*Expert Systems*

WILEY

Expert Systems

ORIGINAL ARTICLE OPEN ACCESS

# Benchmarking Anomaly Detection Methods: Insights From the UCR Time Series Anomaly Archive

Francisco J. Baldán[1,2] | Diego García-Gil[3]

[1]Department of Computer Science and Programming Languages, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Málaga, Málaga, Spain | [2]Department of Computer Science, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Jaén, Jaén, Spain | [3]Department of Software Engineering, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain

**Correspondence:** Francisco J. Baldán (fjbaldan@uma.es)

**Received:** 23 November 2023 | **Revised:** 18 September 2024 | **Accepted:** 10 October 2024

**Funding:** This work was supported by Grant FJC2021-047112-I funded by MICIU/AEI/10.13039/501100011033 and by European Union Next Generation EU/PRTR. Spanish Ministry of Science and Innovation under project TED2021-132702B-C21 funded by MCIN/AEI/10.13039/501100011033 "European Union PRTR" PID2020-119478GB-I00.

**Keywords:** anomaly detection | anomaly types | benchmark | machine learning | time series

## ABSTRACT

Anomaly detection, vital for identifying deviations from normative data patterns, is particularly crucial in sensor-driven real-world applications, which predominantly involve temporal data in the form of time series. Traditional evaluation of anomaly detection methods has relied on public benchmark datasets. Yet, recent revelations have uncovered inherent flaws and inadequacies in these datasets, casting doubt on the perceived progress in the field. To address this challenge, the UCR Time Series Anomaly Archive has been recently proposed—a meticulously curated database comprising 250 time series—designed to provide a robust and error-free benchmark for anomaly detection research. This paper comprehensively evaluates state-of-the-art anomaly detection techniques using the UCR Time Series Anomaly Archive. Our findings demonstrate the efficacy of current methods in accurately detecting anomalies across an important portion of datasets without additional optimization, underscoring the archive's utility as a foundational baseline for future research and development in anomaly detection methodologies.

## 1 | Introduction

In an increasingly interconnected world, and with an ever-growing amount of generated data, being able to detect anomalous behaviours automatically has become a necessity. Applications as diverse as heart anomaly detection Dissanayake et al. (2020), anomalous power consumption patterns Himeur et al. (2021), attacks on recommender systems Si and Li (2020), detection of acoustic anomalies in industrial processes Bayram, Duman, and Ince (2021), or anomalies detection on Internet of Things (IoT) environments Li and Jung (2022) are just a few of the infinite number of problems where the early detection of anomalies plays a crucial role.

Anomalies, or outliers, in data, refer to data points that deviate notably from the majority of the data points Chandola, Banerjee, and Kumar (2009); Aggarwal (2016). These anomalies can be the result of errors in data collection, measurement errors, or true variations in the data that represent interesting and important phenomena. Detecting anomalies is an important task in data analysis as it can help to identify errors and patterns in the data.

In most cases, the anomaly search process is performed on data recorded over time. This specific type of data is known as time series. Each time series value has a certain temporal relationship with its past values that must be preserved Baldan et al. (2018). Due to this, the time series have special properties that must be

considered for a correct processing Baldán and Benítez (2023), such as trend, seasonality, and stationarity, among others. Ignoring these behaviours can lead to misinterpretations, inaccurate models, and unreliable predictions. For example, ignoring information related to trends, cycles, and autocorrelation can result in missing important patterns. In addition, ignoring the seasonality of a time series can prevent the capture of relevant long-term information, resulting in inaccurate models and unreliable forecasts. Therefore, traditional anomaly detection algorithms that do not take this temporal component into account cannot be used.

Anomaly detection in time series has established itself as a very prolific field with a large number of proposals Xu et al. (2019); Pang et al. (2021). However, the vast majority of new proposals are focused on improving the results obtained by previous ones. Although continuous progress in this field can be seen, the proven existence of multiple problems in the state-of-the-art reference datasets, such as unrealistic anomaly density, and mislabelled ground truth, among others, discourage their use as benchmarks. This makes progress in this field just an illusion Wu and Keogh (2021).

To address these issues, the authors of the state-of-the-art univariate Dau et al. (2019) and multivariate Bagnall et al. (2018) time series classification repositories have proposed a new state-of-the-art anomaly detection time series repository, the UCR Time Series Anomaly Archive Wu and Keogh (2021). This repository is composed of 250 different time series from a wide variety of fields. The authors have opted for an approach in which each time series contains a single anomaly, well identified by the experts, facilitating the final evaluation of the results and allowing the repository to be expanded with new time series in a transparent way for future work. In this way, it is only evaluated whether or not an algorithm is capable of detecting the anomaly present in each time series, avoiding some of the problems previously mentioned, such as the case in which in a time interval without changes, part of the values that compose it are considered anomalies while others are not. The anomaly contained in each time series can be of any type, from clear variations in the value of the time series to patterns that are more or less complex to detect.

In this paper, we propose an extensive experimental study of the state-of-the-art anomaly detection methods on the new UCR Time Series Anomaly Archive repository, with the aim of obtaining a robust baseline of results against which the new proposals can be compared. The main advantages of this proposal are:

- A common framework for the comparison of new proposals for anomaly detection in time series.
- Clear and easily comparable results of the main state-of-the-art algorithms from different approaches.
- The number and type of time series included in the study can be easily extendable.

The remainder of this paper is organised as follows: Section 2 depicts the state-of-the-art in time series anomaly detection. In Section 3, the proposed study is explained. Section 4 contains the experimentation performed in detail. Finally, Section 5 concludes the paper.

## 2 | Related Works

This section contains multiple analyses of the state-of-the-art anomaly detection field. First, a study about the anomaly detection problem is included (Section 2.1). Second, the main proposals of the state-of-the-art are analysed (Section 2.2). Finally, the actual problems in the anomalies detection field are explained (Section 2.3).

### 2.1 | Anomaly Detection

Anomaly detection is a set of techniques used to identify data points that deviate from the expected patterns or norms in a dataset Chandola, Banerjee, and Kumar (2009).

