPR method effectively reduces anomaly-induced distortion in both the prediction curves and the squared error.

In both sequential and point anomaly scenarios, the DPR method consistently demonstrated substantial error reduction, highlighting its robustness across different anomaly types. As shown in Table 7, DPR reduced forecasting errors by 88.98% in the presence of sequential anomalies, and by an even higher 92.58% when facing scattered point anomalies. The greater improvement in the latter case may stem from the relative ease of identifying and isolating point anomalies, compared to sustained anomalous segments.

Table 7: Forecasting Error (MSE) and Error Reduction of DPR under Different Anomaly Types

Anomaly Type	Method	MSE	Error Reduction
Sequential	Baseline	0.0203	–
	DPR	0.0022	88.98%
Point	Baseline	0.0215	–
	DPR	0.0016	92.58%

The experimental results demonstrate the strong capability of the DPR method in reducing the adverse effects of anomalies—both sequential and scattered—in time series data, with approximately 90% error reduction in both scenarios.

D.10 Proof of the Universality of the Dynamic Replacement Strategy in Nonlinear Models

Although the above theoretical analysis is based on linear models, this section validates the effectiveness of the dynamic replacement strategy in nonlinear models, specifically a fully connected neural network with ReLU activation. This further demonstrates the universality of the proposed method.

After introducing nonlinear activation functions, the propagation of prediction errors is affected by the nonlinearity. Therefore, the derivation must additionally account for the nonlinear transformation's influence on the prediction error. To this end, we consider the ReLU activation function:

$$\phi(z) = \max(z, 0),$$

and analyze the error propagation mechanism in nonlinear models, comparing it with the linear case to explore the applicability of the dynamic replacement strategy under more complex model structures.

1. Model Definition

Consider a single-layer fully connected network with ReLU activation:

$$\hat{x}_t = \phi\left(\sum_{j=1}^{L} w_j x_{t-j} + b\right), \quad \phi(z) = \max(z, 0).$$

Assume the model has been trained on clean (normal) data, and the weights $\{w_j\}$ and bias b have converged to optimal values such that for normal data:

$$\phi\left(\sum_{j=1}^{L} w_j \sin(t-j) + b\right) \approx \sin(t).$$

2. Effect of Anomalies on Prediction

Control Group (Without Replacement). Assume the input window contains an anomalous value at position $t - i$, such that:

$$x_{t-i} = \sin(t-i) + \varepsilon_{t-i} + \Delta,$$

where Δ represents the anomaly. Then the predicted output becomes:

$$\hat{x}_t = \phi\left(\sum_{j=1}^{L} w_j \sin(t-j) + w_i\Delta + \sum_{j=1}^{L} w_j\varepsilon_{t-j} + b\right).$$

Due to ReLU's nonlinearity, two cases arise:

- **Linear region:** If the expression inside $\phi(\cdot)$ is positive, i.e., normal linear term $+ w_k\Delta > 0$, then the output is a linear combination, and the anomaly directly affects the output.
- **Truncation region:** If the expression is non-positive, i.e., normal linear term $+ w_k\Delta \leq 0$, then $\hat{x}_t = 0$, and the anomaly is completely suppressed.

Experimental Group (With Dynamic Replacement). Replace the anomalous input x_{t-i} with a historical prediction $\hat{x}_{t-i}$:

$$x'_{t-i} = \hat{x}_{t-i} = \phi\left(\sum_{j=1}^{L} w_j x_{t-i-j} + b\right).$$

Since $\hat{x}_{t-i}$ has already been filtered through ReLU, the influence of the anomaly is suppressed in the input window.

3. Error Analysis

Control Group MSE.

- **Case 1 (Linear region):** The prediction error is:

$$e_t = \left(\sum_{j=1}^{L} w_j \varepsilon_{t-j} + w_i \Delta \right) - \varepsilon_t.$$

  The MSE contains a Δ^2 term, similar to the linear model.
- **Case 2 (Truncation region):** The prediction is zero, so the error becomes:

$$e_t = 0 - (\sin(t) + \varepsilon_t),$$

  and

$$\text{MSE} = \sin^2(t) + \sigma^2,$$

  which is significantly higher than in the normal case.

The overall MSE of the control group is a weighted average of the two cases. However, since large Δ values often push the model into the linear region, the MSE remains close to the linear case. If the model enters the truncation region, the MSE increases significantly beyond the linear model's prediction.

Experimental Group MSE. Since the replaced value x'_{t-i} has already been filtered by ReLU, and assuming historical prediction error is small ($\sigma_e^2 \ll \Delta^2$), we have:

$$x'_{t-i} \approx \sin(t-i) + \varepsilon_{t-i},$$

leading to:

$$\hat{x}_t \approx \phi \left(\sum_{j=1}^{L} w_j \sin(t-j) + \sum_{j=1}^{L} w_j \varepsilon_{t-j} + b \right),$$

and thus the prediction error is close to that of the experimental group in the linear model:

$$\text{MSE}_{\text{exp}} = \sigma^2 \left(1 + \sum_{j=1}^{L} w_j^2 \right) + w_i^2 \sigma_e^2 + 2 w_i \sigma^2 \left(\sum_{k=1}^{L-i} w_{i+k} w_k - w_i \right) \tag{17}$$

