DEDE: TIME-SERIES DEEP DECOMPOSITION FOR EXPLAINABLE ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Industrial applications of time-series anomaly detection (TSAD) require both high accuracy and strong explainability to support reliable business decision-making. However, few existing TSAD methods satisfy these two requirements simultaneously. To address this challenge, we propose an end-to-end model named Time-Series **De**ep **De**composition (DeDe) for explainable TSAD. DeDe decomposes a time series into trend, seasonal, and residual components while modeling latent normal structure. Each component is encoded with a multi-scale Transformer, and a forecasting objective augmented with contrastive learning encourages a Stable Latent Manifold Structure (SLMS) of normal dynamics. One-step forecasting error and deviations from the component-wise SLMSs are combined into a point-wise anomaly score. Component-level contributions provide interpretable diagnostic attribution. Experiments on 350 real-world evaluation time series from TSB-AD-U show improvements over the evaluated baselines across multiple metrics. DeDe has also been deployed at scale in industrial production, reliably monitoring millions of time series each day while consistently achieving high recall and precision in real-world operations, demonstrating both strong practical effectiveness and operational robustness.

1 INTRODUCTION

Time-series anomaly detection (TSAD) is a fundamental technique for identifying abnormal behavior in temporal data and is widely used in monitoring applications (Yang et al., 2026; Sylligardos et al., 2023; Ghorbani et al., 2024). As modern environments grow in scale and complexity, TSAD must move beyond flagging outliers to explaining abnormal dynamics for timely and actionable responses.

Univariate TSAD remains important when only one reliable signal is available and additional variables are noisy, delayed, or costly to maintain. In this setting, both detection and explanation must be derived from the temporal structure of the target series itself.

A key observation is that a time series combines heterogeneous dynamics: gradual drift changes its trend, periodic distortions affect seasonality, and abrupt spikes introduce short-term deviations. When these dynamics are modeled jointly, benign variation in one component may obscure an abnormal change in another. Separating trend, seasonal, and residual behavior allows each component to be assessed against its own normal reference. A detector can then ask both whether observed values differ from their predictions and which temporal structure deviates from normality. This provides additional anomaly evidence and a direct basis for component-level attribution.

Existing univariate TSAD methods mainly include reconstruction-based models (Audibert et al., 2020; Yue et al., 2024), forecasting-based methods (Xu et al., 2024; Nie et al., 2023), and density-based variants. Reconstruction methods may interpret benign local variation as anomalous (Yue et al., 2024), while forecasting methods can produce large residuals during volatile but normal regime transitions (Lv et al., 2024). Contrastive methods such as DCdetector (Yang et al., 2023) learn discriminative representations, but do not explicitly separate trend, seasonal, and residual dynamics.

This entanglement also limits explainability: a single score provides little information about which temporal structure has been violated. Recent frequency-domain (Nam et al., 2024) and decomposition-aware methods (Wang et al., 2023; Liu et al., 2025) introduce useful structural priors, motivating further integration of learned component-specific normal representations with anomaly attribution.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

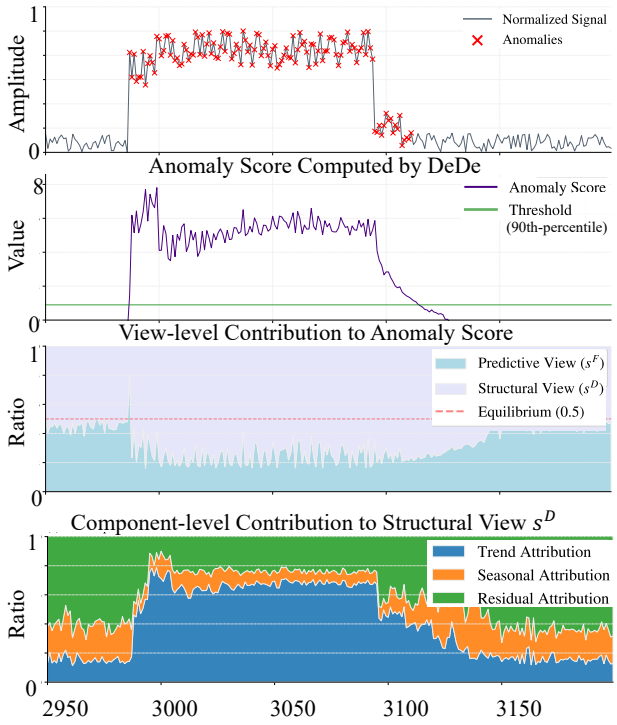


Figure 1: Explainable anomaly detection with DeDe. Given an input time series (top row), DeDe computes the anomaly score (second row) by combining predictive-view and structural-view scores (third row). An anomaly is flagged (red cross, top row) when the score exceeds a predefined threshold (e.g., the 90th percentile, second row). View-level and component-level (bottom row) contributions further provide fine-grained anomaly attribution. More detailed analysis can be found in Section 3.7.

We therefore propose **Time-Series Deep Decomposition (DeDe)**, which integrates decomposition, representation learning, and anomaly scoring. The learnable decomposition parameters and multi-scale encoders are optimized jointly, allowing component representations to adapt to the training objectives. Forecasting encourages retention of predictive temporal information; contrastive learning on concatenated component embeddings encourages consistency across augmented views of the same window. Multi-scale encoding captures dynamics at different temporal resolutions.

At inference, DeDe combines one-step forecasting error with distances from component embeddings to their learned normal distributions. These signals are complementary: a change in temporal organization may leave values within their usual range yet shift a component representation away from its normal reference, providing evidence beyond forecasting error alone. Because structural distances are computed separately for trend, seasonal, and residual representations, the same scoring mechanism supports intrinsic component-level attribution (Fig. 1).

The main contributions of this work are summarized as follows: (1) We introduce a learnable decomposition-based TSAD framework designed to reduce the entanglement of heterogeneous temporal dynamics and enable component-level attribution. (2) We combine multi-scale encoding with forecasting and multi-view contrastive training, and fuse complementary predictive and structural evidence for point-wise detection. (3) Experiments on TSB-AD-U demonstrate improvements over the evaluated baselines; controlled anomaly-injection experiments provide evidence of component-source attribution under the evaluated settings.

2 Related Work

Prior work on time-series anomaly detection (TSAD) spans both shallow and deep paradigms. Shallow methods model normality through statistical or geometric assumptions and identify anomalies via deviations in density, distance, or reconstruction, including KNN/LOF (Ying et al., 2021; Shahrestani

& Sanislav, 2025), Isolation Forest (Liu et al., 2008), and PCA-based approaches (Dani et al., 2024). While efficient and interpretable, they often struggle with non-stationarity and complex multivariate dependencies. Deep TSAD methods instead learn expressive representations of temporal dynamics, typically via reconstruction or forecasting objectives using RNNs or Transformers (Connor et al., 1994; Vaswani et al., 2017), or through contrastive learning to enforce latent consistency (Yang et al., 2023; Yue et al., 2022). Recent foundation models such as Chronos (Ansari et al., 2024) and Moirai/MOMENT (Woo et al., 2024; Goswami et al., 2024) further enhance these approaches with large-scale pretraining. More details are provided in Section A

Another important direction incorporates structural priors through time-series decomposition, separating components such as trend and seasonality to improve modeling (Wu et al., 2021; 2023). However, most existing methods rely on fixed or non-differentiable decomposition schemes, limiting adaptability for anomaly detection, as exemplified by RTdetector (Liu et al., 2025). Meanwhile, explainable TSAD methods mainly focus on post-hoc feature attribution (e.g., LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2017)) or domain-specific priors (Pham et al., 2024; Li et al., 2022; Jeong et al., 2026), requiring external assumptions such as labeled data or domain-specific structures. In contrast, our approach enables end-to-end learning with intrinsic component-level attribution. More details are provided in Section A in Appendix.

3 DEEP DECOMPOSITION (DEDE)

3.1 PROBLEM FORMULATION

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_L\}$ denote a univariate time-series of length L, where each observation $x_t \in \mathbb{R}$. The goal of TSAD is to assign an anomaly score s_t to each timestamp t, such that a higher score indicates a greater likelihood of an anomaly.

3.2 DEDE OVERVIEW

DeDe is an unsupervised model that takes a time series as input and outputs its predicted sequence together with component-wise decomposed embeddings. As shown in Fig. 2, the architecture of DeDe consists of two branches: a prediction branch and a contrastive learning branch. The input signal is fed into both branches and processed by a shared module, namely the Multi-scale Decomposition Encoder (MDE). In the contrastive branch, the MDE further receives augmented views of the input signal. DeDe is trained with a joint objective that combines a prediction loss and a contrastive loss, corresponding to the forecasting task and the contrastive representation learning task, respectively.

