The Power of Anomaly Detection in Predictive Maintenance: [Experiments & Analysis]

ANASTASIOS PAPADOPOULOS, Aristotle University of Thessaloniki, Greece
APOSTOLOS GIANNOULIDIS, ETIS, UMR 8051, CYU, ENSEA, CNRS, France
ANASTASIOS GOUNARIS, Aristotle University of Thessaloniki, Greece
JOHN PAPARRIZOS, The Ohio State University, USA and Aristotle University of Thessaloniki, Greece

Predictive maintenance (PdM) is a key application of the industrial IoT, using sensor-acquired time-series data to forecast equipment failures that cost billions of dollars annually. Time-series anomaly detection (TSAD) offers a promising route to PdM, yet prior work is constrained by: i) reliance on private datasets or case studies, requiring extensive domain-specific engineering; ii) evaluation of only a small number of algorithms, limiting comparative insight; iii) use of a single implementation strategy per algorithm, hindering potential improvements; iv) emphasis on supervised solutions, necessitating annotated data that are costly or unavailable in practice; and v) neglect of runtime and online applicability, raising questions about deployability. To address these limitations, we present an extensive experimental study with rigorous statistical analysis conducted within a common evaluation framework. Our objectives are to: i) provide insights into the accuracy, robustness, and behavior of TSAD techniques in an online PdM setting across four implementation strategies; ii) analyze effectiveness–runtime trade-offs; and iii) rate dataset difficulty and characterize the forms exhibited by anomalies preceding machine breakdowns. Our findings indicate that most industrial cases benefit from an initial calibration phase for operational data collection. Well-established traditional TSAD methods deliver the best trade-off among effectiveness, runtime efficiency, and the ability to predict individual failure types; pre-trained LLMs are still statistically outperformed in this context, suggesting that there is still room for improvement in that direction. Our study serves as a significant step towards disseminating TSAD for PdM and we open source our work to support future research.

CCS Concepts: • **Information systems → Data analytics**; • **Computing methodologies → Anomaly detection**; **Semi-supervised learning settings**; **Online learning settings**.

Additional Key Words and Phrases: Predictive Maintenance; Time-Series

ACM Reference Format:
Anastasios Papadopoulos, Apostolos Giannoulidis, Anastasios Gounaris, and John Paparrizos. 2026. The Power of Anomaly Detection in Predictive Maintenance: [Experiments & Analysis]. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 242 (June 2026), 33 pages. https://doi.org/10.1145/3802119

1 INTRODUCTION

Industry 4.0 integrates Internet of Things (IoT), Artificial Intelligence (AI), and robotics to establish interconnected, automated, and intelligent cyber-physical manufacturing systems. The global IoT market, valued at 18.34 billion USD in 2024, is projected to reach 422.13 billion USD by 2034, driven primarily by advances in connectivity and by data-driven innovations across industrial, healthcare,

Authors' Contact Information: Anastasios Papadopoulos, Aristotle University of Thessaloniki, School of Informatics, Thessaloniki, Greece, apapadoi@csd.auth.gr; Apostolos Giannoulidis, ETIS, UMR 8051, CYU, ENSEA, CNRS, Paris, France, apostolos.giannoulidis@cyu.fr; Anastasios Gounaris, Aristotle University of Thessaloniki, School of Informatics, Thessaloniki, Greece, gounaria@csd.auth.gr; John Paparrizos, The Ohio State University, Columbus, Ohio, USA and Aristotle University of Thessaloniki, School of Informatics, Thessaloniki, Greece, john@paparrizos.org.

ACM 2836-6573/2026/6-ART242
https://doi.org/10.1145/3802119

and agricultural sectors [70, 86, 87, 103, 153, 180, 181]. IoT databases and relevant data management platforms form the operational backbone of these applications, providing storage, indexing, spatial/temporal querying, and ingestion pipelines required for real-time and historical analytics [78, 90, 92, 93, 112, 123, 124, 134, 143, 154, 159, 165, 204, 205, 213, 224, 225]. A persistent challenge in the Industry 4.0 paradigm is industrial maintenance: the problem of ensuring the continuous, safe, and cost-effective operation of heterogeneous assets—including machines, production lines, vehicles, and infrastructure—throughout their lifecycle. Machine failures have been estimated to cost the world's largest manufacturers on the order of 1 trillion USD per year [7].

Discriminating genuine machinery degradation from measurement noise, environmental effects (e.g., altitude-driven changes in pressure and oxygen affect turbocharger behavior), configuration or process shifts, and IoT hardware/software inconsistencies remain challenging and confound the temporal analysis of sensor signals [15, 53, 66, 69, 111, 129, 155, 160, 168, 169]. Traditional maintenance approaches, including reactive [63, 77, 151] and preventive maintenance [16, 122, 219, 232], are often associated with inefficiencies, such as unplanned downtime and unnecessary interventions, both of which contribute to elevated operational costs. In contrast, predictive maintenance (PdM) uses continuous IoT time-series (e.g., temperature, vibration, pressure) and analytics to detect early degradation and enable targeted interventions, with reported reductions in maintenance costs and unplanned downtime of up to 40% and 50%, respectively [19, 75].

Prior research on PdM has predominantly focused on single, often proprietary, datasets or use cases, resulting in heavy reliance on domain-specific knowledge at the expense of generalizability [33, 50, 195, 208, 241]. Most studies treat PdM as a supervised classification problem [97, 194, 238], evaluate only a limited number of algorithms [5, 52, 147, 212]—typically three or fewer—seldom evaluate runtime performance [39, 40, 139, 240], and frequently overlook considerations of online applicability [84, 95, 108, 173]. Several implementation families coexist: simple empirical static thresholds and rule-based systems produce transparent and easily deployable alerts [30, 68, 99, 222], but suffer from high false alarm rates and limited ability to capture complex or nonlinear degradation; fuzzy systems encode expert knowledge with graded membership functions and so improve flexibility and interpretability, yet their design and real-time execution scale poorly as inputs and rules increase [4, 98, 102]; model-based techniques leverage mathematical descriptions of system dynamics and can deliver high accuracy when failure physics are well characterised, but they typically require labeled data and generalise poorly to novel failure modes [2, 55, 203]; survival models estimate degradation trajectories and failure probabilities but depend on consistent, reliable degradation patterns that are often absent in industrial practice [80, 81, 227]. In light of these limitations, time-series anomaly detection (TSAD)—the task of identifying unexpected or infrequent patterns in temporal data [22, 25, 27, 28, 115, 128, 131, 132, 156, 158]—has emerged as a promising, versatile alternative for PdM, combining low time-to-value, easy deployment, and modest data requirements and thus gaining traction in industry and research alike [58, 60, 64–66, 130, 137, 138, 145, 157, 171, 187, 196, 199, 200].

Although comprehensive research has been conducted on TSAD [82, 83, 89, 132, 163, 176, 210, 218], its systematic application in PdM is underexplored. TSAD for PdM differs in three essential respects: i) classical TSAD focuses on identifying unexpected deviations per se, whereas TSAD for PdM seeks anomalies that are precursors to failures [33, 119]; ii) traditional TSAD aims to detect anomalies contemporaneously with their occurrence; TSAD for PdM must account for sufficient lead time to enable maintenance planning [44, 189]; and iii) whereas TSAD datasets offer explicit anomaly annotations, maintenance datasets document failure events, lacking reliable information on preceding anomalous behavior [139, 226]. These differences motivate a systematic study of TSAD for PdM to provide insights into detector behaviour and operational utility. In online settings, a high-level TSAD technique can adopt various operational flavors, a semi-supervised (*online*)

Table 1. Comparison between our study and existing time-series anomaly detection/PdM studies.

Criteria	Exathlon [89]	TSB-UAD [163]	TSB-AD [132]	TimeSeries-Bench [193]	TAB [176]	AutoTSAD [185]	[149]	[208]	[50]	[241]	[195]	**Our study**
Online setting	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓
Failure prediction	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
(Semi/un)-supervised	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Implementation strategies	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✗	✓
Multiple datasets	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓
Real-world datasets	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Experimental datasets	✗	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✓
Synthetic datasets	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Dataset ranking	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
Runtime analysis	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
Analysis across multiple HP	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

approach that trains on segments unlikely to contain failures, incremental window schemes (*sliding*), and fully unsupervised modes—each offering distinct trade-offs. To the best of our knowledge, although these variants have been explored independently in both academic research and the industry [35, 41, 59, 101, 140, 142, 163, 174, 179, 182, 209, 216, 235], a concurrent, comparative evaluation remains absent.

In this study, we present the most extensive experimental analysis of TSAD for PdM to date, offering comprehensive insights into the accuracy, limitations, and operational trade-offs for (semi/un)-supervised TSAD algorithms in PdM. As summarized in Table 1, we close key gaps in prior work by analyzing worst-, typical-, and best-case performance across multiple hyperparameter configurations, rather than the customary best-case-only results. This robustness-focused analysis is motivated by practical constraints—limited compute on edge IoT devices, data-localization restrictions that may prohibit cloud-based tuning, and concept drift that can invalidate previously optimized hyperparameters—under which worst- and typical-case analyses better inform model selection. We also study failure prediction in online settings and compare four distinct implementation strategies, aspects that remain underexplored. In addition, to the best of our knowledge, we integrate for the first time a Formula 1 dataset in a PdM context. Unlike the other 10 datasets, the Formula 1 dataset exhibits non-constant, sub-second sampling rates, thereby extending PdM corpora predominantly with a challenging, real-world workload. Our experimental testbed scales beyond prior PdM studies by combining diverse real, experimental, and synthetic workloads, alongside dataset ranking and runtime analysis.

We begin by describing how TSAD can be applied to a PdM case (Section 2) and reviewing related work (Section 3). Following this, we present our contributions:

- We define the experimental setup of our study (Section 4).
- We provide insights into the accuracy of TSAD techniques and demonstrate how dataset characteristics determine a different flavor choice (Section 5.1).
- We analyze each technique's effectiveness-runtime trade-offs, highlighting how computationally efficient methods compete with more complex alternatives (Section 5.2).
- We examine the robustness to hyperparameter selection of TSAD solutions for PdM (Section 5.3).
- We assess dataset difficulty and characterize the anomaly types present in each dataset (Section 5.4).
- Through a real-world case study, we assess TSAD methods in terms of predictive capability for individual failure types and robustness to adversarial manipulations. (Section 5.5).
- We suggest guidelines for practitioners when applying TSAD solutions to their use cases and identify promising directions to improve TSAD for PdM (Section 6).

Finally, we conclude our work (Section 7). The source code, data, and other artifacts, including the Appendix, are available at [1].

2 PROBLEM STATEMENT

Let S denote a set of (multivariate) time-series where each $s \in S$ is defined over $[s^{t_{\text{start}}}, s^{t_{\text{end}}}]$. The ground truth for each monitored asset s is represented by a set of failure events $F^s = \{f_i^s\}$, where $s^{t_{\text{start}}} \leq f_i^s \leq s^{t_{\text{end}}}$. The complete *maintenance dataset* is thus the collection $\{(s, F^s) \mid s \in S\}$. An illustrative artificial example in Figure 1(a) shows assets with varying numbers of failures, which do not necessarily occur at terminal points. We adopt the evaluation framework of [66] and the VUS metric [21, 158], augmenting the data with a *Predictive Horizon* (PH) and *Lead time* (L). The PH is the interval preceding a failure where raising alarms and observing elevated anomaly scores are desirable, while the lead time L spans from the end of the PH to the failure, ensuring sufficient time for maintenance (Figure 1(b)). Following common practice [186], samples within the union of PH and L are labelled positive (1), and the rest negative (0). As depicted in Figure 1(c), a TSAD method operating in an **online** fashion produces a continuous anomaly score for each incoming sample; applying a threshold to this score yields binary predictions. Note that alarms occurring within L are disregarded (Figure 1(d)) as they provide insufficient time-to-failure.

Computation of the evaluation metrics requires the introduction of *episodes*, defined by the temporal occurrence of failure events. An episode begins either at the $s^{t_{\text{start}}}$ of a time-series s or immediately after a failure, and terminates at the subsequent failure or at $s^{t_{\text{end}}}$. Segmenting the time-series into episodes aids in determining the PH and L intervals. In our experimental study, we focus on the AD1 level [66]. $\text{Precision}_{\text{AD1}}$ is computed as the fraction of detected anomalies that fall within the PH:

$$\text{precision}_{\text{AD1}} = \frac{\#\{\text{anomalies detected inside PH}\}}{\#\{\text{all anomalies detected}\}},$$

while $\text{recall}_{\text{AD1}}$ is defined as the fraction of episodes in which at least one anomaly is detected within the PH:

$$\text{recall}_{\text{AD1}} = \frac{\#\{\text{episodes with } \geq 1 \text{ anomaly in PH}\}}{\#\{\text{episodes}\}}.$$

The area under the PR_{AD1} curve formed by (recall_{AD1},precision_{AD1}) points computed over a set of thresholds, is defined as: AD1 AUC-PR $= \int_0^1 \text{precision}_{\text{AD1}}(r)\, dr$, i.e. larger precision_{AD1} and

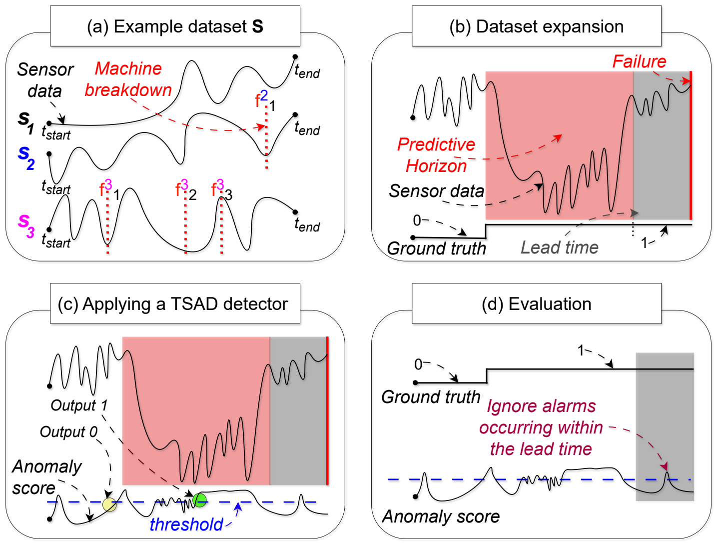


Fig. 1. (a) An artificial illustration of a maintenance dataset. (b) Expansion of a PdM case with the concepts of Predictive Horizon and Lead periods. (c) Anomaly score generation from a TSAD method. (d) Computation of the evaluation metrics.

recall_{AD1} values increase AD1 AUC-PR. For VUS-based evaluation, precision and recall are defined analogously to the AD1 definitions above.

Problem Definition. Given a maintenance dataset S of n (multivariate) time-series (with m channels) and associated failure events, along with fixed PH and L parameters, the goal is to learn a set of functions $\Phi_\theta = \{\phi_s : \mathbb{R}^{s^{t_{end}} \times m} \longrightarrow \mathbb{R}^{s_{t_{end}}} \mid s \in S\}$ with a shared hyperparameterization θ across assets in S. Each function ϕ_s maps a time-series s to a vector of anomaly scores $y \in \mathbb{R}^{s_{t_{end}}}$. The objective is to maximize the AD1 AUC-PR derived from the PR_{AD1} curve.