We can differentiate between five types of anomalous instances Goldstein and Uchida (2016):

- Point Anomalies: Point anomalies refer to individual data points that deviate notably from the rest of the data points in the dataset. These anomalies are the most common type and can be caused by measurement errors, data entry errors, or sampling errors.
- Contextual Anomalies: Contextual anomalies occur when a data point is anomalous in a specific context but is not considered anomalous in other contexts. For example, a high-temperature reading might be anomalous in the context of a hospital room but not anomalous in the context of a sauna.
- Collective Anomalies: Collective anomalies occur when a set of related data points deviates considerably from the expected pattern. These anomalies can be caused by group behaviour or system failures.
- Spatial Anomalies: Spatial anomalies occur when a data point deviates from the expected pattern based on its geographical location. These anomalies are common in environmental monitoring or geographical data analysis.
- Temporal Anomalies: Temporal anomalies occur when a data point deviates from the expected pattern based on its temporal location. These anomalies are common in time-series data analysis or event sequence analysis.

It is important to understand the different types of anomalies to accurately identify them and determine their potential causes. Different types of anomalies may require different detection techniques and methods. In addition, understanding the type of anomaly can help to identify potential causes and develop appropriate mitigation strategies Aggarwal (2016).

Anomalies and noise are two types of irregularities that can affect data. While they may appear similar, there are several key differences between them. Noise is a random error or interference that affects the overall quality of the data by reducing its accuracy and precision. This is why noise must be detected and either corrected or eliminated Luengo et al. (2020); García-Gil et al. (2019). Anomalies, on the other hand, contain valuable information that has to be detected, extracted, and analysed.

Anomaly detection has wide-ranging applications in diverse fields such as security, fraud detection, fault detection, healthcare, and finance.

An anomaly detection algorithm can produce different types of outputs depending on the type of algorithm and the application context Chandola, Banerjee, and Kumar (2009). Here are some of the most common types of outputs that can be produced by an anomaly detection algorithm:

- Binary output: the result is a binary value indicating whether a data point is anomalous or not. This output is useful for simple anomaly detection tasks where the goal is to identify whether a data point is anomalous or not.
- Score output: the result is a score indicating the degree of anomaly for each data point. This output is useful when multiple data points have varying degrees of anomaly, and it is important to prioritise them based on the severity of the anomaly.

As mentioned above, our focus is on anomaly detection for time series. A time series is a collection of chronologically ordered data. Some of its characteristics are large size, high dimensionality, and continuous updating. Anomaly detection in time series has become a hot issue as it is used in a variety of domains.

## 2.2 | Anomalies Detection Algorithms State-Of-The-Art

In this Section, we depict the different types of anomaly detection algorithms according to their internal mechanisms to detect anomalies. We can differentiate between probabilistic algorithms, linear models, proximity-based methods, outlier ensembles, and neural networks:

- Probabilistic: these anomaly detection methods involve the use of statistical models to identify data points or events that deviate notably from the expected or normal behaviour of the data. Probabilistic methods typically assume that the normal behaviour of a system or process can be modelled by a probability distribution and that anomalies can be identified by their low probability of occurrence under this distribution. Examples of these methods are ECOD Li et al. (2022), COPOD Li et al. (2020), and MAD Iglewicz and Hoaglin (1993).
- Linear Model: these methods make use of a linear model to identify anomalies. This linear model assumes that the data can be represented by a linear relationship between the input variables. The most popular method of this family is OCSVM Schölkopf et al. (2001).
- Proximity-Based: distance and similarity measures are the mechanisms used by proximity-based anomaly detection methods. Anomalies can be identified as data points that are considerably different or far from the rest of the data. Examples of proximity-based anomaly detectors are LOF Breunig et al. (2000), and CBLOF He, Xu, and Deng (2003).
- Outlier Ensembles: the combination of multiple anomaly detection methods, in the form of an ensemble, can improve the accuracy and robustness of the anomaly detection process. These methods assume that the combination of all base detectors, along with their weaknesses and strengths, can improve the detection of anomalies. The most popular and widely used methods of this category are Isolation Forest Liu, Ting, and Zhou (2012), and LODA Pevnỳ (2016).
- Neural Networks: these methods are based on the use of neural networks for the identification of anomalies. These anomalies can be detected as data points that have high prediction errors or do not fit well with the learned model. They have the advantage of being able to learn complex and nonlinear relationships between the input variables, without requiring explicit assumptions about normal behaviour. Examples of neural network anomaly detectors are SO_GAAL and MO_GAAL Liu et al. (2019). Another example is DeepSVDD Ruff et al. (2018). It trains a neural network to map input data into a hypersphere of minimal volume, enclosing the majority of normal data points. Anomalies are identified as points that lie outside this hypersphere, with the network learning to extract the common factors of variation in the data. This method optimises an objective that directly targets anomaly detection, distinguishing it from other approaches that rely on reconstruction errors.

The field of anomaly detection has witnessed remarkable advancements in recent years, driven by the proliferation of complex data sources and the demand for robust detection capabilities. State-of-the-art methodologies encompass a diverse range of approaches, including traditional statistical techniques Li et al. (2022), machine learning algorithms Ali et al. (2020); Nassif et al. (2021), and hybrid models Liu, Ting, and Zhou (2012). Recent developments have emphasised the integration of deep learning architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which excel at capturing intricate patterns and temporal dependencies in data Pang et al. (2021). Moreover, the emergence of unsupervised learning paradigms, coupled with advancements in semi-supervised and self-supervised techniques, has expanded the applicability of anomaly detection to scenarios with limited labelled data Zhou et al. (2020). Concurrently, efforts have been made to enhance the interpretability and explainability of anomaly detection models, addressing concerns related to their transparency and trustworthiness in real-world applications Ali et al. (2023). Overall, the latest advancements in anomaly detection methodologies underscore a shift towards more sophisticated, adaptive, and scalable approaches capable of addressing the evolving challenges posed by modern data environments.

## 2.3 | Issues of Anomalies Time Series Datasets

Some state-of-the-art datasets in anomaly detection have specific errors that make them unsuitable for use as benchmarks Wu and Keogh (2021): triviality, unrealistic anomaly density, mislabelled ground truth, and run-to-failure bias.

