HYDRA: A Multi-Level Hierarchy-Driven Approach for Robust Anomaly Detection in Time Series

MINGYI HUANG, The Ohio State University, USA
QINGHUA LIU, The Ohio State University, USA
PAUL BONIOL, Inria, DI ENS, PSL, CNRS, France
JOHN PAPARRIZOS, The Ohio State University, USA and Aristotle University of Thessaloniki, Greece

Time-series anomaly detection is critical across various domains. Despite advances in neural networks and foundation models, recent studies show that traditional data mining methods remain highly competitive due to their effectiveness and scalability. However, these approaches suffer from distinct limitations: discord-based methods fail in the presence of repeated anomalies, whereas clustering-based techniques, though mitigating this issue, struggle to capture fine-grained deviations. Moreover, both approaches rely on distance computation, whose effectiveness fundamentally depends on data normalization, with z-score serving as the de facto standard. However, we observe that while normalization reduces scale bias and can enhance anomaly detectability, it may also suppress amplitude-driven anomalies, making the choice of an appropriate normalization scheme both critical and non-trivial. To address these challenges, we propose HYDRA, a multi-level hierarchical and unsupervised approach that integrates the strengths of distance-based methods while reducing reliance on explicit normalization. HYDRA (i) employs a lightweight approximate nearest-neighbor detector with graph-based selection to identify representative subsequences; (ii) constructs multi-resolution representations of the time series and aggregates anomaly evidence from fine to coarse scales; and (iii) introduces a hierarchical ensemble mechanism that fuses level-wise scores to improve robustness against contamination and scale imbalance. This design allows HYDRA to detect diverse anomaly types, from short, isolated discords to long, persistent deviations, allowing it to detect patterns overlooked by single-scale methods. Extensive evaluation on 40 univariate and multivariate time-series anomaly detection datasets from the TSB-AD benchmark demonstrates that HYDRA achieves state-of-the-art performance, ranking first among 40 competing algorithms, while maintaining scalability to ultra-long sequences. We open-source our code at https://github.com/thedatumorg/HYDRA to facilitate reproducibility.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**; **Machine learning approaches**; **Anomaly detection**; • **Information systems** → **Temporal data**; **Data mining**.

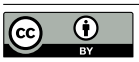
Additional Key Words and Phrases: Time-Series Analysis, Unsupervised Learning, Ensemble Learning

ACM Reference Format:
Mingyi Huang, Qinghua Liu, Paul Boniol, and John Paparrizos. 2026. HYDRA: A Multi-Level Hierarchy-Driven Approach for Robust Anomaly Detection in Time Series. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 197 (June 2026), 30 pages. https://doi.org/10.1145/3802074

1 Introduction

Time-series data arise in a wide range of real-world applications [32, 69, 73, 90, 91] and support many fundamental analysis tasks, including clustering [10, 31, 52, 81, 82, 88, 89, 93], similarity

Authors' Contact Information: Mingyi Huang, The Ohio State University, Columbus, Ohio, USA, huang.5749@osu.edu; Qinghua Liu, The Ohio State University, Columbus, Ohio, USA, liu.11085@osu.edu; Paul Boniol, Inria, DI ENS, PSL, CNRS, Paris, France, paul.boniol@inria.fr; John Paparrizos, The Ohio State University, Columbus, Ohio, USA and Aristotle University of Thessaloniki, Greece, paparrizos.1@osu.edu.



ACM 2836-6573/2026/6-ART197
https://doi.org/10.1145/3802074

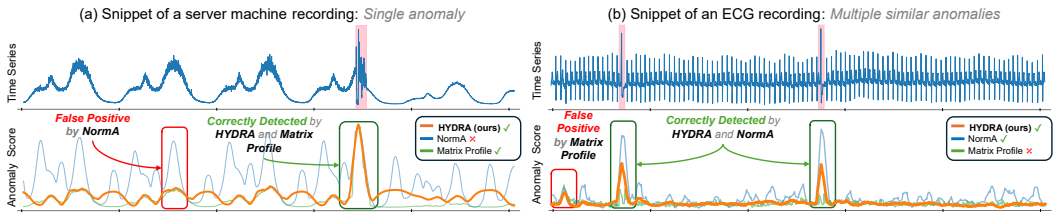


Fig. 1. HYDRA versus state-of-the-art models NormA [13] and Matrix Profile [119] on two time series [33, 106] from TSB-AD benchmark [63]. HYDRA successfully detects anomalies in both cases where either NormA or Matrix Profile fails, highlighting the effectiveness of our proposed multi-level hierarchy-driven strategy.

search [28, 79, 87, 92, 115], forecasting [48, 55], and anomaly detection [14, 16–19, 21, 77, 107, 108]. Among them, time-series anomaly detection (TSAD) plays a particularly vital role, as anomalous patterns often correspond to rare but critical events. In industrial monitoring, early detection of abnormal signals can prevent and explain equipment failures [36, 37, 72, 105, 106]; in financial systems, robust detection mechanisms can uncover fraud and systemic risks [99]; and in healthcare, timely identification of irregular physiological patterns can enable early intervention for severe conditions [23, 38]. Moreover, the proliferation of large-scale sensing and IoT applications [41, 44–46, 49, 56, 57, 64, 74, 75, 85] has led to massive and increasingly heterogeneous time-series data, making TSAD challenging in terms of accuracy, robustness, and evaluation [11, 61, 76, 121].

Motivation. Despite substantial progress in time-series anomaly detection, driven recently by advanced neural networks and foundation models [27, 54, 59, 60], recent benchmark studies [63, 83, 102] reveal that traditional data mining methods remain highly competitive due to their effectiveness and scalability. Among these, distance-based approaches such as Matrix Profile (MP) [119], KMeansAD [65, 114], and NormA [13] often rank among the top-performing methods. However, their underlying assumptions about what constitutes an anomaly introduce structural limitations that hinder robustness in real-world scenarios where anomaly behaviors can be diverse.

Discord-based methods like MP define anomalies as subsequences that are most dissimilar to the rest of the series. This assumption works well for isolated fine-grained anomalies, as illustrated in Figure 1(a). However, when anomalies exhibit similar structure, they become each other's nearest neighbors (i.e., the "twin-freak" problem), leading to reduced distances and suppressed anomaly scores, thereby concealing repeated abnormal patterns (Figure 2(b)).

To mitigate this, clustering-based approaches such as KMeansAD and NormA shift the focus from discord computation to cluster formation. These methods effectively detect repeated or shape-consistent anomalies, as shown in Figure 1(b), and alleviate twin-freak suppression. However, this paradigm assumes anomalies are sparse and well-separated. When anomalies are frequent or shape-consistent, they are absorbed into normal clusters and thus overlooked, leading to false negatives (Figure 2(c)). NormA introduces cluster-level weighting to partially address this issue, yet its coarse cluster abstraction fails to capture fine-grained discords (e.g., subsequence 6 in Figure 2) and remains sensitive to the choice of cluster number, which may induce centroid distortion and contamination by outliers as shown in Figure 2(d).

Collectively, these observations point to two core limitations: (i) both methods are vulnerable to anomaly-induced ***contamination*** (as shown in Figure 2), and (ii) reliance on a ***single*** anomaly semantic assumption limits the ability of existing models to generalize across different anomaly behaviors. This contrast highlights the necessity of a unified framework that inherits the strengths of both paradigms, and motivates a ***contamination-free*** selection mechanism within a ***multi***-level framework that integrates their complementary advantages.

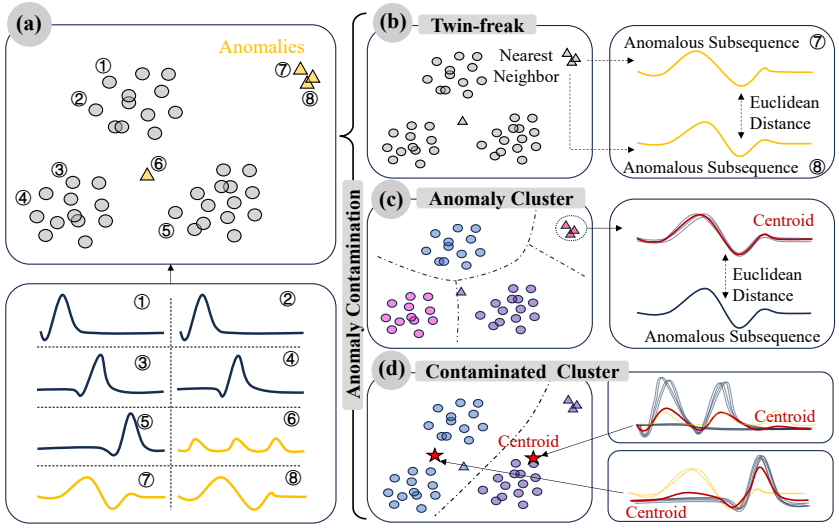


Fig. 2. (a) Two-dimensional subsequence embedding with outliers and duplicate anomalies: the yellow triangles and curves mark anomalies; all other points/curves are normal. (b) Anomalies are not always top discords. (c) Concentrated anomalies are grouped into a single cluster. (d) Misspecified cluster number distorts centroids (red stars/lines), whose shapes deviate from normal subsequences.

A Largely Overlooked Aspect in Data Preprocessing. At the core of distance-based approaches lies normalization, a standard preprocessing step used to reduce scale variance in time-series data. Among available techniques, z-score normalization, which subtracts the mean and divides by the standard deviation, has become a de facto standard in the time-series community [26, 63, 84, 86, 116]. However, when anomalies are amplitude-driven (e.g., spikes), such normalization inadvertently attenuates the large magnitude deviations that signal anomalous behavior. Conversely, for shape-based anomalies (e.g., mean shift, gradual drift), normalization can improve detectability by mitigating scale bias and facilitating more consistent distance comparisons. Since no single normalization strategy is universally effective, committing to a fixed policy such as z-score introduces instability across datasets and anomaly characteristics, motivating the development of methods that reduce reliance on explicit normalization choices.

Table 1. Comparison of state-of-the-art time-series anomaly detection methods across key properties. ✓ denotes support and ✗ denotes lack of support. *Uni* and *Multi* indicate applicability to univariate and multivariate time series. *Unsupervised* denotes label-free operation. *Contamination* indicates resilience to anomaly contamination.

Method	Uni	Multi	Unsupervised	HP-Light	Contamination
MP [119]	✓	✓	✓	✓	✗
NormA [13]	✓	✗	✓	✗	✗
LSTMAD [66]	✓	✓	✗	✗	✗
MOMENT [34]	✓	✓	✓/✗	✓/✗	✗
HYDRA (Ours)	✓	✓	✓	✓	✓

Our Proposed Approach. To address the aforementioned limitations, we introduce the **H**ierarch**Y**-**D**riven Approach for **R**obust **A**nomaly Detection (HYDRA), a novel hierarchical method for time series anomaly detection. HYDRA performs level-wise subsequence selection that retains a subset of candidates to form next-level representatives. This progressive selection induces a hierarchy of

representative subsequences whose upper levels are increasingly purified, ultimately approaching a contamination-free representative set. In parallel, HYDRA constructs multi-resolution representations of the input and, at each level, runs a lightweight approximate-nearest-neighbor detector to produce resolution-specific scores. Fine levels target short, isolated discords, whereas coarse levels capture persistent or clustered deviations. Rather than relying solely on post-hoc score fusion, HYDRA aggregates level-specific, standardized evidence at its native resolution, preserving attribution and maintaining semantic consistency across scales. This cross-resolution consensus enables the detector to reconcile fine-grained local deviations with broader structural irregularities, thereby compensating for scale variation without committing to any single normalization view. In addition, the hierarchical selection process progressively suppresses nearest-neighbor masking among similar anomalies, reduces sensitivity to clustering assumptions, and lessens reliance on clean data by constructing increasingly purified representative sets as the hierarchy ascends.

As shown in Table 1, which summarizes top-performing models from recent benchmarks [63, 102], MP [119] has demonstrated strong detection performance across a wide range of scenarios over years of development and extensive empirical validation, and has evolved from its original univariate formulation to variants that support multivariate settings [118]. However, MP may fail when anomalies recur with similar patterns, revealing a limited tolerance to anomaly contamination. NormA [13] addresses the twin-freak issue but requires an accurate choice of the number of clusters and still struggles to detect subtle deviations. Furthermore, representative neural network-based solutions such as LSTMAD [66] and foundation models like MOMENT [34] have both zero-shot and fine-tuned variants: the zero-shot version is unsupervised and HP-light, whereas the fine-tuned version requires training and hyperparameter tuning. In both cases, however, such models typically assume clean training data and are susceptible to anomaly leakage [39], whereas HYDRA is training-free, parameter-light, and robust under anomaly contamination. Comprehensive evaluation with rigorous statistical analysis on 40 univariate and multivariate datasets from the TSB-AD benchmark [63] demonstrates that HYDRA consistently achieves state-of-the-art performance while maintaining superior scalability on ultra-long time series. A companion demo paper further presents an interactive system for exploring HYDRA and its hierarchical anomaly evidence [42].

Overall, the paper is organized as follows. We first discuss the preliminaries and related work (**Section 2**) and then present our main contributions as follows:

- We revisit the role of normalization and demonstrate that no single strategy generalizes across anomaly types, motivating a hierarchy-driven alternative (**Section 3.1**).
- We introduce the concept of multi-level representations of time series and formalize key properties of hierarchical structure and level-wise score aggregation (**Section 3.2**).
- We present HYDRA, an unsupervised hierarchical approach that constructs representative subsequence sets and fuses level-wise scores through a hierarchical ensemble to detect both subtle discords and persistent deviations without assuming specific anomaly semantics (**Section 4**).
- We conduct extensive experiments on a recent large-scale, heterogeneous, and curated benchmark, accompanied by systematic ablation studies, to validate the robustness and design effectiveness of the proposed approach (**Section 5**).