3 RELATED WORK

Time-Series Anomaly Detection. TSAD remains an active and enduring research area [8, 23, 24, 26, 100, 105, 107, 127, 193, 234]. The diversity of anomaly types, data characteristics, and application domains drives continued interest, given its impact on critical tasks ranging from cost reduction in intrusion detection to disaster prevention. A wide variety of TSAD techniques have been proposed over the years; our intention here is not to exhaustively review them. Instead, we classify the evaluated methods into four broad categories: proximity (P), distribution (D), tree (T), and deep learning (DL) based approaches. Proximity-based methods flag points in sparsely populated localities [6], while distribution-based ones assess deviations from the data distribution. Tree-based methods rely on hierarchical decision structures [14]. DL-based methods employ neural networks, typically reconstructing or forecasting time-series values to compute anomaly scores from the distance between observed and predicted values. Beyond raw performance, explainability [118] is an essential requirement for industrial adoption, since operators must validate alerts before committing to maintenance actions. Accordingly, Section 4.1 describes the evaluated TSAD methods together with their explainability properties. In addition, Section 5.5 showcases how a model-agnostic method can be employed in a PdM case.

Table 2. Categorization of the TSAD techniques. Some techniques can be applied both in a semi-supervised and an unsupervised way. A * symbol denotes streaming capability for the unsupervised variant of a method (after windowing).

Method	Unsupervised	Semisupervised	Built-in distribution shift robustness	Category	Architecture (Sequence-based: S Point-based: P)	Key mechanisms	Streaming Capability	Input type (Univariate: U Multivariate: M)
CHRONOS [10]	✓	✓	✗	DL	S	Quantization, LLM Pretraining	Streaming	U
CHRONOS-2 [9]	✓	✓	✗	DL	S	In-context learning, LLM Pretraining	Streaming	M
IF [126]	✓	✓	✗	T	P	Isolation tree, Ensembling	Streaming*	M
KNN [178]	✓	✓	✗	P	P	K-nearest neighbor	Streaming*	M
LOF [29]	✓	✓	✗	P	P	Density	Streaming*	M
LTSF [231]	✗	✓	✓	DL	S	Fully-connected linear layers	Streaming	M
NP [76]	✓	✓	✗	P	S	Matrix-profile	Static*	U
OCSVM [191]	✗	✓	✗	D	P	Kernel transformation	Streaming	M
PB [66]	✗	✓	✗	P	P	Nearest-neighbor	Streaming	M
SAND [27]	✓	✗	✓	P	S	Time-series clustering	Streaming*	U
TIME-LLM [94]	✗	✓	✗	DL	S	LLM reprogramming	Streaming	M
TIMEMIXER++ [214]	✗	✓	✗	DL	S	Multi-scale decomposition	Streaming	M
TRANAD [206]	✗	✓	✗	DL	S	Attention, Meta/Adversarial learning	Streaming	M
USAD [12]	✗	✓	✗	DL	S	Dual adversarial autoencoder	Streaming	M

Deploying production-grade PdM systems requires end-to-end integration across the engineering stack [79]. Robustness to manipulation is an increasing concern, and adversarial research demonstrates that attackers need not alter model weights to undermine TSAD: false data injection and sensor spoofing strategies that introduce additive noise or induce temporal jittering can decouple observations from physical reality and bypass conventional statistical filters and state estimators [56, 192]. Both classical and DL detectors are vulnerable; discrete optimization attacks can corrupt tree traversals [233], gradient-based, dynamic time warping (DTW), and feature-space perturbations can mask faults or produce spurious alarms [17, 18, 201, 221], while recent black-box attacks show that increasingly popular large pretrained models are also vulnerable [125]. To move beyond heuristic hardening, the community is adopting certified defenses: randomized smoothing [45] and its temporal adaptations yield tractable estimates of a certified radius within which model decisions are provably invariant, while diffusion-based purification aims to tighten safety bounds with lower verification variance [31, 133]. Section 5.5 examines the adversarial robustness of detectors through a real-world case study.

Maintenance Approaches. Maintenance approaches are commonly categorized as reactive, preventive, and predictive [141]; reactive maintenance postpones activities until failure, raising repair costs [135, 215, 230], while preventive maintenance seeks to reduce unexpected breakdowns through scheduled inspection and servicing [114, 198, 217, 220, 236]. In our analysis, we incorporate a methodology designed to represent preventive maintenance approaches. PdM often centers on predicting remaining useful life (RUL) to schedule interventions proactively, but RUL methods, despite their cost-saving potential, are hampered by scarce labeled failure data and limited diversity of failure modes, which constrain generalizability and introduce uncertainty into predictions. A broad literature proposes model-based, data-driven, and hybrid RUL algorithms [47, 57, 113, 117, 223, 239], yet most rely on limited benchmarks such as CMAPSS [184] (which we also use) and struggle to adapt to varying load conditions or handle intermittent faults (transient, unpredictable errors that disappear without maintenance) [150], and often depend on manually engineered, information-rich features [113, 239]; such limitations impede reliable deployment in industrial settings.

4 EXPERIMENTAL SETUP

This section outlines the experimental setup of our study. Our design emphasizes real-world data and evaluation metrics appropriate in TSAD for PdM contexts. By standardizing these elements, we aim to provide a consistent foundation for meaningful operational improvements and address misalignments in prior work [71, 96]. We first present the TSAD algorithms evaluated (Section 4.1) and their various flavors (Section 4.2). We then describe the datasets employed (Section 4.3) and implementation details (Section 4.4).

4.1 Algorithms

In Table 2, we present a categorization of the selected algorithms. It is important to note that some methods are available in both semi-supervised and unsupervised flavors. The objective of this study is not to conduct an exhaustive evaluation, which would require the inclusion of over one hundred TSAD algorithms [176]. Instead, we prioritize methodological diversity by selecting representative algorithms with heterogeneous characteristics (e.g., inductive mechanisms, robustness to distribution shifts) that have demonstrated high effectiveness in both PdM applications [65–67] and traditional TSAD [132, 176]. A comprehensive benchmark encompassing all existing algorithms is left for future work. For the application of univariate techniques to multivariate data, each dimension is processed independently, and the resulting anomaly scores are aggregated using either the mean or the maximum value.

Although foundation models for time-series analysis are a promising and active research direction, we prioritized CHRONOS [10] due to its extensive pretraining and theoretical suitability for the edge via zero-shot inference and quantization [48, 104]. To ensure a comprehensive assessment of this category without diluting the main benchmark with architectures that currently show limited applicability, we also extended our evaluation to CHRONOS-2 [9], TIME-LLM [94], and TIMEMIXER++ [214]. Our preliminary results (Appendix Tables B.1-B.2) confirm that the examined instances generally underperform the traditional TSAD methods on both detection effectiveness and runtime efficiency (TIME-LLM attains the best VUS-PR in only a single case).

CHRONOS. *CHRONOS* [10] tokenizes continuous-valued time-series through scaling and quantization, mapping values to a fixed vocabulary; in this study, we employ the T5-based variant. This reformulates regression as a token classification task, where the model's prediction error defines the anomaly score.

CHRONOS-2. *CHRONOS-2* [9] utilizes in-context learning for multivariate forecasting with covariates via a group attention mechanism; we use past-only covariates. Instead of concatenating inputs, it shares information across the batch dimension, enabling scalable modelling of dependencies between targets and related series.

IF. *Isolation Forest* [126] constructs binary trees based on random space splitting. The nodes with shorter paths to the root are more likely to be anomalies. An ensemble of such randomly constructed isolation trees is employed, and the anomaly score is computed using the average path length across the ensemble.

KNN. In *KNN* [178], the anomaly score of a point is calculated as the distance from its k-st nearest neighbour in a reference set.

LOF. *Local Outlier Factor* [29] detects anomalies by comparing a sample's local density to that of its k-nearest neighbours. The anomaly score is derived from the ratio of the neighbours' Local Reachability Density (LRD) to the sample's LRD; values significantly greater than 1 indicate the point lies in a lower-density region than its surroundings, signalling an outlier.

LTSF. The *LTSF-Linear* family [231] predicts the future values of a time-series. Similar to CHRONOS, the prediction error is used to calculate the anomaly score. LTSF-Linear comes with three variants:

Linear, DLinear and NLinear, with each one being engineered slightly differently for tackling different issues, such as trends in the data (DLinear) and distribution shifts (NLinear).

NP. *Neighbor Profile* [76] was proposed to calculate a more generic matrix-profile [229], supporting k-nearest-neighbor density functions. NP employs a bagging-like subsampling and aggregation strategy over multiple nearest-neighbour distances, yielding smoother and more reliable density estimates for subsequences.

OCSVM. *One-class Support Vector Machine* [191] finds the best-separating hyperplane after transforming samples into a feature space through the use of a kernel function. The anomaly score of a new point is its distance from the hyperplane.

PB. The *Profile-based* [66] method produces the anomaly score of a sample as its distance from a reference set, divided by the maximum distance between data points in this set.

SAND. *SAND* [27] performs online clustering, scoring anomalies based on the distance to temporal centroids. It maintains a weighted subsequence memory updated via K-Shape [152, 161, 162], incorporating both cardinality and recency in cluster weights.

TIME-LLM. *TIME-LLM* [94] repurposes LLMs for forecasting by treating time-series as a language task. It aligns input series with the LLM's capabilities using learnable text prototypes and augments the context with declarative prompts to guide reasoning.

TIMEMIXER++. *TIMEMIXER++* [214] captures multi-scale dynamics by treating time-series as time images. It applies dual-axis attention to separate trend and seasonality, thereby adaptively learning comprehensive time-frequency representations.

TRANAD. *Deep Transformer Networks for Anomaly Detection in Multivariate Time-Series Data* [206] is a transformer-based reconstruction model that identifies anomalies via reconstruction errors, using adversarial training to amplify sensitivity to subtle deviations. It further employs model-agnostic meta-learning and self-conditioning to enable fast, robust generalization across domains.

USAD. *UnSupervised Anomaly Detection on Multivariate Time-Series* [12] is an auto-encoder model that scores anomalies via reconstruction error. It employs adversarial training between decoder stages to amplify deviations and enhance sensitivity to subtle anomalies.

Explainability properties. The proximity-based methods in Table 2 (P category) are distance-based and therefore qualify as white-box, interpretable techniques because the computation of the anomaly score can be directly traced to distances or neighbourhood structures. For instance, a low KNN anomaly score is attributable to nearby reference points, whereas in SAND, the anomaly score reflects proximity to a learned cluster centroid; conversely, high scores occur when observations lie far from the reference set (KNN) or from learned clusters (SAND). By contrast, approaches in the DL and D classes do not provide an explicit, human-readable mapping from input features to anomaly scores: their outputs derive from learned latent representations or from kernel transformations, which motivates the application of inherently interpretable surrogate models or post-hoc explanation methods [106, 144]. IF exposes an interpretable mechanism at the algorithmic level (path lengths in random trees), yet attributing anomalousness to specific input features remains challenging; recent work has therefore proposed techniques to produce feature-level importances and explanations for IF [32]. Previous work [46] reports that model-agnostic approaches are the most widely adopted in practice, while also highlighting the challenge of validating explanations, which typically requires expert annotation of proprietary datasets.

4.2 Flavors

Machine learning (ML) algorithms are commonly classified as supervised, semi-supervised, and unsupervised. The choice among these classes depends on the characteristics of the data and the task. Supervised methods require labelled data, whereas semi-supervised approaches can operate

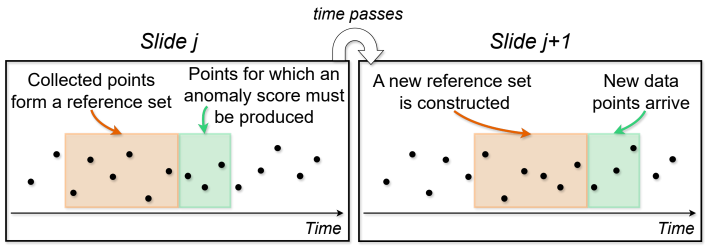


Fig. 2. A single step in the semi-supervised sliding flavor.

Table 3. Flavor prerequisites and zero day applicability.

Methodology	Flavor	Historical Failure Data	Historical Operational Data	Zero Day Applicability
Supervised	Regression	✓	✓	✗
Supervised	Classification	✓	✓	✗
Semi-Supervised	Historical	✗	✓	✗
Semi-Supervised	Online	✗	✗	✓
Semi-Supervised	Sliding	✗	✗	✓
Unsupervised	Unsupervised	✗	✗	✓

with partial labels—typically only for normal (reference) data. Unsupervised methods do not rely on labels. Within semi-supervised settings, different assumptions about normal data dictate how a given detector can be applied.

4.2.1 Semi-supervised. The following flavors capture TSAD methods that require training. The three alternatives are essentially facets of a single higher-level approach: a model is trained on data collected during a presumed healthy operation and subsequently used to produce anomaly scores. For deployment in truly online settings, the training data must temporally precede the data used for scoring (i.e., all training timestamps < scoring timestamps).

Semi-supervised (historical). This approach trains algorithms on healthy historical data. A domain expert must confirm that the training data reflects healthy asset behaviour.

Semi-supervised (online). In this flavor, a compact reference set is constructed for each monitored asset at the start of every episode, obviating the need for long-term historical normal data. The reference set is substantially smaller than a typical historical dataset and therefore only serves as a proxy for normal behavior; it may not be representative and can contain failures or anomalous observations.

Semi-supervised (sliding). Using a sliding-window scheme, the model is retrained on the most recently collected data before generating anomaly scores for new observations. When the window reaches the required size or a predefined time interval elapses, it advances and a new training sample is formed. A graphical illustration is provided in Figure 2. In an extended variant, the reference dataset consists of all historical data accumulated up to that point.

4.2.2 Unsupervised. Techniques in this category employ problem-agnostic assumptions about the data and function without supervision. They may share characteristics with semi-supervised methods, but run independently of any label information.

Table 4. Descriptive statistics for all raw datasets. A * symbol denotes that PH is calculated either as the 10% of the smallest episode or scenario, or, otherwise, it is set based on domain knowledge or external information/requirements.

Dataset	type	#records	Min scenario length	Avg & std scenario length	dimensions	failures	#scenarios w/ failure	#scenarios w/o failure	PH	maximum PH	L
AZURE [20]	synthetic	876100	8761	8761 ± 0	4	761	100	0	96	192	2
BHD 2023 [13]	real	6626869	11	1529 ± 776	5-54	4334	4334	0	10	20	2
CMAPSS [184]	synthetic	265256	19	187 ± 82	14-21	709	709	707	13*	42	2
CNC [172]	real	6304	140	451 ± 259	120	14	14	0	15	48	5
EDP-WT [85]	real	209236	52244	52309 ± 50	79	8	4	0	8640	8640	288
FEMTO [34]	experimental	21493	172	1264 ± 772	44	6	6	11	52*	169	2
FORMULA 1	real	5563727	4809	22344 ± 10595	17	249	249	0	8 minutes	PH + 1920	4 minutes
IMS [110]	experimental	9464	984	3154 ± 2806	88-176	3	3	0	99*	324	2
METROPT-3 [208]	real	1516948	1516948	1516948 ± 0	15	4	1	0	288	1008	12
NAVARCHOS [65]	real	854178	1482	32853 ± 46824	6	22	14	12	15 days	PH + 15	1 day
XJTU-SY [211]	experimental	9216	42	614 ± 853	44	15	15	0	18	26	2

Table 3 summarizes the data prerequisites associated with each flavor and also lists the requirements of the supervised alternatives. As shown, supervised methods for PdM require both operational and failure past examples; semi-supervised methods typically require only data collected during (healthy) operation; and unsupervised methods do not depend on any labelled or prior data. The last column reports applicability from day zero. Supervised methods are not deployable at day zero because they need labelled historical data (hence a higher time-to-value), whereas unsupervised methods can be deployed immediately. The day zero applicability of semi-supervised methods depends on whether a reference set is available; however, for online and sliding flavors, such a reference set can be collected on the fly, enabling deployment from day zero.