- Triviality: problems that are so simple that they could be solved with one line of code (one-liners). Of course, we can use "magic numbers" extracted from the training process, but we should avoid calling already built-in methods as a one-line solution.
- Unrealistic Anomaly Density: time series in which a large number of continuous points are considered anomalies, multiple regions of the same time series are marked as anomalies or different anomalous points which are very close to each other but separated by a normal point.
- Mislabelled Ground Truth: it is possible to find datasets in which the anomaly labels are ambiguous. For example, an extensive area of a time series can have constant values, with no changes, and some of those points are marked as anomalous and not the rest.
- Run-to-Failure: it is common that real-world systems are run-to-failure. In those cases, we can find problems in which the anomalies of the entire dataset are presented at the end of the time series. Any algorithm that marks the last points of any time series as anomalies without additional considerations is able to obtain an excellent performance in those cases.

The problems mentioned above are analysed in-depth in the original work Wu and Keogh (2021). The authors found the typical run-to-failure behaviour problems (Yahoo Laptev, Amizadeh, and Billawala (2015) and NASA Hundman et al. (2018) benchmark datasets), in which a simple naive detector that marks the last point of each time series as anomalous can obtain a competitive accuracy. Additionally, they found that a simple one-liner method could obtain 86.1% accuracy for the case of the Yahoo dataset, which is a competitive result when compared with papers that have used this dataset as a benchmark Qiu, Du, and Qian (2019); Gao et al. (2020). More examples of these behaviours can be found in the previous paper Wu and Keogh (2021). In addition, authors have published a resource Wu and Keogh (2023) where they show additional examples of these kinds of problems in different state-of-the-art datasets (Yahoo, NASA, Numenta Ahmad et al. (2017), and Pei's Lab (OMNI) Su et al. (2019)). According to these results, authors have proposed a new repository, composed of 250 different time series, free of these known errors. Moreover, the authors have encouraged the community to create new sets of tested time series anomaly detection datasets. In addition, authors have also proposed simple anomaly detection models, called one-liners, which we explain in depth in the next section.

## 3 | One-Liners and State-Of-The-Art Algorithms Comparative Framework

This section begins by introducing the UCR Time Series Anomaly Archive one-liners approach (Section 3.1). Then, we explain the selection of the state-of-the-art anomaly detection methods included in this work (Section 3.2). Finally, we describe the metrics employed to evaluate the performance of the state-of-the-art methods (Section 3.3).

**TABLE 1** | One-liners parameters specified in the paper.

| Parameter | UCR proposal | Our implementation |
|---|---|---|
| **u** | $[0,1] \in \mathbb{Z}$ | $[-100,100] \in \mathbb{R}$ |
| **c** | — | $[-100,100] \in \mathbb{R}$ |
| **k** | — | $[10,500] \in \mathbb{Z}$ |
| **b** | $(-\infty,+\infty) \in \mathbb{R}$ | $(-\infty,+\infty) \in \mathbb{R}$ |

### 3.1 | UCR Time Series Anomaly Archive One-Liners

The models proposed as benchmarks in the UCR Time Series Anomaly Archive aim to demonstrate that simple models are able to address anomaly detection problems as effectively as more complex proposals. The UCR Time Series Anomaly Archive proposal includes as benchmark models the next six one-liners:

$$\begin{aligned} abs(diff(TS)) > {} & \mathbf{u}^* movmean(abs(diff(TS)), \mathbf{k}) \\ & + \mathbf{c}^* movstd(abs(diff(TS)), \mathbf{k}) \\ & + \mathbf{b} \end{aligned} \tag{1}$$

$$\begin{aligned} diff(TS) > {} & \mathbf{u}^* movmean(diff(TS), \mathbf{k}) \\ & + \mathbf{c}^* movstd(diff(TS), \mathbf{k}) \\ & + \mathbf{b} \end{aligned} \tag{2}$$

$$abs(diff(TS)) > \mathbf{b} \tag{3}$$

$$\begin{aligned} abs(diff(TS)) > {} & movmean(abs(diff(TS)), \mathbf{k}) \\ & + \mathbf{c}^* movstd(abs(diff(TS)), \mathbf{k}) \\ & + \mathbf{b} \end{aligned} \tag{4}$$

$$diff(TS) > \mathbf{b} \tag{5}$$

$$\begin{aligned} diff(TS) > {} & movmean(diff(TS), \mathbf{k}) \\ & + \mathbf{c}^* movstd(diff(TS), \mathbf{k}) \\ & + \mathbf{b} \end{aligned} \tag{6}$$

In this work, we have re-implemented them, but with slight modifications, because the ranges of values of **k** and **c** parameters were not included in the original paper Wu and Keogh (2021). The information about the parameters used is included in Table 1. We chose to expand the possible values of each parameter with the objective of finding the limitations of those one-liners methods. The *movmean* and *movstd* functions correspond to the calculation of the moving mean and standard deviation, respectively, for a window of size **k**. The diff(ts) method works by calculating the differences between successive observations in the time series. This method is widely used in the time series field and enables focusing on relative changes instead of changes in the absolute values of the time series. In this work, the applied differentiation calculates the difference of each point with the previous one.

The optimization process is performed with the *Optuna* framework Akiba et al. (2019). This process minimises the distance between the predicted anomaly point and the middle point of the known anomaly. The process is composed of three well-differentiated steps:

- First, *Optuna* proposes a combination of **u**, **c**, and **k** parameters whenever available.
- Second, the **b** parameter is calculated to adjust the limit to consider a point as anomalous. In this step, we only obtain one point as the possible anomaly in the input time series.
- Third, based on the distance between the predicted and the middle of the real anomaly points, *Optuna* proposes a new set of parameters combination to minimise this distance.

It is important to mention that these one-liners use a brute-force approach, which optimises the different parameters to obtain the best prediction possible. This approach has the intention to demonstrate that simple models are able to address these kinds of problems but need to know the position of the anomaly to perform its optimization. Because of this, we cannot consider their results in the final state-of-the-art models' evaluation. Additionally, it is relevant to mention the main limitation or bias included in the UCR Time Series Anomaly Archive, which is related to the structure of the proposed data. Each dataset or time series includes a single anomaly, which benefits models that only select as an anomaly a single point that they identify as the most anomalous, leaving out of this classification other points that could also be considered anomalous, such as the one-liners.

### 3.2 | State-Of-The-Art Anomaly Detection Methods

We have extracted the state-of-the-art methods for anomaly detection from the PyOD library Zhao, Nasrullah, and Li (2019). This library has become the reference as far as anomaly detection is concerned. It is being updated actively and regularly with the newest and best-performing anomaly detection methods.