Finally, we conclude with the implications of our work (**Section 6**).

2 Preliminaries and Related Work

We first present the preliminaries of time-series anomaly detection (Section 2.1), followed by a review of related work (Section 2.2).

2.1 Preliminaries

We summarize key concepts and definitions relevant to time-series anomaly detection as follows.

Time Series. Let $\mathbf{y} = (y_1, \ldots, y_T) \in \mathbb{R}^T$ denote a time series of length T, where y_t represents the observation at time step t.
Windowing. Given a window length m with $1 < m \leq T$, the i-th subsequence[1] starts at index $t_i = 1 + (i-1)s$ and is defined as $\mathbf{x}_i = (y_{t_i}, y_{t_i+1}, \ldots, y_{t_i+m-1}) \in \mathbb{R}^m$, for each valid i such that $t_i + m - 1 \leq T$, where s is the stride, defined as follows.
Stride. We use $\lfloor \cdot \rfloor$ for floor. The stride $s \in \mathbb{N}$ controls the overlap between adjacent windows; $s = 1$ yields fully overlapping windows. The total number of windows is $N = \lfloor (T-m)/s \rfloor + 1$. We denote the collection of all windows by $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$. When a per-window score must be aligned back to the timeline, we map $\mathbf{x}_i$ to its center index $c_i = t_i + \lfloor (m-1)/2 \rfloor$, unless stated otherwise.
Distance Measure. We use the *unnormalized Euclidean distance* on raw windows: $d(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_2$. We deliberately avoid per-window normalization so as to preserve absolute magnitude, which is crucial when the discriminative signal lies in amplitude (e.g., spike depth), where normalization can compress the gap between ordinary fluctuations and large spikes, as discussed in Section 3.1.

2.2 Related Work

2.2.1 Time-Series Anomaly Detection. Time-series anomaly detection methods can be broadly categorized according to their underlying detection strategy [14, 76]: (i) distance-based methods, which operate on subsequences and detect anomalies by computing deviations from a reference model or representative set [13, 20, 22]; (ii) density-based methods, which identify anomalies as points residing in low-density regions of the data distribution, rather than relying solely on nearest-neighbor distances [3, 58]; and (iii) prediction-based methods, which train a model on presumed normal data to either reconstruct the input or forecast future observations, flagging anomalies based on reconstruction or prediction residuals [70, 101].

Alternatively, time-series anomaly detection approaches can also be grouped by the underlying learning paradigm they employ: (i) statistical methods, which rely on explicit probabilistic assumptions and treat anomalies as statistical deviations from an expected distribution [2, 103]; (ii) neural network-based methods, which learn implicit representations of normal behavior from anomaly-free training sequences and detect deviations at inference time [66, 106]; and (iii) foundation model-based methods, which leverage large pretrained models, either on time-series data or via cross-modal transfer from other modalities such as texts, to enable generalized anomaly detection without task-specific training [25, 34].

Most existing methods, regardless of their detection strategy or learning paradigm, can be reformulated within a subsequence scoring approach, where overlapping windows are evaluated and assigned abnormality scores. This perspective provides a common view for understanding methods, enables clearer comparison across approaches, and motivates the following discussion.

2.2.2 Subsequence-based Approaches. We assume a sliding-window extraction $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ from $\mathbf{y}$ with stride 1 and window length m, as defined in the preliminaries. Most methods first compute a *window-level* score s_i for each $\mathbf{x}_i$ and then align these scores with the original timeline using an averaging rule defined as follows:

$$S(t) = \frac{1}{|\mathcal{I}(t)|} \sum_{i \in \mathcal{I}(t)} s_i. \tag{1}$$

Here, $\mathcal{I}(t)$ denotes the indices of windows that cover time point t, which can be written as: $\mathcal{I}(t) = \{i : \max(1, t-m+1) \leq i \leq \min(t, N)\}$. We consider two broad types of approaches: (1) methods developed for tabular data and adapted to time series through sliding-window techniques, and (2) algorithms specifically designed for sequential data that exploit temporal structure.

[1] In this paper, we use subsequence and window interchangeably.

Note that the length m is a user-defined parameter for all subsequence anomaly detection methods [13, 15, 119]. It can be set by the domain expert (e.g., cardiologists, who are interested in analyzing heartbeats with a known length) or automatically with the Auto-Correlation Function (ACF). We evaluate the importance of this parameter in Section 5.5. For the rest of this paper, we refer to a method as *hyperparameter-light* if it requires no manual hyperparameter tuning beyond setting the subsequence length m.

Treating Windows as Raw High-Dimensional Points $\mathbf{x}_i \in \mathbb{R}^m$. In this setting, subsequences are embedded directly in $\mathbb{R}^m$, allowing the use of off-the-shelf vector-based anomaly detectors with the distance measure $d(\cdot, \cdot)$ defined earlier.

(i) Proximity-based methods detect anomalies by measuring the distance between a window and its nearest neighbors. A common example is kNN detector [96], where the anomaly score is defined as $s_i = \min_{j \neq i} d(\mathbf{x}_i, \mathbf{x}_j)$ or as the average distance to the k nearest neighbors. While conceptually simple, these methods are highly sensitive to local density fluctuations and overlapping clusters.

(ii) Clustering-based methods assume that normal subsequences form coherent clusters, and anomalies deviate from these cluster structures. For example, KMeansAD [114] computes scores as $s_i = d(\mathbf{x}_i, \boldsymbol{\mu}_{\text{cluster}(\mathbf{x}_i)})$, where $\boldsymbol{\mu}_{\text{cluster}(\mathbf{x}_i)}$ is the assigned cluster centroid. These approaches capture coarse distributional patterns but often fail to isolate subtle or localized deviations.

(iii) Tree-based methods partition the subsequence space recursively and flag windows that are isolated early in the partitioning process. IForest [58] identifies anomalies by their shorter expected partition depth, which offers scalability but relies on axis-aligned separability assumptions that may not hold for complex time-series structures.

These methods produce a sequence of scores $\{s_i\}_{i=1}^{N}$, which are subsequently projected back to the temporal domain via $S(t)$ for evaluation and comparison.

Time-Series-Specific Subsequence Methods. Methods in this category are explicitly designed for time-series data, leveraging temporal overlap, recurrence, or transition structure.

(i) MP [119] defines discord as a subsequence with the largest non-overlapping nearest-neighbor distance. However, MP is prone to the "twin-freak" problem: when multiple anomalous windows exhibit similar patterns, they become nearest neighbors to each other, leading to reduced pairwise distances, and consequently failing to be identified as discords despite being collectively abnormal. Although the Top-k mth-discord strategy, which labels a window as anomalous if it ranks among the k largest mth-nearest distances, can partially mitigate this issue, it assumes prior knowledge of the anomaly count, which is rarely available in practice.

(ii) Series2Graph (S2G) [15] embeds subsequences into a shape-preserving low-dimensional graph where anomaly scores reflect node or transition rarity, effectively handling near-duplicate anomalies and multiple subsequence lengths without supervision. However, S2G assumes structural stationarity, that the time series exhibits stable recurrence patterns such as seasonality or consistent state transitions. When this assumption is violated (e.g., under strong nonstationarity, transient events, weak periodicity, or regime shifts), the graph-based rarity criterion becomes unreliable.

(iii) NormA [13] builds a weighted normal model from representative subsequence clusters and scores anomalies by their weighted distance to this model. While this method captures global data structure, its cluster-level abstraction can obscure fine-grained, localized anomalies that do not significantly influence cluster centroids but remain salient in the raw signal.

Overall, while these time-series–aware methods improve detection accuracy in time series context, they either require structural stationarity, hyperparameter tuning (e.g., cluster count, motif lengths), or assume limited types of anomaly behaviors, which constrains their robustness in heterogeneous real-world scenarios.

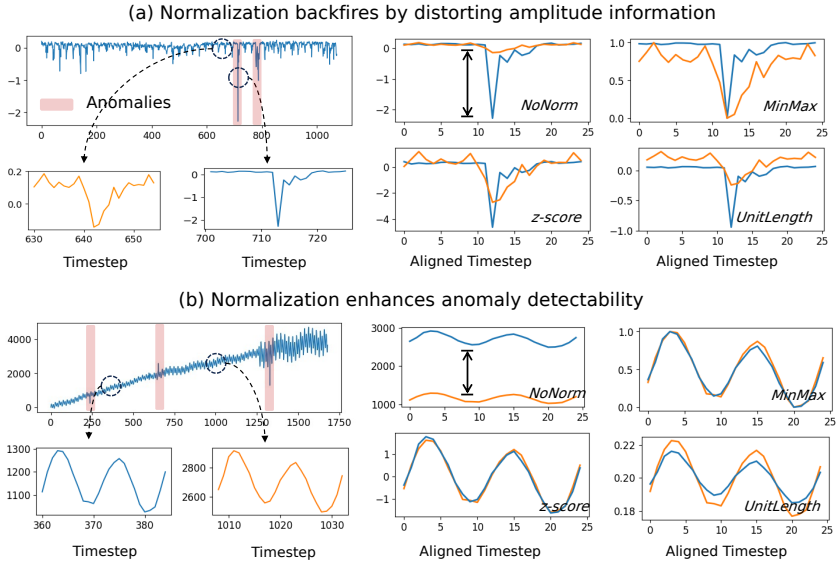


Fig. 3. Effect of normalization on detection performance.

3 Motivation and Problem Formulation

In this section, we outline the motivation behind developing a hierarchical approach for time-series anomaly detection. Specifically, we first discuss the challenges of selecting an appropriate normalization strategy (Section 3.1) and then present empirical and conceptual evidence motivating a hierarchical solution (Section 3.2).

3.1 Normalization: *A Holy Grail Problem*

Normalization refers to the rescaling of time-series data as a preprocessing step to mitigate distortions arising from scale imbalances or offset shifts. In the context of sliding-window-based anomaly detection, normalization is typically applied *per window* rather than over the entire sequence. Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ denote the set of time-series windows, where each window is represented as $\mathbf{x}_i = (y_{t_i}, \ldots, y_{t_i+m-1}) \in \mathbb{R}^m$. Normalization operates on each $\mathbf{x}_i$ to produce a transformed window $\tilde{\mathbf{x}}_i$. Among various techniques, the **Z-score normalization** is the most commonly used, transforming a time series so that the resulting sequence has zero mean and unit variance. Formally, the z-normalized point $\tilde{x}_{i,t}$ is computed as:

$$\tilde{x}_{i,t} = \frac{x_{i,t} - \mu_{\mathbf{x}_i}}{\sigma_{\mathbf{x}_i}}, \quad \text{with} \quad \mu_{\mathbf{x}_i} = \sum_{t=1}^{m} \frac{x_{i,t}}{m}, \sigma_{\mathbf{x}_i} = \sqrt{\sum_{t=1}^{m} \frac{(x_{i,t} - \mu_{\mathbf{x}_i})^2}{m}} \tag{2}$$

An alternative approach, **Min–Max normalization**, rescales values to the $[0, 1]$ range:

$$\tilde{x}_{i,t} = \frac{x_{i,t} - \min_{\mathbf{x}_i}}{\max_{\mathbf{x}_i} - \min_{\mathbf{x}_i}} \tag{3}$$

Unit-length normalization scales the series such that its Euclidean norm equals one:

$$\tilde{x}_{i,t} = \frac{x_{i,t}}{\|\mathbf{x}_i\|_2} = \frac{x_{i,t}}{\sqrt{\sum_{t=1}^{m} \mathbf{x}_i}} \tag{4}$$

However, as illustrated in Figure 3, selecting an appropriate normalization method remains a non-trivial challenge. In some cases (Figure 3(a)), normalization backfires by distorting amplitude information, where high-magnitude spike anomalies are suppressed, reducing their detectability.

In contrast, as shown in Figure 3(b), normalization can enhance anomaly detection by rescaling sequences to a comparable range, thereby facilitating more reliable distance computations among window pairs. This dual effect makes normalization inherently ambiguous, motivating the need for an approach that does not rely on a single normalization choice but instead leverages multi-resolution evidence across hierarchical levels.

In the next section, we demonstrate how our proposed hierarchical approach mitigates the need for explicit normalization. Instead of numerically rescaling values, the multi-level structure induces a form of implicit normalization through consensus across levels, allowing dominant anomaly semantics to emerge naturally without manual preprocessing choices.

3.2 A Hierarchical TSAD Approach

In this section, we present a novel formulation for time-series anomaly detection that eliminates anomaly contamination (Section 3.2.1) and captures diverse types of anomalies (Section 3.2.2).

3.2.1 Multi-level Representation. In this section, we formulate subsequence anomaly detection from a reference-driven perspective. The goal is to construct a compact set of representative windows that capture the typical patterns of the series and to assign anomaly scores to other windows based on their deviation from this reference set. We formalize the problem of robust representative selection under data contamination, introduce a coarse-to-fine hierarchical mechanism that gradually suppresses anomalous representatives, and show that, under mild rarity and majority-consistency assumptions, the procedure converges to an all-normal reference set (without any anomalies). We further define level-wise scoring and develop a scale-stable aggregation strategy that ensures both fine-grained, transient deviations and coarse, persistent irregularities contribute meaningfully to the final anomaly ranking.