4.3 Datasets

Descriptive statistics for the selected raw datasets are summarized in Table 4. The examined datasets are classified by origin into three categories: real-world (collected under operational, uncontrolled conditions), experimental (acquired from physical systems under controlled experiments), and synthetic (generated by simulation). Each dataset may contain multiple monitored assets—for example, in a vehicle-fleet dataset, each vehicle constitutes a distinct scenario. Not all scenarios include failure events; Table 4 reports this using the columns #scenarios_w/_failure and #scenarios_w/o_failure. A run-to-failure scenario is defined as one in which a failure occurs at the end of the time-series, whereas in a non-run-to-failure scenario, any failure that occurs may do so at an arbitrary time.

In some cases the precise PH at which an alarm should be issued is ambiguous. We address this ambiguity by using the maximum PH in conjunction with the VUS metric. Specifically, the PH is set to $0.1 \times L_{min}$, where L_{min} is the length of the shortest scenario in the dataset (for datasets with non-run-to-failure scenarios we use the shortest episode length). The maximum PH is then defined as $\frac{1}{3} \times L_{\min}$. We justify our selections to account for the variability in time-series lengths and sampling rates across datasets. Moreover, in PdM deployments, alarms must be raised within a practically meaningful window—sufficiently early to enable maintenance planning, yet not so far in advance that the warning loses operational relevance. Representative examples can be found in Figure A.1 of the Appendix. We now describe the selected datasets in greater detail.

AZURE. The AZURE dataset [20] comprises hourly measurements of voltage, rotational speed, pressure, and vibration signals collected from 100 machines during 2015. In addition, the dataset includes records of maintenance operations for individual machine components, failure events, and logged error states.

BHD 2023. The BHD 2023 dataset [13] contains S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology) statistics from hard drives operated in Backblaze's data centers.
CMAPSS. The Turbofan Engine Degradation Simulation dataset [184] is generated using the Commercial Modular Aero-Propulsion System Simulation (CMAPSS). It comprises run-to-failure simulation data for four distinct engine fleets, each operating under varying combinations of operating conditions and fault modes.
CNC. The CNC dataset [172] comprises run-to-failure recordings of vibration and electrical drive-current signals acquired during milling of 42CrMo4 steel with 14 cutting tools. Failure events are annotated as tool damage, specifically cutter/blade breakage.
EDP-WT. The Energias de Portugal Wind Turbine dataset [85] comprises one year of SCADA (Supervisory Control and Data Acquisition) measurements from four wind turbines.
FEMTO. The FEMTO dataset [34] incorporates 17 accelerated bearing experiments conducted on a small-scale bearing test rig. This dataset originates from the FEMTO-ST Institute and was employed in the 2012 IEEE Prognostics Challenge.
FORMULA 1. The FORMULA 1 dataset accommodates telemetry data pertaining to the vehicle, including throttle position, brake usage, engine and vehicle speed, and gear selection. In addition, it contains measurements of the distance to the preceding car (in meters) as well as environmental telemetry related to track conditions.
IMS. The IMS dataset [110] was collected using an endurance test rig. Four bearings were mounted on a shaft driven by an AC motor and operated continuously until the failure of one of the bearings.
METROPT-3. The METROPT-3 dataset [208] spans operational data from a metro train. It includes 15 signals recorded over six months from multiple analog and digital sensors installed on the train's compressor air production unit.
NAVARCHOS. The NAVARCHOS dataset [65] contains one year of vehicle operational telemetry and associated service records. It covers data from 40 vehicles; for each vehicle, six distinct signals were recorded at one-minute intervals during operation. Service logs are available for 26 of the 40 vehicles. The dataset also exhibits periods of missing data, reflecting common data-availability challenges in real-world PdM applications.
XJTU-SY. The XJTU-SY dataset [211] reports 15 run-to-failure experiments generated by accelerated bearing tests. It spans three operating conditions and documents several failure modes, notably inner-race wear and outer-race fracture.

4.4 Implementation Details

Main experiments were conducted on an Ubuntu 22.04.3 LTS server equipped with a 16-core 5.2 GHz CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 3090 24GB, while adversarial robustness benchmarks were executed on an HPC Slurm cluster (with 560 cores and 16 NVIDIA A100 40GB GPUs). The software environment utilized the Anaconda distribution with Python 3.9.7. We ported and modified all TSAD techniques into a unified API. DL approaches were implemented using the PyTorch [11] framework. Depending on the algorithm, original codebases were extended to support multivariate data (if needed) or re-implemented from scratch. To align with realistic computational budgets, we evaluated up to 50 distinct hyperparameter configurations for each method–flavor pair, subject to a maximum runtime of one week. A single, global parameterization was applied across all assets of a dataset.

5 EXPERIMENTAL RESULTS AND ANALYSIS

In this section we report and analyze the results of our study. We first summarize the overall accuracy (Section 5.1), runtime performance (Section 5.2) and robustness (Section 5.3) of TSAD methods. Then, we evaluate the relative difficulty of the chosen datasets (Section 5.4), and perform

an in-depth analysis on a real-world PdM case (Section 5.5). Overall, the study is designed to provide answers to the following research questions (**RQ**):
RQ1. Is there a dominant TSAD technique and flavor to employ for PdM, and how do different problem instances benefit from particular flavor choices?
RQ2. What are the empirical effectiveness-runtime trade-offs for TSAD methods when applied to PdM cases?
RQ3. How robust are TSAD for PdM solutions to hyperparameters?
RQ4. How do domain requirements and attributes influence maintenance and, critically, are there identifiable early warning states prior to failures (manifesting as anomalies) that make (semi/un)-supervised TSAD methods effective? If so, how do these warning states typically manifest?

5.1 Overall Effectiveness

Table 5. Experimental results for the AD1 AUC-PR metric per dataset. The green line represents the mean and the orange line the median. The ORACLE represents the best performance achieved per dataset based on all flavors (excluding *ADD* and *ALL*).

Flavor	Method	AZURE	BHD 2023	CMAPSS	CNC	EDP-WT	FEMTO	FORMULA 1	IMS	METROPT-3	NAVARCHOS	XJTU-SY
Online	CHRONOS			out of time								
	IF											
	KNN											
	LOF											
	LTSF											
	NP							out of memory				
	OCSVM											
	PB											
	TRANAD											
	USAD											
Sliding	IF											
	KNN											
	LOF											
	LTSF											
	NP											
	OCSVM											
	PB											
	TRANAD											
	USAD											
Unsupervised	CHRONOS											
	IF											
	KNN											
	LOF											
	NP											
	SAND											
Other	XGBOOST											
	ORACLE	0.488	0.525	0.881	0.743	0.988	0.996	0.548	1.000	0.846	0.519	0.765
	ADD	0.070	0.050	0.160	0.074	0.670	0.210	0.248	0.350	0.290	0.010	0.110
	ALL	0.070	0.010	0.050	0.023	0.320	0.040	0.091	0.030	0.040	0.040	0.020

Table 5 presents boxplots corresponding to different hyperparameters for a given technique-dataset–flavor combination, computed using the AD1 AUC-PR metric, while experimental results for evaluation with VUS-PR can be found in Table B.5 of the Appendix.[1] We include two baselines for completeness: *ALL*, which issues an alert for every timestamp (yielding perfect recall but a maximum false alarm rate), and *ADD*, which implements a standard preventive maintenance policy [109, 202] that triggers intervention once an asset surpasses a fixed count of operational cycles. Including these baselines provides a practical reference for assessing whether PdM methods deliver meaningful improvements for operational decision-making. The ORACLE indicates the best performance achieved per dataset (excluding *ADD* and *ALL*). For readability, we adopt the following notation to streamline the analysis of flavor effects (see Table 2). Methods LOF, KNN, NP, IF, and CHRONOS support both semi-supervised and unsupervised operation and are denoted

[1]Further, results for datasets that contain historical healthy records and enable the historical flavor are presented in Tables B.3 and B.4 of the Appendix.

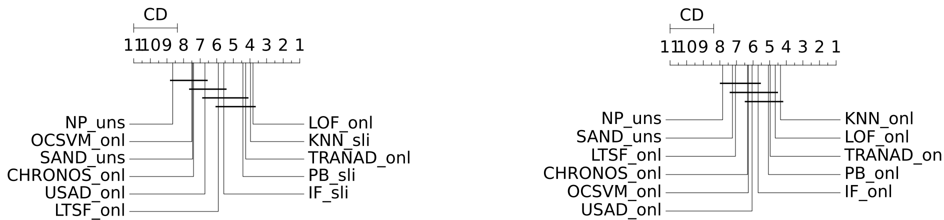
fl_{any}. SAND is limited to unsupervised mode, denoted fl_{uns}, whereas PB, TRANAD, LSTF, USAD, and OCSVM are applicable only in a semi-supervised manner, denoted fl_{semi}.

To provide a holistic view, Table 5 additionally reports the performance of a prevalent algorithm for PdM [62, 91, 120, 121, 188, 190, 207], XGBOOST [42], trained in a supervised classification setting. Supervised algorithms face several practical limitations in PdM contexts: i) labeled failure data are typically rare and insufficient to train robust supervised models; ii) even when historical data exist, obtaining reliable annotations is difficult [228]; iii) collected data may be heterogeneous across assets (e.g., IMS, BHD, CMAPSS), with mismatches in the number and semantics of recorded dimensions; if data cannot be paired for training/testing or the asset provenance is unknown, their utility for supervised learning is limited; and iv) operational dynamics in industrial environments may render historical training data unrepresentative of current normality.

Technique level. We adopt a multifaceted approach, beginning with an analysis of the median of median AD1 AUC-PR for each technique–dataset–flavor combination across datasets (detailed results can be found in Table B.6 of the Appendix), excluding the historical flavor. Subsequent sections provide complementary perspectives that converge on the same methods. KNN, PB, and TRANAD occupy the top five positions. KNN attains median AD1 AUC-PRs of 0.369 (sliding), PB (online) is second with a median of 0.288, and KNN (online) achieves 0.279, while TRANAD records medians of 0.272 with both the online and the sliding flavor. Performance varies substantially across datasets. For example, OCSVM's best AD1 AUC-PRs on CMAPSS and NAVARCHOS are 0.595 and 0.519 (Figure B.6 of the Appendix), respectively, while LTSF attains 0.108 and 0.334 on those datasets. Conversely, on XJTU-SY LTSF achieves the highest performance (0.765), whereas OCSVM's best is 0.335. On the other hand, XGBOOST empirically attains competitive best AD1 AUC–PR results in only two cases: CMAPSS (0.567) and IMS (0.687). CMAPSS is a synthetic dataset with highly constrained conditions and numerous simulations, which favors supervision. We note an important caveat: because supervision requires an explicit train/test split, the XGBOOST results are not directly comparable to the TSAD evaluations (e.g., in IMS XGBOOST was evaluated on one time-series instead of three). This protocol can bias results in favor of supervised methods, yet XGBOOST still performs poorly under few-shot failure scenarios, consistent with prior work [66].

For an unbiased comparison of TSAD techniques, we select each technique–flavor pair as follows. First, for each dataset and flavor, we retain the parameterization yielding the highest performance. Next, among the available flavors of each technique, we select the one with the highest average rank across datasets. In the event of ties, the flavor with the higher median performance is chosen. For each selected pair, we compute the minimum, median, and maximum performance values per dataset to represent worst-case, typical, and peak potential performance, respectively. This procedure produces three performance axes per dataset against which techniques are compared. To assess statistical differences across methods, we apply the Friedman–Nemenyi [51, 61] test to derive average ranks for all method–flavor pairs over the 33 resulting axes, visualized through critical difference (CD) diagrams. In the results, presented in Figure 3, a solid line in each CD diagram connects groups of methods without statistically significant differences (also Figures B.3 and B.4 of the Appendix present a flavor-level view).

Building on the consistency of the XGBOOST findings with prior art [66], we contrast our results with the broader TSAD literature. KNN achieves higher average ranks under VUS-PR in our study (Figure 3b), which agrees with reports of TSAD on multivariate datasets [176], yet it (together with LOF) contradicts earlier multivariate TSAD results [132]. We attribute these discrepancies primarily to differences in hyperparameter search spaces, dataset selection, and evaluation protocols across the two TSAD studies. Conversely, TRANAD demonstrates high competitiveness in our evaluation, challenging the lower performance ceilings reported in traditional TSAD benchmarks [132, 176]. Furthermore, the ranking hierarchy among LOF, KNN, TRANAD, and USAD under VUS-PR shifts



(a) AD1 AUC-PR

(b) VUS-PR

Fig. 3. Critical difference diagrams ($a = 0.05$) for the best (on average) technique-flavor pairs (excluding the historical flavor). Smaller numbers represent higher rankings.

notably compared to previous analyses [132]. TRANAD's resilience aligns with recent reports on robustness against domain shifts [88], which, in our experiments, are analogous to operating-mode switches that methods must handle without inputs of other types (see Section 5.5). LOF has been reported as a strong technique in a recent proprietary PdM case study [67]. TRANAD and LTSF are the only DL methods whose AD1 AUC-PR scores do not differ statistically from alternatives of the P category. Although prior work found no broad statistical gap between LLMs and classical methods [132, 176], our results highlight that CHRONOS performs significantly worse than several established techniques (LOF, KNN, PB) and TRANAD under the AD1 AUC-PR metric.

Figure 3 highlights a distinct behavioral shift: transitioning the evaluation metric from AD1 AUC-PR to VUS-PR redirects the preferred flavor for three P methods (KNN, PB, IF) from sliding to online, whereas LOF and TRANAD retain online as the flavor with the higher average rank. We explain this shift by the differing score statistics induced by the two flavors. For KNN and PB, the online flavor produces larger variance in anomaly scores than the sliding flavor. Machine condition typically evolves smoothly and samples lie on a low-dimensional manifold; a sliding window re-centers the baseline on recent states, so a newly emerging (potentially faulty) regime becomes the local "normal" and scores remain stable. In contrast, the online flavor compares current observations against a static history, causing minor deviations to manifest as score spikes that penalize precision across many thresholds. IF exhibits the same qualitative behaviour despite its bounded scoring: trees constructed from a recent sliding window yield more consistent depths for incoming points, reducing score dispersion and stabilizing precision. VUS-PR fundamentally redefines what we consider an early indicator, and thus score excursions that would be penalized under AD1 AUC-PR are effectively counted as early true positives.