KEY CONCLUSIONS

1. **ReLU's Suppression Effect.** In nonlinear models with ReLU activation, anomalies may cause the model to switch between activation regions, altering the MSE formulation.
   - *Linear region:* When the anomaly drives the model into ReLU's linear regime, the MSE reduction of the experimental group over the control group is consistent with the linear model ($\Delta^2 \gg \sigma_e^2$).
   - *Truncation region:* When the anomaly pushes the model into the zero-output region of ReLU, the control group's prediction collapses to zero, significantly increasing the MSE. In contrast, the dynamic replacement strategy in the experimental group avoids this truncation, substantially lowering MSE.
2. **Comparison Between Nonlinear and Linear Models.**
   - *When the linear region dominates:* If the model mostly operates in the linear region (e.g., due to reasonable weight design), the experimental group still outperforms the control group in MSE, consistent with linear models.
   - *Amplification under extreme anomalies:* Due to ReLU's truncation effect, the control group's MSE in nonlinear models increases even more under large anomalies. Meanwhile, the dynamic replacement strategy amplifies its advantage, showing even greater MSE reduction than in linear cases.

Conclusion: The nonlinear nature of ReLU does not diminish the effectiveness of the dynamic replacement strategy. On the contrary, in specific anomaly patterns, it enhances the advantage of the experimental group. Therefore, the strategy is applicable to a broader range of nonlinear model scenarios.

D.11 PROOF THAT THE PREDICTION ERROR SATISFIES $E[e_t] = 0$ FOR ALL t

PROBLEM RESTATEMENT AND NOTATION

Objective: Prove that after dynamically replacing detected anomalies, the prediction error at each time point

$$e_t = \hat{x}_t - (\sin(t) + \varepsilon_t)$$

satisfies

$$\mathbb{E}[e_t] = 0 \quad \forall t.$$

Base Model and Unbiasedness

Assume the model is trained to convergence on clean (anomaly-free) data. When the input window contains no anomalies, the model prediction satisfies:

$$\hat{x}_t = \sum_{j=1}^{L} w_j x_{t-j} + b,$$

where all weights w_j and bias b are optimized to be unbiased, such that

$$\mathbb{E}[\hat{x}_t] = \sin(t).$$

In the absence of anomalies, the true target value is:

$$x_t = \sin(t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2). \tag{18}$$

Thus, the prediction error is:

$$e_t = \hat{x}_t - (\sin(t) + \varepsilon_t),$$

and the expectation is:

$$\mathbb{E}[e_t] = \mathbb{E}[\hat{x}_t] - \sin(t) - \mathbb{E}[\varepsilon_t] = 0.$$

Mathematical Description of Dynamic Replacement

Suppose there are m anomalies in the input window, located at positions $i_1, i_2, \ldots, i_m$. When an anomaly is detected at $x_{t-i^*} = \sin(t - i^*) + \varepsilon_{t-i^*} + \Delta_{i^*}$, it is replaced by:

$$x'_{t-i^*} = \hat{x}_{t-i^*} = \sin(t - i^*) + \varepsilon_{t-i^*} + e_{t-i^*}.$$

Here, $i^* \in \{i_1, i_2, \ldots, i_m\}$ indicates the position of an anomaly within the input window. The updated input sequence becomes:

$$x'_s = \begin{cases} x_s, & s \neq t - i^* \\ \hat{x}_{t-i^*}, & s = t - i^* \end{cases}.$$

Mathematical Induction Proof of Recursive Unbiasedness

Step 1: Base Case (No Replacement in Window) When the input window contains no anomalies, we have:

$$\mathbb{E}[e_u] = 0 \quad \forall u.$$

Step 2: Inductive Hypothesis Assume that for all times $s \leq k$, the prediction errors satisfy:

$$\mathbb{E}[e_s] = 0 \quad \forall s \leq k.$$

Step 3: Inductive Step ($t = k + 1$) At time $t = k + 1$, the model makes a prediction based on the window $\{x'_{k+1-j}\}_{j=1}^{L}$:

$$\hat{x}_{k+1} = \sum_{j=1}^{L} w_j x'_{k+1-j} + b.$$

Each x'_{k+1-j} in the input window may be:

1. A normal (unreplaced) value:
$$x'_{k+1-j} = \sin(k + 1 - j) + \varepsilon_{k+1-j},$$
2. A replaced value:
$$x'_{k+1-j} = \sin(k + 1 - j) + \varepsilon_{k+1-j} + e_{k+1-j}.$$

By the inductive hypothesis:

$$\mathbb{E}[e_{k+1-j}] = 0 \quad \forall j \geq 1.$$

Therefore, for any x'_{k+1-j}, its expectation is:

$$\mathbb{E}[x'_{k+1-j}] = \sin(k + 1 - j),$$

since $\mathbb{E}[\varepsilon_{k+1-j}] = 0$ and $\mathbb{E}[e_{k+1-j}] = 0$.