Reference-driven formulation. Given a set of windows $\mathcal{X}$, the goal is to select a subset of representative normal windows $\mathcal{R} \subseteq \mathcal{X}$ and score each window by its deviation from $\mathcal{R}$ (i.e., computing the distance to the nearest representative). This design avoids centroid distortion due to misspecified clustering, eliminates the 1-NN suppression effect that plagues MP under near-duplicate anomalies, and removes any dependency on knowing the number of anomalies a priori. As long as $\mathcal{R}$ captures the dominant behavioral patterns, deviations from this reference are detectable without requiring strong stationarity assumptions.

Robust selection under contamination. The key challenge lies in identifying an uncontaminated set of representative normal windows. Equivalently, the task reduces to selecting $\mathcal{R}$ from the set of windows $\mathcal{X}$ in a way that is resilient to the presence of anomalous windows. Contamination can diminish the contrast between normal and abnormal behavior, leading to suppressed anomaly scores and masked outliers. Therefore, the selection process must be conservative, redundancy-aware, and robust to repeated or clustered anomalies as well as mild regime shifts.

Hierarchical screening of representative windows. Building on the above formulation, we introduce a representative-based hierarchical construction. Let $\mathcal{X}^{(0)} = \mathcal{X}$. At level $\ell \geq 0$, we first partition $\mathcal{X}^{(\ell)}$ into local groups, which we call *micro-groups*. Each micro-group contains a small set of neighboring windows, forming a compact cluster, so that windows are partitioned into many small groups and progressively anomalous windows, typically a minority in each group, are filtered out (see Section 4.2 for group building). Formally, we partition the set $\mathcal{X}^{(\ell)}$ into M_ℓ micro-groups denoted as $C^{(\ell)} = \{C_j^{(\ell)}\}_{j=1}^{M_\ell}$, with progressively coarser resolution as ℓ increases. From each group, a representative is selected using the following rule:

$$r_j^{(\ell+1)} \;=\; \text{Select}\Big(C_j^{(\ell)}\Big) \in C_j^{(\ell)}, \qquad j = 1, \ldots, M_\ell \tag{5}$$

The next-level candidate set $\mathcal{X}^{(\ell+1)}$ consists of the representatives selected at level ℓ, formally

defined as:

$$\mathcal{X}^{(\ell+1)} = \mathcal{R}^{(\ell+1)} = \{ r_j^{(\ell+1)} : j = 1, \ldots, M_\ell \} \tag{6}$$

Overall, this process iterates across levels until a single representative remains. Therefore, the challenge is to define a Select function that produces an appropriate set of representatives.

Key Assumption. More precisely, an appropriate Select function should satisfy the following *majority-consistent selection* property. Formally, let each window be associated with a latent label $a_i \in \{0, 1\}$, indicating normal (0) or anomalous (1). We define the set of normal and abnormal windows as follows:

$$\mathcal{N} = \{ \mathbf{x}_i : a_i = 0 \} \qquad \mathcal{A} = \{ \mathbf{x}_i : a_i = 1 \} \tag{7}$$

Moreover, we assume anomaly rarity, i.e., $|\mathcal{A}|/|\mathcal{X}| \leq \eta$ for some small $\eta \leq 0.5$, and a majority-consistent selection rule such that, for a given group C, we have the following property:

$$\text{if } |C \cap \mathcal{N}| > |C \cap \mathcal{A}| \text{ then } \text{Select}(C) \in C \cap \mathcal{N} \tag{8}$$

Under these assumptions, the first hierarchical level produces predominantly normal representatives with only a small number of anomalous ones. As resolution becomes coarser with increasing ℓ, anomalous representatives are absorbed into groups dominated by normal windows and, by the majority-consistency property, are unlikely to be re-selected; consequently, the number of anomalous representatives is monotonically non-increasing across levels and there exists a finite level L such that $\mathcal{X}^{(L)} = \mathcal{R}^{(L-1)} \subseteq \mathcal{N}$, yielding a fully normal representative set suitable for reference-driven scoring. Intuitively, near-duplicate windows tend to fall into the same micro-group, and each group contributes one representative. Even if nearby anomalies yield an anomalous representative, it is re-grouped at the next level with other (mostly normal) representatives and becomes a minority, so it is unlikely to be selected again, yielding progressively purified representative sets. This purification prevents anomalous windows from contaminating the reference set, in contrast to clustering methods that may group nearby anomalies into a cluster and use its centroid as a prototype (avoid centroid smearing). Moreover, because our scores are computed from windows to representatives (rather than window to window), an anomalous window is unlikely to have another anomaly as its nearest reference (avoiding nearest-neighbor suppression).

Figure 4 illustrates this process: at **Level 0**, all windows are initially considered as representatives. At **Level 1**, a stricter selection criterion retains only a reduced subset of representative windows, and due to anomaly rarity, only a single anomalous element persists. After one further hierarchical reduction, this remaining anomalous representative is eliminated at **Level 2**, resulting in an all-normal reference set. With such a clean reference established, deviation-based scoring (e.g., nearest-representative distance) effectively separates anomalous windows from normal behavior.

3.2.2 Score Aggregation. Although there is a level L such that the set of representatives contains only normal windows, relying on a single hierarchy level L is suboptimal for the following two reasons. First, the optimal resolution for detection is unknown a priori in unsupervised settings, making a fixed-level choice prone to failure. Second, a single representative may be insufficient to capture the full spectrum of anomalies. For instance, anomalies with higher amplitude can dominate distance-based scores, suppressing subtle but semantically meaningful deviations.

The hierarchical construction naturally yields multiple levels of representatives: (i) fine-resolution levels, containing many representatives, are sensitive to localized and low-magnitude deviations, while (ii) coarse-resolution levels, with fewer representatives, emphasize persistent or aggregated irregularities. Ensembling both fine and coarse-resolution levels is the only viable solution to accurately identify all types of anomalies. As an example, Figure 4(b) depicts the score (i.e., distance to the nearest representative) for levels 0 and 3. While the score for level 0 (in blue) accurately identifies small and isolated anomalies, the score of level 3 (in orange) correctly identifies the

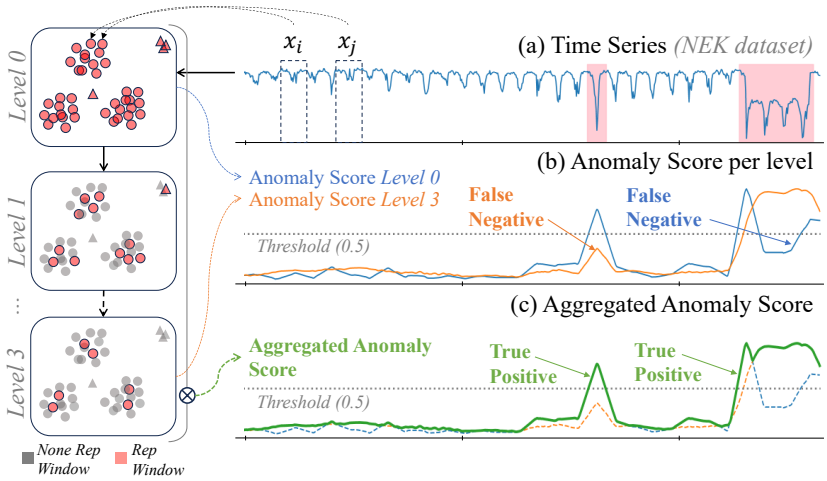


Fig. 4. Hierarchical anomaly detection on a univariate time series. Level 0 and 3 scores are depicted in (b), and the aggregated score of all levels in (c). Single levels suffer scale mismatches that create false negatives. Aggregating scores across levels recovers all anomalies.

large-scale anomaly. Only the aggregation of all levels (in green in Figure 4(c)) identifies both types of anomalies. These observations suggest that results should be combined across levels to capitalize on the strengths of each resolution.

Indeed, a growing body of work in time-series anomaly detection employs score-ensemble strategies to combine the complementary strengths of diverse detectors [62]. However, these approaches typically interact only at the final anomaly scores—a common format that facilitates combination but confines aggregation to the very last stage. This late fusion substantially limits the attainable gains. In contrast, we propose (see Section 4.5) an aggregation mechanism that is native to—and exploits—the internal structure of our algorithm. This allows integration at earlier, semantically richer stages, thereby yielding a more effective and robust aggregation than conventional ensemble methods. In the next section, we detail the hierarchical construction and its score-aggregation mechanism, along with its approximate and multivariate extensions.

4 HYDRA: *Proposed Approach*

Building on the motivation outlined earlier, we present HYDRA, a hierarchical, unsupervised, and parameter-light framework that integrates multi-resolution evidence for time-series anomaly detection. Section 4.1 provides an overview of the pipeline, followed by detailed descriptions of each component: group construction (Section 4.2), representative selection (Section 4.3), score computation and aggregation (Sections 4.4 and 4.5), approximate solution for scaling (Section 4.6), and multivariate extension (Section 4.7)

4.1 Overview

As depicted in Figure 5, starting from the windowed collection $\mathcal{X}$, HYDRA iteratively (i) forms conservative, redundancy-aware groups of windows at the current resolution; (ii) selects a single representative per group to produce a coarser reference set; (iii) computes reference-driven, distance-based scores against that set; and (iv) aggregates scores to capture anomalies across multiple temporal resolutions. Repeating (i)–(ii) yields a sequence of levels that progressively suppress anomalous influence while preserving prevalent structure, thus exposing complementary signals

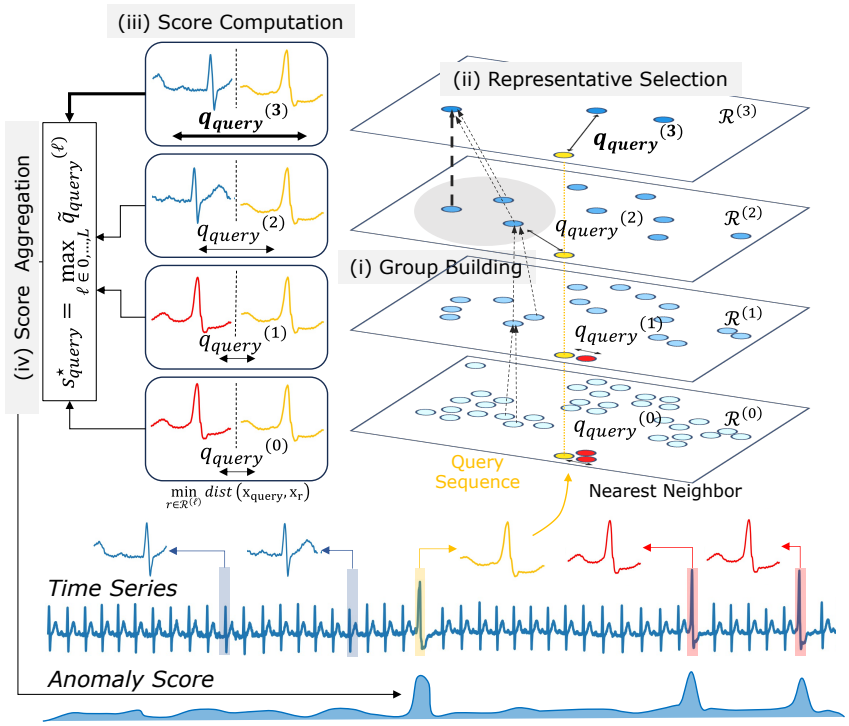


Fig. 5. Overview of HYDRA steps. HYDRA constructs multi-level representative sets via (i) grouping and (ii) selection, (iii) computes reference-based anomaly scores at each level, followed by (iv) robust aggregation to capture anomalies across multiple temporal resolutions.

from fine to coarse scales. The resulting level-wise scores are subsequently normalized and aggregated to balance heterogeneous anomaly types without prespecifying their number. This design counteracts nearest-neighbor suppression among near-duplicate anomalies, reduces sensitivity to degraded cluster representatives caused by anomaly contamination or by grouping multiple anomalies together, and scales to long series under minimal assumptions.

4.2 Group Building

As motivated in Section 3, we first form conservative groups at each level to attenuate anomalous influence before selecting representative windows. We denote the total number of levels by L (automatically determined by HYDRA), index levels by $\ell = \{0, \ldots, L\}$, and the number of representatives in level ℓ by N_ℓ. Given the current index set $\mathcal{I}_\ell \subseteq \{1, \ldots, N_\ell\}$ and the distance $d(\cdot, \cdot)$, we define:

$$\phi^{(\ell)}(i) \;=\; \arg\min_{j \in \mathcal{I}_\ell \setminus \{i\}} d(\mathbf{x}_i, \mathbf{x}_j), \qquad i \in \mathcal{I}_\ell \tag{9}$$

$\phi^{(l)}(i)$ denotes the index of the nearest neighbor of x_i within layer ℓ. The corresponding in-degree counts are defined as:

$$\kappa^{(\ell)}(j) \;=\; \left|\{\, i \in \mathcal{I}_\ell :\; \phi^{(\ell)}(i) = j \,\}\right| \tag{10}$$

For each pair $(i, \phi^{(\ell)}(i))$, we apply the following mapping:

$$\psi^{(\ell)}(i) \;=\; \begin{cases} \phi^{(\ell)}(i), & \text{if } \kappa^{(\ell)}(\phi^{(\ell)}(i)) > \kappa^{(\ell)}(i), \\ i, & \text{if } \kappa^{(\ell)}(\phi^{(\ell)}(i)) < \kappa^{(\ell)}(i), \\ \min\{i, \phi^{(\ell)}(i)\}, & \text{if ties occur.} \end{cases} \tag{11}$$

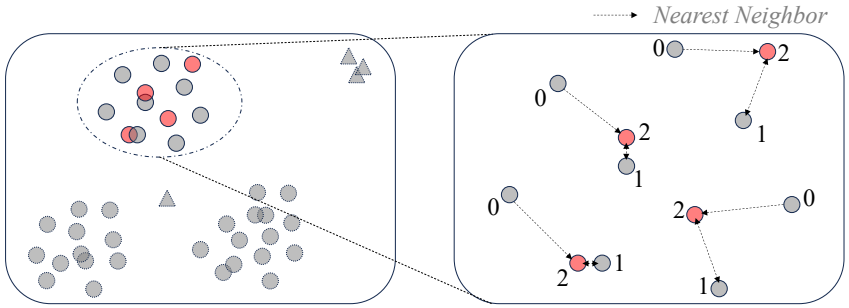


Fig. 6. Windows are grouped into local neighborhoods, and a representative is selected. Arrows indicate nearest-neighbor relationships, and labels denote the in-degree counts.