In the case of LOF, the sliding window approach renders the algorithm hypersensitive to subtle density fluctuations—such as those introduced by incipient fault data—thereby increasing the false alarm rate. Conversely, the online flavor maintains a more stable global density estimate, producing smoother anomaly scores that effectively isolate the faulty condition near failure, typically amplifying the anomaly score relative to earlier fluctuations. In several assets, we observed that LOF's score increase occurred too late (i.e., within the L time), a behaviour we also encountered for TIME-LLM in preliminary experiments [94]. TRANAD behaves better under the online flavor more consistently, although its sliding-flavor performance is only marginally worse (see Figure B.6 in the Appendix). We ascribe this stability to its model-agnostic meta-learning (MAML) mechanism and adversarial training, which enable the model to infer the manifold of normality from the initial training, effectively obviating the need for frequent retraining. This challenges the prevailing assumption that retraining consistently yields performance gains. For instance, while [175] reported that utilizing temporally distant data (analogous to our online flavor) degraded accuracy in a specific edge-to-cloud TSAD context, our results align more closely with [37], which demonstrates TRANAD's ability to improve even with limited exposure to new distributions, and [170], which discusses TRANAD's ability to maintain stable detection performance under recurring normal patterns with varying frequencies. Consequently, while continuous weight adaptation remains a promising avenue for future PdM research, our findings suggest TRANAD already captures the core data manifold. Synthesizing these insights can help us define the state-of-the-art: methods that occupy the top clique of the CD diagrams and also exhibit superior secondary characteristics, which we analyze in the following sections.

Dataset level. At the dataset level, the online flavor appears most appropriate for AZURE. CNC, EDP-WT, FEMTO, IMS, METROPT-3 and XJTU-SY exhibit negligible sensitivity to flavor choice, such that the selection of technique becomes the primary determinant of performance. Although prior TSAD work indicates LLMs can capture general temporal patterns and perform well on trend- and shift-dominated data [176], we observed no accuracy improvement on FEMTO/XJTU-SY, which exhibit such characteristics. For BHD 2023, only SAND and NP achieve nontrivial detection performance, with maximum AD1 AUC-PRs of 0.219 and 0.525, respectively. Performance on CMAPSS and FORMULA 1 depends on an interaction between flavor and technique. Finally, NAVARCHOS benefits from semi-supervision: techniques generally perform better under semi-supervised flavors—for example, NP's best result increases by 0.308 (from 0.055 in fully unsupervised batching mode to 0.363 when using a sliding reference set). We extend the analysis to VUS-based evaluation for datasets without a fixed PH (CMAPSS, FEMTO, and IMS). SAND achieves the highest performance on both CMAPSS and FEMTO, consistent with the AD1 AUC-PR results, attaining VUS-PR scores of 0.625 and 0.743, respectively. In the IMS dataset, four techniques reach the maximum AD1 AUC-PR value of 1. Among these, IF and USAD exhibit the best overall performance with varying PHs, achieving VUS-PR scores of 0.806 and 0.734, respectively.

Flavor level. Considering the best parameterizations, KNN and CHRONOS (fl_{any}) achieve better performance when applied with the online flavor on the CMAPSS dataset. Similarly, semi-supervised methods (PB, LTSF and OCSVM; fl_{semi}) preferentially adopt the online flavor over sliding, whereas TRANAD is an exception, performing marginally better with the sliding flavor. This behaviour is explained by the characteristics of the CMAPSS dataset: it is a run-to-failure corpus that exhibits a consistent degradation trajectory (18% mean absolute percentage change across scenarios). By contrast, NAVARCHOS is subject to strong concept drift (e.g., weather and driver behavior); here, the sliding flavor—by using a more temporally local reference set—provides more relevant context. In the BHD 2023 dataset, records are similar at the beginning and end of asset lifecycles; SAND and NP therefore perform best because they exploit temporal record sequences rather than individual observations. For the historical flavor, we perform a separate analysis, as its applicability is dataset-dependent. Only CMAPSS and FEMTO include healthy operational data, enabling the use of this flavor. In CMAPSS, the historical flavor improves the performance of LOF and USAD, while in FEMTO, it enhances CHRONOS, PB, and TRANAD. Despite the availability of healthy data, other flavors are superior, and the overall best performance in both cases is achieved without using the historical flavor. Pareto frontiers further detail the power of the online flavor (Figure B.2 of the Appendix).

RQ1 Answer. Our analysis reveals substantial variability in the relative performance of TSAD techniques across datasets, highlighting the absence of a universal, out-of-the-box solution. Flavor selection is case dependent and should align with the underlying temporal characteristics of the data—such as monotonic degradation or nonstationary concept drift. Despite this variability, proximity-based methods (such as KNN and LOF) consistently rank among the top-performing detectors with the appropriate flavor. Moreover, the online flavor demonstrates strong performance, particularly when combined with proximity-based approaches or with TRANAD.

5.2 Runtime Analysis

Figure 4 reports execution time and normalized AD1 AUC–PR, grouped by flavor and dataset. For each configuration, the normalized AD1 AUC–PR is computed by dividing its AD1 AUC–PR by the maximum AD1 AUC–PR observed within the same dataset. For visual clarity, combinations with more than 30 points are sorted twice—once by AD1 AUC–PR and once by execution time—and each ordering is partitioned into 10 approximately equal-sized bins; the median point of each bin is retained. Moreover, for every combination, we always preserve the configurations with the maximum and minimum AD1 AUC-PR/execution time. The KDE and histogram panels that depict runtime distributions are computed over the complete set of experimental runs[2].

Fig. 4. Log_{10}-scale execution time and normalized AD1 AUC–PR, grouped by flavor and dataset.

[2]Results for the datasets that enable the historical flavor are presented in Figure B.1 of the Appendix.

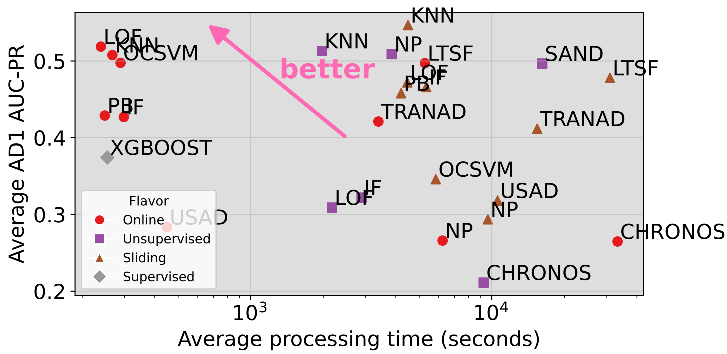


Fig. 5. The relationship between the average AD1 AUC-PR and processing time across datasets.

In the online flavor, TRANAD is substantially slower than other well-performing methods (approximately by one to two orders of magnitude), while LTSF also consistently exhibits high runtime across datasets. In contrast, KNN, LOF, IF and OCSVM are typically faster while preserving competitive detection performance. In the historical flavor, analogous performance patterns are observed for TRANAD, KNN, LOF, and IF. The sliding flavor exhibits greater variability in runtime (vertical axis), primarily due to window-size selection: smaller windows increase the number of method invocations and thus runtime. Overall, patterns similar to the online flavor persist. NP attains competitive performance on roughly half of the datasets but incurs higher execution times. PB, together with KNN and OCSVM, offers a favorable trade-off between effectiveness and computational cost. KNN exhibits superior performance in the unsupervised flavor and provides an efficient approach for computing anomaly scores; IF and LOF also perform well on several datasets. NP and SAND can yield competitive if not top AD1 AUC-PR but suffer from runtimes at least an order of magnitude higher. Similarly, CHRONOS is up to two orders of magnitude slower, corroborating [132, 176]. These inferior runtimes stem, among others, from the methods' limitation to univariate input.

Figure 5 illustrates the trade-off between mean AD1 AUC–PR and average total runtime across datasets. For each configuration, we use the hyperparameters with the highest AD1 AUC–PR, utilizing runtime as a tie-breaker. This visualization can guide model selection under operational constraints, revealing that high performance need not compromise speed. It also shows an opportunity for novel sequence-based architectures to occupy the currently sparse region defined by intermediate runtimes and moderately superior accuracy. The results highlight the efficiency of online variants of LOF, KNN, PB, IF, and OCSVM. Notably, LOF (online) achieves 94.8% of the best average AD1 AUC–PR while incurring only 5.3% of the average computational cost.

To validate statistical stability, Figure 6 presents CD diagrams stratified by flavor (we utilize the flavor-level view because the online flavor has the lowest overhead; interested readers can find global-level results in Figure B.5 of the Appendix). For each technique–dataset–flavor combination, we compute the minimum, median, and maximum of execution time, yielding 33 comparison axes per flavor in the general case (six axes for the historical flavor). We then apply the Friedman–Nemenyi test independently for each flavor. As depicted in Figure 6, LOF, KNN, PB, IF, and OCSVM under the online flavor constitute a group with significantly superior computational performance (in terms of runtime) compared with the remaining techniques. Under fully unsupervised settings, KNN, LOF, and IF are significantly faster than the other detectors.

RQ2 Answer. LOF, KNN, PB, IF, and OCSVM exhibit the best effectiveness-runtime trade-offs.

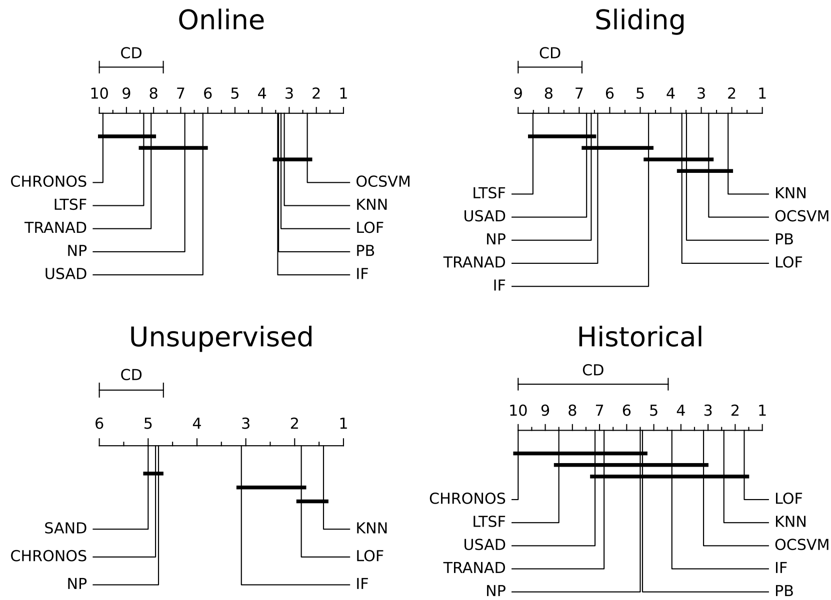


Fig. 6. Critical difference diagrams ($a = 0.05$) based on the execution time of TSAD techniques, grouped by flavor. Smaller numbers represent higher rankings.

5.3 Robustness Analysis

We first analyze results at the dataset and flavor level. In the online flavor (Figure 4), TRANAD is relatively insensitive to hyperparameter choices in most cases, while LTSF shows marked sensitivity to hyperparameter selection. PB displays some variability in performance yet is generally robust. USAD and NP are also sensitive to parameterization and frequently underperform across many datasets. Fine-tuning CHRONOS yields only marginal gains, which we attribute to CHRONOS's large model capacity being ill-suited to few-shot failure scenarios. For the VUS–PR metric, OCSVM paired with the online flavor exhibits large per-dataset variance; this behavior is explained by the RBF kernel's superior ability to project data points into a higher-dimensional feature space where they become more easily separable, while alternative kernels deliver inferior results. With a suitable configuration, OCSVM can accommodate varying PHs. NP again demonstrates hyperparameter-sensitivity. Counterintuitively, XGBOOST displays variance even on some of the easier cases w.r.t AD1 AUC-PR (see Section 5.4).

We next discuss the worst-case gap to ORACLE across datasets (details in Tables B.8–B.10 and B.12–B.14 of the Appendix). In all semi-supervised flavors, TRANAD, KNN, PB, LOF, and IF achieve the smallest average worst-case gap to ORACLE for both AD1 AUC-PR and VUS-PR. While we note a permutation in method rankings between the two metrics in the unsupervised flavor, the inter-method performance disparity is markedly reduced. XGBOOST generally underperforms, exhibiting an average worst-case gap to ORACLE of 0.678 for AD1 AUC-PR and 0.456 for VUS-PR.
RQ3 Answer. Overall, our analysis indicates that TRANAD is the most robust and consistently effective method at a dataset and flavor level. LOF, KNN, PB, and IF also exhibit strong robustness (especially w.r.t runtime), although achieving optimized performance with these methods typically requires careful hyperparameter selection.

5.4 Case Requirements and Complexity

We analyze dataset complexity using PH_r, defined as the fraction of records that lie within the PH, where the detailed results can be found at Table B.7 of the Appendix. NAVARCHOS exhibits a relatively low $PH_r = 0.035$, which constrains the temporal window within which anomalies must occur prior to failure, thereby increasing task difficulty. By contrast, EDP-WT has a high $PH_r = 0.33$, providing a larger interval for anomalies to manifest, which facilitates higher achievable performance. Although both datasets originate from real-world settings, differences in the domain expert requirements materially influence the complexity of the PdM task. For the comparative analysis reported below, and to ensure fairness across methods, we represent each technique–flavor pair by the parameterization with the median performance on each dataset. Employing a Bradley–Terry model [148] under a maximum-likelihood estimation framework, we then produce an elo-inspired scoring system of datasets from pairwise comparisons: dataset i is judged to "win" against dataset j when a given method–flavor combination attains a higher AD1 AUC–PR on i than on j. The resulting ordering (Table B.7 of the Appendix) provides a data-driven characterization of relative dataset difficulty and whether datasets contain early-warning anomalies. The kinds of methods that achieve competitive top performance (Figure B.6 of the Appendix) further inform the likely form of those anomalies (point- vs. sequence-based).

The ranking indicates that two of the three experimental datasets (FEMTO and IMS) are easier (i.e., ranked higher) than XJTU-SY, suggesting that more observable wear precedes failures in FEMTO and IMS, whereas XJTU-SY contains preceding anomalies to a lesser extent—possibly because different sensors convey less information about the underlying physical state. Synthetic datasets occupy the middle ranks, implying that assumptions made during dataset synthesis strongly affect detectability. On CMAPSS, SAND, NP, and OCSVM top AD1 AUC-PR scores (0.595–0.881) show early indicators exist, but finding the optimal configuration to capture such patterns remains the challenge. Because SAND attains leading performance across both metrics, we interpret the detectable signs in CMAPSS predominantly as sequence-based anomalies. Across real-world datasets the presence and form of early signs vary: EDP–WT and FORMULA 1 exhibit both point- and sequence-based anomalies (with FORMULA 1 also containing some hard cases with no preceding signs), METROPT-3 is biased toward point- and CNC toward sequence-based anomalies, and BHD 2023 contains little sequence-based anomalies prior to failure. NAVARCHOS is characterized by point-based anomalies but is further complicated by concept drift and data gaps, which hinder anomaly capture by TSAD methods.