At inference time, DeDe computes the anomaly score by combining the prediction error with the deviation of the test embeddings from the learned component-wise normal distributions estimated on the training set. The prediction error and the component-level deviation scores are also used to provide interpretable explanations, indicating both the magnitude and the semantic source (trend/seasonal/residual) of the detected anomaly.

3.3 MULTI-SCALE DECOMPOSITION ENCODER

The primary challenge in TSAD lies in the entanglement of heterogeneous temporal patterns. To address this, DeDe processes the input using a sliding window of length W, denoted as $\mathbf{x} \in \mathbb{R}^W$, and employs a Differentiable STL (Seasonal-Trend decomposition using Loess; D-STL) module to disentangle $\mathbf{x}$ into three additive and interpretable components: trend $\mathbf{x}^{(\mathbf{T})}$, seasonal $\mathbf{x}^{(\mathbf{S})}$, and residual $\mathbf{x}^{(\mathbf{R})}$, each retaining the dimensionality of $\mathbf{x}$. This decomposition is performed along the temporal dimension and integrated into end-to-end training. Period selection is discrete; subsequent operations are differentiable conditional on the selected period.

Trend Component Extraction. The trend component $\mathbf{x}^{(\mathbf{T})}$ captures the low-frequency, long-term progression of the series. To isolate this, we employ a 1D convolutional operator whose kernel is initialized as a weight-averaging filter to provide an explicit inductive bias for low-pass filtering.

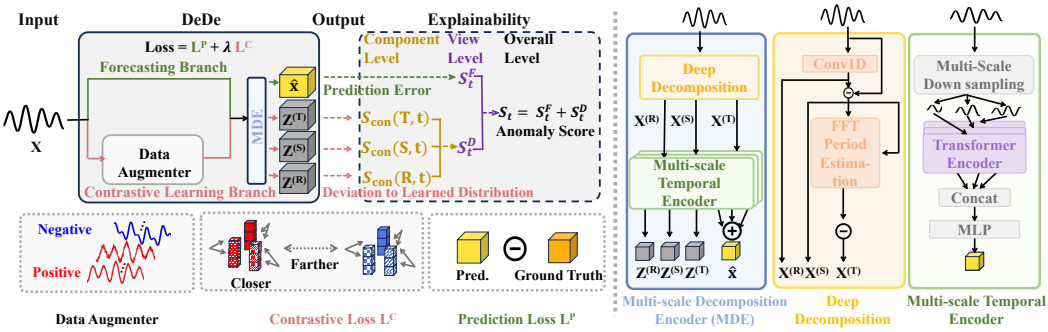


Figure 2: Overview of DeDe which computes a point-wise score for the observation following each input window ($H = 1$). Score subscripts in the schematic denote target timestamps. Left: Two-branch training and inference. The forecasting branch trains MDE by minimizing prediction loss, while the embedding branch trains a projection head with contrastive loss using augmented views and estimates component-wise distribution for trend/seasonality/residual. At inference, time-series are scored by combining prediction error and the Mahalanobis distance of embeddings to the learned distribution. Right: The Multi-scale Decomposition Encoder (MDE) performs deep decomposition into Trend $\mathbf{x}^{(\mathbf{T})}$ Seasonal $\mathbf{x}^{(\mathbf{S})}$ and Residual $\mathbf{x}^{(\mathbf{R})}$, encodes each component with a multi-scale temporal encoder, and outputs predictions (sum) and embeddings (concat).

Formally, for a kernel size k, the trend at timestamp t is computed as:

$$x_t^{(\mathbf{T})} = \sum_{i=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \omega_i x_{t+i}, \quad \text{where } \omega_i^{(0)} = \frac{1}{k}. \tag{1}$$

Unlike static decomposition, these weights remain learnable during training, allowing the model to adaptively refine the smoothing response based on the data distribution. The kernel size k is dynamically scaled with the window length W (e.g., $k = \min(25, \lfloor W/4 \rfloor \cdot 2 + 1)$) to ensure sufficient receptive fields. We apply replicate padding at the boundaries to preserve the temporal resolution and mitigate edge artifacts.

Adaptive Seasonal Decomposition. Unlike standard decomposition which assumes a fixed period, DeDe adaptively estimates the seasonality from the detrended signal $\mathbf{x}' = \mathbf{x} - \mathbf{x}^{(\mathbf{T})}$ through the following steps:

i) *Period Estimation.* To identify the main cyclic pattern, we transform the detrended signal into the frequency domain via the Fast Fourier Transform (FFT). The dominant period τ is derived from the frequency index $\hat{f}$ possessing the maximum power:

$$\tau = \frac{W}{\hat{f}}, \quad \text{where } \hat{f} = \arg\max_{f>0} |\text{FFT}(\mathbf{x}')_f| \tag{2}$$

The period τ specifies a temporal scale. The single-sine projection below estimates its signed amplitude, but does not explicitly represent arbitrary phase.

ii) *Sinusoidal Projection.* To bridge the gap between the estimated frequency and the time-domain representation, we use τ to construct a sinusoidal basis $\mathbf{M} \in \mathbb{R}^W$, where $M_t = \sin(2\pi t/\tau)$. This basis serves as a structural prior, capturing the component aligned with the selected sine wave. Specifically, we project $\mathbf{x}'$ onto $\mathbf{M}$ using a differentiable least-squares formulation to compute the raw seasonal amplitude a:

$$a = \frac{\langle \mathbf{x}', \mathbf{M} \rangle}{\|\mathbf{M}\|^2 + \epsilon} = \frac{\sum_{t=1}^{W} x'_t M_t}{\sum_{t=1}^{W} M_t^2 + \epsilon}, \quad \mathbf{S}_{raw} = a \cdot \mathbf{M}, \tag{3}$$

where ϵ is used to avoid division by zero.

The projection gives $\mathbf{S}_{raw}$ the selected period τ, but a phase-shifted signal orthogonal to the sine basis can have a near-zero projection.

iii) *Seasonal Refinement.* Since real-world seasonality is rarely perfectly sinusoidal, we apply a seasonal refinement network Φ_{seas} (consisting of 1D convolutions and ReLU activations) to learn complex periodic shapes:

$$\mathbf{x}^{(\mathbf{S})} = \Phi_{seas}(\mathbf{S}_{raw}). \tag{4}$$

Since Φ_{seas} operates on the periodic base $\mathbf{S}_{raw}$ rather than the noisy raw signal, it refines this restricted representation without guaranteeing recovery of phase information lost during projection.

Residual Component. The residual component $\mathbf{R}$ captures the stochastic fluctuations, irregular noise, and most importantly, the local outliers that are not explained by the trend or seasonal patterns. It is defined as the remaining signal after removing the other two components:

$$\mathbf{x}^{(\mathbf{R})} = \mathbf{x} - \mathbf{x}^{(\mathbf{T})} - \mathbf{x}^{(\mathbf{S})}. \tag{5}$$

3.4 Multi-scale Temporal Encoder

Following the decomposition layer, we employ three independent encoder-decoder networks, each dedicated to a specific component $c \in \mathcal{C} = \{\mathbf{T}, \mathbf{S}, \mathbf{R}\}$ (trend, seasonal, residual).

Formally, given the historical component window $\mathbf{x}^{(c)}_{t-W+1:t} \in \mathbb{R}^W$, the model learns a mapping to a compact latent vector $\mathbf{z}^{(c)}_t$ and subsequently predicts the next observation $\hat{x}^{(c)}_{t+1}$ with forecast horizon $H = 1$. The forecast given the original signal is the summation of component-wise predictions:

$$\hat{x}_{t+1} = \sum_{c \in \{\mathbf{T},\mathbf{S},\mathbf{R}\}} g_c \left(\phi_c \left(\mathbf{x}^{(c)}_{t-W+1:t} \right) \right), \tag{6}$$

where $\phi_c(\cdot)$ and $g_c(\cdot)$ denote the encoder and prediction head for component c, respectively.

Standard Transformers operating on a fixed time resolution often struggle to simultaneously capture global dependencies and local variations. We address this via a Multi-Resolution Parallel Encoder, which processes the input window across disjoint temporal granularities.

Multi-Granularity Resampling. Given a set of scaling factors $\Gamma = \{0.1, 0.3, 0.5\}$, we resample the input window into multiple scales via 1-D linear interpolation. For each scale $\gamma \in \Gamma$, the rescaled sequence $\tilde{\mathbf{x}}^{(c)}_\gamma$ has a length of $W_\gamma = \max(1, \lfloor W \cdot \gamma \rfloor)$.