Thus, the expected prediction is:

$$\mathbb{E}[\hat{x}_{k+1}] = \sum_{j=1}^{L} w_j \mathbb{E}[x'_{k+1-j}] + b = \sum_{j=1}^{L} w_j \sin(k + 1 - j) + b = \sin(k + 1).$$

The prediction error is:

$$e_{k+1} = \hat{x}_{k+1} - (\sin(k + 1) + \varepsilon_{k+1}),$$

and its expectation is:

$$\mathbb{E}[e_{k+1}] = \mathbb{E}[\hat{x}_{k+1}] - \sin(k + 1) - \mathbb{E}[\varepsilon_{k+1}] = 0.$$

Step 4: Inductive Conclusion By mathematical induction, combining the base case

$$\mathbb{E}[e_u] = 0 \quad \forall u,$$

with the inductive hypothesis and inductive step, we conclude:

$$\mathbb{E}[e_t] = 0 \quad \forall t.$$

D.12 Analytical Derivation of the Optimal Weights $\boldsymbol{w}$

D.12.1 Problem Setup and Notation

We consider a simple single-layer fully connected neural network for predicting a sine function based on past inputs. The setup is as follows:

- Input window:
$$\boldsymbol{x}_t = [\sin(t-1), \sin(t-2), \ldots, \sin(t-L)]\,.$$
- Output target:
$$x_t = \sin(t).$$
- Model (no bias term, since $\mathbb{E}[x_t] = 0$):
$$x_{\text{pred}} = \sum_{i=1}^{L} w_i \sin(t-i).$$
- Objective: minimize the expected mean squared error:
$$\mathcal{L}(\boldsymbol{w}) = \mathbb{E}\left[\left(\sum_{i=1}^{L} w_i \sin(t-i) - \sin(t)\right)^2\right].$$

Assume that $t \sim \mathcal{U}[0, 2\pi)$, i.e., t is uniformly distributed over one period.

D.12.2 Orthogonality Conditions

Since the model is trained using the gradient descent strategy, the partial derivative of the loss function with respect to each weight w_j can be considered zero when the weights reach a local optimum.

$$\frac{\partial \mathcal{L}}{\partial w_j} = 0, \quad \forall j = 1, 2, \ldots, L.$$

We expand the loss:

$$\mathcal{L}(\boldsymbol{w}) = \mathbb{E}\left[\left(\sum_{i=1}^{L} w_i \sin(t-i)\right)^2\right] - 2\mathbb{E}\left[\sin(t)\sum_{i=1}^{L} w_i \sin(t-i)\right] + \mathbb{E}[\sin^2(t)].$$

Taking the derivative w.r.t. w_j:

$$\frac{\partial \mathcal{L}}{\partial w_j} = 2\mathbb{E}\left[\left(\sum_{i=1}^{L} w_i \sin(t-i)\right)\sin(t-j)\right] - 2\mathbb{E}[\sin(t)\sin(t-j)] = 0.$$

Rewriting:

$$\mathbb{E}\left[\left(\sum_{i=1}^{L} w_i \sin(t-i) - \sin(t)\right)\sin(t-j)\right] = 0, \quad \forall j.$$

This yields a system of linear equations:

$$\sum_{i=1}^{L} w_i \mathbb{E}[\sin(t-i)\sin(t-j)] = \mathbb{E}[\sin(t)\sin(t-j)], \quad \forall j. \tag{19}$$

D.12.3 SIMPLIFYING THE EXPECTATIONS

We now compute the expectations in Equations equation 19. Since $t \sim \mathcal{U}[0, 2\pi)$, we have:

$$\mathbb{E}[\sin(t-i)\sin(t-j)] = \frac{1}{2\pi}\int_0^{2\pi} \sin(t-i)\sin(t-j)\,dt.$$

Using the trigonometric identity:

$$\sin A \sin B = \frac{1}{2}\left[\cos(A-B) - \cos(A+B)\right].$$

Apply it to $\sin(t-i)\sin(t-j)$:

$$\sin(t-i)\sin(t-j) = \frac{1}{2}\left[\cos(j-i) - \cos(2t-(i+j))\right].$$

Then:

$$\begin{aligned}\mathbb{E}[\sin(t-i)\sin(t-j)] &= \frac{1}{4\pi}\left[2\pi\cos(j-i) + \underbrace{\int_0^{2\pi} -\cos(2t-(i+j))\,dt}_{=0}\right] \\ &= \frac{1}{2}\cos(j-i).\end{aligned}$$

Similarly:

$$\mathbb{E}[\sin(t)\sin(t-j)] = \frac{1}{2}\cos(j). \tag{20}$$

Substituting Equations equation 19 and equation 20, we obtain:

$$\sum_{i=1}^{L} w_i \cos(i-j) = \cos(j), \quad \forall j = 1, 2, \ldots, L. \tag{21}$$

D.12.4 SOLVING BY HYPOTHESIS

We hypothesize a solution of the form:

$$w_i = k\cos(i).$$

Substitute into Equation equation 21:

$$\sum_{i=1}^{L} k\cos(i)\cos(j-i) = k\sum_{i=1}^{L}\cos(i)\cos(j-i).$$

Using identity:

$$\cos(a-b)\cos(b) = \frac{1}{2}[\cos(a) + \cos(a-2b)].$$

We obtain:

$$\begin{aligned}k\sum_{i=1}^{L}\cos(i)\cos(j-i) &= \frac{k}{2}\sum_{i=1}^{L}[\cos(j) + \cos(j-2i)] \\ &= \frac{kL}{2}\cos(j) + \frac{k}{2}\sum_{i=1}^{L}\cos(j-2i).\end{aligned}$$