We next evaluate the existence of anomalies prior failures and the practical value of TSAD for PdM within the general context of industrial maintenance, by comparing it against the *ADD* policy. For a fair comparison, we contrast both the median and the best AD1 AUC-PR performance achieved by each TSAD technique (each employed with the best flavor per dataset) against *ADD* through pairwise comparison, where the detailed results can be found at Table B.16 of the Appendix. Only LOF and PB achieve a median performance statistically better than *ADD*, whereas SAND, although it achieves statistically worse median performance, is the only one to win in all 11 cases with the best parameterization. Finally, CHRONOS is the only TSAD method whose best performance is not statistically better than the *ADD* policy.

RQ4 Answer. The complexity of industrial maintenance is influenced by domain-specific requirements—such as the PH and system attributes like IoT network Quality of Service (e.g., data gaps)—as well as by the presence of early-warning anomalies within the PH. Although higher PH_r values typically facilitate prediction—as seen in EDP–WT, this does not always hold. For example, BHD 2023 is the most challenging dataset, even with extended PH under VUS-PR, due to weak or absent pre-failure anomalies. In contrast, FEMTO and IMS, despite relatively low PH_r values,

Table 6. Best AD1 AUC–PR performance per failure type for each technique on the EDP–WT dataset, including log excerpts for a generator failure. The maximum performance for each failure type is highlighted in bold.

Technique	Avg AD1 AUC-PR	AD1 AUC-PR performance for each failure type					Generator Failure Logs	
		Transformer fan damaged	Gearbox bearings damaged	Oil leakage in Hub	Generator damaged	Hydraulic group error in the brake circuit	Timestamp	Description
CHRONOS	0.36	0.16	**0.50**	0.60	0.28	0.24	09:00:21	Pause -1.9kW 0RPM
IF	**0.54**	0.36	**0.50**	**0.83**	**0.50**	0.50	09:03:18	Emergency circuit open
KNN	0.46	0.21	**0.50**	**0.83**	**0.50**	0.26	09:03:20	Emergency -4.7kW 0RPM
LOF	0.45	0.24	**0.50**	0.82	**0.50**	0.19	09:52:07	Pause pressed on keyboard
LTSF	0.41	0.27	**0.50**	0.17	**0.50**	0.63	09:52:08	Pause -4.7kW 0RPM
NP	0.43	0.19	**0.50**	0.26	**0.50**	0.71	09:54:36	Stop -11.3kW 22RPM
OCSVM	0.47	0.12	**0.50**	**0.83**	**0.50**	0.42	10:00:42	Pause pressed on keyboard
PB	0.46	0.22	**0.50**	**0.83**	**0.50**	0.23	10:00:43	Pause -11.3kW 22RPM
SAND	0.50	0.14	**0.50**	0.65	**0.50**	0.70	10:01:04	Stop -2.6kW 46RPM
TRANAD	0.42	**0.50**	**0.50**	0.26	**0.50**	0.35	10:26:44	Pause -2.6kW 46RPM
USAD	0.42	0.26	**0.50**	0.10	**0.50**	**0.75**	10:26:44	Pause pressed on keyboard

are easier to handle because of clear and consistent degradation patterns. More complex datasets, such as NAVARCHOS, AZURE, and FORMULA 1, demonstrate greater difficulty, driven by dynamic operating conditions, sparse anomaly occurrences, and short PHs. Importantly, the presence of anomalies across datasets supports the practical relevance of TSAD for PdM, as its statistically significant improvements over preventive maintenance demonstrate that these anomalies can be effectively leveraged to improve industrial maintenance.

5.5 A Multifaceted Real-World Case Study

Predictive Capability. Table 6 reports the AD1 AUC–PR performance for each failure type, based on the best parameterization and flavor evaluated for each technique. Most methods achieve average AD1 AUC–PR values between 0.4 and 0.5, reflecting their ability to capture different aspects of system dynamics; strong performance on a single failure type tends to elevate the overall average. IF attains the highest average performance, performing exceptionally well on one failure type while maintaining reasonable performance on the others. TRANAD and USAD demonstrate an ability to model complex relationships, achieving the best performance on two specific failure types. In experimental conditions, bearing-related failures appear easier to predict, whereas in real-world scenarios they are more challenging. These observations underscore the potential value of ensemble methods for handling a broader spectrum of failure types. The bounded performance observed across certain failure types can be explained by lack of data fusion: log traces indicate that generator failures are likely associated with electrical issues and multiple human interventions prior to failure—events that are not represented in the time-series channels supplied to the detectors. Finally, CHRONOS shows no bias toward specific failure types, consistent with prior research on diverse anomalies [176].

Explainability. Model-agnostic methods, such as SHAP [136], have attracted significant attention [46, 73, 74] due to their applicability across diverse architectures. Figure 7 illustrates the application of SHAP to *IF* predictions, presenting both local explanations (feature importance for individual anomaly scores) and global explanations, where the top 10 influential features are retained. Blue and red colors indicate features with decreasing or increasing effects on the anomaly score, respectively. Examining the local explanations in Figure 7a, we can observe distinct temporal dynamics: feature

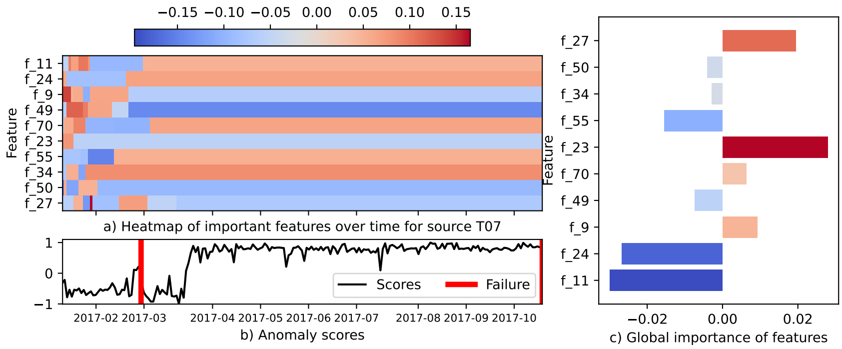


Fig. 7. Example of local/global explanations using SHAP.

f27 undergoes a polarity shift—from negative to positive—immediately before the first failure, matching with a rise in the anomaly score, whereas feature importance remains stable during the high-score period preceding the second failure. On a macro level, Figure 7c depicts global importance, quantifying the cumulative impact of specific features across the timeline of the T07 wind turbine.

Adversarial Robustness. We evaluate the best model configurations on EDP-WT against out-of-order attacks [72, 237] and noise injection [116] (Figure 8). The former simulates temporal corruption via *Max Lag* (the maximum position a point can be swapped: 6, 12, and 24 hours) and *Prob* (swap probability: 0.1, 0.5, 0.75); the latter injects zero-mean Gaussian noise of varying standard deviation, while dashed markers indicate raw data performance. Results reveal a structural dichotomy: sequence-based architectures (e.g., LTSF) degrade significantly as temporal coherence is eroded, whereas point-based methods remain robust. KNN proves the most resilient, due to the order-invariant distance-based decision rule, while IF shows moderate sensitivity as reordering alters tree-construction subsamples. Under noise injection, TRANAD and LTSF (alongside NP) outperform other sequence-based methods, implying robust latent feature learning. In contrast, IF and OCSVM prove significantly more brittle than their point-based counterparts. Notably, KNN and LOF reaffirm their status as robust, high-performing solutions, consistent with our broader analysis of secondary characteristics throughout this work. We also note instances where noise paradoxically aids detection, likely via noise-induced regularization and enhanced separability in the feature space [3].

To mitigate the susceptibility of specific models to noise, we investigate randomized smoothing [45]. While standard randomized smoothing estimates the Gaussian mean, we employ median smoothing via Monte Carlo sampling to better handle outliers and variance [36, 43]. Furthermore, utilizing a denoising preprocessing step before the smoothing function has demonstrated efficacy in other domains [133, 183]. Tailoring this to online (semi/un)-supervised PdM, we introduce a causal moving average denoising filter (with window size 100) before anomaly scoring. We empirically examine IF, OCSVM, and TRANAD (see Appendix Table B.17) using 100 noisy dataset copies. The impact is method-dependent: OCSVM gains substantially under high-noise conditions (improving AD1 AUC-PR from 0.385 to 0.758), likely as smoothing restores the data manifold, pulling perturbed points back within the hypersphere boundary. Conversely, TRANAD performance degrades uniformly with denoising, suggesting the filter attenuates essential high-frequency temporal transients. IF exhibits a distinct accuracy-robustness tradeoff, where smoothing generally reduces performance except at extreme noise levels, where improvements are negligible. Although injecting a simple

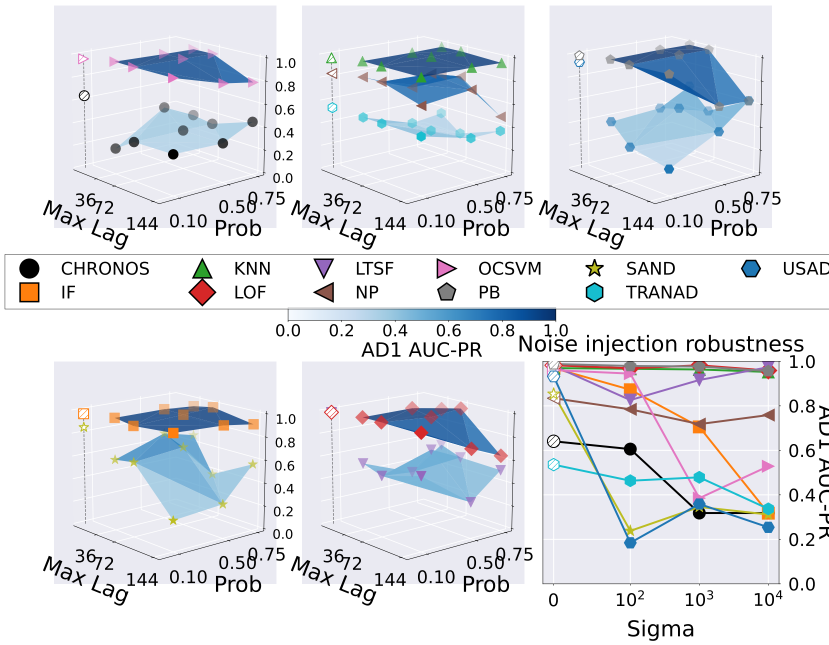


Fig. 8. The AD1 AUC-PR performance of TSAD methods after out-of-order and noise injection perturbations.

denoising layer, such as the moving average filter, appears promising in this case study, further research is required to fully delineate the impact and trade-offs of such preprocessing steps in the context of online PdM.

6 DISCUSSION AND FUTURE WORK

For a new use case, inter-dataset similarity measures can be employed [38, 49, 54, 146, 164, 166, 167, 177, 197], and our analysis provides insights into the behavior of TSAD techniques, as well as the most suitable flavor. In general, we recommend the online flavor, as it ranks among the top three in approximately half of the cases and is paired with most methods in our selection process (Figure 3a). Additionally, it imposes lower runtime overhead than other flavors (Figure 4). When experimentation time is limited and TRANAD's latency is acceptable, it is a robust choice at the technique level due to its reduced sensitivity to hyperparameters. For intensive applications, faster alternatives such as LOF, KNN, IF, or PB may be preferred, as they achieve higher average rankings, offer a favorable effectiveness-runtime trade-off, and exhibit the smallest average worst-case gap to ORACLE under all semi-supervised flavors. The sliding flavor is better suited to dynamic environments with varying operating modes and can be effectively combined with KNN, PB, or IF. Finally, we would like to note that results may vary with different hardware, preprocessing pipelines, and hyperparameter search spaces.

In many experiments, individual methods produced rising anomaly scores during the PH, indicating successful capture of degradation. However, variability in score magnitudes across assets prevents the use of a single global decision threshold and thus degrades final performance. For example, on CMAPSS (online flavor), KNN and LOF achieve AD1 AUC–PRs of 0.445 and 0.357, respectively; if their scores were post hoc min–max normalized (which requires access to future maximum and is therefore infeasible in online deployment), the corresponding AD1 AUC–PRs would increase to 0.808 and 0.759. This motivates further investigation of thresholding and post-processing strategies—such as adaptive or instance-wise normalization and calibration—to improve detector performance. Also, we observed that, in the online flavor, abrupt changes in an asset's input-data scale can cause the anomaly score to spike and effectively suppress the potential successful detection for other assets.

7 CONCLUSION

We investigate the application of (semi/un)-supervised TSAD methods for PdM in an online setting. Prior PdM studies typically evaluate a small set of competitors, consider single implementation strategies, and focus on specialised use cases, often neglecting runtime performance, generalisability, and robustness. To address these gaps and to spotlight the use of TSAD for PdM, we compiled 11 publicly available maintenance datasets from diverse domains and provided insights into the empirical behavior of TSAD algorithms. Our experimental study shows that established detectors—LOF, KNN, IF, and PB—when deployed with an initial warm-up period, frequently offer superior trade-offs between runtime efficiency and effectiveness across a range of contexts. In some cases, transformer-based approaches can better model complex latent relationships but do so at substantially greater computational cost, and pre-trained LLMs have not yet demonstrated competitive performance on this task. Overall, no single TSAD method proved universally optimal in terms of both accuracy and computational efficiency across datasets. An analysis of a real-world case calls for further research on ensemble techniques to accommodate the heterogeneous characteristics of different failure modes, and enhancing TSAD models' robustness to adversarial manipulations. This work therefore provides a foundation for more comprehensive research into TSAD for PdM.

Acknowledgments

We thank the anonymous reviewers for their insightful comments, which greatly improved the manuscript. Some results presented in this work have been produced using the Aristotle University of Thessaloniki (AUTh) High Performance Computing Infrastructure and Resources. Generative AI tools were used for polishing parts of the text and figures.

References

[1] 2026. The Power of Anomaly Detection in Predictive Maintenance. https://github.com/apapadoi/tsad-for-pdm.
[2] Ammar N Abbas, Georgios C Chasparis, and John D Kelleher. 2022. Interpretable input-output hidden Markov model-based deep reinforcement learning for the predictive maintenance of turbofan engines. In *Int'l Conf. on Big Data Analytics and Knowledge Discovery*. Springer, 133–148.
[3] Reyhaneh Abdolazimi, Shengmin Jin, Pramod K. Varshney, and Reza Zafarani. 2024. Harnessing the Power of Noise: A Survey of Techniques and Applications. arXiv:2410.06348 [cs.LG] https://arxiv.org/abs/2410.06348
[4] Ashraf Aboshosha, Ayman Haggag, Neseem George, and Hisham A Hamad. 2023. IoT-based data-driven predictive maintenance relying on fuzzy system and artificial neural networks. *Scientific Reports* 13, 1 (2023), 12186.
[5] Sulaiman Aburakhia, Tareq Tayeh, Ryan Myers, and Abdallah Shami. 2022. Similarity-based predictive maintenance framework for rotating machinery. In *2022 5th Int'l Conf. on Communications, Signal Processing, and their Applications (ICCSPA)*. IEEE, 1–6.
[6] Charu C Aggarwal. 2016. Proximity-based outlier detection. In *Outlier analysis*. Springer, 111–147.
[7] Steven Aliano. 2021. *World's Largest Manufacturers Lose $1 Trillion/Year to Machine Failure.* https://blog.isa.org/worlds-largest-manufacturers-lose-1-trillion/year-to-machine-failure Accessed: 2025-09-16.
[8] Sarah Alnegheimish, Laure Berti-Equille, and Kalyan Veeramachaneni. 2023. Making the end-user a priority in benchmarking: Orionbench for unsupervised time series anomaly detection. *arXiv preprint arXiv:2310.17748* (2023).
[9] Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. 2025. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821* (2025).
[10] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao

Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the Language of Time Series. *arXiv preprint arXiv:2403.07815* (2024).