Here, we have carried out a selection among all algorithms available in the PyOD library. This selection is based on the ability of the methods to be run on all the benchmark datasets without the need to perform any parameter optimization, only using their default values. Since our focus is to present a baseline benchmark of the different anomaly detection methods available in the literature, we will not focus on obtaining the best result for each tested method for each of the 250 available datasets. In Table 2, we show the selected 21 methods.

### 3.3 | Anomaly Detection Functions

To perform the detection of the anomalies, we include two different criteria:

- *maxScore*: the anomaly point is chosen based on the maximum anomaly score provided by the different methods.
- *maxScoreWindow*: a sliding window is used for the selection of the anomaly point. In particular, we select the middle point of the sliding window with the maximum accumulated score value.

On the one hand, the *maxScore* criterium is the simplest and more demanding criterion, but it is sensible to little variations in data and parameters. On the other hand, *maxScoreWindow* needs to process a higher number of data points but provides a robust final decision since it includes context information about the anomaly evaluation.

## 4 | Experimentation

In this section, the experimental framework and results are presented. First, the experimentation setup is explained (Section 4.1). Second, the datasets used are described in Section 4.2. Third, the metrics and evaluation criteria selected are depicted (Section 4.3). Finally, the results obtained are shown and analysed in Section 4.4.

### 4.1 | Experimentation Setup

The one-liners experimentation has been performed with the equations and parameters specified in Section 3.1. When applying the maxScoreWindow anomaly detection function, we selected the following window lengths based on the 100-point anomaly detection threshold defined in the original paper: 10, 100, 200, and 500. This allows us to analyse the impact of using window widths both above and below the specified threshold. Additionally, we had limited the maximum number of iterations in *Optuna* for each time series to 1000. The state-of-the-art anomaly detection methods have been set to the default parameters offered by the PyOD library.

The source code of the experimentation has been developed in Python 3.9.13 and can be found in the online repository.[1]

### 4.2 | Datasets

The UCR Time Series Anomaly Archive Wu and Keogh (2021) is composed of 250 different univariate time series. Each time series contains only one anomaly, and its position is known: the start and end point of the anomaly subsequence. Additionally, the training and test subsets of each time series are specified. The training part is free of anomalies.

It is important to note that the types of anomalies and time series considered in this dataset are diverse. The time series of this repository have lengths that vary between 6684 and 900,000 data points.

### 4.3 | Metrics and Evaluation Criteria

Following the indications of the dataset archive authors, one prediction is evaluated as correct only if it is located at a distance equal to or less than 100 points from any point on the

**TABLE 2** | State-of-the-art anomaly detection methods selected.

| Type | Name | Year | Reference |
|---|---|---|---|
| Probabilistic | ECOD | 2022 | Li et al. (2022) |
| | COPOD | 2020 | Li et al. (2020) |
| | MAD | 1993 | Iglewicz and Hoaglin (1993) |
| | KDE | 2007 | Latecki, Lazarevic, and Pokrajac (2007) |
| | Sampling | 2013 | Sugiyama and Borgwardt (2013) |
| Linear model | OCSVM | 2001 | Scholkopf et al. (2001) |
| Proximity-based | LOF | 2000 | Breunig et al. (2000) |
| | CBLOF | 2002 | He, Xu, and Deng (2003) |
| | HBOS | 2012 | Goldstein and Dengel (2012) |
| | kNN | 2000 | Ramaswamy, Rastogi, and Shim (2000) |
| | AvgKNN | 2002 | Ramaswamy, Rastogi, and Shim (2000) |
| | MedKNN | 2002 | Ramaswamy, Rastogi, and Shim (2000) |
| | ROD | 2020 | Almardeny, Boujnah, and Cleary (2020) |
| Outlier ensembles | IForest | 2008 | Liu, Ting, and Zhou (2012) |
| | INNE | 2018 | Bandaragoda et al. (2018) |
| | LODA | 2016 | Pevnỳ (2016) |
| Neural networks | AutoEncoder | 2015 | Aggarwal (2016) |
| | VAE | 2013 | Kingma and Welling (2013) |
| | So_GAAL | 2019 | Liu et al. (2019) |
| | MO_GAAL | 2019 | Liu et al. (2019) |
| | DeepSVDD | 2018 | Ruff et al. (2018) |

anomaly. This simple idea aims to facilitate the comparison of the results between different proposals, which is one of the most important limitations of the actual anomaly detection state-of-the-art, allowing them all to use the same metric. In this way, they suggest performing the comparison by evaluating the total number of anomalies detected, similarly to the accuracy. It is calculated as the total number of correctly detected anomalies divided by the total number of time series. We do not perform additional evaluations about the distance of prediction to the anomaly.

The Critical Difference diagrams (CD) Demšar (2006) have been included to analyse the results obtained from a statistical point of view. CD allows for the comparison of results among different models from a statistical standpoint. In this diagram, each model is ordered based on its average rank, which is visible on the top line of the graph. The models connected by a bold line are deemed to exhibit no statistically significant differences in their outcomes at a specified confidence level $\alpha$. For this study, we have opted for a 95% confidence level, setting $\alpha$ to 0.05. The R *scmamp* package Calvo and Santafé (2016) has been employed to compute the average rank and the CD.

Additional information regarding the metrics used in our evaluation can be found in GitHub repository.[1]

## 4.4 | Results

In this section, the results obtained for the one-liners are shown (Section 4.4.1). Results achieved by the state-of-the-art anomaly detection methods are also described in detail (Section 4.4.2).

### 4.4.1 | One-Liners Results

Table 3 contains the results obtained for the one-liner approach over the 250 time series that composes the UCR Anomaly Detection repository. We show that the one-liners with the highest number of parameters or components (Equations 1 and 2) are able to obtain a high accuracy in the experimentation performed. These results are expected since Equations 3 and 4 are a simplification or special case of Equation 1, and Equations 5 and 6 are a simplification of Equation 2. Higher degrees of freedom in the equations allow better results to be obtained.

It is important to note that, for this approach, it is necessary to know the anomaly position in each case because this information is used to optimise the one-liner parameters. The objective of this approach is not to be part of the state-of-the-art time series anomaly detection but to show that simple models are able to describe a substantial part of typical time series anomaly detection problems.