This operation acts as a temporal filter: lower scales suppress high-frequency noise to highlight structural dependencies, while higher scales preserve transient details essential for anomaly detection.

Scale-Specific Encoding and Fusion. Following resampling, we employ parallel Transformer branches to extract features from each resolution. For a specific scale γ, the sequence $\tilde{\mathbf{x}}^{(c)}_\gamma$ is first mapped to a hidden dimension d_{model} via a linear projection and augmented with sinusoidal positional encodings ($\mathbf{PE}$) to preserve sequential order.

The augmented tokens are then processed by a scale-specific Transformer Encoder, which utilizes multi-head self-attention to capture temporal dependencies within that granularity:

$$\mathbf{H}^{(c)}_\gamma = \text{TransformerEncoder}_\gamma \left(\text{Linear}(\tilde{\mathbf{x}}^{(c)}_\gamma) + \mathbf{PE} \right), \tag{7}$$

where $\mathbf{H}^{(c)}_\gamma$ represents the sequence of hidden states. Note that we employ independent Transformer weights for each scale to prevent feature interference between different resolutions.

To distill a compact summary, we extract the representation of the final time step, $\mathbf{h}^{(c)}_{\gamma,\text{last}} \in \mathbb{R}^{d_{\text{model}}}$. These summaries are first aggregated via mean pooling to unify the multi-scale temporal views, followed by a non-linear MLP projection. While the attention layers capture temporal dependencies, this specific projection step explicitly models the spatial interactions across the latent feature dimensions:

$$\mathbf{z}^{(c)}_t = \text{MLP} \left(\frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \mathbf{h}^{(c)}_{\gamma,\text{last}} \right). \tag{8}$$

This fusion mechanism ensures the final embedding $\mathbf{z}^{(c)}_t$ encodes both robust temporal dynamics and the intrinsic spatial geometry of the latent manifold.

3.5 TASK-COUPLED TRAINING

We train DeDe by a hybrid objective that simultaneously minimizes forecasting error in the observation space and maximizes representation consistency in the latent space.

Let $\mathcal{B}$ denote the set of window endpoints in a training batch. The overall loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \, \mathcal{L}_{\text{cl}}. \tag{9}$$

where λ controls the weight of the contrastive loss $\mathcal{L}_{\text{cl}}$.

Forecasting loss (observation space). The first objective ensures the model follows the additive decomposition principle. We minimize the Mean Squared Error (MSE) between the next observation x_{t+1} and the sum of one-step component forecasts $\hat{x}_{t+1} = \sum_{c \in \mathcal{C}} \hat{x}_{t+1}^{(c)}$, and minimize the prediction error:

$$\mathcal{L}_{\text{pred}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} (\hat{x}_{t+1} - x_{t+1})^2. \tag{10}$$

Multi-view contrastive loss. To ensure the latent space is robust to noise and local variations (see Fig.4), we employ a multi-view contrastive loss. For a given input batch, we randomly generate $V \geq 2$ augmented views (e.g., via masking, jittering or noise injection). Let $\tilde{\mathbf{z}}_t^{(v)}$ denote the concatenated embedding vector of all components for the v-th view:

$$\tilde{\mathbf{z}}_t^{(v)} = \text{concat}\big(\mathbf{z}_t^{(\mathbf{T},v)}, \mathbf{z}_t^{(\mathbf{S},v)}, \mathbf{z}_t^{(\mathbf{R},v)}\big). \tag{11}$$

To simplify the formulation, we first define the element-wise contrastive loss for a specific anchor window t in view v, and its positive counterpart in view v':

$$\ell(t, v, v') = -\log \frac{\exp(\text{sim}(\tilde{\mathbf{z}}_t^{(v)}, \tilde{\mathbf{z}}_t^{(v')})/\tau_{\text{cl}})}{\sum_{(t',u) \neq (t,v)} \exp(\text{sim}(\tilde{\mathbf{z}}_t^{(v)}, \tilde{\mathbf{z}}_{t'}^{(u)})/\tau_{\text{cl}})}. \tag{12}$$

Here, the denominator sums over all augmented samples in the batch excluding the anchor itself, with $t' \in \mathcal{B}$ and $u \in \{1, \ldots, V\}$. Here sim is cosine similarity and $\tau_{\text{cl}} > 0$ is the contrastive temperature, distinct from the period τ.

The final objective is computed by averaging this loss over all time windows and all positive view pairs:

$$\mathcal{L}_{\text{cl}} = \frac{1}{|\mathcal{B}| V (V-1)} \sum_{t \in \mathcal{B}} \sum_{v=1}^{V} \sum_{v' \neq v} \ell(t, v, v'). \tag{13}$$

3.6 DUAL-BRANCH INFERENCE

At inference time, we compute a point-wise score s_{t+1} for the next observation after history $\mathbf{x}_{t-W+1:t}$ ($H = 1$) by combining two complementary signals: (i) observation-space inconsistency measured by forecasting error, and (ii) latent-space deviation measured by distances to a stable latent manifold structure (SLMS). Each window scores one target timestamp, so no window-to-timestamp aggregation is needed. The score becomes available when x_{t+1} is observed.

SLMS estimation on normal data. We model the normal manifold by estimating the mean $\boldsymbol{\mu}^{(c)}$ and covariance $\boldsymbol{\Sigma}^{(c)}$ of the embeddings for each component $c \in \{\mathbf{T}, \mathbf{S}, \mathbf{R}\}$ using the training set. At inference time, for a test window ending at t, we measure its deviation from this manifold via the regularized Mahalanobis distance, summing the deviations across all components:

$$s_{\text{con}}(c, t) = \sqrt{(\mathbf{z}_t^{(c)} - \boldsymbol{\mu}^{(c)})^\top (\boldsymbol{\Sigma}^{(c)} + \epsilon I)^{-1} (\mathbf{z}_t^{(c)} - \boldsymbol{\mu}^{(c)})}, \tag{14}$$

$$s_{\text{slms}}(t) = \sum_{c \in \{\mathbf{T},\mathbf{S},\mathbf{R}\}} s_{\text{con}}(c, t) \tag{15}$$

Dual-Branch Aggregation (Anomaly score computation). Simultaneously, we quantify observation-space inconsistency using the one-step forecasting error:

$$s_{\text{pred}}(t) = \left(\hat{x}_{t+1} - x_{t+1}\right)^2. \tag{16}$$

To reconcile the scale discrepancy between latent distance and prediction error, we standardize both metrics using statistics derived from the training data to obtain the final point-wise anomaly score s_{t+1}:

$$\begin{cases} s_{t+1} = s_t^D + s_t^F, \\ s_t^D = \dfrac{s_{\text{slms}}(t) - \mu_{\text{slms}}}{\sigma_{\text{slms}}}, \\ s_t^F = \dfrac{s_{\text{pred}}(t) - \mu_{\text{pred}}}{\sigma_{\text{pred}}}. \end{cases} \tag{17}$$

Branch scores are indexed by forecast origin t; the final score is assigned to target $t+1$. Here μ_m and σ_m ($m \in \{\text{slms}, \text{pred}\}$) denote the mean and standard deviation of the respective scores computed on the training data.

3.7 Explainability

The interpretability of DeDe is established through the interaction between the point-wise prediction error (s_t^F) and the latent structural deviation (s_t^D). By monitoring how these scores evolve as the sliding window traverses the anomaly, the model provides component-level diagnostic attribution. Fig.1 provides a multi-faceted view including the true anomalies (marked with red crosses), the anomaly scores, the contribution ratio of s_t^F and s_t^D, and the fine-grained attribution across STL components.

3.7.1 Predictive and Structural Contribution Calculation

To enhance the interpretability of our framework, we implement a component-wise attribution mechanism to quantify the influence of different branches on the final anomaly score s_{t+1}. Since the prediction error s_t^F and the structural deviation s_t^D are aligned within a standardized Z-score space, we can evaluate their relative magnitudes to characterize detected events. The contribution ratio ψ_k for a specific component $k \in \{F, D\}$ at target time $t+1$ is calculated based on the relative magnitude of its standardized score:

$$\psi_k(t+1) = \frac{|s_t^k|}{|s_t^F| + |s_t^D|}. \tag{18}$$

Furthermore, by leveraging the D-STL decomposition, we extend this attribution to a hierarchical level. For magnitude allocation, define raw distances $s_t^{(c)} := s_{\text{con}}(c, t)$, $c \in \mathcal{C}$, distinct from the standardized component distances used in the injection evaluation. The fine-grained attribution ψ_i ($i \in \{\mathbf{T}, \mathbf{S}, \mathbf{R}\}$) is defined as:

$$\psi_i(t+1) = \psi_D(t+1) \cdot \frac{s_t^{(i)}}{\sum_{c \in \mathcal{C}} s_t^{(c)}}. \tag{19}$$

Set a ratio to zero if its denominator vanishes. These ratios describe score magnitudes, not additive signed-score contributions or causal effects.