[11] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. 2024. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM Int'l Conf. on Architectural Support for Programming Languages and Operating Systems, Vol. 2 (ASPLOS '24)*. ACM. doi:10.1145/3620665.3640366

[12] 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 Int'l Conf. on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. ACM, NY, USA, 3395–3404. doi:10.1145/3394486.3403392

[13] Backblaze. [n. d.]. Hard Drive Test Data. Backblaze Cloud Storage Resources. https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data Accessed: 2025-01-30.

[14] Tommaso Barbariol, Filippo Dalla Chiara, Davide Marcato, and Gian Antonio Susto. 2022. A review of tree-based approaches for anomaly detection. *Control charts and machine learning for anomaly detection in manufacturing* (2022), 149–185.

[15] 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.

[16] Ernnie Illyani Basri, Izatul Hamimi Abdul Razak, Hasnida Abdul Samat, and Shahrul Kamaruddin. 2017. Preventive Maintenance (PM) planning: a review. *Journal of Quality in Maintenance Engineering* 23 (05 2017). doi:10.1108/JQME-04-2016-0014

[17] Taha Belkhouja and Janardhan Rao Doppa. 2022. Adversarial framework with certified robustness for time-series domain via statistical features. *Journal of Artificial Intelligence Research* 73 (2022), 1435–1471.

[18] Taha Belkhouja, Yan Yan, and Janardhan Rao Doppa. 2023. Dynamic Time Warping Based Adversarial Framework for Time-Series Domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 6 (2023), 7353–7366. doi:10.1109/TPAMI.2022.3224754

[19] Steve Bennett. 2025. *Predictive Maintenance Software Statistics 2025.* https://llcbuddy.com/data/predictive-maintenance-software-statistics/#source-9 Accessed: 2025-09-16.

[20] Arnab Biswas. [n. d.]. Microsoft Azure Predictive Maintenance. Kaggle Dataset. https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance Accessed: 2025-06-01.

[21] 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.

[22] 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).

[23] 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. *Proc. VLDB Endow.* 15, 12 (2022), 3702–3705.

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

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

[26] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND in Action: Subsequence Anomaly Detection for Streams. *Proc. VLDB Endow.* 14, 12 (2021), 2867–2870.

[27] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J. Franklin. 2021. SAND: streaming subsequence anomaly detection. *Proc. VLDB Endow.* 14, 10 (jun 2021), 1717–1729. doi:10.14778/3467861.3467863

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

[29] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. *SIGMOD Rec.* 29, 2 (may 2000), 93–104. doi:10.1145/335191.335388

[30] Andrea Brunello, Dario Della Monica, Angelo Montanari, Nicola Saccomanno, and Andrea Urgolo. 2023. Monitors that learn from failures: Pairing STL and genetic programming. *IEEE Access* 11 (2023), 57349–57364.

[31] Yuanpu Cao, Lu Lin, and Jinghui Chen. 2024. Adversarially robust industrial anomaly detection through diffusion model. *arXiv preprint arXiv:2408.04839* (2024).

[32] Mattia Carletti, Matteo Terzi, and Gian Antonio Susto. 2023. Interpretable Anomaly Detection with DIFFI: Depth-based feature importance of Isolation Forest. *Engineering Applications of Artificial Intelligence* 119 (2023), 105730. doi:10.1016/j.engappai.2022.105730
[33] Jacinto Carrasco, David López, Ignacio Aguilera-Martos, Diego García-Gil, Irina Markova, Marta Garcia-Barzana, Manuel Arias-Rodil, Julián Luengo, and Francisco Herrera. 2021. Anomaly detection in predictive maintenance: A new evaluation framework for temporal unsupervised anomaly detection algorithms. *Neurocomputing* 462 (2021), 440–452.
[34] NASA Ames Research Center. [n. d.]. FEMTO Bearing Dataset. NASA Prognostics Data Repository, Moffett Field, CA.
[35] Sayan Chakraborty, Smit Shah, Kiumars Soltani, Anna Swigart, Luyao Yang, and Kyle Buckingham. 2020. Building an Automated and Self-Aware Anomaly Detection System. arXiv:2011.05047 [cs.LG] https://arxiv.org/abs/2011.05047
[36] Zakariya Chaouai and Mohamed Tamaazousti. 2024. Universal robustness via median randomized smoothing for real-world super-resolution. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 9059–9068.
[37] Feiyi Chen, Zhen Qin, Mengchu Zhou, Yingying Zhang, Shuiguang Deng, Lunting Fan, Guansong Pang, and Qingsong Wen. 2024. Lara: A light and anti-overfitting retraining approach for unsupervised time series anomaly detection. In *Proceedings of the ACM Web Conf. 2024*. 4138–4149.
[38] Hongjie Chen, Akshay Mehra, Josh Kimball, and Ryan A. Rossi. 2025. Measuring Time-Series Dataset Similarity using Wasserstein Distance. arXiv:2507.22189 [cs.LG] https://arxiv.org/abs/2507.22189
[39] Jianjun Chen, Weihao Hu, Di Cao, Zhenyuan Zhang, Zhe Chen, and Frede Blaabjerg. 2023. A Meta-Learning Method for Electric Machine Bearing Fault Diagnosis Under Varying Working Conditions With Limited Data. *IEEE Transactions on Industrial Informatics* 19, 3 (2023), 2552–2564. doi:10.1109/TII.2022.3165027
[40] Jiao Chen, Ruyi Huang, Zuohong Lv, Jianhua Tang, and Weihua Li. 2025. Faultgpt: Industrial fault diagnosis question answering system by vision language models. *arXiv preprint arXiv:2502.15481* (2025).
[41] Run-Qing Chen, Guang-Hui Shi, Wan-Lei Zhao, and Chang-Hui Liang. 2021. A joint model for IT operation series prediction and anomaly detection. *Neurocomputing* 448 (2021), 130–139.
[42] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd int'l conf. on knowledge discovery and data mining*. 785–794.
[43] Ping-yeh Chiang, Michael Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. 2020. Detection as regression: Certified object detection with median smoothing. *Advances in Neural Information Processing Systems* 33 (2020), 1275–1286.
[44] Heejeong Choi, Donghwa Kim, Jounghee Kim, Jina Kim, and Pilsung Kang. 2022. Explainable anomaly detection framework for predictive maintenance in manufacturing systems. *Applied Soft Computing* 125 (2022), 109147.
[45] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *int'l conf. on machine learning*. PMLR, 1310–1320.
[46] Logan Cummins, Alexander Sommers, Somayeh Bakhtiari Ramezani, Sudip Mittal, Joseph Jabour, Maria Seale, and Shahram Rahimi. 2024. Explainable predictive maintenance: A survey of current methods, challenges and opportunities. *IEEE access* 12 (2024), 57574–57602.
[47] K. Deng, Y. Zhang, and H. Wang. 2022. A remaining useful life prediction method with automatic feature extraction for aircraft engines. *Journal of Aerospace Engineering* 35, 1 (2022), 04021001.
[48] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit Optimizers via Block-wise Quantization. *CoRR* abs/2110.02861 (2021). arXiv:2110.02861 https://arxiv.org/abs/2110.02861
[49] 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.
[50] Mouhamadou Saliou Diallo, Sid Ahmed Mokeddem, Agnès Braud, Gabriel Frey, and Nicolas Lachiche. 2021. Identifying benchmarks for failure prediction in industry 4.0. In *Informatics*, Vol. 8. MDPI, 68.
[51] Critchlow E Douglas and Fligner A Michael. 1991. On distribution-free multiple comparisons in the one-way analysis of variance. *Communications in Statistics-theory and Methods* 20, 1 (1991), 127–139.
[52] Xusheng Du and Jiong Yu. 2022. Graph neural network-based early bearing fault detection. *arXiv preprint arXiv:2204.11220* (2022).
[53] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. 2019. Band-limited training and inference for convolutional neural networks. In *Int'l Conf. on Machine Learning*. PMLR, 1745–1754.
[54] 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 Int'l Conf. on Data Engineering Workshops (ICDEW)*. IEEE, 107–112.
[55] Youssof Fassi, Vincent Heiries, Jerome Boutet, and Sebastien Boisseau. 2023. Toward physics-informed machine-learning-based predictive maintenance for power converters—a review. *IEEE Transactions on Power Electronics* 39, 2

(2023), 2692–2720.
[56] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. 2019. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems.. In *NDSS*. 1–15.
[57] Carlos Ferreira and Gil Gonçalves. 2022. Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. *Journal of manufacturing systems* 63, May (2022), 550–562.
[58] Fiix Software Community. n.d.. *A deep dive into predictive maintenance (Part 2): The four types*. https://community.fiixsoftware.com/supercharge-your-maintenance-strategy-24/a-deep-dive-into-predictive-maintenance-part-2-the-four-types-1122 Accessed June 2025.
[59] Vitaly Filimonov. 2015. Unsupervised Anomaly Detection for Arbitrary Time Series. https://patentimages.storage.googleapis.com/US20150269050A1.pdf
[60] Jonatan Flyckt, Tony Gorschek, Daniel Mendez, and Niklas Lavesson. 2025. Identifying key AI challenges in make-to-order manufacturing organisations: A multiple case study. *Journal of Systems and Software* 230 (2025), 112559. doi:10.1016/j.jss.2025.112559
[61] 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.
[62] Premanand Ghadekar, Aman Manakshe, Sarthak Madhikar, Sushrut Patil, Mehvish Mukadam, and Tejas Gambhir. 2024. Predictive maintenance for industrial equipment: Using XGBoost and local outlier factor with explainable AI for analysis. In *2024 14th Int'l Conf. on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 25–30.
[63] Yousof Gholipour. 2025. A Comprehensive Review of Maintenance Strategies: From Reactive to Proactive Approaches. *Available at SSRN 5349871* (2025).
[64] Apostolos Giannoulidis and Anastasios Gounaris. 2023. A context-aware unsupervised predictive maintenance solution for fleet management. *Journal of Intelligent Information Systems* 60, 2 (2023), 521–547.
[65] Apostolos Giannoulidis, Anastasios Gounaris, and Ioannis Constantinou. 2024. Exploring unsupervised anomaly detection for vehicle predictive maintenance with partial information. In *EDBT*. 753–761.
[66] Apostolos Giannoulidis, Anastasios Gounaris, Athanasios Naskos, Nikodimos Nikolaidis, and Daniel Caljouw. 2024. Engineering and evaluating an unsupervised predictive maintenance solution: a cold-forming press case-study. *Journal of Intelligent Manufacturing* (2024), 1–19.
[67] Apostolos Giannoulidis, Anastasios Gounaris, Athanasios Naskos, Nikodimos Nikolaidis, and Daniel Caljouw. 2025. Leveraging Feedback and Causality-Enriched Multimodal Context for Predictive Maintenance. *IEEE Access* 13 (2025), 132269–132286. doi:10.1109/ACCESS.2025.3592775
[68] Apostolos Giannoulidis, Anastasios Gounaris, Nikodimos Nikolaidis, Athanasios Naskos, and Daniel Caljouw. 2022. Investigating thresholding techniques in a real predictive maintenance scenario. *SIGKDD Explor.* 24, 2 (2022), 86–95. doi:10.1145/3575637.3575651
[69] 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.
[70] GII Research. 2024. IoT Sensors Market - Global Industry Analysis, Size, Share, Growth, Trends, and Forecast 2031. https://www.giiresearch.com/report/fmr1591714-iot-sensors-market-global-industry-analysis-size.html. Accessed: 2025-09-16.
[71] Dario Guidotti, Laura Pandolfo, and Luca Pulina. 2025. A systematic literature review of supervised machine learning techniques for predictive maintenance in Industry 4.0. *IEEE Access* (2025).
[72] Ammara Gul and Stephen D. Wolthusen. 2017. Measurement re-ordering attacks on power system state estimation. In *2017 IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)*. 1–6. doi:10.1109/ISGTEurope.2017.8260145
[73] Suchit Gupte and John Paparrizos. 2025. ShapX Engine: A Demonstration of Shapley Value Approximations. In *Companion of the 2025 Int'l Conf. on Management of Data*. 107–110.
[74] 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.
[75] Luke Hamer. 2025. *The Latest Statistics from the Maintenance Industry [2025]*. https://www.getsockeye.com/blog/maintenance-statistics/ Accessed: 2025-09-16.
[76] Yuanduo He, Xu Chu, and Yasha Wang. 2020. Neighbor Profile: Bagging Nearest Neighbors for Unsupervised Time Series Mining. In *2020 IEEE 36th Int'l Conf. on Data Engineering (ICDE)*. 373–384. doi:10.1109/ICDE48307.2020.00039
[77] Tarek Hegazy, Shipra Singh Ahluwalia, and Mohamed Attalla. 2010. Two condition indicators for building components based on reactive-maintenance data. *Journal of Facilities Management* 8, 1 (2010), 64–74.
[78] 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.
[79] Michaël Hoarau, Moulham Zahabi, and Fauzan K. 2023. Deploy a predictive maintenance solution for airport baggage handling systems with Amazon Lookout for Equipment. AWS Machine Learning

Blog. https://aws.amazon.com/blogs/machine-learning/deploy-a-predictive-maintenance-solution-for-airport-baggage-handling-systems-with-amazon-lookout-for-equipment/

[80] Axel Hochstein, Hyung-Il Ahn, Ying Tat Leung, and Matthew Denesuk. 2013. Survival analysis for HDLSS data with time dependent variables: Lessons from predictive maintenance at a mining service provider. In *Proceedings of 2013 IEEE Int'l Conf. on Service Operations and Logistics, and Informatics*. IEEE, 372–381.

[81] Olov Holmer, Erik Frisk, and Mattias Krysander. 2023. Energy-based survival models for predictive maintenance. *IFAC-PapersOnLine* 56, 2 (2023), 10862–10867.

[82] 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 Int'l Conf. on Management of Data (SIGMOD Companion '26)*. doi:10.1145/3788853.3801594

[83] Mingyi Huang, Qinghua Liu, Paul Boniol, and John Paparrizos. 2026. HYDRA: A Multi-Level Hierarchy-Driven Approach for Robust Anomaly Detection in Time Series. *Proceedings of the ACM on Management of Data* 4, 3, Article 197 (June 2026), 30 pages. doi:10.1145/3802074

[84] Julio Hurtado, Dario Salvati, Rudy Semola, Mattia Bosio, and Vincenzo Lomonaco. 2023. Continual learning for predictive maintenance: Overview and challenges. *Intelligent Systems with Applications* 19 (2023), 200251.

[85] EDP Innovation. [n. d.]. Innovation — Data. https://www.edp.com/en/innovation/data Accessed: 2025-06-30.

[86] Global Market Insights. 2025. IoT Sensors Market Size & Share, Statistics Report 2025-2034. https://www.gminsights.com/industry-analysis/iot-sensors-market. Accessed: 2025-09-16.