#### 4.4.2 | State-Of-The-Art Algorithm Results

Table 4 gathers the accuracy results achieved by the different anomaly detection algorithms employed in this study. In view of these results, we can draw the following conclusions:

**TABLE 3** | One-liners results.

| One-liner | Anomalies detected | Accuracy |
|---|---|---|
| Equation 1 | 212 | 0.848 |
| Equation 2 | 208 | 0.832 |
| Equation 3 | 76 | 0.304 |
| Equation 4 | 196 | 0.784 |
| Equation 5 | 74 | 0.296 |
| Equation 6 | 195 | 0.780 |

- As we can see, the use of the differentiation method is improving the results of the two prediction functions used, *maxScore* and *maxScoreWindow*. All methods are achieving their best results using the differentiation method. Some cases reflect an improvement of close to 100%. This result is expected since the differentiation method reduces the gap between consecutive data points, making the time series flatter, and therefore, the algorithms can easily identify the real anomalies.
- DeepSVDD stands as the best-performing method for all metrics. It is capable of detecting correctly the anomalies in more than half the datasets tested. Moreover, the use of different metrics shows a slight variance in the results.
- If we attend to the length of the window for the *maxScoreWindow* metric, we can see two different trends: algorithms that benefit from the increase of available data, and algorithms that suffer from this increase in data. Distance-based methods, as well as DeepSVDD, are improving their results with increasing window lengths.

**TABLE 4** | State-of-the-art algorithms accuracy results.

| Differentiation method | Not applied | | | | | Applied | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Prediction function** | **maxScore** | **maxScoreWindow** | | | | **maxScore** | **maxScoreWindow** | | | |
| Length | — | 10 | 100 | 200 | 500 | — | 10 | 100 | 200 | 500 |
| ECOD | 0.184 | 0.164 | 0.124 | 0.112 | 0.080 | 0.324 | 0.344 | 0.272 | 0.264 | 0.132 |
| COPOD | 0.128 | 0.140 | 0.160 | 0.108 | 0.080 | 0.256 | 0.224 | 0.220 | 0.252 | 0.152 |
| MAD | 0.128 | 0.140 | 0.196 | 0.184 | 0.100 | 0.304 | 0.320 | 0.320 | 0.324 | 0.164 |
| KDE | 0.196 | 0.192 | 0.224 | 0.232 | 0.156 | 0.400 | 0.392 | 0.396 | 0.408 | 0.204 |
| Sampling | 0.116 | 0.160 | 0.184 | 0.160 | 0.132 | 0.336 | 0.344 | 0.388 | 0.388 | 0.196 |
| OCSVM | 0.104 | 0.120 | 0.156 | 0.160 | 0.104 | 0.292 | 0.284 | 0.288 | 0.288 | 0.156 |
| LOF | 0.096 | 0.208 | 0.292 | 0.288 | 0.244 | 0.188 | 0.336 | 0.404 | 0.396 | 0.304 |
| CBLOF | 0.112 | 0.148 | 0.176 | 0.172 | 0.120 | 0.312 | 0.320 | 0.372 | 0.376 | 0.168 |
| HBOS | 0.056 | 0.128 | 0.144 | 0.108 | 0.104 | 0.024 | 0.280 | 0.248 | 0.256 | 0.148 |
| kNN | 0.180 | 0.172 | 0.192 | 0.200 | 0.164 | 0.384 | 0.412 | 0.452 | 0.460 | 0.260 |
| AvgKNN | 0.184 | 0.168 | 0.204 | 0.204 | 0.180 | 0.392 | 0.416 | 0.464 | 0.464 | 0.276 |
| MedKNN | 0.176 | 0.172 | 0.200 | 0.200 | 0.172 | 0.388 | 0.412 | 0.468 | 0.472 | 0.272 |
| ROD | 0.176 | 0.156 | 0.164 | 0.116 | 0.092 | 0.312 | 0.344 | 0.328 | 0.336 | 0.156 |
| IForest | 0.060 | 0.120 | 0.124 | 0.120 | 0.096 | 0.100 | 0.252 | 0.276 | 0.280 | 0.148 |
| INNE | 0.176 | 0.156 | 0.092 | 0.108 | 0.076 | 0.356 | 0.292 | 0.212 | 0.152 | 0.092 |
| LODA | 0.040 | 0.116 | 0.160 | 0.148 | 0.104 | 0.052 | 0.156 | 0.228 | 0.240 | 0.168 |
| AutoEncoder | 0.172 | 0.176 | 0.168 | 0.152 | 0.124 | 0.304 | 0.308 | 0.304 | 0.296 | 0.232 |
| VAE | 0.172 | 0.188 | 0.164 | 0.176 | 0.140 | 0.296 | 0.300 | 0.292 | 0.304 | 0.228 |
| So_GAAL | 0.028 | 0.036 | 0.044 | 0.032 | 0.032 | 0.132 | 0.088 | 0.088 | 0.088 | 0.048 |
| MO_GAAL | 0.020 | 0.020 | 0.028 | 0.032 | 0.020 | 0.100 | 0.104 | 0.104 | 0.092 | 0.072 |
| DeepSVDD | **0.440** | **0.472** | **0.496** | **0.504** | **0.364** | **0.472** | **0.500** | ***0.520*** | ***0.520*** | **0.416** |
| Average | 0.140 | 0.160 | 0.176 | 0.167 | 0.128 | 0.273 | 0.306 | 0.316 | 0.317 | 0.190 |

*Note:* The highest accuracy value per configuration is stressed in **bold**. The best result overall is stressed in *italic*. The best result for each method is highlighted in blue.

- It is expected that the best results are achieved by the *maxScoreWindow* metric since the *maxScore* metric is much more aggressive with the results. However, as we can see, with just the maximum anomaly score for each dataset, in the case of DeepSVDD, it is possible to correctly predict the anomalies in 47.2% of the datasets.