If L is large and $\cos(j-2i)$ is approximately uniformly distributed, the second term averages to 0:

$$\Rightarrow \frac{kL}{2}\cos(j) \approx \cos(j) \quad \Rightarrow \quad k = \frac{2}{L}.$$

Hence, the optimal weights are:

$$w_i = \frac{2}{L}\cos(i).$$

Note: If the input vector is reindexed as $[\sin(t-L), \ldots, \sin(t-1)]$ corresponding to weights $[w_0, w_1, \ldots, w_{L-1}]$, then:

$$w_x = \frac{2}{L}\cos(x-L).$$

D.12.5 Empirical Validation

Experimental Setup

This experiment investigates the empirical weight patterns learned by a neural network trained on pure sine signals. Specifically:

- Generate sine function data as time series.
- Use a single fully connected layer (linear layer) neural network.
- Use sliding window input with window sizes $L \in \{40, 45, 50, 55, 60, 65\}$.
- Run 500 independent training trials with random initializations.
- Analyze the mean and distribution of learned weights.

Research Objectives

- Examine whether the network consistently learns a similar weight pattern.
- Compare the learned weights with the theoretically optimal solution $w_x = \frac{2}{L}\cos(x - L)$.

Experimental Conclusion

Across multiple training runs, the learned weights converge to a highly consistent pattern. The mean curve of the weights aligns closely with the theoretically optimal cosine function $w_x = \frac{2}{L}\cos(x - L)$, confirming the analytical derivation. The following plot 7 illustrates the results: the blue line is the empirical mean, while the red dashed line is the theoretical cosine shape.

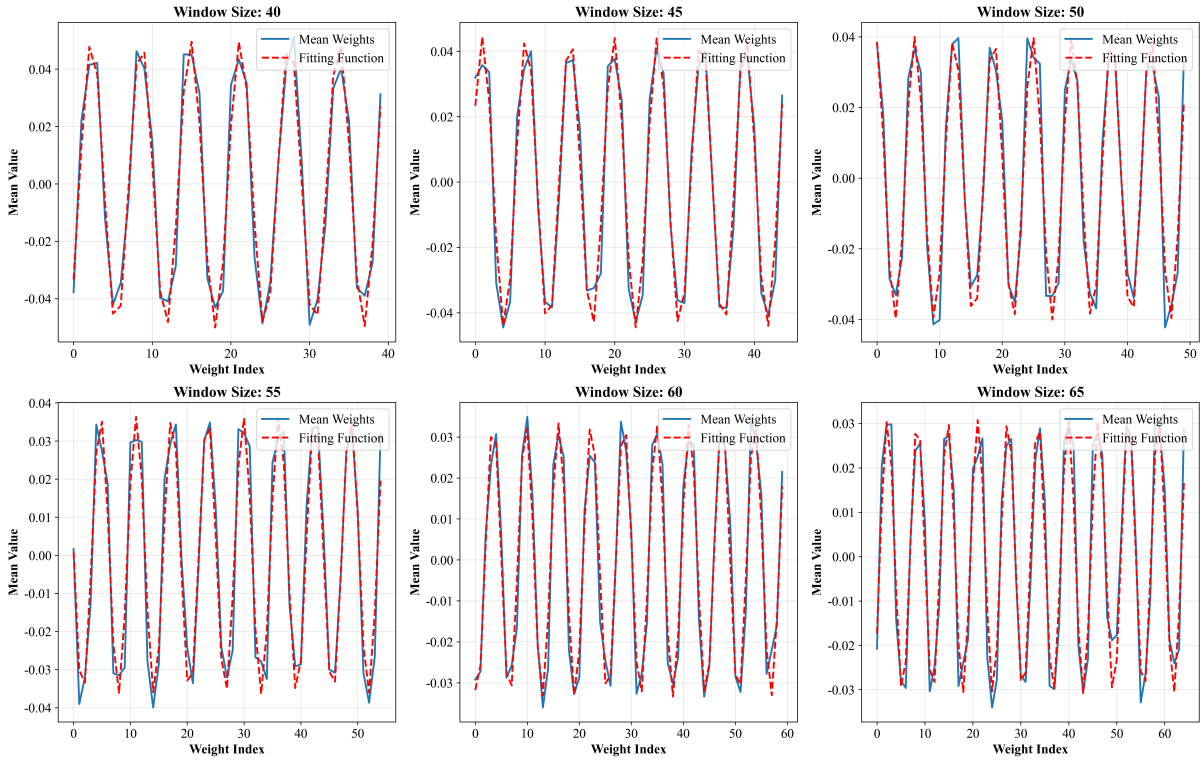

Figure 7: Weight distribution patterns across different window sizes (L = 40, 45, 50, 55, 60, 65) after training a linear model on sinusoidal data. Each subplot displays the mean weights (blue solid line) from 500 independent training runs and a theoretical fitting curve 2/L·cos(x-L) (red dashed line). The remarkable alignment between empirical weight distributions and theoretical predictions demonstrates that the learned representations consistently converge to optimal sinusoidal predictors regardless of the window size. This supports our hypothesis that linear predictors implicitly encode trigonometric representations when trained on time-series with cyclic patterns.

