Scalable and accurate online multivariate anomaly detection

Rebecca Salles, Benoit Lange, Reza Akbarinia, Florent Masseglia, Eduardo Ogasawara, Esther Pacitti

▶ **To cite this version:**

Rebecca Salles, Benoit Lange, Reza Akbarinia, Florent Masseglia, Eduardo Ogasawara, et al.. Scalable and accurate online multivariate anomaly detection. Information Systems, 2025, 131, pp.102524. ⟨10.1016/j.is.2025.102524⟩. ⟨lirmm-05332273⟩

HAL Id: lirmm-05332273

https://hal-lirmm.ccsd.cnrs.fr/lirmm-05332273v1

Submitted on 26 Oct 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Scalable and Accurate Online Multivariate Anomaly Detection

Rebecca Salles
INRIA
rebeccapsalles@acm.org

Benoit Lange
INRIA
benoit.lange@inria.fr

Reza Akbarinia
INRIA
reza.akbarinia@inria.fr

Florent Masseglia
INRIA
florent.masseglia@inria.fr

Eduardo Ogasawara
CEFET/RJ
eogasawara@ieee.org

Esther Pacitti
INRIA
esther.pacitti@lirmm.fr

Abstract

The continuous monitoring of dynamic processes generates vast amounts of streaming multivariate time series data. Detecting anomalies within them is crucial for real-time identification of significant events, such as environmental phenomena, security breaches, or system failures, which can critically impact sensitive applications. Despite significant advances in univariate time series anomaly detection, scalable and efficient solutions for online detection in multivariate streams remain underexplored. This challenge becomes increasingly prominent with the growing volume and complexity of multivariate time series data in streaming scenarios. In this paper, we provide the first structured survey primarily focused on scalable and online anomaly detection techniques for multivariate time series, offering a comprehensive taxonomy. Additionally, we introduce the Online Distributed Outlier Detection (2OD) methodology, a novel well-defined and repeatable process designed to benchmark the online and distributed execution of anomaly detection methods. Experimental results with both synthetic and real-world datasets, covering up to hundreds of millions of observations, demonstrate that a distributed approach can enable centralized algorithms to achieve significant computational efficiency gains, averaging tens and reaching up to hundreds in speedup, without compromising detection accuracy.

1 Introduction

Processes monitored in almost every scientific field evolve over time, leading to a collection of sequential data referred as time series. When a process demands continuous monitoring, the time series is collected in a continuous flow, characterizing a streaming of time series data. Given the increasing volume of generated data and bandwidth, we anticipate that most future time series analysis work will probably be carried out on data streams. However, most of the currently available methods are not adapted to the dynamic nature of streaming time series [Esling and Agon, 2012].

In addition, most processes are tracked from different angles, with variables that together represent the process as well as possible. For example, climatic data can include temperature, sunshine, precipitation, wind, hygrometry, etc. To take multiple dimensions of a process into account, we need to work with multivariate time series. Unlike univariate time series, for multivariate time series multiple variables are measured in each observation. In that case, one might be interested in finding associations among the different variables.

For most real-world streaming time series, it is often possible to observe the occurrence of a significant change in the behavior of a time series at a certain time t or time interval $\{b, \ldots, e\}$, where b and e represent the beginning and end of the interval, respectively. Such an abnormal behavior change commonly characterizes the occurrence of an anomaly at t or $\{b, \ldots, e\}$ [Guralnik and Srivastava, 1999]. Unlike noise, an anomaly can represent a phenomenon with a specific meaning. For example, it may represent a breach of security or a failure in an industrial process. An anomaly is

defined as a rare pattern or an observation that deviates from most other observations as if generated by a different phenomenon [Wang et al., 2019].

Anomaly detection is the process of identifying anomalous events in time series. With this process, we may be interested in learning about past events (offline) [Pimentel et al., 2014, Ding et al., 2014, Ahmed et al., 2016, Alevizos et al., 2017, Sebestyen et al., 2018, Zhou et al., 2019, Pang et al., 2021, Wang et al., 2019], identifying events in real-time (online) [Zhang et al., 2010, Ahmad et al., 2017, Ariyaluran Habeeb et al., 2019, Munir et al., 2019a], or even predicting future events before they occur [Yan et al., 2004, Salfner et al., 2010, Molaei and Keyvanpour, 2015, Gmati et al., 2019, Zhao, 2021]. The detection of anomalies is a basic function in many surveillance and monitoring applications such as biodiversity monitoring, weather forecasting, agricultural plant surveillance, seismic activity monitoring, medical surveillance, fault detection in industrial systems, intrusion detection, etc [Pimentel et al., 2014].