3.7.2 Value-Pattern Discordance Logic

- **High s_t^F, Low s_t^D (Transient Magnitude Shocks).** This represents an isolated point anomaly (*Spike*). When a spike first appears in the prediction horizon, s_t^F surges due to the immediate numerical mismatch. However, because the historical look-back window still contains normal data, the latent components (trend and seasonal) remain within their SLMS, resulting in a low s_t^D. As the window slides, the spike moves into the historical sequence. The Differentiable STL module subsequently isolates this impulse into the residual component. Consequently, the attribution shifts from prediction error to the residual score, until the spike exits the window.



(a) Dataset_816 (b) Dataset_816

Figure 4: t-SNE of SLMS embeddings on Dataset_816 with (a) and without (b) CL.

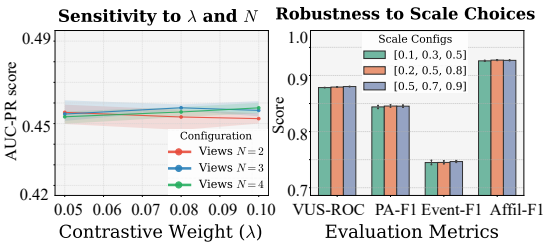


Figure 5: Sensitivity analysis of DeDe regarding hyperparameters. The plotted N denotes the number of views V.

- **Low s_t^F, High s_t^D (Pattern Corruptions (see Fig.1)).** This identifies a *Contextual Anomaly* where the observed values remain within the historical magnitude range but violate the underlying temporal logic (e.g., a Phase Shift or Frequency Modulation). In such cases, s_t^F may remain low because the absolute values are 'plausible', but the latent representations of the trend or seasonal encoders deviate significantly from the SLMS.

To further demonstrate the practical utility of our framework, we provide a detailed exposition of Value-Pattern Consistency Scenarios (see Appendix D) and illustrative industrial scenarios (see Appendix F) in the supplementary material. These sections illustrate the logical mapping between diverse anomaly attributions and their corresponding operational actions.

4 Experiments

4.1 Experimental Settings

Experiments are conducted on the univariate subset of the TSB-AD benchmark (TSB-AD-U) (Liu & Paparrizos, 2024), which provides 350 evaluation time series for comprehensive evaluation. DeDe is evaluated against 12 representative high-performing methods reported in TSB-AD under the univariate setting, together with five more recent TSAD approaches. The standard TSB-AD evaluation protocol is adopted, reporting AUC-ROC, AUC-PR, VUS-ROC, VUS-PR, and multiple F1 variants. All results are produced using the official TSB-AD implementations. Further information on the dataset, baselines, implementation, failure case analysis, and evaluation metrics is provided in the Supplementary Material.

4.2 Results

Comparison with baselines. DeDe improves over the evaluated baselines across all reported metrics on TSB-AD-U, including point-wise, range-aware, and event-level measures. These results support both anomaly ranking and interval localization. TSPulse (Ekambaram et al., 2026) separately reports a fine-tuned TSB-AD-U VUS-PR of 0.52, exceeding our 0.495; this is not a controlled reproduction under our pipeline. TSPulse and PaAno (Park & Kang, 2026) are not included in our experiments.

Ablation Study. We evaluate the contribution of each module in Table 2. The comparison yields two key insights regarding our inference mechanism and architectural design.

First, the *Dual-view scoring strategy* improves detection performance. Relying solely on forecasting error (Score: MSE, Row 4) causes the most significant performance degradation (AUC-PR drops to 0.396), compared with 0.436 when removing contrastive learning while retaining Dual scoring (Row 2; Table A1). These controlled comparisons support both contrastive learning and structural scoring.

Second, *Deep Decomposition* (Row 1) improves detection and enables component-level attribution; *multi-scale encoding* (Row 3) also improves detection. Experiment results show that DeDe achieves better anomaly detection performance with the proposed decomposition module than with classical

Table 1: Performance comparison on TSB-AD-U (higher is better). Best is in **bold**; second-best is underlined. The ML category stands for machine learning, corresponding to the shallow methods. The NN means neural networks, indicating deep methods.

Category	Method	AUC-PR	AUC-ROC	VUS-PR	VUS-ROC	Standard-F1	PA-F1	Event-F1	R-based-F1	Affiliation-F
ML	Sub_PCA (2017)	0.375	0.714	0.424	0.761	0.415	0.559	0.492	0.410	0.850
	Sub_KNN (2000)	0.272	0.757	0.350	0.792	0.339	0.607	0.426	0.318	0.835
	IForest (2008)	0.313	0.708	0.349	0.772	0.349	0.542	0.433	0.324	0.821
	KShapeAD (2017)	0.356	0.743	0.398	0.763	0.399	0.598	0.471	0.402	0.830
NN	USAD (2020)	0.323	0.661	0.361	0.711	0.366	0.502	0.425	0.399	0.836
	AnomalyTran. (2022)	0.080	0.507	0.126	0.563	0.123	0.531	0.329	0.133	0.764
	TimesNet (2023)	0.182	0.611	0.267	0.722	0.238	0.678	0.469	0.210	0.856
	FITS (2024)	0.171	0.611	0.259	0.728	0.227	0.653	0.412	0.195	0.860
	Lag-Llama (2023)	0.252	0.651	0.271	0.720	0.303	0.772	0.590	0.313	0.882
	TimesFM (2024)	0.283	0.671	0.304	0.741	0.342	0.845	0.633	0.347	0.894
	$\text{MOMENT}_{\text{ZS}}$ (2024)	0.302	0.688	0.383	0.756	0.354	0.616	0.493	0.363	0.862
	$\text{MOMENT}_{\text{FT}}$ (2024)	0.305	0.694	0.392	0.766	0.351	0.650	0.495	0.356	0.864
	TimeVQVAE. (2024)	0.277	0.685	0.372	0.768	0.327	0.650	0.452	0.304	0.860
	DADA (2025)	0.302	0.686	0.316	0.750	0.341	0.774	0.571	0.327	0.865
	KAN-AD(2025)	0.354	0.745	0.371	0.816	0.409	0.841	0.699	0.368	0.903
	RTDetector(2025)	0.163	0.556	0.215	0.662	0.225	0.576	0.398	0.199	0.826
	GBOC (2026)	0.246	0.682	0.301	0.757	0.296	0.792	0.613	0.297	0.887
	DeDe (Ours)	**0.461**	**0.834**	**0.495**	**0.880**	**0.513**	**0.850**	**0.748**	**0.436**	**0.927**

Table 2: Ablation study of DeDe on TSB-AD. We evaluate the contribution of Deep Decomposition (**Decomp**), Contrastive Learning objective (**CL**), and Multi-scale features (**Scale**). The **Score** column denotes whether the anomaly score uses the dual-view metric or solely forecasting error (*MSE*). The ablation results with a comprehensive set of evaluation metrics are provided in the Supplementary Material.

Configuration				Performance			
Decomp	CL	Scale	Score	AUC-PR	AUC-ROC	VUS-PR	Std-F1
×	✓	✓	Dual	0.440	0.827	0.477	0.494
✓	×	✓	Dual	0.436	0.826	0.480	0.486
✓	✓	×	Dual	0.438	0.823	0.485	0.499
✓	✓	✓	MSE	0.396	0.772	0.396	0.447
✓	✓	✓	**Dual**	**0.461**	**0.834**	**0.495**	**0.513**

STL. We further evaluate component-wise anomaly attribution through controlled anomaly injection. Detailed results are provided in Appendix C.

Sensitivity Analysis Figure 5 (left) shows stable AUC-PR across contrastive-loss weights $\lambda \in [0.05, 0.10]$ and different numbers of views V. Moderate weights and $V = 3$ or 4 perform well in the sweep; reported experiments use $\lambda = 0.05$. Across multi-scale settings (right), VUS-ROC, PA-F1, Event-F1, and Affiliation-F remain consistent.

5 Conclusion

DeDe jointly learns decomposition and anomaly detection, separating time series into trend, seasonal, and residual components. It combines forecasting error with regularized Mahalanobis distances in the learned embedding spaces, enabling component-level and view-level diagnostic attribution.