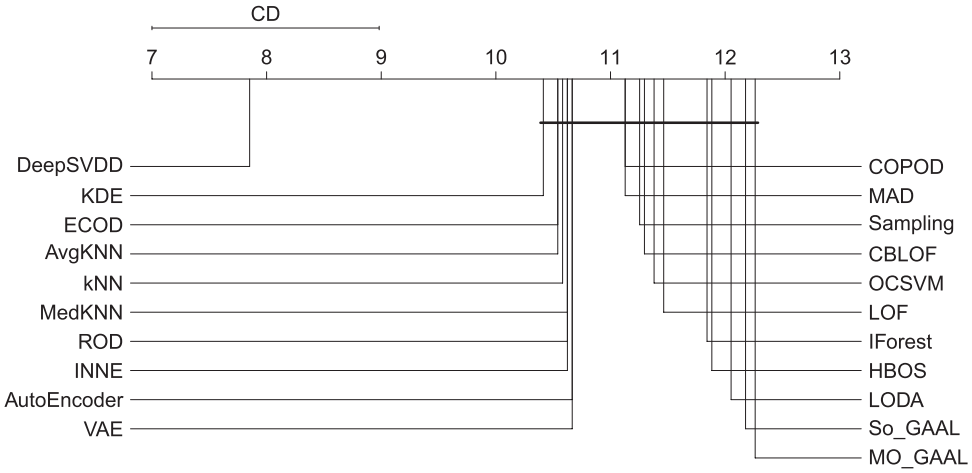
Figures 1 and 2 contain the CD extracted from each metric and case of interest in Table 4. These diagrams allow us to analyse from a statistical point of view the obtained results.

For the *maxScore* metric, Figure 1 shows two well-differentiated behaviours. In the first case (Figure 1a), where no additional preprocessing technique was applied to the original time series, and the maximum value of *maxScore* provided by each method is used to identify the anomaly, DeepSVDD provided results that are statistically distinguishable of the remaining methods. In the second case (Figure 1b), we can see a larger number of statistically non-distinguishable relationships between the results of the different methods. This differentiation preprocessing step improves the results of most cases overall and increases the competitiveness between them.

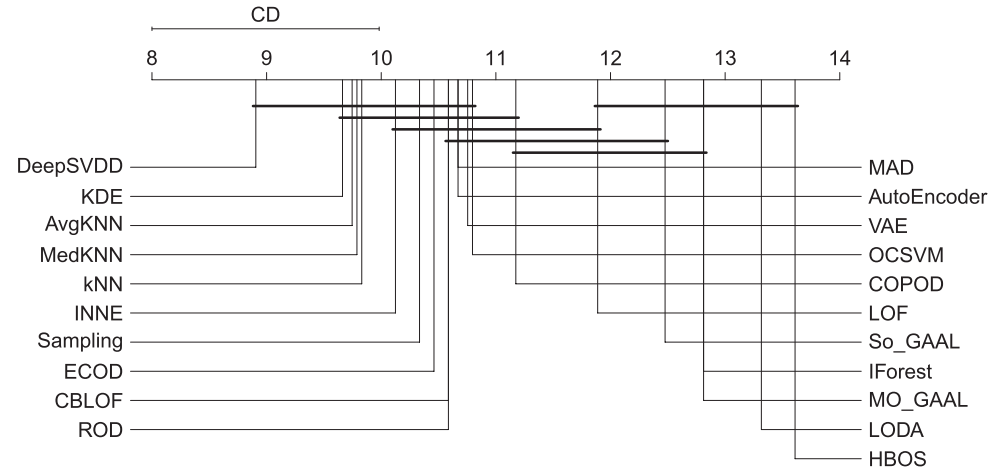
In the *maxScoreWindow* case (Figure 2), the differentiation preprocessing step improves the final results, causing the results of each model to be less statistically distinguishable from the other models. This behaviour can be observed by comparing Figure 2a,b. On the first hand, Figure 2a shows that DeepSVDD provides results statistically distinguishable from the remainder methods, similarly as in the previous case (Figure 1a). On the other hand, in Figure 2b, we can observe that DeepSVDD provides results statistically non-distinguishable from seven other methods.

In the earlier cases (Figure 1b), DeepSVDD provides results statistically non-distinguishable from the other 13 methods. The number of methods with statistically non-distinguishable results has increased in the later comparisons. Due to this, the benefits of applying the differentiation step are, therefore, proven. Additionally, we can appreciate the AvgKNN algorithm as a robust third-best anomaly detection method, while the second position is subject to continuous changes.

These results show the challenging problem of finding a unique approach for this type of problem and the benefit of having a benchmark as wide as possible to allow a robust evaluation of the new proposals under the same parameters.

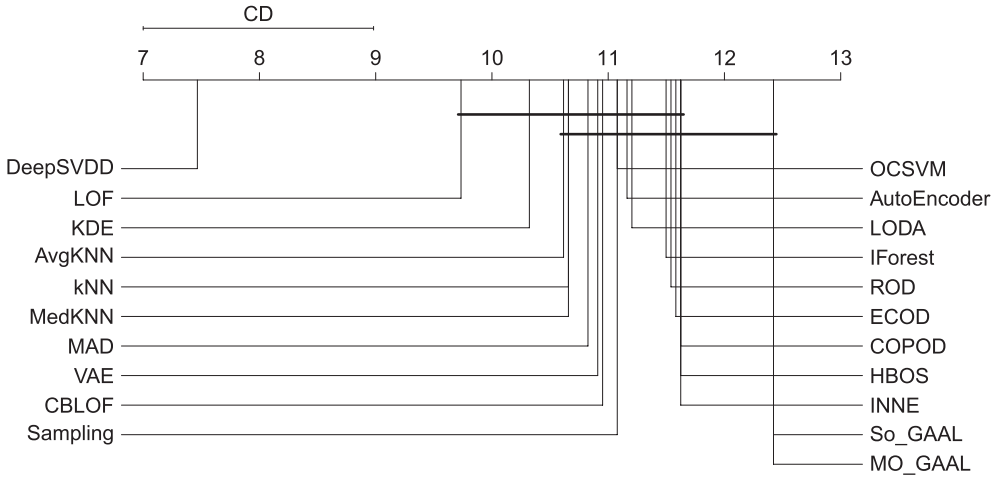



(a) CD without differentiation *maxScore* case.