The mapping $\psi^{(\ell)}$ defines pairwise associations between samples. By repeatedly applying $\psi^{(\ell)}$, elements connected through these associations converge to the same representative index $p^{(\ell)}(i)$, which serves as the group label for i. The resulting equivalence classes form the following disjoint micro-groups:

$$C^{(\ell)} \;=\; \Big\{ C_u^{(\ell)} = \{\, i \in \mathcal{I}_\ell :\; p^{(\ell)}(i) = u \,\} \;:\; u \in p^{(\ell)}(\mathcal{I}_\ell) \Big\} \tag{12}$$

This construction is parameter-free (relying on nearest neighbor distance calculation), computationally scalable, and under the anomaly rarity assumption, forms groups that are predominantly normal, providing a stable foundation for conservative reference selection. **Algorithm 1** depicts the overall group building process.

Algorithm 1: BuildGroups$(W, \mathcal{I})$;
(self-excluding 1-NN, majority-favoring unions)

Input: Window matrix $W \in \mathbb{R}^{N\times m}$; current index set $\mathcal{I} \subseteq \{1, \ldots, N\}$
Output: Parent map $p : \mathcal{I} \to \mathcal{I}$ (global indices)

Compute self-excluding 1-NN indices $nn[i] \in \mathcal{I} \setminus \{i\}$ for all $i \in \mathcal{I}$ w.r.t. $d(\cdot,\cdot)$;
Compute in-degrees $c[j] \leftarrow |\{\, i \in \mathcal{I} : nn[i] = j \,\}|$ for all $j \in \mathcal{I}$;
for $i \in \mathcal{I}$ **do**
    $j \leftarrow nn[i]$;
    if $c[j] > c[i]$ **then** $p[i] \leftarrow j$;
    else if $c[j] < c[i]$ **then** $p[i] \leftarrow i$;
    else $p[i] \leftarrow \min\{i, j\}$;
update group labels until convergence (i.e., $p[i] \leftarrow p[p[i]]$);
return p;

4.3 Representative Selection

We initialize the hierarchy at level 0 with the full set of windows: the index set $\mathcal{I}_0$ contains all windows and the representative set equals it, i.e., $\mathcal{R}^{(0)} = \mathcal{I}_0$. From each group $C_u^{(\ell)}$ introduced in the previous section, we select exactly one representative to form the representative set $\mathcal{R}^{(\ell+1)}$, formally defined as follows:

$$\mathcal{R}^{(\ell+1)} \;=\; \big\{\, r_u^{(\ell+1)} \in C_u^{(\ell)} \;:\; u \in p^{(\ell)}(\mathcal{I}_\ell) \,\big\}, \qquad \mathcal{I}_{\ell+1} \;=\; \mathcal{R}^{(\ell+1)} \tag{13}$$

The one-per-group rule retains coverage while limiting redundancy so that $|\mathcal{I}_{\ell+1}| \le \frac{|\mathcal{I}_\ell|}{2}$. Under the majority-consistency assumption, anomalous representatives become non-increasing across

levels; early levels preserve fine structure (i.e., many reps), whereas deeper levels retain only persistent, widely supported patterns (i.e., few reps). This stratification exposes heterogeneous anomaly evidence across resolutions. The selection process is illustrated in Figure 6 and detailed in **Algorithm 2**.

Algorithm 2: SelectRepsAndScore$(W, \mathcal{I})$;
(one-per-group reps & reference-driven scoring)

Input: All windows $W \in \mathbb{R}^{N\times m}$; current index set $\mathcal{I}$
Output: Rep set $\mathcal{R} \subseteq \mathcal{I}$; window scores $q \in \mathbb{R}^N$; series score $y \in \mathbb{R}^T$

$p \leftarrow$ BuildGroups$(W, \mathcal{I})$ `// c.f. Alg. 1 ;`
Let $\mathcal{U} \leftarrow \{u : u = p[u] \text{ and } u \in \mathcal{I}\}$ be the set of roots;
$\mathcal{R} \leftarrow \{r_u : r_u \in C_u = \{i \in \mathcal{I} : p[i] = u\},\ \forall u \in \mathcal{U}\}$ `// e.g., choose` $r_u = u$;
for $i \in \{1, \ldots, N\}$ **do**
  $q[i] \leftarrow \min_{r\in\mathcal{R}} \|W[i] - W[r]\|_2$ `// reference-driven distance`
return $(\mathcal{R}, q)$;

4.4 *Per-level* Score Computation

Given the level-ℓ representatives $\mathcal{R}^{(\ell)}$, each window $\mathbf{x}_i$ receives a reference-driven distance score $q_i^{(\ell)}$, formally computed as follows:

$$q_i^{(\ell)} = \min_{r\in\mathcal{R}^{(\ell)}} d(\mathbf{x}_i, \mathbf{x}_r), \qquad i = 1, \ldots, N. \tag{14}$$

Compared with all-pairs or global-prototype scoring, reference-based scores (i) avoid nearest-neighbor suppression among near-duplicate anomalies—windows compete against a stable reference set—and (ii) reduce sensitivity to misspecified cluster numbers or centroid smearing, since references arise from conservative grouping rather than global prototype fitting. The per-level scoring computation is incorporated in **Algorithm 2**.

Beyond the reference-driven mode, we introduce a complementary setting: We continue until a single representative remains ($|\mathcal{R}^{(L)}| = 1$) and then combine $\{q^{(\ell)}\}_{\ell=0}^{L}$ across levels (i.e., score aggregation, further described in the following section). In practice, early levels highlight fine-grained discords, middle levels surface small, localized clusters of deviations, and the deepest levels—retaining only a few representatives—approximate a global baseline, which is effective for detecting pronounced departures.

A key design choice is to compute distances on *unnormalized* windows. Normalizing can suppress amplitude-driven anomalies and thus weaken sensitivity to spikes; conversely, it can help in explicit mean-shift scenarios. Our hierarchy reduces reliance on such preprocessing: in HYDRA, early levels retain many locally sampled representatives, so each query window is compared primarily against representatives drawn from similar local mean regions (as determined by the 1-NN and in-degree–based grouping). As a simple example, consider a series with two normal regimes that differ mainly by a constant offset: global comparisons may over-emphasize the offset, whereas HYDRA's local grouping makes the offset largely shared within each micro-group, thereby reducing the need for explicit per-window normalization. This intuition is also supported by the empirical results in Section 5.4. The full-hierarchy mode is typically more robust across heterogeneous anomaly types, at the cost of additional computation. We next introduce the aggregation mechanism.

4.5 Score Aggregation

Having constructed the hierarchy and obtained per-level window scores, we summarize each *window across levels*—rather than aggregating only at the final timeline—to preserve evidence

visible at any resolution. Let $q_i^{(\ell)}$ denote the score of window i at level $\ell = 0, \ldots, L$, with windows aligned across levels to a common index $i = 1, \ldots, N$. To balance scale disparities between levels, we standardize scores *within* each level:

$$\tilde{q}_i^{(\ell)} = \frac{q_i^{(\ell)} - \mu_\ell}{\sigma_\ell + \delta}, \qquad \mu_\ell = \frac{1}{N}\sum_{i=1}^{N} q_i^{(\ell)}, \quad \sigma_\ell^2 = \frac{1}{N}\sum_{i=1}^{N}\left(q_i^{(\ell)} - \mu_\ell\right)^2 \tag{15}$$

We introduce a small $\delta > 0$ for numerical stability (levels with $\sigma_\ell \approx 0$ contribute nearly neutrally). We then aggregate *across levels* by applying a maximum selection per window and per level:

$$s_i^\star = \max_{\ell=0,\ldots,L} \tilde{q}_i^{(\ell)} \tag{16}$$

Thus, a window that is strongly anomalous at *any* level is retained. This mitigates dominance by high-amplitude patterns, improves sensitivity to heterogeneous anomaly types, and preserves per-level interpretability. Finally, we map the fused window scores $\{s_i^\star\}_{i=1}^N$ back to the timeline using the averaging rule $S(t)$, yielding the final anomaly curve. The overall computational steps of HYDRA are summarized in **Algorithm 3**. Taken together, this design transforms normalization from a brittle data preprocessing choice into a structurally emergent property of its hierarchy. By combining raw-scale distances, level-wise score normalization, and hierarchical reference consensus, HYDRA achieves normalization-free robustness across both amplitude- and shape-driven anomalies.

Algorithm 3: HYDRA;
(training-free, parameter-light)

Input: Series X; window size m; stride s; target K; flag `includeL0`
Output: Final series score $z \in \mathbb{R}^T$

$W \leftarrow$ sliding windows of X with (m, s); $N \leftarrow$ #rows of W; $\mathcal{I}_0 \leftarrow \{1, \ldots, N\}$;
Initialize lists $\mathcal{Q} \leftarrow [\,]$ `// per-level window-score vectors`
if `includeL0` **then**
  Compute $q^{(0)}[i] \leftarrow$ 1-NN distance in W for $i = 1..N$; `// In practice, it is done during Algorithm 1`
  append $q^{(0)}$ to $\mathcal{Q}$;
$\ell \leftarrow 0$;
while $|\mathcal{I}_\ell| > K$ **do**
  $(\mathcal{R}_{\ell+1}, q^{(\ell+1)}) \leftarrow$ SelectRepsAndScore$(W, \mathcal{I}_\ell)$;
  Append $q^{(\ell+1)}$ to $\mathcal{Q}$;
  $\mathcal{I}_{\ell+1} \leftarrow \mathcal{R}_{\ell+1}$; $\ell \leftarrow \ell + 1$;
Level-wise normalization & aggregation;
Stack $\mathcal{Q}$ into matrix $Q \in \mathbb{R}^{L \times N}$ (rows = levels);
for $\ell = 1..L$ **do**
  if $\sigma_\ell > 0$ **then** $Q_{\ell i} \leftarrow (Q_{\ell i} - \mu_\ell)/\sigma_\ell$ for all i;
  else $Q_{\ell i} \leftarrow 0$ for all i `// levels with` $\sigma_\ell = 0$;
$s_i^\star \leftarrow \max_{\ell=1..L} Q_{\ell i}$ for $i = 1..N$ `// Aggregate levels`
$z \leftarrow$ AvgOverlapMap$(s^\star)$ `// fused window scores`
return z;

4.6 HYDRA at Scale: *An Approximate Solution*

As introduced in **Section** 4.2, our detector uses the 1-NN distance as anomaly score. The exact version, **HYDRA_exact**, computes pairwise dissimilarities among n windows of length m, requiring $O(n^2m)$ time at the first level. Although one could precompute an $n \times n$ distance matrix to reuse across hierarchy levels, this quickly becomes infeasible for long sequences, as storing $O(n^2)$ entries exceeds memory limits. Hence, distances are recomputed at each level, since the total work across levels differs from the first level only by a constant factor, the overall complexity remains $O(n^2m)$.

However, the key intuition lies in the representative-based hierarchy: at each level, adjacent or highly similar windows are grouped together, and only one representative is lifted to the next level. As shown in **Section** 3.2, this design gradually filters out anomalous windows. As anomalies are rare, they become minorities within mixed groups when the resolution coarsens. Therefore, the selection rule remains valid even when the neighborhood structure is only approximate—the anomalous representatives converge faster, while the normal representatives persist.

Based on this observation, we replace the exact KNN computation with *approximate nearest neighbor search (ANNS)*, such as hierarchical navigable small-world (HNSW) [67, 80, 110], yielding the variant **HYDRA_approx.** In our implementation, we index the current representatives with HNSW and reuse the same index for both selection and scoring queries, reducing the end-to-end cost from $O(n^2m)$ to $O(n \log n \cdot m)$ and yielding clear wall-clock speedups. We show in Section 5.5.6 that HYDRA_approx matches the accuracy of HYDRA_exact while being significantly faster on long time series. In this paper, we primarily use HYDRA_approx, as it achieves scalability on long time series while maintaining comparable accuracy. Unless otherwise specified, all instances of HYDRA in our experiments refer to HYDRA_approx.

4.7 HYDRA for Multivariate Time Series

In this section, we introduce a strategy that enables our algorithm to be easily applied to a multivariate scenario. For a C-channel time series $Y \in \mathbb{R}^{C \times T}$ and a window length m, we first *standardize each channel globally* using z-score normalization to ensure cross-channel comparability, denoted as $\tilde{Y}$. Importantly, as we discussed, to preserve the window *magnitude/energy*, we do **not** normalize at the window level. Given two windows $X_i, X_j \in \mathbb{R}^{C \times m}$ extracted from $\tilde{Y}$, we compute the multivariate distance as follows:

$$d(X_i, X_j) \;=\; \sqrt{\sum_{c=1}^{C} \sum_{\tau=1}^{m} \left(X_i^{c,\tau} - X_j^{c,\tau}\right)^2} \tag{17}$$

The multivariate distance treats all channels equally after global standardization and preserves absolute within-channel magnitudes across windows. Although our multivariate extension is intentionally simple and fully unsupervised, richer alternatives typically demand practitioner supervision or task-specific preprocessing (e.g., channel weighting, feature fusion, or selection). We defer the design of adapted multivariate distance measures and channel-aware normalization to future work. Nevertheless, experimental results show that even this naive extension already outperforms strong existing baselines, suggesting that the hierarchical design of HYDRA remains effective beyond the univariate setting.