Reproducibility statement

We provide detailed descriptions of the proposed method, experimental settings, datasets, baselines, evaluation metrics, and implementation details in the main paper and Appendix. Additional results and ablation studies are also included in the Appendix to facilitate reproducibility. The code will be made publicly available upon acceptance.

AI use statement

LLM tools assisted with language polishing and reference checking. The authors are responsible for verifying all AI-assisted edits and for the final content of this work.

REFERENCES

Charu C Aggarwal and Saket Sathe. Which outlier detection algorithm should i use? In *Outlier Ensembles: An Introduction*, pp. 207–274. Springer, 2017.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

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.

Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

Saurav Kumar Dani, Chander Thakur, Naman Nagvanshi, and Gurwinder Singh. Anomaly detection using pca in time series data. In *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, volume 2, pp. 1–6. IEEE, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj Dayama, Wesley Gifford, and Jayant Kalagnanam. TSPulse: Tiny pre-trained models with disentangled representations for rapid time-series analysis. In *International Conference on Learning Representations*, pp. 62090–62133, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6572721b15834cb37599abec23b5784f-Paper-Conference.pdf.

Naghmeh Ghanooni, Waleed Mustafa, Dennis Wagner, Sophie Fellenz, Anthony Lin, and Marius Kloft. Mitigating spurious features in contrastive learning with spectral regularization. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 107856–107888. Curran Associates, Inc., 2025. doi: 10.52202/085713-3597. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/9aeabac8f13e3ea49eef15cf1154b3c4-Paper-Conference.pdf.

Ramin Ghorbani, Marcel J.T. Reinders, and David M.J. Tax. Pate: Proximity-aware time series anomaly evaluation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 872–883, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671971. URL https://doi.org/10.1145/3637528.3671971.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.

Zelin He, Sarah Alnegheimish, and Matthew Reimherr. Harnessing vision-language models for time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*,

volume 40, pp. 21690–21698, 2026. doi: 10.1609/aaai.v40i26.39319. URL https://ojs.aaai.org/index.php/AAAI/article/view/39319.

Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003.

Woojin Jeong, Geonwoo Shin, and Jaewook Lee. JoCE: Joint counterfactual explanations for interpretable time series anomaly detection. *Pattern Recognition*, 177:113324, 2026. doi: 10.1016/j.patcog.2026.113324. URL https://www.sciencedirect.com/science/article/pii/S003132032600289X.

HyunGi Kim, Jisoo Mok, Hyungyu Lee, Juhyeon Shin, and Sungroh Yoon. CANDI: Curated test-time adaptation for multivariate time-series anomaly detection under distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15018–15026, 2026. doi: 10.1609/aaai.v40i17.38524. URL https://ojs.aaai.org/index.php/AAAI/article/view/38524.

Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. Towards foundation models for zero-shot time series anomaly detection: Leveraging synthetic data and relative context discrepancy. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://github.com/thu-sail-lab/Time-RCD.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Explainable time series anomaly detection using masked latent generative modeling. *Pattern Recognition*, 156:110826, 2024. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2024.110826. URL https://www.sciencedirect.com/science/article/pii/S0031320324005776.

Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. Causal inference-based root cause analysis for online service systems with intervention recognition. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 3230–3240, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539041. URL https://doi.org/10.1145/3534678.3539041.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 108231–108261. Curran Associates, Inc., 2024. doi: 10.52202/079017-3437. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c3f3c690b7a99fba16d0efd35cb83b2c-Paper-Datasets_and_Benchmarks_Track.pdf.

Xinhong Liu, Xiaoliang Li, Yangfan Li, Fengxiao Tang, and Ming Zhao. Rtdetector: Deep transformer networks for time series anomaly detection based on reconstruction trend. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 5788–5796. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/644. URL https://doi.org/10.24963/ijcai.2025/644. Main Track.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Changze Lv, Yansen Wang, Dongqi Han, Xiaoqing Zheng, Xuanjing Huang, and Dongsheng Li. Efficient and effective time-series forecasting with spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Youngeun Nam, Susik Yoon, Yooju Shin, Minyoung Bae, Hwanjun Song, Jae-Gil Lee, and Byung Suk Lee. Breaking the time-frequency granularity discrepancy in time-series anomaly detection. In

Proceedings of the ACM Web Conference 2024, WWW '24, pp. 4204–4215, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645556. URL https://doi.org/10.1145/3589334.3645556.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pp. 1855–1870, 2015.

Jinju Park and Seokho Kang. PaAno: Patch-based representation learning for time-series anomaly detection. In *International Conference on Learning Representations*, pp. 54464–54491, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5991785875cf5eee9c29e2ed7d26d1c0-Paper-Conference.pdf.

Luan Pham, Huong Ha, and Hongyu Zhang. Root cause analysis for microservice system based on causal inference: How far are we? In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pp. 706–715, 2024.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, Sahil Garg, Alexandre Drouin, Nicolas Chapados, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. URL https://openreview.net/forum?id=jYluzCLFDM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

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, 2018.

Udo Schlegel and Daniel A. Keim. A deep dive into perturbations as evaluation technique for time series xai. In Luca Longo (ed.), *Explainable Artificial Intelligence*, pp. 165–180, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-44070-0.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.

Shahed Shahrestani and Ioan Sanislav. How does dimensionality influence outlier detection effectiveness in multivariate geochemical data? insights from lof and if methods. *Earth Science Informatics*, 18(1):27, 2025.

Wenlong Shang, Shihao Tian, Xutong Wan, and Peng Chang. COGNOS: Universal enhancement for time series anomaly detection via constrained gaussian-noise optimization and smoothing. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://github.com/giansha/COGNOS.

Lifeng Shen, Liang Peng, Ruiwen Liu, Shuyin Xia, and Yi Liu. Finding time series anomalies using granular-ball vector data description. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25295–25303, 2026.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *ICLR*, 2025.

Vardhan Shorewala and Shivam Shorewala. Anomaly detection and improvement of clusters using enhanced k-means algorithm. *arXiv preprint arXiv:2505.24365*, 2025.

Emmanouil Sylligardos, Paul Boniol, John Paparrizos, Panos Trahanias, and Themis Palpanas. Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. *Proc. VLDB Endow.*, 16(11):3418–3432, July 2023. ISSN 2150-8097. doi: 10.14778/3611479.3611536. URL https://doi.org/10.14778/3611479.3611536.

Qitai Tan, Yiyun Chen, Mo Li, Ruiwen Gu, Yilin Su, and Xiao-Ping (Steven) Zhang. SynTSBench: Rethinking temporal pattern learning in deep learning models for time series. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4860. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d60361f3d0021c76626b2597957261d6-Paper-Datasets_and_Benchmarks_Track.pdf. Datasets and Benchmarks Track.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dennis Wagner, Tobias Michels, Florian CF Schulz, Arjun Nair, Maja Rudolph, and Marius Kloft. Timesead: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research*, 2023.

Chengsen Wang, Zirui Zhuang, Qi Qi, Jingyu Wang, Xingyu Wang, Haifeng Sun, and Jianxin Liao. Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10758–10774. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/22f5d8e689d2a011cd8ead552ed59052-Paper-Conference.pdf.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Shiwang Xing, Jianwei Niu, and Tao Ren. DoKnowAD: Calibrating normal representations with refined domain knowledge to enhance time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27126–27134, 2026. doi: 10.1609/aaai.v40i32.39927. URL https://ojs.aaai.org/index.php/AAAI/article/view/39927.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with $10k$ parameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bWcnvZ3qMb.

Jiawei Yang, Jingdong Chen, and Susanto Rahardja. Test-time learning for outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 38(8):5420–5432, 2026. doi: 10.1109/TKDE.2026.3689274.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proc. 29th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2023)*, pp. 3033–3045, 2023.

Shi Ying, Bingming Wang, Lu Wang, Qingshan Li, Yishi Zhao, Jianga Shang, Hao Huang, Guoli Cheng, Zhe Yang, and Jiangyi Geng. An improved knn-based efficient log anomaly detection method with automatically labeled samples. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(3):1–22, 2021.

Wenzhen Yue, Xianghua Ying, Ruohao Guo, DongDong Chen, Ji Shi, Bowei Xing, Yuqing Zhu, and Taiyan Chen. Sub-adjacent transformer: improving time series anomaly detection with reconstruction error from sub-adjacent neighborhoods. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/279. URL https://doi.org/10.24963/ijcai.2024/279.

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.