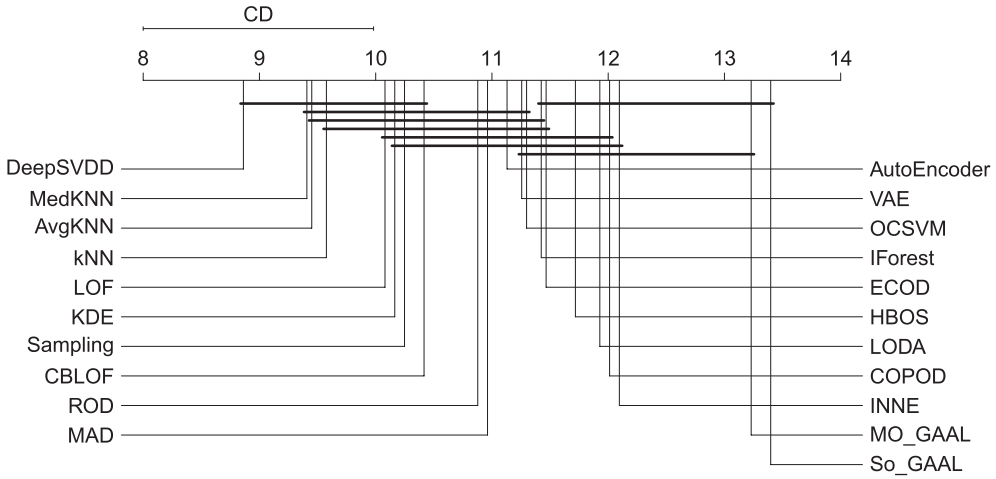



(b) CD with differentiation *maxScore* case

**FIGURE 1** | CD for *maxScore* metric of each case of interest in Table 4.




(a) CD without differentiation *maxScoreWindow* with window 200 case




(b) CD with differentiation *maxScoreWindow* with window 100 case

**FIGURE 2** | CD for *maxScoreWindow* metric of each case of interest in Table 4.

## 5 | Conclusions

In this work, we have presented a new benchmark for the time series anomaly detection field. This benchmark provides a new starting point for any new time series anomaly detection proposal, based on datasets free of typical issues and limitations that recommend discontinuing the use of the main state-of-the-art datasets. The main state-of-the-art methods have been included, without additional considerations or optimization, facilitating the comparison of new proposals.

The experimentation code has been published to facilitate the reproducibility of results and future comparisons. The approach, datasets, and metrics chosen for this work allow a simple and easy comparison between different methods, as well as the inclusion of new datasets. The results obtained show that the DeepSVDD method is the best state-of-the-art method. If the differentiation method is applied to the original data, DeepSVDD remains the best-performing anomaly detection method, but it is not statistically better than others. The aforementioned simple differentiation step provides an improvement of the results obtained and shows a more competitive environment among the methods used.

### Author Contributions

**Francisco J. Baldán:** conceptualization, methodology, data curation, software, writing – review and editing, validation. **Diego García-Gil:** software, writing – review and editing, validation.

### Acknowledgements

This work has been partially supported by project PID2020-119478GB-I00. This work has also been partially supported by the Spanish Ministry of Science and Innovation under project TED2021-132702B-C21 funded by MCIN/AEI/10.13039/501100011033 "European Union PRTR." Francisco J. Baldán was supported by grant FJC2021-047112-I funded by MICIU/AEI/10.13039/501100011033 and by European Union Next Generation EU/PRTR. Funding for open access charge: Universidad de Málaga / CBUA.

### Conflicts of Interest

The authors declare no conflicts of interest.

## Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## Endnotes

[1] Time series Anomaly Detection repository. https://github.com/fjbaldan/tsAD.

## References


Aggarwal, C. C. 2016. *Outlier Analysis*. 2nd ed. Switzerland: Springer Publishing Company.

Ahmad, S., A. Lavin, S. Purdy, and Z. Agha. 2017. "Unsupervised Real-Time Anomaly Detection for Streaming Data." *Neurocomputing* 262: 134–147.

Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama. 2019. "Optuna: A Next-Generation Hyperparameter Optimization Framework." In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

Ali, W. A., K. Manasa, M. Bendechache, M. Fadhel Aljunaid, and P. Sandhya. 2020. "A Review of Current Machine Learning Approaches for Anomaly Detection in Network Traffic." *Journal of Telecommunications and the Digital Economy* 8: 64–95.

Ali, S., T. Abuhmed, S. El-Sappagh, et al. 2023. "Explainable Artificial Intelligence (Xai): What We Know and What Is Left to Attain Trustworthy Artificial Intelligence." *Information Fusion* 99: 101805.

Almardeny, Y., N. Boujnah, and F. Cleary. 2020. "A Novel Outlier Detection Method for Multivariate Data." *IEEE Transactions on Knowledge and Data Engineering* 34: 4052–4062.

Bagnall, A., H. A. Dau, J. Lines, et al. 2018. "The UEA Multivariate Time Series Classification Archive, 2018." *arXiv Preprint*. arXiv:1811.00075.

Baldán, F. J., and J. M. Benítez. 2023. "Complexity Measures and Features for Times Series Classification." *Expert Systems with Applications* 213: 119227.

Baldan, F. J., S. Ramirez-Gallego, C. Bergmeir, F. Herrera, and J. M. Benitez. 2018. "A Forecasting Methodology for Workload Forecasting in Cloud Systems." *IEEE Transactions on Cloud Computing* 6: 929–941.

Bandaragoda, T. R., K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells. 2018. "Isolation-Based Anomaly Detection Using Nearest-Neighbor Ensembles." *Computational Intelligence* 34: 968–998.

Bayram, B., T. B. Duman, and G. Ince. 2021. "Real Time Detection of Acoustic Anomalies in Industrial Processes Using Sequential Autoencoders." *Expert Systems* 38: e12564.

Breunig, M. M., H.-P. Kriegel, R. T. Ng, and J. Sander. 2000. "Lof: Identifying Density-Based Local Outliers." In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104.

Calvo, B., and G. Santafé. 2016. "Scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems." *R Journal* 8: 248–256.

Chandola, V., A. Banerjee, and V. Kumar. 2009. "Anomaly Detection: A Survey." *ACM Computing Surveys (CSUR)* 41: 1–58.

Dau, H. A., A. Bagnall, K. Kamgar, et al. 2019. "The UCR Time Series Archive." *IEEE/CAA Journal of Automatica Sinica* 6: 1293–1305.

Demšar, J. 2006. "Statistical Comparisons of Classifiers Over Multiple Data Sets." *Journal of Machine Learning Research* 7: 1–30.

Dissanayake, T., T. Fernando, S. Denman, S. Sridharan, H. Ghaemmaghami, and C. Fookes. 2020. "A Robust Interpretable Deep Learning Classifier for Heart Anomaly Detection Without Segmentation." *IEEE Journal of Biomedical and Health Informatics* 25: 2162–2171.