5 Experimental Evaluation

We conduct a comprehensive set of experiments to evaluate the effectiveness, robustness, and scalability of our proposed approach. The evaluations are organized as follows.

Section 5.1 describes the experimental setup, including datasets, baselines, and evaluation metrics. **Section 5.2** reports the overall accuracy results and statistical significance analysis. **Section 5.3** examines the performance across different anomaly domains, revealing how the method adapts to

diverse anomaly types. **Section 5.4** investigates the impact of normalization strategies on detection stability and magnitude sensitivity. **Section 5.5** studies the effects of key parameters, including subsequence length, hierarchical depth, and approximation strategy. Finally, **Section 5.6** evaluates the method's runtime scalability on large datasets.

Together, these experiments provide a holistic understanding of the framework's performance, demonstrating its generality and efficiency under diverse settings, as well as its robustness to different anomaly characteristics and parameter configurations.

5.1 Experimental Setup

5.1.1 Platform. We conduct our experiments on a server with the following configuration: 2xAMD EPYC 7713 64-Core. The server has two Nvidia A100 GPUs and runs Ubuntu 22.04.3 LTS (64-bit). We implemented HYDRA and baselines in Python 3.11 with the main dependencies including: Pytorch 2.3.0 [94] and scikit-learn 1.3.2 [95]. We open-source the implementation [8].

5.1.2 Datasets. The limitations in the quality and construction of existing time-series anomaly detection benchmarks, particularly issues such as mislabeling, dataset bias, and lack of practical feasibility, have substantially impeded progress in evaluation and benchmarking [63, 83, 111]. To ensure reliable assessment, we evaluate the effectiveness of HYDRA on the recently released, heterogeneous, and manually curated TSB-AD benchmark [63]. TSB-AD consists of 870 univariate (TSB-AD-U) and 200 multivariate (TSB-AD-M) time series spanning nine diverse domains, including web services [5, 111], medical [33, 35], facility monitoring [29, 68], synthetic [50, 51], human activity [9, 98], sensor [6, 43], environmental [1, 111], finance [104, 109], and traffic [5]. All baselines are implemented following the official TSB-AD protocol, with their parameters tuned on the tuning set and evaluated on the corresponding test set to ensure fair comparison and reproducibility.

5.1.3 Baselines. We evaluate **HYDRA**[2] on both univariate and multivariate time series. For comparison, we consider the top half of the best-performing methods reported in **TSB-AD**, spanning classical/statistical, neural, and foundation-model baselines. Univariate baselines: KMeansAD [114], Spectral Residual (SR) [97], NormA [13], Sub-KNN [96], Matrix Profile (MP/STOMP) [119], Series2Graph [15], POLY [53], Sub-PCA [2], USAD [7], CNN [70], LSTM-AD [66], MOMENT (fine-tuned; MOMENT-FT) [34], and TimesFM [25]. Multivariate baselines: KMeansAD [114], AutoEncoder [101], FITS [113], KNN [96], PCA [2], USAD [7], CNN [70], LSTM-AD [66], Donut [112], OCSVM [103], MCD [100], CBLOF [40], and OmniAnomaly [106]. In addition, motivated by recent progress in contrastive learning, several studies inject synthetic anomalies to impose training constraints and to help models learn a boundary between normal and abnormal patterns [117, 120]. We integrate representative injection-based methods within our evaluation pipeline and report results for the three strongest performers in our setting, namely CTAD [47], TFAD [122], and CARLA [24]. Unless otherwise stated, we use the TSB-AD implementations and the same preprocessing, windowing, and evaluation protocol to ensure a fair comparison. Additionally, we do not apply normalization to any subsequence-based method for fairness, except Series2Graph, which is inherently designed based on correlation. Notably, foregoing normalization can yield better results, as shown in Section 5.4. To balance fairness with brevity, all following experiments are conducted on the top-five methods identified in Table 2 univariate benchmarks. This selection preserves competitive coverage while keeping the evaluation compact and interpretable.

5.1.4 Multivariate extension. We apply *exactly the same* preprocessing and distance as **Section** 4.7 described to all multivariate baselines: (i) **Sub-KNN**: the k-NN search uses the above $d(\cdot, \cdot)$ between

[2]Unless stated otherwise, all accuracy results report HYDRA_approx; HYDRA_exact is used only in the exact-vs-approx comparison (Fig. 12).

Table 2. Overall results. Left: Univariate. Right: Multivariate. Best per column in **gold** and **bold**, second best in **silver**.

	TSB-AD-U (univariate)						**TSB-AD-M** (multivariate)				
	AUC-PR	AUC-ROC	VUS-PR	VUS-ROC	F1		AUC-PR	AUC-ROC	VUS-PR	VUS-ROC	F1
CARLA	0.2443	0.6753	0.2891	0.7075	0.3017	FITS	0.1539	0.5831	0.2150	0.6591	0.2159
TimesFM	0.2844	0.6741	0.3003	0.7435	0.3356	CARLA	0.1952	0.6056	0.2469	0.6523	0.2534
SR	0.3157	0.7420	0.3238	0.8115	0.3800	Donut	0.2041	0.6438	0.2602	0.7110	0.2802
LSTMAD	0.3147	0.6757	0.3250	0.7560	0.3682	OCSVM	0.2274	0.6090	0.2648	0.6703	0.2793
CNN	0.3301	0.7127	0.3429	0.7872	0.3812	MCD	0.2709	0.6548	0.2711	0.6885	0.3255
TFAD	0.2458	0.6693	0.3462	0.7358	0.3057	CBLOF	0.2763	0.6697	0.2731	0.7026	0.3185
USAD	0.3231	0.6606	0.3612	0.7107	0.3662	AutoEncoder	0.2960	0.6698	0.2950	0.6937	0.3415
MOMENT	0.3010	0.6921	0.3857	0.7625	0.3526	USAD	0.2564	0.6367	0.3041	0.6832	0.3063
Series2Graph	0.3281	0.7574	0.3881	0.8046	0.3762	LSTMAD	0.3118	0.7030	0.3066	0.7370	0.3579
POLY	0.3067	0.7259	0.3898	0.7609	0.3686	OmniAnomaly	0.2662	0.6456	0.3122	0.6879	0.3204
Sub-PCA	0.3625	0.7965	0.4826	0.8400	0.4394	CNN	0.3167	0.7292	0.3130	0.7591	0.3731
NormA	0.4646	0.8127	0.4905	0.8305	0.4897	CTAD	0.3058	0.6752	0.3157	0.7083	0.3555
Sub-KNN	0.4445	0.8291	0.5416	0.8570	0.5103	KMeansAD	0.3015	0.7260	0.3441	0.7657	0.3529
MatrixProfile	0.4454	0.8378	0.5480	0.8693	0.5057	Sub-PCA	0.3100	0.7098	0.3858	0.7591	0.3878
KMeansAD	0.5033	0.8346	0.5576	0.8570	0.5376	Sub-KNN	0.3538	0.6904	0.4296	0.7393	0.4159
HYDRA_exact	0.6102	0.9086	0.6655	0.9276	0.6502	**HYDRA_exact**	0.4006	0.7636	0.4583	0.8002	0.4570
HYDRA	**0.6071**	**0.9102**	**0.6620** (+18.7%)	**0.9286**	**0.6466**	**HYDRA**	**0.4110**	**0.7829**	**0.4691**(+9.2%)	**0.8181**	**0.4636**

windows; (ii) **KMeans**: assignment and centroid updates operate on the same window representation and distance; (iii) **Sub-PCA**: each window is vectorized by concatenation ($\mathbb{R}^{C\times m} \to \mathbb{R}^{Cm}$) after the global per-channel standardization (with no window re-scaling), and PCA is performed on these vectors. This ensures a fair, consistent multivariate treatment across methods.

5.1.5 Statistical Validation. To validate the statistical difference of performance among *multiple* anomaly detection models across *multiple* datasets, we apply the Friedman test [30], followed by the post-hoc Nemenyi test [71] at a 95% confidence level. In the Critical Difference (CD) diagram, methods that do not exhibit statistical differences are connected by black lines.

5.1.6 Accuracy Evaluation Measures. We report results on **AUC-PR**, **AUC-ROC**, **VUS-PR**, **VUS-ROC**, **Standard-F1 (F1)**. Following TSB-AD's recommendation, we *focus on VUS-PR* [12, 78] as the primary metric due to its threshold-independence and robustness to misalignment and class imbalance, while the other threshold-based or point-adjusted metrics may suffer from various biases. Formally, let $\mathcal{B}$ be the set of buffer lengths used to extend anomaly ranges; for each $b \in \mathcal{B}$, compute the range-aware PR curve and its area AP_b. Then VUS-PR $= \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} AP_b$, i.e., the volume under the PR surface obtained by varying both the threshold and the buffer length. Thus, VUS-PR is threshold-independent and range-aware, averaging performance across tolerance settings to remain robust to temporal misalignment and severe class imbalance.

5.2 Overall Accuracy Evaluation

5.2.1 Accuracy Evaluation. Table 2 reports results on **TSB-AD-U** (univariate) and **TSB-AD-M** (multivariate). Across both settings, **HYDRA** attains the best average accuracy across all measures-*AUC-PR*, *AUC-ROC*, *VUS-PR*, *VUS-ROC*, and *Standard-F1*—with the top score in each column highlighted in gold and the second best in silver. On the primary metric *VUS-PR*, HYDRA reaches **0.6620** on TSB-AD-U, improving over the strongest baseline (KMeansAD, 0.5576) by **+18.7%**. On TSB-AD-M, HYDRA achieves **0.4691**, exceeding the best baseline (Sub-KNN, 0.4296) by **+9.2%**. Among classical subsequence baselines on the univariate suite, KMeansAD is the strongest overall; MatrixProfile ranks well on ROC-type metrics but lags on *VUS-PR*. In the multivariate suite, CNN often delivers the strongest *AUC-ROC* among baselines, whereas Sub-KNN tends to lead on *VUS-PR*/*Standard-F1*. While Sub-KNN attains competitive accuracy in the multivariate case, it also tends to introduce a higher rate of false positives, which is undesirable for practical deployment (i.e., higher alert

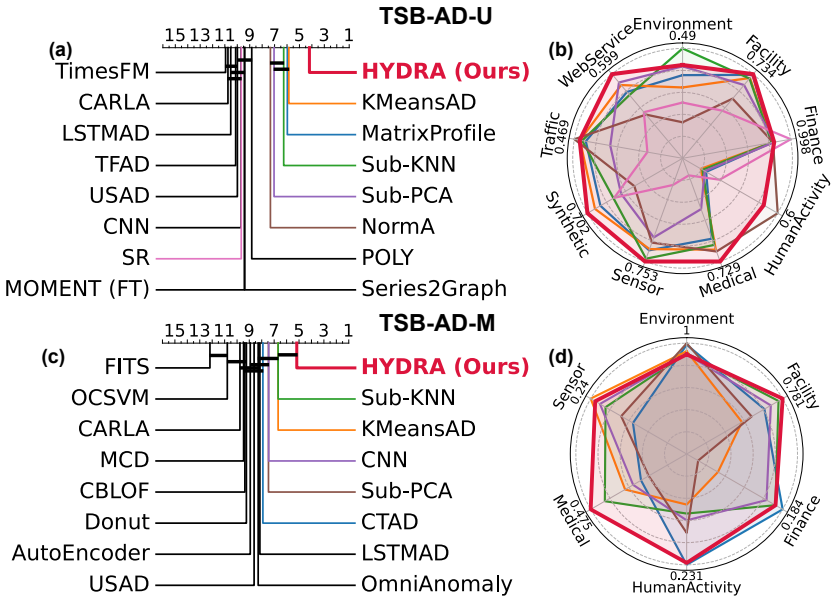


Fig. 7. (a)(c) Critical Difference Diagrams. (b)(d) Domain-wise results on TSB-AD with per-domain max-scaling—each axis is normalized to that domain's best.

burden for operators). Overall, HYDRA delivers the most balanced performance across point-based and range-based metrics while maintaining clear margins on *VUS-PR* in both the univariate and multivariate regimes, making it a more reliable choice when both accuracy and stability matter.

5.2.2 Statistical Significance Test. Figure 7(a) and 7(c) presents CD diagrams of average ranks on *VUS-PR* ($\alpha = 0.05$; Friedman test with Nemenyi post-hoc). On **TSB-AD-U**, HYDRA attains the best average rank and its bar is separated from all baselines by more than the CD, indicating a *statistically significant* improvement. On **TSB-AD-M**, we have 2× fewer datasets; hence, the power to distinguish significance is smaller given the same number of methods. However, HYDRA still achieves the best average rank, but the gaps to the strongest baselines do *not* exceed the CD; consequently, the advantage is *not statistically significant* at $\alpha = 0.05$ (but significant at $\alpha = 0.1$). Overall, the CD diagrams corroborate Table 2: HYDRA consistently ranks first, with statistical significance at $\alpha = 0.05$ on the univariate suite and at $\alpha = 0.1$ on the multivariate suite.