Quan Zhou, Changhua Pei, Fei Sun, HanJing, Zhengwei Gao, Haiming Zhang, Gaogang Xie, Dan Pei, and Jianhui li. KAN-AD: Time series anomaly detection with kolmogorov–arnold networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=LWQ4zu9SdQ.

A Related Work

In this section, we first review representative TSAD methods, including both shallow and deep learning-based approaches. We then discuss time-series decomposition techniques and recent progress in explainable TSAD, which are closely related to DeDe.

A.1 Shallow TSAD methods

Shallow TSAD methods model normality via simple geometric or statistical regularities and score anomalies by density/distance, one-class boundaries, reconstruction residuals, or deviation from cluster prototypes. Representative examples include neighborhood- and density-based detectors (e.g., KNN[1]/LOF (Ying et al., 2021; Shahrestani & Sanislav, 2025)), one-class boundary methods such as One-Class SVM (Schölkopf et al., 1999) and Isolation Forest (Liu et al., 2008), subspace/reconstruction approaches (e.g., PCA/RobustPCA (Dani et al., 2024)), and clustering-based detectors such as CBLOF (He et al., 2003) or enhanced k-means (Shorewala & Shorewala, 2025). These methods are often lightweight and interpretable, but can be sensitive to non-stationarity and multivariate temporal dependencies, and may require feature engineering for complex time-series.

A.2 Deep TSAD methods

Deep TSAD architectures generally learn robust representations of normal dynamics and can be broadly categorized by their primary training objective: (i) minimizing explicit prediction errors via forecasting or reconstruction, (ii) enforcing latent-space consistency through contrastive learning, and (iii) compactness-based one-class objectives that constrain normal representations to a compact region (e.g., a hypersphere), such as Deep SVDD (Ruff et al., 2018) and GBOC (Shen et al., 2026). Our method is most closely related to categories (i) and (ii), and representative approaches from these lines of work are reviewed below.

A.2.1 Reconstruction- and Forecasting-based methods

Forecasting-based approaches utilize temporal predictors (e.g., RNNs or Transformers (Connor et al., 1994; Vaswani et al., 2017)) to model sequential dependencies, scoring anomalies based on large prediction residuals or low likelihood. Reconstruction-based approaches, conversely, learn to recover the prototype of normal patterns via auto-encoding objectives, flagging anomalies by high reconstruction error. Recently, time-series foundation models have extended this paradigm by leveraging large-scale pretraining: Chronos (Ansari et al., 2024) tokenizes time-series values and pretrains T5-based models with a cross-entropy objective, while Moirai/MOMENT employ masked

[1]Throughout this paper, the abbreviations of methods follow the definitions used in their original papers

reconstruction (Woo et al., 2024; Goswami et al., 2024), both naturally compatible with traditional residual-based anomaly scoring. TSPulse (Ekambaram et al., 2026) learns disentangled temporal, spectral, and semantic representations through masked pretraining for downstream tasks including anomaly detection.

Recent work also addresses residual modeling and distribution shift. COGNOS constrains reconstruction residuals toward Gaussian white noise and applies adaptive Kalman smoothing (Shang et al., 2026). To address distribution shift in multivariate TSAD, CANDI selects unlabeled test samples to update input and output adapters around a frozen reconstruction backbone (Kim et al., 2026). For pretrained anomaly detection, TimeRCD (Lan et al., 2026) uses synthetic anomaly labels, while VLM4TS (He et al., 2026) applies pretrained vision and vision-language models to time-series plots.

A.2.2 Contrastive-Learning-based methods

As an alternative to point-wise error, contrastive learning (Ghanooni et al., 2025) aims to capture discriminative representations by enforcing consistency between views in the latent space. Representative methods include DCdetector (Yang et al., 2023), which uses asymmetric dual attention and a contrastive objective to learn permutation-invariant representations, and DoKnowAD (Xing et al., 2026), which uses selected auxiliary datasets from the same domain to calibrate normal representations through joint reconstruction and contrastive learning. PaAno (Park & Kang, 2026) learns patch embeddings using triplet and pretext losses and scores deviations from a memory bank of normal patches. These methods generally avoid the accumulation of uncertainty in the forecasting, but require a careful design of the metric space to ensure that anomalies are separable from normal variations.

A.3 Decomposition-based methods

Decomposition-based models inject structure by separating heterogeneous dynamics (e.g., trend and seasonality), but most designs keep the decomposition rule fixed. Such ideas have been incorporated into deep architectures for forecasting and anomaly detection (e.g., Autoformer (Wu et al., 2021) and TimesNet (Wu et al., 2023)). Recent RTdetector (Liu et al., 2025) further introduces global statistics (e.g., mean/std) to enhance trend-aware reconstruction, using predefined statistical summaries for trend modeling. In contrast, we propose a *learnable decomposition operator* integrated into end-to-end training, with discrete dominant-period selection, enabling task-adaptive decoupling tailored to anomaly detection.

A.4 Explainable TSAD

Explainability in TSAD typically follows several paradigms. However, they rarely provide the intrinsic component-level attribution we target. **Post-hoc Feature Attribution:** Model-agnostic explainers such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) highlight influential segments (Schlegel & Keim, 2023), but are post-hoc and often require fine-grained labels for validation. **Topology-based Root Cause Analysis:** AIOps methods trace anomalies via causal graphs/service topologies over "Golden Signals" (Pham et al., 2024; Li et al., 2022), yet depend on domain-specific structural priors. **Interpretable Classical Detectors:** Classical decomposition- or rule-based detectors are interpretable, but generally lack an end-to-end mechanism to learn representations and provide component-wise attributions under nonstationarity. More recently, JoCE (Jeong et al., 2026) explains anomalies through counterfactual input changes, while VLM4TS (He et al., 2026) provides natural-language justifications for detected intervals. Neither explicitly decomposes anomaly scores into contributions from learned temporal components. In contrast, DeDe provides intrinsic Trend/Seasonal/Residual attribution without external labels or topological priors.

B Experimental settings

Dataset. While earlier studies relied on traditional benchmarks (Wagner et al., 2023), these legacy collections are now limited by their smaller scale and known labeling inaccuracies. To ensure a more rigorous assessment, we conduct our experiments on the TSB-AD benchmark (Liu & Paparrizos,

2024), a large-scale, unified benchmark tailored to rectify these label quality issues and support comprehensive, multi-metric evaluation. The benchmark organizes datasets by dimensionality into univariate (TSB-AD-U) and multivariate (TSB-AD-M) groups. In this work, we use only the univariate subset (TSB-AD-U). For each group, the benchmark further provides a *Tuning* split for hyperparameter selection and an *Eval* split for final performance assessment. In addition, every time series dataset in the collection comes with a predefined cut-off point that determines the training portion, i.e., the segment before the split is used for model fitting. All results in this study are reported using the benchmark's predefined training split (i.e., training on the segment before the cut-off point).

Implementation Details The architecture employs a multi-scale spatio-temporal encoder operating on three distinct temporal scales defined by the set $\Gamma = \{0.1, 0.3, 0.5\}$, utilizing a hidden dimension of $d_{model} = 64$ and $n_{\text{head}} = 4$ attention heads. The dual-branch objective combines a forecasting Mean Squared Error (MSE) loss and a multi-view InfoNCE contrastive loss. For the contrastive formulation, we define the temperature parameter $\tau_{\text{cl}} = 0.5$, the weighting coefficient $\lambda = 0.05$ for the contrastive loss term. We optimize the model using AdamW with an initial learning rate of 1×10^{-4} and weight decay of 1×10^{-5}. Training is performed with a batch size of 256 for a maximum of 50 epochs on an NVIDIA A100 GPU with 80 GB memory, utilizing early stopping with a patience of 5 epochs and standard Z-score normalization (with a 20% validation split). To ensure a fair comparison, we adopt the optimal hyperparameter settings reported in the TSB-AD-U benchmark for all baselines and fix the sliding-window size to 50. DeDe uses $H = 1$: each history ending at t yields one score for target $t + 1$, without window-to-timestamp aggregation. The maximum number of training epochs is set to 50 for all deep models to maintain a consistent computational budget. For the latent distribution branch, the component-wise mean $\mu^{(c)}$ and covariance $\Sigma^{(c)}$ are estimated from training embeddings, and we add diagonal loading with $\epsilon = 10^{-6}$, i.e., $\Sigma^{(c)} + \epsilon I$, to ensure numerical stability when inverting the covariance matrix.

Contrastive Augmentations For each input window, we randomly sample V augmented views from three transformations that are shared across datasets: (1) jittering, which adds Gaussian noise $\mathcal{N}(0, 0.03)$; (2) masking, which randomly zeros out 10% of time steps; and (3) scaling, which multiplies the series by a factor sampled from $\mathcal{U}(0.9, 1.1)$. These augmentations are used uniformly across the benchmark to avoid dataset-specific tuning of the contrastive branch.