D.13 Theoretical Limitations and Future Directions

Theoretical Limitations: Although this paper provides a comprehensive empirical evaluation of the dynamic replacement strategy, its theoretical analysis relies on several simplifying assumptions. In particular, the current

proof assumes that the observation noise is independently and identically distributed (i.i.d.) and that each input window contains at most a single outlier. However, in practical scenarios, noise may exhibit autocorrelation or heteroscedasticity, and outliers may appear in clusters. Under such non-ideal conditions, the present error analysis may become biased, thereby affecting the accuracy of performance assessment for the proposed strategy. Consequently, the current theoretical results may not generalize well to situations involving clustered anomalies or non-i.i.d. noise, which limits our understanding of the method's behavior in more diverse settings.

Future work: We plan to extend our analysis to scenarios where multiple outliers or clustered anomalies appear within the input window. This direction is expected to provide a more comprehensive theoretical foundation for evaluating the robustness and applicability of the strategy in real-world environments. Due to the complexity of modeling non-i.i.d. noise, we leave its exploration to future work depending on application-specific demands.

E Proof of Complexity Reduction in the SN Module

E.1 Empirical Validation of PELT Complexity

To empirically validate the theoretical time complexity of the PELT algorithm, we conducted experiments by varying the input sequence length n and recording the elapsed runtime. The measured data were then fitted against several candidate complexity models, namely $O(n)$, $O(n \log n)$, $O(n^2)$, and $O(\log n)$. The fitting quality was evaluated using the coefficient of determination (R^2).

For each input size n, we executed the standard PELT algorithm and measured the elapsed time in seconds. The sequence length was varied from 100 to 40,000. The observed runtimes are reported in Table 8.

Table 8: Runtime of PELT under different input sizes.

n	Elapsed Time (s)
100	0.0104
500	0.1938
1000	0.6426
2000	1.9772
4000	4.7989
8000	12.775
12000	23.5916
16000	42.1717
20000	63.5235
24000	81.3433
40000	196.9832

Model Fitting. The recorded data were fitted to multiple complexity models. Table 9 reports the R^2 scores for each candidate model. The quadratic model $O(n^2)$ achieves the best fit with $R^2 = 0.9992$, significantly outperforming the alternatives.

Table 9: Goodness-of-fit of different complexity models.

Complexity Model	R^2 Score
$O(n^2)$	0.999206
$O(n \log n)$	0.956115
$O(n)$	0.936277
$O(\log n)$	0.425482

The empirical results strongly corroborate the theoretical analysis: the runtime of the PELT algorithm scales quadratically with input size n. The $O(n^2)$ model yields an almost perfect fit ($R^2 = 0.9992$), confirming that PELT exhibits quadratic time complexity in practice.

E.2 Problem Definition

Given a time series of length n, the task is to detect changepoints within the sequence. We compare the computational complexity of two approaches:

1. **Direct PELT Method:** The standard PELT algorithm with time complexity $O\big(n^2\big)$.

2. **Two-Stage Method:**
   - *Coarse Detection Stage:* Apply PELT with a jump parameter jump $= \lfloor 0.001n \rfloor$ to reduce computational cost.
   - *Refined Detection Stage:* Around each coarse changepoint, perform local detection on a subsequence of size n_{local} using a window-based segmentation method. The time complexity is $O(w \cdot n_{\text{local}})$, where w denotes the window size parameter.

The goal is to derive and compare the time complexities of these two approaches.

E.3 Direct PELT Method

The PELT algorithm detects changepoints via dynamic programming. Its standard time complexity is

$$T_{\text{direct}} = O\big(n^2\big)\,,$$

which grows quadratically with the sequence length n.

E.4 Two-Stage Method

Coarse Detection Stage

Using PELT with a jump parameter $\text{jump} = \epsilon n$, where $\epsilon = 0.001$, reduces computational cost. Specifically, the complexity decreases from $O\big(n^2\big)$ to $O\Big(\frac{n^2}{\text{jump}}\Big)$.

Substituting $\text{jump} = 0.001n$ yields:

$$T_{\text{coarse}} = O\Big(\tfrac{n^2}{0.001n}\Big) = O(1000n) = O(n)\,.$$

Thus, the coarse detection stage achieves linear time complexity.

Refined Detection Stage

Suppose the coarse stage identifies K changepoints $\tau_1, \tau_2, \ldots, \tau_K$. For each τ_i, a local refinement is performed in the neighborhood $[\tau_i - S, \tau_i + S]$, where S is the local window radius (chosen as a fixed value much smaller than n).

The number of points in each neighborhood is $n_{\text{local}} = 2S$, leading to a total refined sample size:

$$W = \sum_{i=1}^{K} n_{\text{local}} = 2KS.$$

(assuming non-overlapping neighborhoods, or equivalently $K \ll n$ so that $W \approx 2KS$).

The window-based segmentation method has complexity $O(w \cdot n_{\text{local}})$ per neighborhood. Hence, the total refined detection cost is

$$T_{\text{fine}} = O\left(\sum_{i=1}^{K} w \cdot n_{\text{local}}\right) = O(w \cdot W)\,.$$

Since $W = 2KS$, and typically $K \ll n$, $S \ll n$, and $w \ll n$, we have

$$T_{\text{fine}} = O(wKS)\,.$$

If K, S, and w are constants or grow much slower than n, then $T_{\text{fine}} = O(1)$.