Online anomaly detection is the problem of detecting anomalies in streaming time series data in real-time, or as soon as possible. Online methods run concurrently with the monitored process and each data observation is processed as it becomes available. Most available anomaly detection methods do not address this scenario [Gama et al., 2014, Ariyaluran Habeeb et al., 2019], especially for large and high-dimensional time series, which demand efficient approaches concerning both computational complexity and memory requirements [Tafazoli and Keogh, 2023]. In particular, the existing work largely overlooks scalability and online multivariate-focused methodologies [Olteanu et al., 2023]. Others do not consider the unique demands of streaming data applications [Schmidl et al., 2022, Zamanzadeh Darban et al., 2024], or focus on univariate detection [Ntroumpogiannis et al., 2023b, Ahmad et al., 2017, Ariyaluran Habeeb et al., 2019, Mason et al., 2019].

This paper addresses these unmet needs by presenting a comprehensive survey that integrates scalability, online anomaly detection, and multivariate time series approaches. We provide a unified taxonomy that categorizes state-of-the-art techniques based on detection strategy, data input, scalability, and algorithm execution. Furthermore, we introduce the Online Distributed Outlier Detection (2OD) methodology for conducting and benchmarking scalable distributed online multivariate anomaly detection. Based on data-window processing [Chen et al., 2016] and inspired by edge computing solutions such as federated learning [Abreha et al., 2022, Agrawal et al., 2022], 2OD enables the analysis of the efficiency-accuracy trade-off of adopting a distributed online anomaly detection approach, especially for multivariate time series anomaly detection methods. It is applicable to most methods available in the literature, even if they were originally designed for offline and centralized execution. 2OD is used to conduct an experimental analysis of the scenarios in which time series anomaly detection benefits from parallel and distributed computing over high-throughput and/or high-dimensional streaming multivariate datasets containing both synthetic and real-world anomalous data, covering up to hundreds of millions of observations. With a distributed setup, the algorithms were able to gain substantial computational efficiency, reaching on average tens and up to hundreds in speedup without compromising detection accuracy performances. These results indicate the potential for distributed online anomaly detection over multivariate time series and motivate future advancements in this area of study. Our contributions are summarized as follows:

- a comprehensive survey on scalable and online multivariate anomaly detection, encompassing a review and contextualization of almost 200 references.
- unified taxonomies of both online multivariate anomaly detection, used for categorizing over 30 state-of-the-art methods, and scalable online multivariate anomaly detection, including the most recent and increasingly prominent techniques.
- 2OD, a well-defined methodology describing an innovative process for conducting and benchmarking distributed online anomaly detection, providing scalability to traditional anomaly detection methods across high-dimensional, high-frequency, high-throughput streaming data contexts. It enables the evaluation of centralized offline, online and distributed approaches regarding efficiency and accuracy, extending the applicability of distributed analysis to previously offline or centralized algorithms.
- an extensible implementation of 2OD with a Spark-like data partitioning distribution setup made publicly available at Github[1].
- an experimental analysis of our methodology suggesting that a distributed approach, particularly for local anomaly detection, provides a scalable solution for real-time applications without a meaningful loss of accuracy.

The remainder of this paper is organized as follows. Sections 2 and 3 present fundamental concepts and a taxonomy regarding multivariate online anomaly detection, respectively. Section 4 describes the problems and current solutions

[1] `https://github.com/RebeccaSalles/2OD`

to scalable and distributed online multivariate anomaly detection. Section 5 presents our proposed methodology and an experimental analysis of the trade-off between efficiency and accuracy in distributed online multivariate anomaly detection. Section 6 contextualizes this paper with related work. Finally, Section 7 concludes.

2 Fundamental concepts on multivariate anomalies

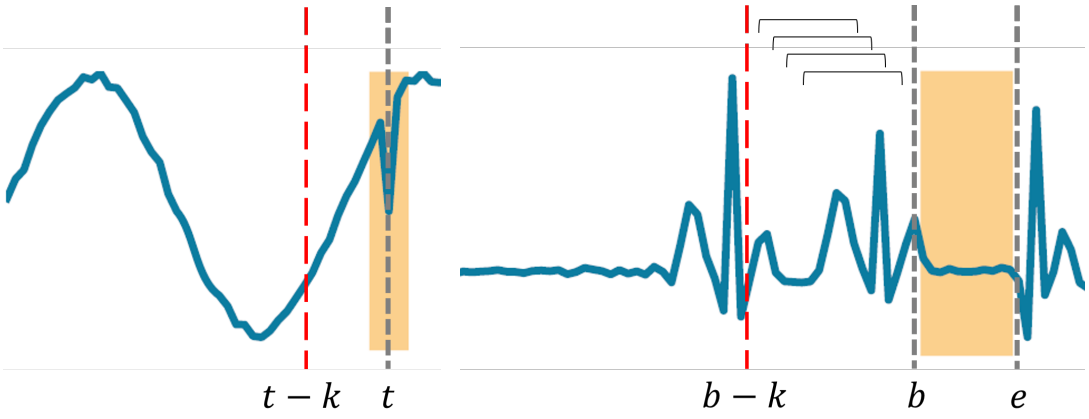
This section presents fundamental concepts of multivariate time series anomalies.