Baseline methods. We evaluate DeDe against 12 representative high-performing detectors provided by the TSB-AD benchmark implementation (univariate setting), and further consider five more recent TSAD methods.

The TSB-AD baselines span three groups: (i) shallow detectors, including Sub_PCA Aggarwal & Sathe (2017), Sub_KNN Ramaswamy et al. (2000), IForest Liu et al. (2008), and KShapeAD Paparrizos & Gravano (2015); (ii) neural approaches, including USAD Audibert et al. (2020), AnomalyTransformer Xu et al. (2022), TimesNet Wu et al. (2023), and FITS Xu et al. (2024); and (iii) time-series foundation models, including Lag-Llama Rasul et al. (2023), TimesFM Das et al. (2024), and MOMENT Goswami et al. (2024)(zero-shot and fine-tuned settings). In addition, we include five recent baselines: KAN-AD Zhou et al. (2025), TimeVQVAE-AD Lee et al. (2024), DADA Shentu et al. (2025), RTDetector Liu et al. (2025), and GBOC Shen et al. (2026).

All comparisons follow the standard TSB-AD protocol and use the official benchmark implementations. Standard shallow and deep baselines are trained under the benchmark's recommended tuning procedure on the provided tuning split, while foundation models are evaluated in their officially supported transfer settings within TSB-AD rather than through extra ad hoc fine-tuning. DeDe uses the same data split, preprocessing, and evaluation pipeline as the compared methods.

Evaluation Metrics. Following the standard TSB-AD evaluation protocol, AUC-ROC, AUC-PR, VUS-ROC, VUS-PR, and F1 variants were used. The definitions of these metrics are consistent with the TSB-AD Benchmark.

Statistical Stability Note. In addition to the aggregate benchmark evaluation over 350 evaluation time series, we performed internal multi-run checks on representative subsets to assess training stability under stochastic initialization and augmentation. These checks showed low variability in AUC-PR (below 0.01 standard deviation on the evaluated subsets), indicating that the reported gains

are stable beyond a single run, although we do not provide benchmark-wide error bars for every compared method in the current submission.

Compute Note. The main training runs were conducted on an NVIDIA A100 GPU with 80 GB memory, using batch size 256, a maximum training budget of 50 epochs, and early stopping with patience 5. These settings define the compute budget used for the reported benchmark runs. As an additional execution-time reference, DeDe achieved 4.52 ms/sample at batch size 1 and 0.035 ms/sample at batch size 128 on a single RTX 4090 with 24 GB memory in our latency benchmark. We do not provide a full project-wide accounting of every exploratory run, but the hardware, memory, training budget, and inference-time measurements above describe the main reported experiments.

Table A1: Ablation study of DeDe on TSB-AD. We evaluate the contribution of Deep Decomposition (**Decomp**), Contrastive Learning objective (**CL**), and Multi-scale features (**Scale**). The **Score** column denotes whether the anomaly score uses the dual-view metric or solely forecasting error (MSE).

Configuration				Performance								
Decomp	**CL**	**Scale**	**Score**	**AUC-PR**	**AUC-ROC**	**VUS-PR**	**VUS-ROC**	**Std-F1**	**PA-F1**	**Event-F1**	**R-F1**	**Affil.-F**
×	✓	✓	Dual	0.440	0.827	0.477	0.876	0.494	0.836	0.737	0.404	0.923
✓	×	✓	MSE	0.446	0.828	0.477	0.876	0.496	0.836	0.730	0.414	0.920
✓	✓	×	Dual	0.438	0.823	0.485	0.876	0.499	0.825	0.724	0.423	0.920
✓	✓	✓	MSE	0.396	0.772	0.396	0.834	0.447	0.845	0.747	0.392	0.916
✓	×	✓	Dual	0.436	0.826	0.480	0.875	0.486	0.820	0.716	0.399	0.918
✓	✓	✓	**Dual**	**0.461**	**0.834**	**0.495**	**0.880**	**0.513**	**0.850**	**0.748**	**0.436**	**0.927**

Table A2: Deep decomposition versus classical STL on the evaluated TSB-AD subset. Both variants retain contrastive learning and Dual scoring. Higher values are better; bold indicates the higher mean. F1 metrics follow the benchmark's oracle-threshold evaluation protocol.

Decomposition	AUC-PR	AUC-ROC	VUS-PR	VUS-ROC	Std-F1	PA-F1	Event-F1	R-F1	Affil.-F
Classical STL	0.404	0.800	0.459	0.863	0.458	0.830	0.731	0.393	0.910
Deep (DeDe)	**0.461**	**0.834**	**0.495**	**0.880**	**0.513**	**0.850**	**0.748**	**0.436**	**0.927**

C Additional Results

C.1 Comparison with Classical STL

To assess the contribution of deep decomposition, we replace it with classical STL, which uses LOESS smoothing to separate each input window into trend, seasonal, and residual components. The seasonal period is estimated by FFT after moving-average detrending. We retain the same downstream architecture, contrastive learning objective, and Dual scoring rule to compare the two decomposition approaches within the same detection framework.

Table A2 shows higher mean scores for DeDe across all nine metrics on the TSB-AD subset. AUC-PR increases from 0.404 to 0.461 and VUS-PR from 0.459 to 0.495, corresponding to gains of 5.7 and 3.6 percentage points, respectively. Std-F1 and R-F1 also increase by 5.5 and 4.3 percentage points. These improvements support the contribution of deep decomposition to detection performance within the same downstream framework.

C.2 Component-wise Anomaly Attribution

We evaluate whether DeDe's component-wise structural distances identify the source of an anomaly. Using the basic signal forms in SynTSBench Tan et al. (2025), we construct sequences by combining a linear trend, sinusoidal seasonality, and Gaussian noise. We inject a trend level shift, a seasonal amplitude increase, or a residual amplitude increase into one component at a time, leaving the others unchanged. This provides a known anomaly source against which to evaluate attribution. Each injected anomaly has smooth onset and offset transitions. We generate 50 sequences of 3,000 observations, using the first 2,000 for training and the remaining 1,000 for evaluation. Anomalies are injected into the evaluation segment, and one model is trained per sequence for each of three random seeds.

For each event, we predict the source as the component with the largest mean standardized structural distance over the evaluation interval. The 50 sequences, three training seeds, and three anomaly categories yield 450 attribution evaluations, with 150 per category.

Figure A1 shows the observed signals and component-wise responses to three anomalies injected separately into the same sequence. Panels (a)–(c) correspond to trend, seasonal, and residual anomalies, respectively. In each case, the corresponding component has the largest mean standardized structural distance over the highlighted evaluation interval, correctly identifying the injected source. Table A3 summarizes the predicted and injected anomaly sources across the evaluated sequences

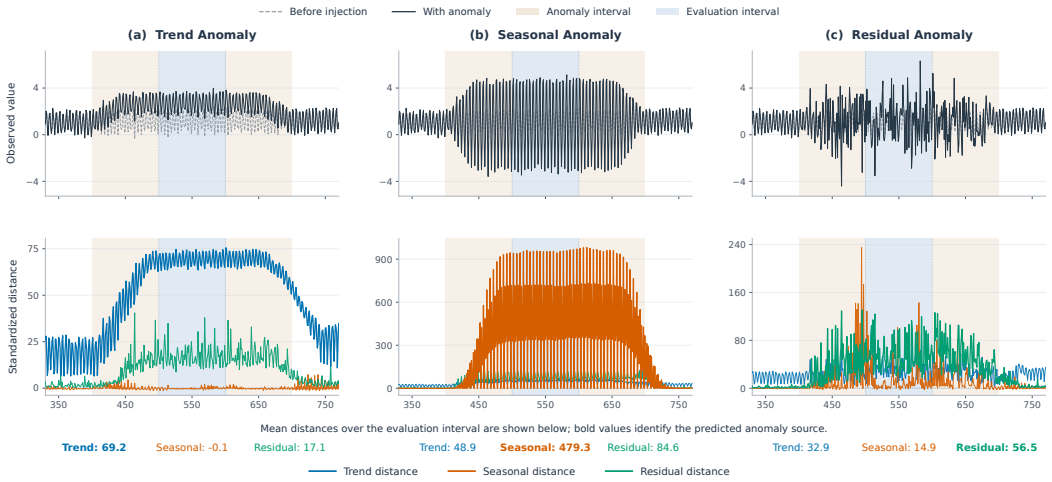


Figure A1: Component-wise distances identify trend, seasonal, and residual anomalies in the illustrated sequence. Top: observations before and after anomaly injection. Bottom: standardized structural distances. For each anomaly, the corresponding component achieves the highest mean distance over the evaluation interval, yielding correct source attribution. Beige and blue shading indicate the anomaly and evaluation intervals, respectively. Values below the panels are interval-mean distances, with the maximum in bold. Distance axes vary across columns.