E.4.1 Total Complexity of the Two-Stage Method

The overall cost is

$$T_{\text{proposed}} = T_{\text{coarse}} + T_{\text{fine}} = O(n) + O(wW)\,.$$

Since $W \ll n$ and $w \ll n$, the second term is dominated by the linear term, leading to

$$T_{\text{proposed}} = O(n)\,.$$

E.5 Complexity Comparison

In summary, the direct PELT method scales quadratically as $O\big(n^2\big)$, whereas the proposed two-stage method achieves linear complexity $O(n)$. This represents an improvement by a factor of $O(n)$.

F Related works

F.1 Time Series Anomaly Detection Methods

To tackle the problem of *unsupervised time series anomaly detection*, a variety of techniques have been proposed, including *forecasting-based methods* Chen et al. (2021); Zhao et al. (2020); Zhang et al. (2022); Deng & Hooi (2021b), *reconstruction-based methods* Tuli et al. (2022); Zhang et al. (2021b); Xu et al. (2021); Audibert et al. (2020), *density estimation* approaches Zong et al. (2018); Dai & Chen (2022), and *clustering-based methods* Xu et al. (2024); Ruff et al. (2018a).

Forecasting-based Methods. Forecasting-based anomaly detection is one of the most extensively studied directions, where the core challenge lies in extracting informative features from input sequences. To enhance the modeling capacity, prior works have incorporated mechanisms such as *contrastive learning* Yue et al. (2022), *2D convolutions* Wu et al. (2022) to improve the representation of multivariate sequences. In addition to feature extraction, some approaches introduce auxiliary mechanisms to improve detection performance. For instance, CAT Zhang et al. (2022) integrates *one-class classification loss* Guo et al. (2021); Wang et al. (2021) into the forecasting objective; MTAD-GAT Zhao et al. (2020) trains two networks jointly for forecasting and reconstruction; GDN Deng & Hooi (2021a) transforms prediction errors into normalized graph-structured deviation scores; and LSTM-NDT Hundman et al. (2018b) proposes a dynamic thresholding method based on exponential smoothing. While these methods incorporate various enhancements beyond forecasting, their primary focus remains on the design of forecasting models, with other components playing a supportive role. In contrast, this work aims to propose a *more general forecasting framework*, rather than improving a specific model.

Density Estimation Methods. These methods assume that anomalies lie in low-probability regions and thus exhibit low data density. Early methods such as LOF Breunig et al. (2000) and COF Tang et al. (2002) estimate sample density based on the k-nearest neighbors. DAGMM Zong et al. (2018) combines reconstruction errors from autoencoders with Gaussian Mixture Models (GMMs) to jointly model low-dimensional embeddings and reconstruction loss. More recently, GANF Dai & Chen (2022) utilizes Bayesian networks with *normalizing flows* for density estimation, learning flow parameters to improve estimation accuracy.

Clustering-based Methods. These methods assume that normal data points cluster densely, while those far from the center are likely anomalies. Typical approaches include SVDD Ruff et al. (2018b) and its deep variant Deep SVDD Ruff et al. (2018a). THOC Shen et al. (2020) extends this idea by introducing multiple latent spaces and computing weighted distances to all centers as anomaly scores. CPOD Tran et al. (2020) propose enhancements from the perspectives of efficiency and streaming data processing, respectively. COUTA Xu et al. (2024) generates pseudo-anomalies via data augmentation to guide the model in learning decision boundaries for anomalies.

Reconstruction-based Methods. These approaches train models to reconstruct the original time series, under the assumption that anomalies are harder to reconstruct and thus can be identified. To prevent models from simply learning identity mappings, various techniques have been introduced to enhance anomaly discriminability. Most existing methods are based on generative models such as *Variational Autoencoders (VAEs)* Kingma et al. (2013) and *Generative Adversarial Networks (GANs)* Goodfellow et al. (2014). LSTM-VAE Park et al. (2018) is a representative method that combines sequential modeling with the VAE framework. Omni-Anomaly Shi et al. (2023) and InterFusion Li et al. (2021) further integrate techniques such as normalizing flows, hierarchical structures, and bidirectional temporal modeling to improve detection performance. GAN-based methods often adopt adversarial training strategies, with implementations ranging from multi-objective min-max optimization to more complex variants Tuli et al. (2022); Xu et al. (2021); Audibert et al. (2020); Li et al. (2019); Geiger et al. (2020); Bashar & Nayak (2020).

F.2 Time Series Forecasting Methods

Time series forecasting models can be broadly categorized based on the neural network architecture they employ, including: (1) *Transformer-based models* Wu et al. (2022); Wang et al. (2024); Huang & Liu (2024), (2) *Multi-Layer Perceptrons (MLPs)* Zeng et al. (2022); Challu et al. (2023); Zhou et al. (2023c) , (3) *Recurrent Neural Networks (RNNs)* Salinas et al. (2020); Lai et al. (2018), (4) *Convolutional Neural Networks (CNNs)* Luo & Wang (2024); Liu et al. (2022a), and (5) *Graph Neural Networks (GNNs)* Zhou et al. (2023a); Liu et al. (2022b).It is important to note that this categorization is not exhaustive. As these directions are beyond the scope of this work, we do not elaborate on them here.

In our experiments, we further demonstrate that DRPAD can be seamlessly integrated into all of the above forecasting models, effectively transforming them into anomaly detection methods.