Gao, J., X. Song, Q. Wen, P. Wang, L. Sun, and H. Xu. 2020. "Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks." *arXiv Preprint*. arXiv:2002.09545.

García-Gil, D., J. Luengo, S. García, and F. Herrera. 2019. "Enabling Smart Data: Noise Filtering in Big Data Classification." *Information Sciences* 479: 135–152.

Goldstein, M., and A. Dengel. 2012. "Histogram-Based Outlier Score (Hbos): A Fast Unsupervised Anomaly Detection Algorithm." *KI-2012: Poster and Demo Track* 1: 59–63.

Goldstein, M., and S. Uchida. 2016. "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data." *PLoS One* 11: e0152173.

He, Z., X. Xu, and S. Deng. 2003. "Discovering Cluster-Based Local Outliers." *Pattern Recognition Letters* 24: 1641–1650.

Himeur, Y., K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira. 2021. "Artificial Intelligence Based Anomaly Detection of Energy Consumption in Buildings: A Review, Current Trends and New Perspectives." *Applied Energy* 287: 116601.

Hundman, K., V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. 2018. "Detecting Spacecraft Anomalies Using Lstms and Nonparametric Dynamic Thresholding." In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 387–395.

Iglewicz, B., and D. C. Hoaglin. 1993. *Volume 16: How to Detect and Handle Outliers*. Milwaukee, Wisconsin: Quality Press.

Kingma, D. P., and M. Welling. 2013. "Auto-Encoding Variational Bayes." *arXiv Preprint*. arXiv:1312.6114.

Laptev, N., S. Amizadeh, and Y. Billawala. 2015. "S5-a Labeled Anomaly Detection Dataset, Version 1.0 (16m)".

Latecki, L. J., A. Lazarevic, and D. Pokrajac. 2007. "Outlier Detection With Kernel Density Functions." In *MLDM, International Workshop on Machine Learning and Data Mining in Pattern Recognition* vol. 7, 61–75. Berlin, Heidelberg: Springer Berlin Heidelberg.

Li, G., and J. J. Jung. 2022. "Dynamic Graph Embedding-Based Anomaly Detection on Internet of Things Time Series." *Expert Systems* 41, no. 2: e13083. https://doi.org/10.1111/exsy.13083.

Li, Z., Y. Zhao, N. Botta, C. Ionescu, and X. Hu. 2020. "Copod: Copula-Based Outlier Detection." In *2020 IEEE International Conference on Data Mining (ICDM)*, 1118–1123, IEEE.

Li, Z., Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen. 2022. "Ecod: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions." *IEEE Transactions on Knowledge and Data Engineering* 35: 12181–12193.

Liu, F. T., K. M. Ting, and Z.-H. Zhou. 2012. "Isolation-Based Anomaly Detection." *ACM Transactions on Knowledge Discovery From Data (TKDD)* 6: 1–39.

Liu, Y., Z. Li, C. Zhou, et al. 2019. "Generative Adversarial Active Learning for Unsupervised Outlier Detection." *IEEE Transactions on Knowledge and Data Engineering* 32: 1517–1528.

Luengo, J., D. Garíca-Gil, S. Ramírez-Gallego, S. García, and F. Herrera. 2020. *Big Data Preprocessing—Enabling Smart Data*. Switzerland: Springer.

Nassif, A. B., M. A. Talib, Q. Nasir, and F. M. Dakalbab. 2021. "Machine Learning for Anomaly Detection: A Systematic Review." *IEEE Access* 9: 78658–78700.

Pang, G., C. Shen, L. Cao, and A. V. D. Hengel. 2021. "Deep Learning for Anomaly Detection: A Review." *ACM Computing Surveys (CSUR)* 54: 1–38.

Pevnỳ, T. 2016. "Loda: Lightweight On-Line Detector of Anomalies." *Machine Learning* 102: 275–304.

Qiu, J., Q. Du, and C. Qian. 2019. "Kpi-Tsad: A Time-Series Anomaly Detector for Kpi Monitoring in Cloud Applications." *Symmetry* 11: 1350.

Ramaswamy, S., R. Rastogi, and K. Shim. 2000. "Efficient Algorithms for Mining Outliers From Large Data Sets." In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 427–438.

Ruff, L., R. Vandermeulen, N. Goernitz, et al. 2018. "Deep One-Class Classification." In *International Conference on Machine Learning*, 4393–4402, PMLR.

Schölkopf, B., J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. 2001. "Estimating the Support of a High-Dimensional Distribution." *Neural Computation* 13: 1443–1471.

Si, M., and Q. Li. 2020. "Shilling Attacks Against Collaborative Recommender Systems: A Review." *Artificial Intelligence Review* 53: 291–319.

Su, Y., Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei. 2019. "Robust Anomaly Detection for Multivariate Time Series Through Stochastic Recurrent Neural Network." In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2828–2837.

Sugiyama, M., and K. Borgwardt. 2013. "Rapid Distance-Based Outlier Detection via Sampling." In *Advances in Neural Information Processing Systems*, edited by C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, 26. New York: Curran Associates, Inc.

Wu, R., and E. J. Keogh. 2021. "Current Time Series Anomaly Detection Benchmarks Are Flawed and Are Creating the Illusion of Progress." *IEEE Transactions on Knowledge and Data Engineering* 35: 2421–2429.

Wu, R., and E. Keogh. 2023. *Supporting Page for Current Time Series Anomaly Detection Benchmarks Are Flawed and Are Creating the Illusion of Progress*. https://wu.renjie.im/research/anomaly-benchmarks-are-flawed/.

Xu, X., H. Liu, M. Yao, et al. 2019. "Recent Progress of Anomaly Detection." *Complexity* 2019: 2686378. https://doi.org/10.1155/2019/2686378.

Zhao, Y., Z. Nasrullah, and Z. Li. 2019. "Pyod: A Python Toolbox for Scalable Outlier Detection." *Journal of Machine Learning Research* 20: 1–7.

Zhou, X., W. Liang, S. Shimizu, J. Ma, and Q. Jin. 2020. "Siamese Neural Network Based Few-Shot Learning for Anomaly Detection in Industrial Cyber-Physical Systems." *IEEE Transactions on Industrial Informatics* 17: 5790–5798.