[87] Mordor Intelligence. 2025. IoT Sensor Market - Size, Share & Forecast 2025-2030. https://www.mordorintelligence.com/industry-reports/iot-sensor-market. Accessed: 2025-09-16.

[88] Vincent Jacob and Yanlei Diao. 2025. Unsupervised Anomaly Detection in Multivariate Time Series across Heterogeneous Domains. *Proc. VLDB Endow.* 18, 6 (Feb. 2025), 1691–1704. doi:10.14778/3725688.3725699

[89] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. 2020. Exathlon: A benchmark for explainable anomaly detection over time series. *arXiv preprint arXiv:2010.05073* (2020).

[90] 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 Int'l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing*. IEEE, 107–114.

[91] Zhiyang Jia, Zhibo Xiao, and Yijin Shi. 2021. Remaining useful life prediction of equipment based on xgboost. In *Proceedings of the 5th Int'l Conf. on Computer Science and Application Engineering*. 1–6.

[92] 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.

[93] 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 Int'l Conf. on Management of Data*. 843–856.

[94] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-fang Li, Shirui Pan, et al. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *Int'l Conf. on Learning Representations*.

[95] Nicolas Jourdan, Sagar Sen, Erik Johannes Husom, Enrique Garcia-Ceja, Tobias Biegel, and Joachim Metternich. 2021. On the reliability of machine learning applications in manufacturing environments. *arXiv preprint arXiv:2112.06986* (2021).

[96] Eda Jovicic, Daria Primorac, Marko Cupic, and Alan Jovic. 2023. Publicly available datasets for predictive maintenance in the energy sector: A review. *IEEE access* 11 (2023), 73505–73520.

[97] Dovile Juodelyte, Veronika Cheplygina, Therese Graversen, and Philippe Bonnet. 2022. Predicting bearings degradation stages for predictive maintenance in the pharmaceutical industry. In *Proceedings of the 28th acm sigkdd conf. on knowledge discovery and data mining*. 3107–3115.

[98] Mehmet Karakose and Orhan Yaman. 2020. Complex fuzzy system based predictive maintenance approach in railways. *IEEE Transactions on industrial informatics* 16, 9 (2020), 6023–6032.

[99] Dohyeong Kim, Soyeon Caren Han, Yingru Lin, Byeong Ho Kang, and Sungyoung Lee. 2018. RDR-based knowledge based system to the failure detection in industrial cyber physical systems. *Knowledge-Based Systems* 150 (2018), 1–13.

[100] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. 2022. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI conf. on artificial intelligence*, Vol. 36. 7194–7201.

[101] B Ravi Kiran. 2017. Multi-scale streaming anomalies detection for time series. arXiv:1706.06910 [stat.AP] https://arxiv.org/abs/1706.06910

[102] Ranganath Kothamasu and Samuel H Huang. 2007. Adaptive Mamdani fuzzy model for condition-based maintenance. *Fuzzy sets and Systems* 158, 24 (2007), 2715–2733.

[103] 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.
[104] Eldar Kurtic, Alexandre Noll Marques, Shubhra Pandit, Mark Kurtz, and Dan Alistarh. 2025. "Give Me BF16 or Give Me Death"? Accuracy-Performance Trade-Offs in LLM Quantization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 26872–26886.
[105] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting time series outlier detection: Definitions and benchmarks. In *Thirty-fifth conf. on neural information processing systems datasets and benchmarks track (round 1).*
[106] Kausik Lakkaraju, John A Aydin, Biplav Srivastava, et al. 2025. On Identifying Why and When Foundation Models Perform Well on Time-Series Forecasting Using Automated Explanations and Rating. In *Proceedings of the AAAI Symp. Series*, Vol. 7. 273–282.
[107] Alexander Lavin and Subutai Ahmad. 2015. Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark. In *2015 IEEE 14th int'l conf. on machine learning and applications (ICMLA).* IEEE, 38–44.
[108] Minh-Huong Le-Nguyen, Fabien Turgis, Pierre-Emmanuel Fayemi, and Albert Bifet. 2023. Exploring the potentials of online machine learning for predictive maintenance: a case study in the railway industry. *Applied Intelligence* 53, 24 (2023), 29758–29780.
[109] LeanTeams. 2025. Barclay Chemicals — Lean Business Case Study 1: Preventive Maintenance. Case Study, LeanTeams. https://leanteams.com/barclay-chemicals-case-study-preventive-maintenance/ Accessed October 2025.
[110] Jay Lee, H Qiu, G Yu, J Lin, et al. 2007. IMS, University of Cincinnati. "Bearing Data Set", NASA Ames Prognostics Data Repository. *NASA Ames Research Center, Moffett Field, CA* (2007).
[111] 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
[112] Huan Li, Bo Tang, Hua Lu, Muhammad Aamir Cheema, and Christian S Jensen. 2022. Spatial data quality in the iot era: management and exploitation. In *Proceedings of the 2022 Int'l Conf. on Management of Data.* 2474–2482.
[113] Huiqin Li, Zhengxin Zhang, Tianmei Li, and Xiaosheng Si. 2024. A review on physics-informed data-driven remaining useful life prediction: Challenges and opportunities. *Mechanical systems and signal processing* 209 (2024), 111120.
[114] Jiale Li, Guohui Yin, Xuefei Wang, and Weixi Yan. 2022. Automated decision making in highway pavement preventive maintenance based on deep learning. *Automation in Construction* 135 (2022), 104111. doi:10.1016/j.autcon.2021.104111
[115] Lin Li, Hongchun Qu, Zhaoni Li, Jian Zheng, Xiaoming Tang, and Ping Liu. 2024. Data Reconstruction via Consensus Graph Learning for Effective Anomaly Detection in Industrial IoT. *IEEE Transactions on Industrial Informatics* 20, 3 (2024), 3996–4006. doi:10.1109/TII.2023.3316220
[116] Mingyuan Li, Jiahao Wang, Guangsheng Yu, Xu Wang, Qianrun Chen, Wei Ni, Lixiang Li, and Haipeng Peng. [n. d.]. RobustLight: Improving Robustness via Diffusion Reinforcement Learning for Traffic Signal Control. In *Forty-second Int'l Conf. on Machine Learning.*
[117] X. Li, W. Zhang, and Q. Ding. 2021. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliability Engineering & System Safety* 182 (2021), 208–218.
[118] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. 2023. A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2023), 1–54.
[119] Elad Liebman. 2024. Pattern-Based Time-Series Risk Scoring for Anomaly Detection and Alert Filtering–A Predictive Maintenance Case Study. *arXiv preprint arXiv:2405.17488* (2024).
[120] Soram Lim and Seokho Chi. 2019. Xgboost application on bridge management systems for proactive damage estimation. *Advanced Engineering Informatics* 41 (2019), 100922.
[121] Kuo-Yi Lin, Yu-He Hong, Meng-Hua Li, Yongjiang Shi, and Kotomichi Matsuno. 2025. Predictive maintenance in industrial systems: an XGBoost-based approach for failure time estimation and resource optimization. *Journal of Industrial and Production Engineering* (2025), 1–24.
[122] Biyu Liu, Jie Pang, Haidong Yang, and Yilin Zhao. 2024. Optimal condition-based maintenance policy for leased equipment considering hybrid preventive maintenance and periodic inspection. *Reliability Engineering & System Safety* 242 (2024), 109724. doi:10.1016/j.ress.2023.109724
[123] 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.
[124] Chunwei Liu, John Paparrizos, and Aaron J Elmore. 2024. AdaEdge: A Dynamic Compression Selection Framework for Resource Constrained Devices. In *2024 IEEE 40th Int'l Conf. on Data Engineering (ICDE).* IEEE, 1506–1519.
[125] Fuqiang Liu, Sicong Jiang, Luis Miranda-Moreno, Seongjin Choi, and Lijun Sun. 2024. Adversarial vulnerabilities in large language models for time series forecasting. *arXiv preprint arXiv:2412.08099* (2024).
[126] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *Proceedings of the 2008 Eighth IEEE Int'l Conf. on Data Mining (ICDM '08).* IEEE Computer Society, USA, 413–422. doi:10.1109/ICDM.2008.17
[127] Jinyang Liu, Wenwei Gu, Zhuangbin Chen, Yichen Li, Yuxin Su, and Michael R Lyu. 2024. MTAD: Tools and benchmarks for multivariate time series anomaly detection. *arXiv preprint arXiv:2401.06175* (2024).

[128] 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.

[129] 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).

[130] 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.

[131] Qinghua Liu, Seunghak Lee, and John Paparrizos. 2025. Tsb-autoad: Towards automated solutions for time-series anomaly detection. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4364–4379.

[132] 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.

[133] Shijie Liu, Tansu Alpcan, Christopher Leckie, and Sarah Monazam Erfani. 2025. Fortifying Time Series: DTW-Certified Robust Anomaly Detection. In *The Thirty-ninth Annual Conf. on Neural Information Processing Systems*.

[134] 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.

[135] Andreas Livera, Marios Theristis, Alexios Charalambous, Joshua S Stein, and George E Georghiou. 2021. Decision support system for corrective maintenance in large-scale photovoltaic systems. In *2021 IEEE 48th Photovoltaic Specialists Conf. (PVSC)*. IEEE, 0306–0311.

[136] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st Int. Con. on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[137] Arnav V. Malawade, Nathan D. Costa, Deepan Muthirayan, Pramod P. Khargonekar, and Mohammad A. Al Faruque. 2021. Neuroscience-Inspired Algorithms for the Predictive Maintenance of Manufacturing Systems. *IEEE Transactions on Industrial Informatics* 17, 12 (2021), 7980–7990. doi:10.1109/TII.2021.3062030

[138] Fernando Mateo, Joan Vila-Francés, Emilio Soria-Olivas, Marcelino Martínez-Sober, Juan Gómez-Sanchis, and Antonio José Serrano-López. 2025. Dynamic classifier auditing by unsupervised anomaly detection methods: an application in packaging industry predictive maintenance. *Applied Sciences* 15, 2 (2025), 882.

[139] Lukas Meitz, Julia Senge, Tim Wagenhals, Thorsten Schöler, Jörg Hähner, Janick Edinger, and Christian Krupitzer. 2025. A Literature Review Framework and Open Research Challenges for Predictive Maintenance in industry 4.0. *Computers & Industrial Engineering* 206 (2025), 111193. doi:10.1016/j.cie.2025.111193

[140] Nesryne Mejri, Laura Lopez-Fuentes, Kankana Roy, Pavel Chernakov, Enjie Ghorbel, and Djamila Aouada. 2024. Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods. *Expert Systems with Applications* 256 (2024), 124922. doi:10.1016/j.eswa.2024.124922

[141] Oana Merkt. 2019. On the use of predictive models for improving the quality of industrial maintenance: An analytical literature review of maintenance strategies. In *2019 Federated Conf. on Computer Science and Information Systems (FedCSIS)*. IEEE, 693–704.

[142] Grégory Mermoud. 2016. Warm-Start with Knowledge and Data-Based Grace Period for Live Anomaly Detection Systems. https://patentimages.storage.googleapis.com/78/1c/1f/81aa028de66c19/US20160352764A1.pdf

[143] Luís Manuel Meruje Ferreira, Fabio Coelho, and José Pereira. 2024. Databases in edge and fog environments: A survey. *Comput. Surveys* 56, 11 (2024), 1–40.

[144] Dang Minh, H. Xiang Wang, Y. Fen Li, and Tan N. Nguyen. 2022. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review* 55, 5 (01 Jun 2022), 3503–3568. doi:10.1007/s10462-021-10088-y

[145] MonoM. 2023. *Modelos no supervisados en mantenimiento predictivo*. https://monom.ai/en/modelos-no-supervisados-en-mantenimiento-predictivo/ Accessed June 2025.

[146] Michael D Morse and Jignesh M Patel. 2007. An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM SIGMOD int'l conf. on Management of data*. 569–580.

[147] Bruno Mota, Pedro Faria, and Carlos Ramos. 2022. Predictive maintenance for maintenance-effective manufacturing using machine learning approaches. In *Int'l Workshop on Soft Computing Models in Industrial and Environmental Applications*. Springer, 13–22.

[148] Sahand Negahban, Sewoong Oh, and Devavrat Shah. [n. d.]. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.). Curran Associates, Inc.

[149] Antonios Ntroumpogiannis, Michail Giannoulis, Nikolaos Myrtakis, Vassilis Christophides, Eric Simon, and Ioannis Tsamardinos. 2023. A meta-level analysis of online anomaly detectors. *The VLDB Journal* 32, 4 (2023), 845–886.

[150] JDMD Editorial Office, Nagi Gebraeel, Yaguo Lei, Naipeng Li, Xiaosheng Si, and Enrico Zio. 2023. Prognostics and Remaining Useful Life Prediction of Machinery: Advances, Opportunities and Challenges. *Journal of Dynamics, Monitoring and Diagnostics* 2, 1 (Feb. 2023), 1–12. doi:10.37965/jdmd.2023.148

[151] Demet Özgür-Ünlüakın, Busenur Türkali, Ayşe Karacaörenli, and S Çağlar Aksezer. 2019. A DBN based reactive maintenance model for a complex system in thermal power plants. *Reliability Engineering & System Safety* 190 (2019), 106505.

[152] Ioannis Paparrizos. 2018. *Fast, scalable, and accurate algorithms for time-series analysis.* Ph. D. Dissertation. Columbia University.

[153] Ioannis Paparrizos, Stylianos Basagiannis, and Sophia Petridou. 2011. Quantitative analysis for authentication of low-cost RFID tags. In *2011 IEEE 36th Conf. on Local Computer Networks*. IEEE, 295–298.

[154] Ioannis Paparrizos, Hoyoung Jeung, and Karl Aberer. 2011. Advanced search, visualization and tagging of sensor metadata. In *2011 IEEE 27th Int'l Conf. on Data Engineering*. IEEE, 1356–1359.

[155] John Paparrizos and Sai Prasanna Teja Reddy Bogireddy. 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.

[156] 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 Conf. on Knowledge Discovery and Data Mining V. 2*. 6151–6161.

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

[158] 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.

[159] 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 Int'l Conf. on Data Engineering (ICDE)*. IEEE, 2969–2983.

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

[161] John Paparrizos and Luis Gravano. 2015. k-Shape: Efficient and Accurate Clustering of Time Series. In *Proceedings of the 2015 ACM SIGMOD Int'l Conf. on Management of Data*. 1855–1870.

[162] John Paparrizos and Luis Gravano. 2017. Fast and Accurate Time-Series Clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 1–49.

[163] 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.

[164] 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).

[165] 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*.

[166] 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 Int'l Conf. on Management of Data*. 1887–1905.

[167] 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.

[168] 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.

[169] 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).

[170] Jongjun Park, Fei Chiang, and Mostafa Milani. 2025. Adaptive Anomaly Detection in the Presence of Concept Drift. *arXiv preprint arXiv:2506.15831* (2025).

[171] Kiwon Park, Myoung Gyo Lee, Sung Yong Cho, Yoon Jang, and Young Tae Choi. 2024. Development of Anomaly Detection Technology Applicable to Various Equipment Groups in Smart Factory. In *PHM Society European Conf.*, Vol. 8. 12–12.