F.3 Change Point Detection Methods

Change Point Detection (CPD) aims to identify positions in a time series where statistical properties—such as mean, variance, or distribution—undergo significant changes. CPD has found wide applications in fields such as finance, industrial monitoring, and anomaly detection. Existing approaches can be broadly categorized into *supervised* and *unsupervised* methods.

Supervised methods typically formulate CPD as a classification task, training classifiers based on labeled data. Depending on the problem formulation, these methods can be further divided into multi-class classifiers (e.g., decision trees Reddy et al. (2010), k-nearest neighbors Wei & Keogh (2006), Hidden Markov Models (HMM) Cleland et al. (2014)) and binary classifiers (e.g., SVM Desobry et al. (2005); Feuz et al. (2014), Naïve Bayes Feuz et al. (2014), logistic regression Feuz et al. (2014)). Although supervised methods generally perform well when high-quality labeled data are available, their applicability is limited due to the scarcity of such data in real-world scenarios.

In contrast, **unsupervised methods** do not rely on labeled data, making them more generalizable in practice. Based on different modeling strategies, mainstream unsupervised CPD approaches can be grouped into the following categories: (1) *Likelihood-ratio-based methods*, which detect change points by computing the difference or ratio of probability densities before and after a segment (e.g., KLIEP Liu et al. (2013), uLSIF Liu et al. (2013)); (2) *Subspace modeling methods* (e.g., SI Liu et al. (2013), PELT Killick et al. (2012)), which analyze structural variations in the embedded space of the time series; (3) *Probabilistic modeling methods* (e.g., Gaussian Processes Saatçi et al. (2010)), which estimate changes from a generative modeling perspective; (4) Other methods based on kernel techniques, graph-based structures, or clustering under sliding windows.

These methods exhibit different strengths and are suited for varying data characteristics and application scenarios.

In this study, we adopt a strategy that combines both global and local features: We first perform coarse-grained detection using the PELT Killick et al. (2012) (Pruned Exact Linear Time) algorithm. PELT is an unsupervised subspace modeling method that minimizes a weighted cost function, allowing linear-time detection while preserving optimality. This makes it suitable for large-scale time series. To further improve precision, we introduce a local refinement strategy based on a sliding window Truong et al. (2020), which scans the candidate change point regions at a finer granularity. This hybrid mechanism significantly enhances the robustness and accuracy of segmentation, providing high-quality structural support for subsequent *segment-based normalization*.

F.4 Comparison with Related Work

The AFMF framework Shen et al. (2024) introduces a technique called *Local Instance Normalization (LIN)* with a similar goal to our proposed *Segment-wise Normalization (SN)*: both aim to mitigate the effect of varying data scales during anomaly detection. LIN independently normalizes data within each fixed-length input window, reducing the influence of amplitude shifts on detection performance.

When the data distributions across adjacent windows differ significantly—for instance, if the previous window contains large-magnitude values while the next has small-scale fluctuations—LIN effectively balances the scale across windows, thereby improving the overall Mean Squared Error (MSE) performance. This helps prevent small-amplitude anomalies from being undetected due to diminished MSE values in such regions.

However, LIN has limitations in another common scenario. As illustrated in Figure 1, when large-valued points dominate the early portion of the input window, the normalization scale is skewed, causing subsequent small-scale anomalies to be masked, with reduced MSE and thus harder to detect.

To address this issue, our proposed SN method employs *change point detection* to adaptively segment the sequence. Normalization is then performed *within each segment*, preserving local scale variations. This segment-aware normalization effectively alleviates the problem of large values "overshadowing" small anomalies, leading to improved robustness and precision in anomaly detection.

Moreover, the AFMF framework introduces a mechanism called *Progressive Adjacent Masking (PAM)* that works in conjunction with LIN to further enhance anomaly detection performance. The normalization in LIN adjusts the data toward a zero-centered distribution, laying the foundation for PAM's zero-masking operation. The core idea of PAM is to observe how masking affects prediction error, helping distinguish between false positives caused by nearby anomalies and true anomalies.

Specifically, when anomalies are surrounded by adjacent anomalous points, masking these neighboring values reduces prediction errors significantly. Conversely, masking normal data introduces noise and increases the prediction error. PAM leverages this behavior by comparing the prediction errors before and after masking to better separate true anomalies from false positives.

Despite its conceptual validity, PAM's rigid zero-masking strategy risks distorting the input, especially in smoothly varying sequences. Such abrupt changes may disrupt the continuity and introduce unnatural patterns that did not appear during training, making it harder for models to generalize and potentially causing misclassifications.

To resolve this, we propose a novel *Dynamic Prediction Replacement* mechanism: when an anomaly is detected, it is directly replaced by the model's predicted value, which is then used as input for subsequent steps. This smooth substitution suppresses the propagation of anomalous information, maintaining continuity and stability in the input sequence. Particularly in scenarios with consecutive anomalies or frequent distribution shifts, the replacement mechanism allows real-time window updates and enhances the adaptability of the detection process.

G Detailed description and sources of baselines and datasets

The following provides a detailed introduction to the nine real-world time series anomaly detection benchmarks, with numerical details summarized in Table 10. The processing methods for all datasets are consistent with AFMF Shen et al. (2024).

SMD (Server Machine Dataset) Su et al. (2019) is a one-minute-level dataset consisting of 38 dimensions, collected from a large Internet company over a period of five weeks.