Table A3: Anomaly-source attribution for DeDe across 450 evaluations (150 per category). Rows indicate injected sources and columns indicate predicted sources. Each entry reports the event count and row percentage in parentheses; diagonal entries are in bold.

	Predicted source		
Injected source	Trend	Seasonal	Residual
Trend	**141 (94.00%)**	4 (2.67%)	5 (3.33%)
Seasonal	2 (1.33%)	**138 (92.00%)**	10 (6.67%)
Residual	7 (4.67%)	4 (2.67%)	**139 (92.67%)**

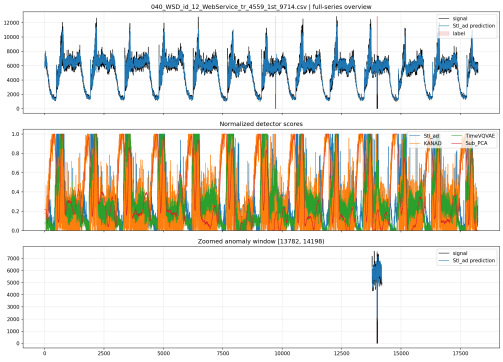
and training seeds. DeDe achieves attribution accuracies of 94.00%, 92.00%, and 92.67% for trend, seasonal, and residual anomalies, respectively, with an overall accuracy of 92.89%.

D Value-Pattern Consistency Scenarios

- **Low s_t^F, Low s_t^D (Stable Alignment).** Low predictive and structural scores indicate agreement with the training reference under both views. Negative Z-scores reduce the combined score, but do not by themselves guarantee a normal label or a low false-positive rate.
- **High s_t^F, High s_t^D (Total Breach).** High scores in both views indicate simultaneous predictive and structural deviations. This supports further investigation but does not establish a systemic failure, its severity, or its physical cause.

E Failure Cases and Benchmark-Level Difficulty

The following examples illustrate representative difficult cases that were challenging not only for DeDe but also for strong baseline detectors. Rather than DeDe-specific outliers, these cases reflect benchmark-level difficulty regimes where anomaly evidence is weak relative to the background dynamics or where the labels only partially align with the underlying structural irregularity.

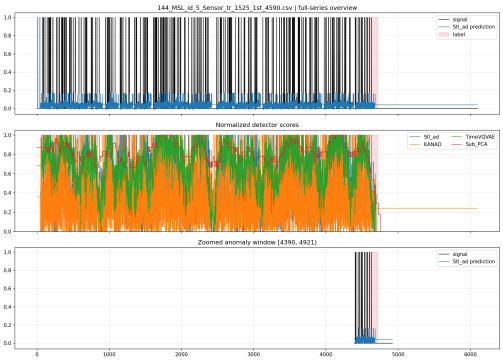


(a) Overview and detector-score comparison.

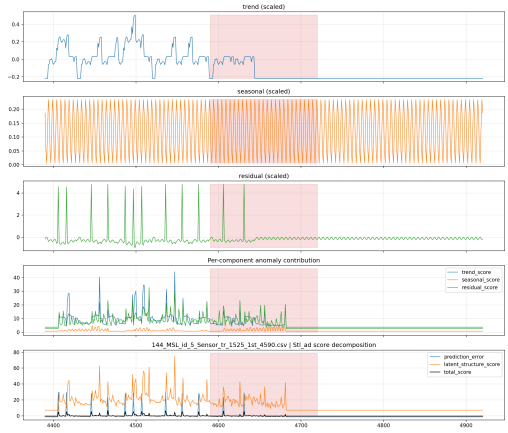


(b) Component attribution and DeDe score decomposition.

Figure A2: WSD failure case. The anomaly consists of several very short events embedded in strong periodic background dynamics. DeDe and strong baselines all struggle to isolate the labeled events consistently, indicating that the dominant difficulty is short-duration anomaly detection under heavy seasonality rather than a method-specific failure.



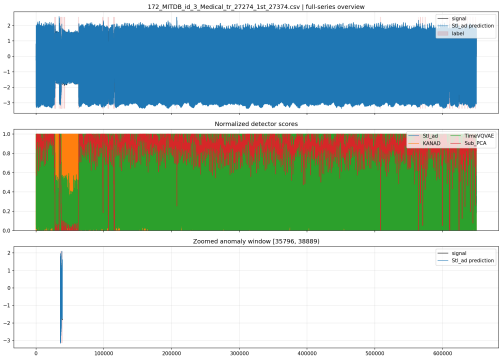
(a) Overview and detector-score comparison.



(b) Component attribution and DeDe score decomposition.

Figure A3: MSL failure case. The labeled anomalous block has low contrast relative to frequent normal fluctuations, so detector scores remain diffuse across the neighborhood. This example illustrates how weak anomaly-to-background contrast can limit all compared methods even when the model decomposition remains interpretable.

Observed failure modes. Across the selected cases, we repeatedly observe three patterns: weak anomaly-to-background contrast, very short anomalous events under strong periodic structure, and long structured sequences with repeated abnormal waveforms that remain morphologically close to normal behavior. These cases are consistent with the limitations discussed in the main paper: the benchmark contains events whose detectability depends strongly on annotation granularity, background seasonality, and the local contrast between anomalous and normal subsequences.



(a) Overview and detector-score comparison.



(b) Component attribution and DeDe score decomposition.

Figure A4: MITDB failure case. The sequence is long and contains many repeated abnormal events whose morphology stays close to normal beats, which makes it difficult to separate point-wise labels from broader structural irregularity. In this regime, both DeDe and the baselines exhibit unstable localization despite repeated score peaks.

F Illustrative Industrial Scenarios

Table A4 presents illustrative mappings from anomaly attribution to possible industrial explanations and actions for expert review. These examples are not deployment evidence or causal diagnoses; component attribution alone cannot establish a physical root cause.

Table A4: Illustrative industrial examples: anomaly attribution, possible explanations, and actions for review.

Category	Sub-type	Diagnostic Attribution	Possible Causes & Actions for Review
Point	**Spike / Surge**	Fidelity Score $s^F \uparrow$	**Causes:** Sensor glitches, EMI, voltage fluctuations. **Action:** Suppress false alarm; inspect sensor shielding.
Interval	**Trend Shift**	Structural Comp. $\psi_T \uparrow$	**Causes:** Tool wear, filter clogging, aging, leakages. **Action:** Schedule maintenance; replace consumables.
	Seasonal Error	Structural Comp. $\psi_S \uparrow$	**Causes:** Multi-machine desync, batch process delays. **Action:** Recalibrate control logic; optimize cycle timing.
	Residual Burst	Structural Comp. $\psi_R \uparrow$	**Causes:** Loose components, unstable vibrations. **Action:** Perform Vibration Analysis (VSA); stabilize process.
Mixed	**Regime Shift**	Synchronized Breach ($s^F \uparrow$ and $s^D \uparrow$)	**Causes:** Major mechanical failure, raw material change. **Action:** Emergency shutdown; Implement multi-dimensional RCA.

G Limitations and Broader Impacts

Limitations. This study focuses on univariate TSAD and reports results on the TSB-AD-U benchmark, which provides a controlled basis for comparison but does not by itself fully characterize multivariate scenarios or cross-domain generalization. To provide a more transparent evaluation, the supplementary material includes representative benchmark-level failure cases, such as weak anomaly contrast, strong periodicity, and repeated waveform structures, where both DeDe and competitive baselines remain challenged. In addition, DeDe's decomposition and multi-scale design introduce higher computational cost than simpler single-branch detectors. While this cost remains practical for offline analysis and moderate-rate monitoring, further optimization may be needed for latency-critical deployment.

Broader impacts. This work can have a positive impact by improving early anomaly detection and diagnosis in industrial monitoring, infrastructure maintenance, and other safety-critical forecasting workflows where timely interventions reduce downtime and operational risk. At the same time, incorrect deployment may create harms: false positives can trigger unnecessary inspections or shutdowns, false negatives can delay responses to real failures, and explainable anomaly scores could be misused in high-stakes monitoring scenarios without sufficient domain oversight. We therefore recommend that deployments use human review, domain-specific validation, and clear operating policies before acting on model outputs.