5.3 Evaluation Across Domains

To further examine the generality of our method, we adopt the domain categorization provided in the TSB-AD benchmark and evaluate models on each domain separately. Beyond the top-five algorithms identified in Section 5.2, we also include SR, a high-performance method tailored to point-based anomalies only, to provide a fuller comparison within each domain. As shown in Figure 7(b), NormA performs particularly well on domains dominated by collective anomalies, while MP achieves stronger results on datasets characterized by a single anomaly. In contrast, HYDRA maintains consistently high performance across all domains, demonstrating its robustness to different anomaly types and domain characteristics. Although certain domains still favor specific approaches (e.g., NormA in Medical data), the overall balance between multiple collective and single anomaly detection highlights the adaptability of our framework. Notably, in the Human Activity domain, there exist anomalies that deviate significantly from the global mean yet share highly similar temporal shapes with each other. Our method achieves substantially better performance than other distance-based baselines in this domain, which highlights the advantage of our hierarchical

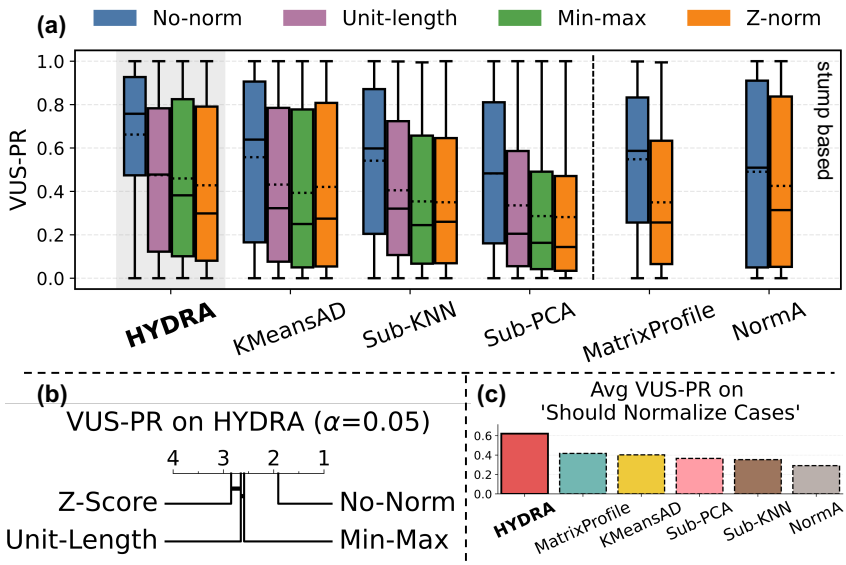


Fig. 8. (a) Effect of normalization across different methods. (b) Critical Diagram of HYDRA on different normalizations. (c) Average VUS-PR of non-normalized variants on a filtered TSB-AD-U subset (should normalize cases), where No-norm typically fails while Z-norm typically succeeds.

structure in capturing diverse types of anomalies—from localized deviations to structurally similar but globally shifted patterns. These results indicate that our framework effectively integrates local and global views, leading to superior adaptability across heterogeneous real-world datasets.

5.4 Normalization Impact

Figure 8(b) presents a comprehensive comparison of normalization effects on detection performance. The CD diagram summarizes the pairwise rankings of four normalization policies applied to HYDRA, showing that **No-norm** significantly outperforms Z-score, Unit-length, and Min–Max normalization at $\alpha = 0.05$. These results indicate that per-window normalization consistently weakens HYDRA's ability to preserve discriminative anomaly evidence, leading to a measurable loss in detection performance.

Figure 8(a) extends this comparison across top-5 sequence-based baselines, including KMeansAD, MP, Sub-KNN, NormA, and Sub-PCA. A consistent trend emerges: the **non-normalized** variant achieves the highest median VUS-PR across all methods, while Z-score and Min-Max normalization typically degrade performance by suppressing amplitude information. Unit-length normalization (which preserves more magnitude information; see Figure 3) behaves similarly to NoNorm on a few datasets but is less stable overall. Taken together, these results demonstrate that normalization will greatly influence detectability and that distance-based detectors—including HYDRA—are generally more reliable without per-window normalization.

Although skipping normalization is generally beneficial, some time series still gain from it. Our proposed HYDRA, however, mitigates the penalty of choosing an ill-suited normalization variant. To make this concrete, we construct a *normalization-sensitive* subset by filtering instances on which multiple baselines fail without normalization (VUS-PR < 0.3) but succeed with Z-score normalization (VUS-PR > 0.7). This subset isolates cases where the normalization decision materially changes outcomes. Figure 8(c) reports, for each method, the average VUS-PR of its *non-normalized* variant on these filtered cases. HYDRA attains the highest accuracy by a clear margin, indicating robustness

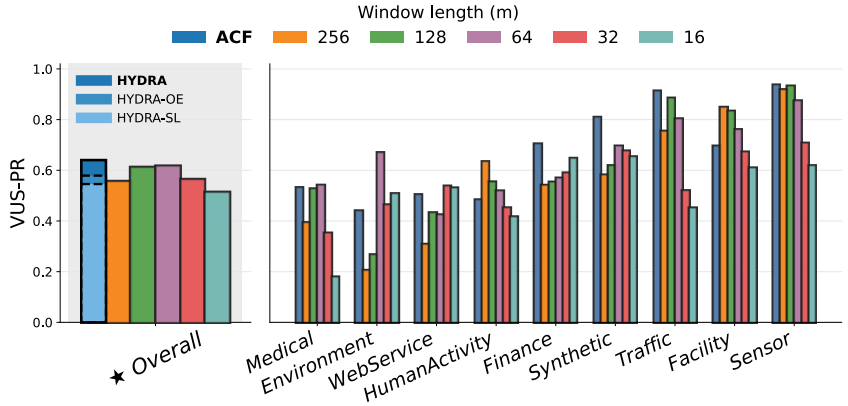


Fig. 9. Comparison of fixed window sizes and ACF-adaptive windows across domains. The Overall bar (ACF setting) compares HYDRA-SL (no aggregation), HYDRA-OE (score ensembling), and the proposed HYDRA.

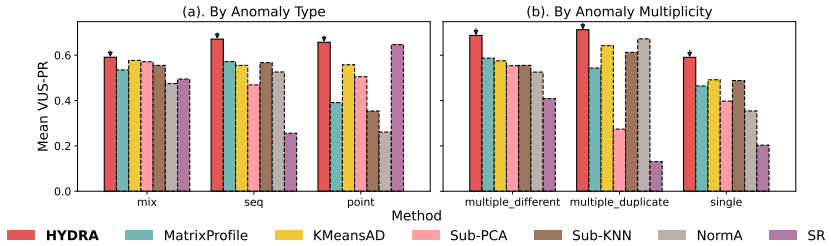


Fig. 10. Mean VUS-PR by anomaly type. We group datasets by (a) anomaly type and (b) anomaly multiplicity, and compare HYDRA with representative baselines.

to normalization choice: even without per-window normalization, it detects many anomalies that competing methods recover only after normalization.

5.5 Ablation Study

We analyze the effect of key parameters—including subsequence length, hierarchy depth, and other hyperparameters—on model performance. The results show that deeper hierarchies consistently improve detection, while the optimal subsequence length can be determined automatically, reducing the need for manual tuning.

5.5.1 Subsequence length. To investigate the effect of the window size (i.e., subsequence length) m, we evaluate our model using different fixed window sizes and compare them with the ACF-based adaptive window selection. While fixed-window methods require manually setting m, the ACF strategy (i.e., Auto-Correlation Function) automatically determines the window length according to the temporal correlation structure of each dataset, making our method require no manual hyperparameter tuning in this aspect.

As shown in Figure 9, the ACF-based version achieves the highest overall VUS-PR score and maintains relatively stable performance across all domains. Although some fixed window settings (e.g., m=64) yield competitive results in certain domains, their performance fluctuates considerably across datasets. In contrast, the ACF configuration provides a better trade-off between adaptability and robustness. This result suggests that automatic window selection not only avoids manual tuning, but also improves consistency across diverse application scenarios.

5.5.2 Aggregation. As shown in Figure 9, the leftmost bars under the ACF-adaptive setting compare three variants of HYDRA (i) we tuned a best-performing single-level detector, where anomaly

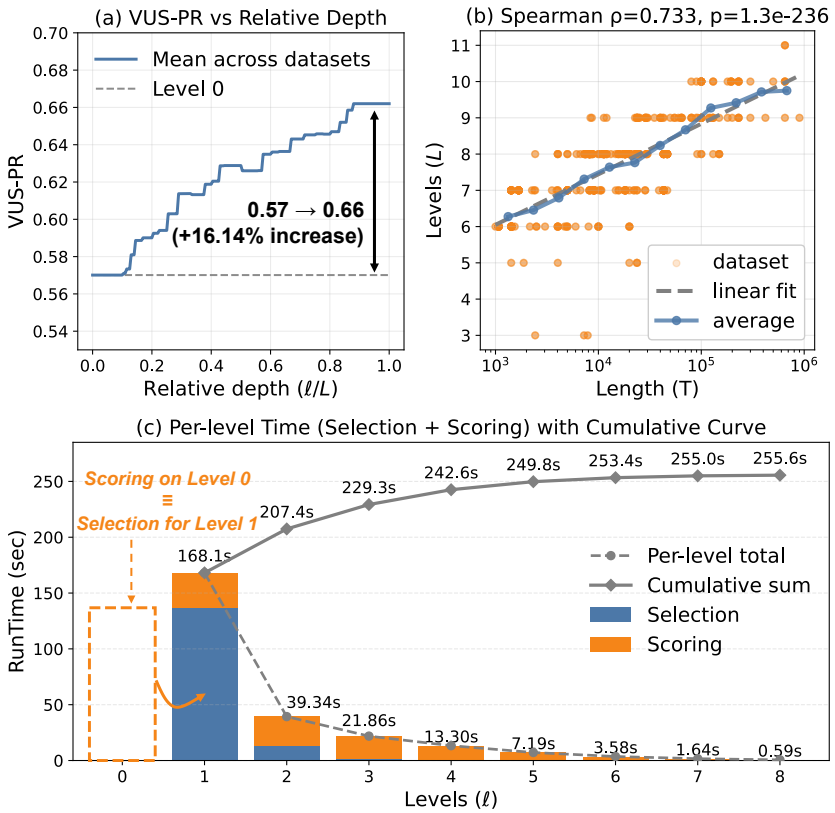


Fig. 11. Hierarchy depth vs. (a) accuracy and (b) time series length. (c) Per-level and cumulative runtime timeline (L1–L8) on a synthetic 1M-length time series. All these experiments are based on HYDRA_approx.

scores were computed from an empirically selected level determined by iterative grouping until the number of representative windows fell below 1% of the total, denoted as HYDRA-SL; (ii) we adopted the optimal ensemble strategy proposed in TSB-AutoAD [62] (i.e., Outlier Ensemble [4]) to aggregate scores across levels, denoted as HYDRA-OE; (iii) We applied the aggregation mechanism introduced in this work. Obviously, multi-level aggregation shows better results than single-level design, and our proposed mechanism further improves over the previous best method.

5.5.3 Anomaly Types. To better understand performance under heterogeneous anomalies, we partition the benchmark datasets along two complementary aspects. First, we group datasets by anomaly type into point, subsequence/pattern anomalies (seq), and mix anomalies. Second, we group datasets by anomaly multiplicity into multiple distinct anomaly patterns (multiple_different), repeated/near-duplicate anomalies (multiple_duplicate), and a single dominant anomalous event (single). Figure 10 summarizes the mean VUS-PR within each group. Across both partitionings, HYDRA consistently achieves the highest mean VUS-PR, indicating robust performance across diverse anomaly characteristics. In addition, Figure 14a presents representative successful cases of HYDRA across different anomaly types, illustrating its robustness under a unified framework. As shown, HYDRA can consistently detect heterogeneous anomalies, ranging from isolated global outliers and repeated patterns to trend shifts, periodicity changes, and contextual anomalies, highlighting the benefit of aggregating anomaly evidence across hierarchical resolutions.

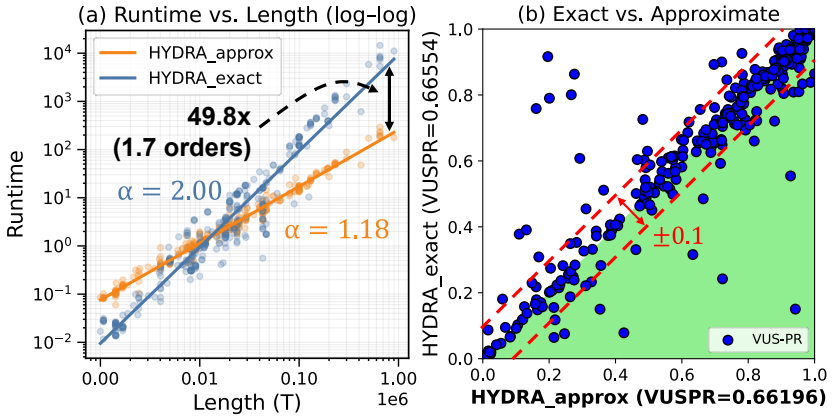


Fig. 12. HYDRA_exact vs. HYDRA_approx for (a) runtime (seconds) and (b) accuracy (VUS-PR).