[172] Grzegorz Piecuch and Tomasz Żabiński. 2025. A new open dataset from a milling process–data for classification and estimation of tool life. *Scientific Data* 12, 1 (2025), 650.

[173] Nicolò Oreste Pinciroli Vago, Francesca Forbicini, and Piero Fraternali. 2024. Predicting Machine Failures from Multivariate Time Series: An Industrial Case Study. *Machines* 12, 6 (2024). doi:10.3390/machines12060357

[174] Lorena Poenaru-Olaru, Natalia Karpova, Luis Cruz, Jan S. Rellermeyer, and Arie van Deursen. 2024. Is Your Anomaly Detector Ready for Change? Adapting AIOps Solutions to the Real World. In *Proceedings of the IEEE/ACM 3rd Int'l Conf. on AI Engineering - Software Engineering for AI (CAIN 2024)*. ACM, 222–233. doi:10.1145/3644815.3644961

[175] SM Qasim, C Hankendi, M Sherman, K Keahey, G Stringhini, and AK Coskun. 2025. Lessons Learned from Anomaly Detection in Chameleon Cloud. In *2025 IEEE Int'l Conf. on Cloud Engineering (IC2E)*. IEEE, 54–64.

[176] Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, et al. 2025. Tab: Unified benchmarking of time series anomaly detection methods. *arXiv preprint arXiv:2506.18046* (2025).

[177] Muhammad Rajabinasab, Anton Lautrup, and Arthur Zimek. 2025. Metrics for Inter-Dataset Similarity with Example Applications in Synthetic Data and Feature Selection Evaluation. In *Proceedings of the 2025 SIAM Int'l Conf. on Data Mining (SDM)*. SIAM, 527–537.

[178] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.* 29, 2 (may 2000), 427–438. doi:10.1145/335191.335437

[179] 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 int'l conf. on knowledge discovery & data mining*. 3009–3017.

[180] Research and Markets. 2025. IoT Sensors Market Opportunity, Growth Drivers, Industry Trend Analysis, and Forecast 2025-2034. https://www.researchandmarkets.com/reports/6163648/iot-sensors-market-opportunity-growth-drivers. Accessed: 2025-09-16.

[181] Precedence Research. 2025. IoT Sensors Market Size and Forecast 2025 to 2034. https://www.precedenceresearch.com/iot-sensors-market. Accessed: 2025-09-16.

[182] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2019. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694* (2019).

[183] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. 2020. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems* 33 (2020), 21945–21957.

[184] A. Saxena and K. Goebel. 2008. Turbofan Engine Degradation Simulation Data Set. NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA.

[185] Sebastian Schmidl, Felix Naumann, and Thorsten Papenbrock. 2024. AutoTSAD: Unsupervised Holistic Anomaly Detection for Time Series Data. *Proceedings of the VLDB Endowment* 17, 11 (2024), 2987–3002.

[186] 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.

[187] Sensorise. n.d.. *Predictive maintenance strategies for manufacturing*. https://sensorise.de/analytics/predictive-maintenance-for-manufacturing/ Accessed June 2025.

[188] Nagoor Basha Shaik, Kittiphong Jongkittinarukorn, and Kishore Bingi. 2024. XGBoost based enhanced predictive model for handling missing input parameters: A case study on gas turbine. *Case studies in chemical and environmental engineering* 10 (2024), 100775.

[189] Faizan Shaikh, Bestoun S Ahmed, and Agne Swerin. 2025. Unsupervised detection of faults in industrial pumps from multivariate time series. *Machine Learning with Applications* (2025), 100784.

[190] Ali Shehadeh, Odey Alshboul, Rabia Emhamed Al Mamlook, and Ola Hamedat. 2021. Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression. *Automation in Construction* 129 (2021), 103827.

[191] Hyun Joon Shin, Dong-Hwan Eom, and Sung-Shick Kim. 2005. One-class support vector machines—an application in machine fault detection and classification. *Computers & Industrial Engineering* 48 (2005), 395–408. doi:10.1016/j.cie.2005.01.009

[192] Yasser Shoukry, Paul Martin, Yair Yona, Suhas Diggavi, and Mani Srivastava. 2015. Pycra: Physical challenge-response authentication for active sensors under spoofing attacks. In *Proceedings of the 22nd ACM SIGSAC Conf. on Computer and Communications Security*. 1004–1015.

[193] 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 Int'l Symp. on Software Reliability Engineering (ISSRE)*. IEEE, 61–72.

[194] Moses Laksono Singgih, Fauziyyah Firdausi Zakiyyah, et al. 2024. Machine Learning for Predictive Maintenance: A Literature Review. In *2024 Seventh Int'l Conf. on Vocational Education and Electrical Engineering (ICVEE)*. IEEE, 250–256.

[195] Philip Stahmann, Maximilian Nebel, and Christian Janiesch. 2025. AI-based real-time anomaly detection in industrial engineering: A structured literature review, taxonomy, and research agenda. *Computers & Industrial Engineering* (2025), 111236.

[196] statworx GmbH. 2025. *Enhancing Operational Performance through Anomaly Detection and Predictive Maintenance*. https://www.statworx.com/en/case-studies/enhancing-operational-performance-through-anomaly-detection-and-predictive-maintenance Accessed: 2025-09-16.

[197] Marieke Stolte, Franziska Kappenberg, Jörg Rahnenführer, and Andrea Bommert. 2024. Methods for quantifying dataset similarity: a review, taxonomy and comparison. *Statistic Surveys* 18 (2024), 163–298.

[198] Haimeng Sun, Deqiang He, Jiecheng Zhong, Zhenzhen Jin, Zexian Wei, Zhenpeng Lao, and Sheng Shan. 2023. Preventive maintenance optimization for key components of subway train bogie with consideration of failure risk. *Engineering Failure Analysis* 154 (2023), 107634. doi:10.1016/j.engfailanal.2023.107634

[199] 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.

[200] 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.

[201] Shahroz Tariq, Binh M Le, and Simon S Woo. 2022. Towards an awareness of time series anomaly detection models' adversarial vulnerability. In *Proceedings of the 31st ACM Int'l Conf. on Information & Knowledge Management*. 3534–3544.

[202] TBM. 2025. Preventive Maintenance Program Helped a US Manufacturer Reduce Equipment Breakdowns and Improve Productivity. White Paper / Case Study, TBM. https://tbmcg.com/case-study/maintenance-intervention-cuts-downtime/ Accessed October 2025.

[203] Georges Tod, Agusmian Partogi Ompusunggu, and Erik Hostens. 2023. An improved first-principle model of AC powered solenoid operated valves for maintenance applications. *ISA transactions* 135 (2023), 551–566.

[204] Theodoros Toliopoulos and Anastasios Gounaris. 2022. Explainable Distance-Based Outlier Detection in Data Streams. *IEEE Access* 10 (2022), 47921–47936. doi:10.1109/ACCESS.2022.3172345

[205] Theodoros Toliopoulos, Anastasios Gounaris, Kostas Tsichlas, Apostolos Papadopoulos, and Sandra Sampaio. 2020. Continuous outlier mining of streaming data in flink. *Inf. Syst.* 93 (2020), 101569. doi:10.1016/J.IS.2020.101569

[206] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. 2022. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proc. VLDB Endow.* 15, 6 (jun 2022), 1201–1214. doi:10.14778/3514061.3514067

[207] Wisdom Udo and Yar Muhammad. 2021. Data-driven predictive maintenance of wind turbine based on SCADA data. *IEEe Access* 9 (2021), 162370–162388.

[208] Bruno Veloso, João Gama, Rita P Ribeiro, and Pedro M Pereira. 2022. A Benchmark dataset for predictive maintenance. *arXiv preprint arXiv:2207.05466* (2022).

[209] Miryam Elizabeth Villa-Pérez, Miguel A Alvarez-Carmona, Octavio Loyola-González, Miguel Angel Medina-Pérez, Juan Carlos Velazco-Rossell, and Kim-Kwang Raymond Choo. 2021. Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. *Knowledge-Based Systems* 218 (2021), 106878.

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

[211] Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. 2020. A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings. *IEEE Transactions on Reliability* 69, 1 (2020), 401–412. doi:10.1109/TR.2018.2882682

[212] Chuanping Wang, Yudong Li, Jiajia Wang, Yuzhe Wang, Yufeng Liu, Ling Han, Fan Yang, and Xiaoyong Gao. 2025. A Diffusion Model-Empowered CNN-Transformer for Few-Shot Fault Diagnosis in Natural Gas Wells. *Processes* 13, 8 (2025), 2608.

[213] Chen Wang, Jialin Qiao, Xiangdong Huang, Shaoxu Song, Haonan Hou, Tian Jiang, Lei Rui, Jianmin Wang, and Jiaguang Sun. 2023. Apache iotdb: A time series database for iot applications. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–27.

[214] Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. 2025. TimeMixer++: A General Time Series Pattern Machine for Universal Predictive Analysis. In *ICLR*.

[215] Yuanhang Wang, Chao Deng, Jun Wu, Yingchun Wang, and Yao Xiong. 2014. A corrective maintenance scheme for engineering equipment. *Engineering Failure Analysis* 36 (2014), 269–283. doi:10.1016/j.engfailanal.2013.10.006

[216] Zhongmin Wang, Yue Wang, Cong Gao, Fengwei Wang, Tingwu Lin, and Yanping Chen. 2022. An adaptive sliding window for anomaly detection of time series in wireless sensor networks. *Wirel. Netw.* 28, 1 (Jan. 2022), 393–411. doi:10.1007/s11276-021-02852-3

[217] Shuaichong Wei, Mustapha Nourelfath, and Nabil Nahas. 2023. Analysis of a production line subject to degradation and preventive maintenance. *Reliability Engineering & System Safety* 230 (2023), 108906. doi:10.1016/j.ress.2022.108906

[218] Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. 2022. Timeeval: A benchmarking toolkit for time series anomaly detection algorithms. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3678–3681.

[219] Jye West, Milind Siddhpura, Ana Evangelista, and Assed Haddad. 2024. Improving Equipment Maintenance—Switching from Corrective to Preventative Maintenance Strategies. *Buildings* 14, 11 (2024). doi:10.3390/buildings14113581

[220] Shen Yuong Wong, Xiaofeng Ye, Fengkai Guo, and Hui Hwang Goh. 2022. Computational intelligence for preventive maintenance of power transformers. *Applied Soft Computing* 114 (2022), 108129. doi:10.1016/j.asoc.2021.108129

[221] Tao Wu, Xuechun Wang, Shaojie Qiao, Xingping Xian, Yanbing Liu, and Liang Zhang. 2022. Small perturbations are enough: Adversarial attacks on time series prediction. *Information Sciences* 587 (2022), 794–812. doi:10.1016/j.ins.2021.11.007

[222] Zhenyu Wu, Hao Luo, Yunong Yang, Peng Lv, Xinning Zhu, Yang Ji, and Bian Wu. 2018. K-PdM: KPI-oriented machinery deterioration estimation framework for predictive maintenance using cluster-based hidden Markov model. *IEEE Access* 6 (2018), 41676–41687.

[223] M. Xia, T. Li, and T. Shu. 2021. A two-stage approach for the remaining useful life prediction of bearings using deep neural networks. *IEEE Transactions on Industrial Informatics* 17, 6 (2021), 3703–3711.

[224] 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.

[225] 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.

[226] Hong Yang and Travis Desell. 2022. A large-scale annotated multivariate time series aviation maintenance dataset from the ngafid. *arXiv preprint arXiv:2210.07317* (2022).

[227] Zhen Yang, Juho Kanniainen, Tomi Krogerus, and Frank Emmert-Streib. 2022. Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment. *Scientific reports* 12, 1 (2022), 8529.

[228] Yu Yao, Mingming Gong, Yuxuan Du, Jun Yu, Bo Han, Kun Zhang, and Tongliang Liu. 2023. Which is better for learning with noisy labels: the semi-supervised method or modeling label noise?. In *Int'l conf. on machine learning*. PMLR, 39660–39673.

[229] 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 int'l conf. on data mining (ICDM)*. Ieee, 1317–1322.

[230] Pedro Yepez, Basel Alsayyed, and Rafiq Ahmad. 2019. Intelligent assisted maintenance plan generation for corrective maintenance. *Manufacturing Letters* 21 (2019), 7–11. doi:10.1016/j.mfglet.2019.06.004

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

[232] Yanqing Zeng, Zeqiang Zhang, Yu Zhang, Wei Liang, and Haoxuan Song. 2025. Modelling and optimization of line efficiency for preventive maintenance of robot disassembly line. *Journal of Manufacturing Systems* 79 (2025), 347–363. doi:10.1016/j.jmsy.2025.01.021

[233] Chong Zhang, Huan Zhang, and Cho-Jui Hsieh. 2020. An efficient adversarial attack for tree ensembles. *Advances in neural information processing systems* 33 (2020), 16165–16176.

[234] Chaoli Zhang, Yingying Zhang, Lanshu Peng, Qingsong Wen, Yiyuan Yang, Chongjiong Fan, Minqi Jiang, Lunting Fan, and Liang Sun. 2024. Advancing Multivariate Time Series Anomaly Detection: A Comprehensive Benchmark with Real-World Data from Alibaba Cloud. In *Proceedings of the 33rd ACM Int'l Conf. on Information and Knowledge Management*. 5410–5414.

[235] Lingyu Zhang, Jiabao Zhao, and Wei Li. 2019. Online and unsupervised anomaly detection for streaming data using an array of sliding windows and PDDs. *IEEE Transactions on Cybernetics* 51, 4 (2019), 2284–2289.

[236] Xian Zhao, Xiaofei Chai, Shuai Cao, and Qingan Qiu. 2023. Dynamic loading and condition-based maintenance policies for multi-state systems with periodic inspection. *Reliability Engineering & System Safety* 240 (2023), 109586. doi:10.1016/j.ress.2023.109586

[237] Ziming Zhao, Zhaoxuan Li, Jialun Jiang, Fengyuan Yu, Fan Zhang, Congyuan Xu, Xinjie Zhao, Rui Zhang, and Shize Guo. 2023. ERNN: Error-Resilient RNN for Encrypted Traffic Detection towards Network-Induced Phenomena. *IEEE Transactions on Dependable and Secure Computing* (2023), 1–18. doi:10.1109/TDSC.2023.3242134

[238] Haowen Zheng, Hui Liu, Zhenyu Liu, and Jianrong Tan. 2024. Federated temporal-context contrastive learning for fault diagnosis using multiple datasets with insufficient labels. *Advanced Engineering Informatics* 60 (2024), 102432. doi:10.1016/j.aei.2024.102432

[239] Jianghong Zhou, Jiahong Yang, Quan Qian, and Yi Qin. 2024. A comprehensive survey of machine remaining useful life prediction approaches based on pattern recognition: taxonomy and challenges. *Measurement Science and Technology* 35, 6 (2024), 062001.

[240] Tianwen Zhu, Yongyi Ran, Xin Zhou, and Yonggang Wen. 2019. A survey of predictive maintenance: Systems, purposes and approaches. *arXiv preprint arXiv:1912.07383* (2019).

[241] Patrick Zschech, Jonas Bernien, and Kai Heinrich. 2019. Towards a Taxonomic Benchmarking Framework for Predictive Maintenance: The Case of NASA's Turbofan Degradation. In *ICIS*.

Received October 2025; revised January 2026; accepted February 2026