PSM (Pooled Server Metrics) Abdulaal et al. (2021) contains 25 dimensions and is collected from internal nodes of multiple application servers at eBay.

MSL (Mars Science Laboratory Rover) and **SMAP** (Soil Moisture Active Passive Satellite) Hundman et al. (2018a) are public datasets originating from Incident Surprise Anomalies (ISA) and contain telemetry anomaly data from spacecraft monitoring systems, with 55 and 25 dimensions, respectively.

SWaT (Secure Water Treatment) Mathur & Tippenhauer (2016) is a dataset collected from a water treatment plant, containing 51 dimensions, including 7 days of normal operation and 4 days of artificially induced attack scenarios.

WADI (Water Distribution) Ahmed et al. (2017) is an extended testbed of SWaT, involving 123 sensors and actuators. The dataset includes 14 days of normal operation and 2 days of attack scenarios.

MBA (MIT-BIH Supraventricular Arrhythmia Database) Moody & Mark (2001) is a popular large-scale dataset comprising electrocardiogram (ECG) recordings from four patients, including two types of arrhythmias (supraventricular premature beats and premature ventricular contractions).

NAB (Numenta Anomaly Benchmark) Ahmad et al. (2017) is a dataset containing multiple univariate sub-datasets, such as ambient temperature and CPU usage.

MSDS (Multi-Source Distributed System) Nedelkoski et al. (2020) records CPU, memory, and load metrics from a distributed IT system consisting of one controller and four computing nodes.

We re-conducted all experiments related to other baselines under their default experimental settings. Their source codes origins are given in Table 11. Some changes are made to DAGMM in the project of TranAD according to another code implementation of DAGMM `https://github.com/danieltan07/dagmm` to avoid 'nan' losses. The only modification was replacing their threshold selection strategies with ours, namely determining anomaly detection thresholds based on a fixed percentile. Additionally, all window size settings were kept consistent with those used in the AFMF framework.

The LF component employed in DRPAD is adapted from the AFMF framework, and we follow its original configuration when applying it. When integrating Transformer-based models with DRPAD, the values of discrete variates at prediction timestamps are not used as decoder inputs. Transformer-based architectures typically require decoder inputs at prediction timestamps to be initialized, often utilizing representations such as trend features derived from encoder inputs. This initialization strategy conflicts with the design of LF, which provides masked continuous variates and full discrete variates to the decoder. Therefore, following its original configuration, we abandon the use of discrete variates' values at prediction timestamps as decoder inputs and continue to use the initialization method of Transformer-based models when combining it with DRPAD.

Table 10: Detailed information of the nine benchmarks.

Benchmark	Application	N (Dimensions)	Window Size	Train	Validation	Test	Anomalies (%)	r (%)	δ
SMD	Server	38	720	566,724	141,681	708,420	4.16	0.5	100
PSM	Server	25	48	105,984	26,497	87,841	27.76	1.5	200
MSL	Space	55	24	46,653	11,664	73,729	10.72	1.5	30
SMAP	Space	25	720	108,146	27,037	427,617	13.13	1.5	20
SWaT	Water	51	720	396,000	99,000	449,919	11.98	0.5	100
WADI	Water	123	100	627,656	156,915	172,801	5.99	0.5	100
MBA	ECG	2	100	6,144	1,536	7,680	5.60	1.5	5
NAB	Various	1	360	2,325	807	4,032	0.60	0.5	50
MSDS	Server	2	720	249,168	62,293	14,457	3.24	2.5	50

Table 11: Baseline and Source Code Origin

Baseline	Source Code Origin
DAGMM	https://github.com/imperial-qore/TranAD
MEMTO	https://github.com/gunny97/MEMTO
CAE-M Zhang et al. (2021a)	https://github.com/imperial-qore/TranAD
GDN	https://github.com/d-ailin/GDN
AFMF	https://github.com/OrigamiSL/AFMF?tab=readme-ov-file
FEDformer	https://github.com/MAZiqing/FEDformer
Autoformer	https://github.com/OrigamiSL/AFMF?tab=readme-ov-file
DLinear	https://github.com/OrigamiSL/AFMF?tab=readme-ov-file
RTNet	https://github.com/OrigamiSL/AFMF?tab=readme-ov-file
DeepAR	https://github.com/OrigamiSL/AFMF?tab=readme-ov-file
GTA	https://github.com/ZEKAICHEN/GTA

Table 12: Details of hyper-parameters and experimental settings

Hyper-parameters/Settings	Values/Mechanisms
Dropout	0.1
Loss function	MSE
Batch size	128
Initial learning rate	1×10^{-4}
Optimizer	AdamW
Weight decay	1×10^{-4}
Gradient clipping	Max norm = 0.5
NaN handling	Reduce LR by half and skip current batch
Learning rate scheduler	OneCycleLR (cosine annealing)
Max LR	2×10^{-4}
Warm-up proportion	30% of total steps
Initial LR	2×10^{-5} (max_lr/10)
Final LR	2×10^{-6} (max_lr/100)
Anneal strategy	Cosine
Epsilon (numerical stability)	1×10^{-8}
AMSGrad	False
Fused implementation	False
Training epochs	As specified by args.train_epochs
Repetition strategy	5 independent runs, results averaged
Platform	Python 3.12.7, PyTorch 2.5.0
Device	4 × NVIDIA GeForce RTX 4090 (24GB)