5.5.4 One versus multiple levels. To examine how performance evolves with increasing levels, we progressively aggregate windows starting with level 0, then levels (0,1), (0,1,2), and so forth until maximum depth L. Because datasets vary in their absolute number of levels, depth is normalized to $[0, 1]$. As an example, for a dataset with three levels, the normalized ranges $[0, 0.33)$, $[0.33, 0.66)$, and $[0.66, 1]$ correspond to levels 0, 1, and 2, respectively.

As shown in Figure 11(a), the x-axis denotes normalized depth and the y-axis reports VUS-PR. The solid blue curve gives the mean performance across datasets, while the dashed red line marks the level-0 baseline. We observe a clear upward trend: as relative depth increases, aggregated accuracy converges to higher values. This demonstrates that multi-level aggregation consistently improves performance, with stronger gains as deeper levels are included.

Complementing this, Figure 11(c) shows that later levels are *lightweight* in computational cost, consistent with our runtime analysis in **Section** 4.6. Taken together, these two views indicate that our framework leverages deeper levels that are both (i) increasingly informative for detection and (ii) inexpensive to compute. Hence, the proposed hierarchical aggregation is well-justified: it captures complementary temporal structures across scales, yielding robust accuracy improvements at a modest marginal cost per level.

5.5.5 Length versus Hierarchy Depth. The hierarchy depth L is an *emergent* property automatically determined by data characteristics rather than a preset hyperparameter. Theoretically, L grows approximately linearly with $\log n$ (where $n = T - m + 1$), and is at most $log_2 n$ in the worst case, implying that longer sequences naturally yield deeper hierarchies. Empirically, Figure 11(b) confirms this log–linear trend: sequence length shows the strongest correlation with hierarchy depth ($\rho = 0.733$), and the smoothed curve reveals a clear monotonic rise with occasional local plateaus. These plateaus often correspond to datasets with periodic or repetitive structures, where aggregation saturates early because recurring patterns are already consolidated at shallow levels. This suggests that hierarchy depth is influenced not only by sequence length, but also by the amount of new structure that remains to be integrated.

5.5.6 Exact versus Approximate. In **Section** 4.6 we introduce an approximate version of the proposed HYDRA. Now we compare the accuracy and runtime between HYDRA_exact and HYDRA_approx, here, HYDRA_approx builds indices based on HNSW with default hyperparameters. As demonstrated in Figure 12, this approximate variant achieves up to 50× faster runtime (approximately 1.7 orders of magnitude faster) on 1-million-length time series, while maintaining nearly identical (or even slightly superior) detection quality in VUS-PR performance. Additionally, the

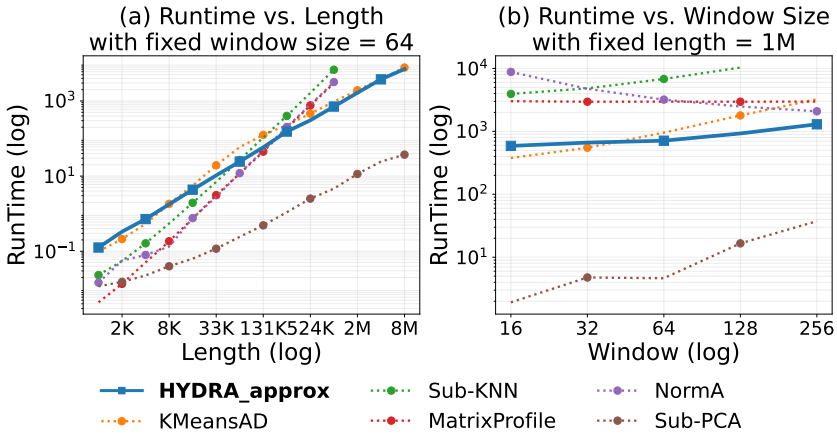


Fig. 13. Runtime (seconds) vs. (a) time series length at a fixed window size, (b) window size at a fixed time series length. Both panels use log–log axes.

fitted log–log slope for HYDRA_approx is close to 1, indicating nearly linear scaling with length. This is consistent with our assumption in **Section** 4.6, and confirms that the hierarchical framework is inherently robust to approximate neighborhoods and scales efficiently to large datasets. We also note that on short series, HYDRA_approx can be slower because HNSW index construction overhead can dominate runtime.

5.6 Scalability Evaluation

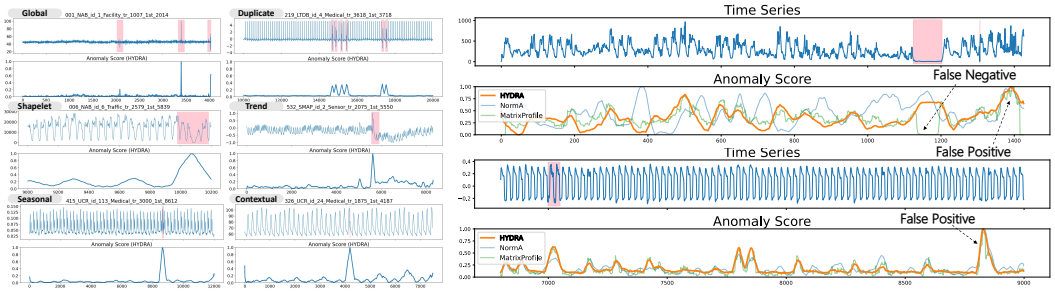
To ensure fair and reproducible comparisons, all methods run single-threaded by forcing numerical backends to use one thread.[3] For stump-based approaches (MP and NormA), we perform a one-shot warm-up prior to timing to amortize JIT/cache initialization. A 4-hour wall-clock limit is set per configuration; runs beyond it are marked failed. To characterize scaling with data size, we vary the time-series length T from $\mathbf{2^{10}}$ to $\mathbf{2^{23}}$ and set the window size to $m \in \{16, 32, 64, 128, 256\}$. Figure 13 shows two representative slices under identical threading constraints at $T = 1\text{M}$ and $m = 64$.

As shown in Figure 13(a), on short sequences, HYDRA_approx may appear slower than brute-force KNN due to the index construction and conservative ANN parameters chosen for robustness on small inputs. As the sequence length increases, HYDRA_approx amortizes its indexing cost and becomes faster, outperforming MP and approaching KMeansAD. Following that, Figure 13(b) evaluates the runtime for both HYDRA variants, which increases with window size, reflecting the higher per-query cost at larger m. Among baselines, MP shows stable scaling due to its FFT-based formulation, while NormA slightly benefits from fewer samples at longer windows. Across all methods, **HYDRA** maintains competitive efficiency, ranking second only to Sub-PCA, and demonstrates strong scalability across increasing series lengths.

5.7 Limitations and Future Work

Figure 14b presents two representative failure cases of HYDRA, alongside SOTA baselines, NormA and MP. In the upper panel of Figure 14b, the normal behavior exhibits substantial heterogeneity, while the anomalous pattern remains relatively flat. Under this setting, HYDRA's representative window screening may fail to fully purify normal prototypes, leading to inflated pairwise distances among normal sequences and, consequently, an increased number of false positives. Similar challenges arise for MP and NormA, whose underlying assumptions are likewise violated, resulting

[3]We set representative environment variables (e.g., `OMP_NUM_THREADS=1`, `MKL_NUM_THREADS=1`, and `OPENBLAS_NUM_THREADS=1`), with others configured analogously.



(a) HYDRA detects multiple anomaly types.

(b) Failure case analysis.

Fig. 14. Case studies of HYDRA on different anomaly patterns and failure scenarios.

in degraded separation between normal and anomalous score distributions. The lower panel of Figure 14b illustrates another failure mode, in which anomalies manifest as low-amplitude additive perturbations accompanied by high-frequency jitter. Such patterns induce only subtle discrepancies under Euclidean distance, yielding weak anomaly-score contrast across HYDRA, MP, and NormA. This observation highlights a shared limitation of time-domain, distance-based heuristics when confronted with spectrally localized or high-frequency anomalies.

Despite these observations, HYDRA's multi-resolution hierarchical design, grounded in simple yet effective time-domain heuristics, demonstrates strong potential as a general anomaly detection framework. A promising direction for future work is to extend HYDRA's multi-level integration mechanism to incorporate a broader family of anomaly heuristics. In particular, frequency-domain discrepancy measures could be naturally integrated to better capture oscillatory or jittering anomalies that are more salient in the spectral domain.

5.8 Key Takeaways

Extensive experiments on the curated TSB-AD benchmark demonstrate consistent, state-of-the-art accuracy: on univariate series, HYDRA improves VUS-PR by +18.7% over the strongest baseline; on multivariate series, it achieves a +9.2% gain. Ablation studies further show that (i) aggregating deeper levels monotonically increases accuracy at modest marginal cost, (ii) non-normalized window distances are generally more reliable than z-score and unit-length normalizations, and (iii) an ACF-based window selection provides stable performance across domains. These results suggest that hierarchical reference consensus is effective for subsequence-based detection.

6 Conclusion

In this paper, we introduced HYDRA, a multi-level and unsupervised approach for time-series anomaly detection that is robust to contamination, hyperparameter-light, and scalable to long sequences. By hierarchically selecting representative windows and aggregating anomaly scores across fine-to-coarse resolutions, HYDRA unifies the strengths of discord- and clustering-based paradigms while implicitly mitigating normalization bias. Unlike prior methods that rely on a single anomaly assumption or clean training data, HYDRA progressively refines normal references and detects both subtle discords and persistent deviations. Extensive experiments on 40 datasets from the TSB-AD benchmark show that HYDRA outperforms state-of-the-art methods, offering a practical and principled solution to detect anomalies in time series. Future work should extend the framework to non-Euclidean similarity measures and explore adaptive, uncertainty-aware methods for streaming deployment. Enhancing multivariate coupling and developing automatic segmentation for long-duration events also present key opportunities for improvement.

References

[1] Pacific Marine Environmental Laboratory (PMEL) [n. d.]. *TAO. https://www.pmel.noaa.gov/*. Pacific Marine Environmental Laboratory (PMEL). https://www.pmel.noaa.gov/

[2] Charu C. Aggarwal. 2017. *Outlier Analysis* (2 ed.). Springer International Publishing. doi:10.1007/978-3-319-47578-3

[3] Charu C Aggarwal and Charu C Aggarwal. 2017. *An introduction to outlier analysis*. Springer.

[4] Charu C. Aggarwal and Saket Sathe. 2015. Theoretical Foundations and Algorithms for Outlier Ensembles. *SIGKDD Explor. Newsl.* 17, 1 (Sept. 2015), 24–47. doi:10.1145/2830544.2830549

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

[6] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. 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*. 25–28.

[7] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 3395–3404.

[8] Anonymous Authors. 2025. HYDRA Github. https://github.com/thedatumorg/HYDRA.

[9] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster. 2009. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2009), 436–446.

[10] Mohini Bariya, Alexandra von Meier, John Paparrizos, and Michael J Franklin. 2021. k-ShapeStream: Probabilistic Streaming Clustering for Electric Grid Events. In *2021 IEEE Madrid PowerTech*. IEEE, 1–6.

[11] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A Review on outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–33.

[12] Paul Boniol, Ashwin K Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S Tsay, Aaron Elmore, Michael J Franklin, and John Paparrizos. 2025. VUS: effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal* 34, 3 (2025), 32.

[13] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* (2021), 1–23.

[14] Paul Boniol, Qinghua Liu, Mingyi Huang, Themis Palpanas, and John Paparrizos. 2024. Dive into time-series anomaly detection: A decade review. *arXiv preprint arXiv:2412.20512* (2024).

[15] Paul Boniol and Themis Palpanas. 2020. Series2Graph: Graph-based Subsequence Anomaly Detection for Time Series. *PVLDB* 13, 11 (2020).

[16] Paul Boniol, John Paparrizos, Yuhao Kang, Themis Palpanas, Ruey S Tsay, Aaron J Elmore, and Michael J Franklin. 2022. Theseus: navigating the labyrinth of time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3702–3705.

[17] Paul Boniol, John Paparrizos, and Themis Palpanas. 2023. New Trends in Time Series Anomaly Detection.. In *EDBT*. 847–850.

[18] Paul Boniol, John Paparrizos, and Themis Palpanas. 2024. An interactive dive into time-series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5382–5386.

[19] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND in action: subsequence anomaly detection for streams. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2867–2870.

[20] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND: streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1717–1729.

[21] Paul Boniol, Emmanouil Sylligardos, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2024. Adecimo: Model selection for time series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5441–5444.

[22] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. *ACM SIGMOD Record* 29, 2 (May 2000), 93–104. doi:10.1145/335191.335388

[23] S. Chauhan and L. Vig. 2015. Anomaly Detection in ECG Time Signals via Deep Long Short-Term Memory Networks. In *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*. 1–7. doi:10.1109/DSAA.2015.7344872

[24] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu C. Aggarwal, and Mahsa Salehi. 2025. CARLA: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition* 157 (2025), 110874. doi:10.1016/j.patcog.2024.110874

[25] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*.

[26] Jens E d'Hondt, Haojun Li, Fan Yang, Odysseas Papapetrou, and John Paparrizos. 2025. A Structured Study of Multivariate Time-Series Distance Measures. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–29.
[27] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. 2019. Band-limited training and inference for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 1745–1754.
[28] Jens E d'Hondt, Odysseas Papapetrou, and John Paparrizos. 2024. Beyond the Dimensions: A Structured Evaluation of Multivariate Time Series Distance Measures. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 107–112.
[29] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. 2016. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. arXiv:1612.06676 [cs.LG]
[30] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32, 200 (1937), 675–701.
[31] Apostolos Giannoulidis, Anastasios Gounaris, and John Paparrizos. 2025. BURST: Rendering Clustering Techniques Suitable for Evolving Streams. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4054–4063.
[32] Rahul Goel, Sandeep Soni, Naman Goyal, John Paparrizos, Hanna Wallach, Fernando Diaz, and Jacob Eisenstein. 2016. The social dynamics of language change in online networks. In *International conference on social informatics*. Springer, 41–57.
[33] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* 101, 23 (2000), e215–e220.
[34] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning*.
[35] Scott David Greenwald. 1990. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information.* Thesis. Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/29206 Accepted: 2005-10-07T20:45:22Z.
[36] Suchit Gupte and John Paparrizos. 2025. ShapX Engine: A Demonstration of Shapley Value Approximations. In *Companion of the 2025 International Conference on Management of Data*. 107–110.
[37] Suchit Gupte and John Paparrizos. 2025. Understanding the Black Box: A Deep Empirical Dive into Shapley Value Approximations for Tabular Data. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–31.
[38] Medina Hadjem, Farid Naït-Abdesselam, and Ashfaq A. Khokhar. 2016. ST-segment and T-wave anomalies prediction in an ECG data using RUSBoost. In *Healthcom*.
[39] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 32142–32159.
[40] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern recognition letters* 24, 9-10 (2003), 1641–1650.
[41] Kaisei Hishida, Chunwei Liu, John Paparrizos, and Aaron Elmore. 2025. Beyond Compression: A Comprehensive Evaluation of Lossless Floating-Point Compression. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4396–4409.
[42] Mingyi Huang, Qinghua Liu, Paul Boniol, and John Paparrizos. 2026. GlassboxAD: An Interactive System for Dissecting Hierarchical Time-Series Anomaly Detection. In *Companion of the International Conference on Management of Data (SIGMOD Companion '26)*. doi:10.1145/3788853.3801594
[43] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 387–395.
[44] Hoyoung Jeung, Sofiane Sarni, Ioannis Paparrizos, Saket Sathe, Karl Aberer, Nicholas Dawes, Thanasis G Papaioannou, and Michael Lehning. 2010. Effective metadata management in federated sensor networks. In *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. IEEE, 107–114.
[45] Hao Jiang, Chunwei Liu, Qi Jin, John Paparrizos, and Aaron J Elmore. 2020. Pids: attribute decomposition for improved compression and query performance in columnar storage. *Proceedings of the VLDB Endowment* 13, 6 (2020), 925–938.
[46] Hao Jiang, Chunwei Liu, John Paparrizos, Andrew A Chien, Jihong Ma, and Aaron J Elmore. 2021. Good to the Last Bit: Data-Driven Encoding with CodecDB. In *Proceedings of the 2021 International Conference on Management of Data*. 843–856.
[47] HyunGi Kim, Siwon Kim, Seonwoo Min, and Byunghan Lee. 2024. Contrastive Time-Series Anomaly Detection. *TKDE* 36, 10 (2024), 5053–5065. doi:10.1109/TKDE.2023.3335317
[48] Xiangjie Kong, Zhenghao Chen, Weiyao Liu, Kaili Ning, Lechao Zhang, Syauqie Muhammad Marier, Yichen Liu, Yuhao Chen, and Feng Xia. 2025. Deep learning for time series forecasting: a survey. *International Journal of Machine Learning and Cybernetics* 16, 7 (2025), 5079–5112.
[49] Sanjay Krishnan, Aaron J Elmore, Michael Franklin, John Paparrizos, Zechao Shang, Adam Dziedzic, and Rui Liu. 2019. Artificial intelligence in resource-constrained and shared environments. *ACM SIGOPS Operating Systems Review*

53, 1 (2019), 1–6.
[50] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In *NeurIPS Track on Datasets and Benchmarks*.
[51] N. Laptev, S. Amizadeh, and Y. Billawala. 2015. *S5 - A Labeled Anomaly Detection Dataset, version 1.0(16M).* https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70
[52] Haojun Li and John Paparrizos. 2026. MUFASA: Fast and Accurate Multivariate Time-Series Clustering. *Proceedings of the ACM on Management of Data* 4, 3, Article 213 (June 2026), 29 pages. doi:10.1145/3802090
[53] Zhi Li, Hong Ma, and Yongbing Mei. 2007. A Unifying Method for Outlier and Change Detection from Data Streams Based on Local Polynomial Fitting. In *Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science)*, Zhi-Hua Zhou, Hang Li, and Qiang Yang (Eds.). Springer, Berlin, Heidelberg, 150–161. doi:10.1007/978-3-540-71701-0_17
[54] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 6555–6565.
[55] Bryan Lim and Stefan Zohren. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* 379, 2194 (2021), 20200209.
[56] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2586–2598.
[57] Chunwei Liu, John Paparrizos, and Aaron J Elmore. 2024. AdaEdge: A Dynamic Compression Selection Framework for Resource Constrained Devices. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1506–1519.
[58] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. IEEE, 413–422.
[59] Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. 2024. Time-Series Anomaly Detection: Overview and New Trends. *Proceedings of the VLDB Endowment (PVLDB)* 17, 12 (2024), 4229–4232.
[60] Qinghua Liu, Sam Heshmati, Zheda Mai, Zubin Abraham, John Paparrizos, and Liu Ren. 2025. Mllm4ts: Leveraging vision and multimodal language models for general time-series analysis. *arXiv preprint arXiv:2510.07513* (2025).
[61] Qinghua Liu, Seunghak Lee, and John Paparrizos. 2025. Easyad: A demonstration of automated solutions for time-series anomaly detection. *Proceedings of the VLDB Endowment* 18, 12 (2025), 5431–5434.
[62] Qinghua Liu, Seunghak Lee, and John Paparrizos. 2025. TSB-AutoAD: Towards Automated Solutions for Time-Series Anomaly Detection. *PVLDB* 18, 11 (2025), 4364–4379.
[63] Qinghua Liu and John Paparrizos. 2024. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. *Advances in Neural Information Processing Systems* 37 (2024), 108231–108261.
[64] Shinan Liu, Tarun Mangla, Ted Shaowang, Jinjin Zhao, John Paparrizos, Sanjay Krishnan, and Nick Feamster. 2023. AMIR: Active Multimodal Interaction Recognition from Video and Network Traffic in Connected Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–26.
[65] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.
[66] Pankaj Malhotra, L. Vig, Gautam M. Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN*.
[67] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.
[68] Aditya P Mathur and Nils Ole Tippenhauer. 2016. 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)*. IEEE, 31–36.
[69] Kathy McKeown, Hal Daume III, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R Fleischmann, et al. 2016. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology* 67, 11 (2016), 2684–2696.
[70] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. 2018. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access* 7 (2018), 1991–2005.
[71] Peter Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph. D. Dissertation. Princeton University.
[72] Anastasios Papadopoulos, Apostolos Giannoulidis, Anastasios Gounaris, and John Paparrizos. 2026. The Power of Anomaly Detection in Predictive Maintenance. *Proceedings of the ACM on Management of Data* 4, 2, Article 242 (June 2026), 34 pages. doi:10.1145/3802119
[73] Ioannis Paparrizos. 2018. *Fast, scalable, and accurate algorithms for time-series analysis*. Ph. D. Dissertation. Columbia University.
[74] Ioannis Paparrizos, Stylianos Basagiannis, and Sophia Petridou. 2011. Quantitative analysis for authentication of low-cost RFID tags. In *2011 IEEE 36th Conference on Local Computer Networks*. IEEE, 295–298.

[75] Ioannis Paparrizos, Hoyoung Jeung, and Karl Aberer. 2011. Advanced search, visualization and tagging of sensor metadata. In *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 1356–1359.

[76] John Paparrizos, Paul Boniol, Qinghua Liu, and Themis Palpanas. 2025. Advances in Time-Series Anomaly Detection: Algorithms, Benchmarks, and Evaluation Measures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 6151–6161.

[77] John Paparrizos, Paul Boniol, Qinghua Liu, and Themis Palpanas. 2026. A Comprehensive Guide to Time-Series Anomaly Detection. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*. 1363–1366.

[78] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. 2022. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2774–2787.

[79] John Paparrizos, Ikraduya Edian, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2022. Fast Adaptive Similarity Search through Variance-Aware Quantization. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2969–2983.

[80] John Paparrizos and Michael J Franklin. 2019. GRAIL: efficient time-series representation learning. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1762–1777.

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

[82] John Paparrizos and Luis Gravano. 2017. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 1–49.

[83] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. 2022. TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1697–1711.

[84] John Paparrizos, Haojun Li, Fan Yang, Kaize Wu, Jens E d'Hondt, and Odysseas Papapetrou. 2024. A survey on time-series distance measures. *arXiv preprint arXiv:2412.20574* (2024).

[85] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikraduya Edian, Aaron J Elmore, Michael J Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices.. In *CIDR*.

[86] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2020. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.

[87] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2023. Querying Time-Series Data: A Comprehensive Comparison of Distance Measures. *Data Engineering* (2023), 69.

[88] John Paparrizos and Sai Prasanna Teja Reddy. 2023. Odyssey: An Engine Enabling the Time-Series Clustering Journey. *Proceedings of the VLDB Endowment* 16, 12 (2023), 4066–4069.

[89] John Paparrizos and Sai Prasanna Teja Reddy. 2025. Time-Series Clustering: A Comprehensive Study of Data Mining, Machine Learning, and Deep Learning Methods. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4380–4395.

[90] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Detecting devastating diseases in search logs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 559–568.

[91] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Screening for pancreatic adenocarcinoma using signals from web search logs: Feasibility study and results. *Journal of Oncology Practice* 12, 8 (2016), 737–744.

[92] John Paparrizos, Kaize Wu, Aaron Elmore, Christos Faloutsos, and Michael J Franklin. 2023. Accelerating Similarity Search for Elastic Measures: A Study and New Generalization of Lower Bounding Distances. *Proceedings of the VLDB Endowment* 16, 8 (2023), 2019–2032.

[93] John Paparrizos, Fan Yang, and Haojun Li. 2024. Bridging the Gap: A Decade Review of Time-Series Clustering Methods. *arXiv preprint arXiv:2412.20582* (2024).

[94] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[95] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

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

[97] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 3009–3017.

[98] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 233–240.

[99] Peter Rousseeuw, Domenico Perrotta, Marco Riani, and Mia Hubert. 2019. Robust monitoring of time series with application to fraud detection. *Econometrics and statistics* 9 (2019), 108–121.

[100] Peter J Rousseeuw and Katrien Van Driessen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 3 (1999), 212–223.

[101] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*. 4–11.

[102] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment* 15, 9 (2022), 1779–1797.

[103] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99)*. MIT Press, Cambridge, MA, USA, 582–588.

[104] Iman Sharafaldin, Arash Habibi Lashkari, Ali A Ghorbani, et al. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* 1 (2018), 108–116.

[105] Haotian Si, Jianhui Li, Changhua Pei, Hang Cui, Jingwen Yang, Yongqian Sun, Shenglin Zhang, Jingjing Li, Haiming Zhang, Jing Han, et al. 2024. Timeseriesbench: An industrial-grade benchmark for time series anomaly detection models. In *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 61–72.

[106] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2828–2837. doi:10.1145/3292500.3330672

[107] Emmanouil Sylligardos, Paul Boniol, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2023. Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3418–3432.

[108] Emmanouil Sylligardos, John Paparrizos, Themis Palpanas, Pierre Senellart, and Paul Boniol. 2025. MSAD: A deep dive into model selection for time series anomaly detection. *The VLDB Journal* 34, 6 (2025), 1–25.

[109] Luan Tran, Liyue Fan, and Cyrus Shahabi. 2016. Distance-based outlier detection in data streams. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1089–1100.

[110] Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. 2021. A Comprehensive Survey and Experimental Comparison of Graph-Based Approximate Nearest Neighbor Search. arXiv:2101.12631 [cs.IR] https://arxiv.org/abs/2101.12631

[111] Renjie Wu and Eamonn J Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering* 35, 3 (2021), 2421–2429.

[112] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*. 187–196.

[113] Zhijian Xu, Ailing Zeng, and Qiang Xu. 2023. FITS: Modeling Time Series with $10k$ Parameters. In *The Twelfth International Conference on Learning Representations*.

[114] Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. 2001. Fault detection by mining association rules from house-keeping data. In *proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Vol. 18. Citeseer, 21.

[115] Fan Yang and John Paparrizos. 2025. SAIL: A Voyage to Symbolic Approximation Solutions for Time-Series Analysis. *Proceedings of the VLDB Endowment* 18, 12 (2025), 5419–5422.

[116] Fan Yang and John Paparrizos. 2025. SPARTAN: Data-Adaptive Symbolic Time-Series Approximation. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–30.

[117] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2023. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 3033–3045.

[118] Chin-Chia Michael Yeh, Audrey Der, Uday Singh Saini, Vivian Lai, Yan Zheng, Junpeng Wang, Xin Dai, Zhongfang Zhuang, Yujie Fan, Huiyuan Chen, et al. 2024. Matrix Profile for Anomaly Detection on Multidimensional Time Series. In *2024 IEEE International Conference on Data Mining (ICDM)*. IEEE, 911–916.

[119] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee, 1317–1322.

[120] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *AAAI*, Vol. 36. 8980–8987.

[121] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. 2024. Deep learning for time series anomaly detection: A survey. *Comput. Surveys* 57, 1 (2024), 1–42.

[122] Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2022. TFAD: A Decomposition Time Series Anomaly Detection Architecture with Time-Frequency Analysis. In *CIKM (CIKM '22)*. ACM, 2497–2507. doi:10.1145/3511808.3557470

Received October 2025; revised January 2026; accepted February 2026