A time series is defined as a sequence of observations of a process collected over time. Each variable of a monitored process produces a sequence of observations referenced as a univariate time series. Generally, a univariate time series X can be considered as a sequence of n random variables, $<x_1, x_2, x_3, \ldots, x_n>$, where x_i represents the value at the ith time point [Esling and Agon, 2012, Shumway and Stoffer, 2017]. The length n of a time series X is also represented by $|X|$.

Anomalies Anomalies correspond to an irregular phenomenon predefined in a particular domain. In the context of time series, anomalies represent significant changes in expected behavior at a certain time or interval [Guralnik and Srivastava, 1999, Chandola et al., 2009]. Thus, the anomalies can be modeled as isolated observations of the remaining nearby data. In general, punctual anomalies of a given time series X can be identified in a simplified way by $pa_u(X, k, \tau)$ using Equation 1, where k is the length of nearby observations, τ is a given threshold, and $ep(x_t, k)$ is the expected value for x_t based on the previous k observations, *i.e.*, $ep(x_t, k) = \mathbb{E}(x_t \mid x_{t-k}, \ldots, x_{t-1})$ [Salles et al., 2024, Ogasawara et al., 2024]:

$$pa_u(X, k, \tau) = \{t, |x_t - ep(x_t, k)| > \tau\} \tag{1}$$

If x_t escapes $ep(x_t, k)$ above a threshold τ with respect to previous k observations [Gujarati, 2021, Lima et al., 2022], it can be considered an anomaly candidate. An example is given in Figure 1a. In streaming applications, k can be defined by the sliding window size.



(a) Punctual anomaly candidate

(b) Sequence anomaly candidate

Figure 1: Examples of different anomaly types with a defined context of k observations (adapted from Wenig et al. [2024])

This formalization can also be adapted for subsequence anomalies, which are collections of contiguous observations with anomalous patterns. In that case, individual values might be within expectations, but their co-occurrence is anomalous [Lai et al., 2021]. Let $X_{(b,e)} =< x_b, \ldots, x_e >$ be a subsequence of size $|e - b|$, with $1 \leq b < e \leq n$, where b and e define the beginning and end of a period of time, respectively. Figure 1b illustrates the example. Subsequence anomalies in time series are also referred to as discords [Linardi et al., 2020]. Let $\Delta p(X_{(b,e)}, k)$ be the set of distances between $X_{(b,e)}$ and all subsequences of the same size $p = |e - b|$ that can be drawn from the previous k observations. If the minimum distance between $X_{(b,e)}$ and any subsequences of the same size from previous k observations is above a threshold τ, then $X_{(b,e)}$ can be considered a subsequence anomaly candidate:

$$sa_u(X, k, \tau) = \{(b, e), min(\Delta p(X_{(b,e)}, k)) > \tau\} \tag{2}$$

The latest time series anomaly taxonomy proposed by Wenig et al. [2024] distinguishes univariate anomalies on two levels regarding anomaly types and locality, namely global and contextual. Global anomalies escape global

expectations, while contextual anomalies differ significantly from their surrounding observations/patterns (context). Equations 1 and 2 define anomalies within the context k, but can be generalized to global anomalies by defining k to the limit of n.

Multivariate anomalies A time series that contains observations related to multiple variables is referred to as a multivariate time series. A multivariate time series S can be defined as the set $\{X_1, X_2, \ldots, X_m\}$, where each X_v ($1 \leq v \leq m$) is a univariate time series corresponding to the m monitored variables contained in S. An example can be drawn from sensor networks, where each X_v corresponds to a specific sensor [Ang et al., 2023]. It is often represented as a matrix $S = (X_1, X_2, \ldots, X_m)^\top$, where X_v is a n-dimensional vector $X_v = x_{v,1}, \ldots, x_{v,n}$ and $x_{v,t}$ ($1 \leq t \leq n$) is an observation of variable X_v in the time point t. The matrix S is therefore of size m by n. Multivariate time series variables are commonly referred to as features or dimensions, and a high number of m indicates that it presents high-dimensionality.

If there is a punctual anomaly in time t for any X_v, it is also considered an anomaly for S. In particular, if $t \in pa_u(X_v, k, \tau)$ for multiple values of v, t is considered a multivariate punctual anomaly as defined by $pa_m(S, k, \tau)$ in Equation 3. A z-ary anomaly is present in z variables of S. Analogously, if (b, e) defines a sequence anomaly based on $sa_u(X_v, k, \tau)$ for any X_v, it also defines a sequence anomaly in S, and a multivariate sequence anomaly if this is true for multiple X_v according to $sa_m(S, k, \tau)$ in Equation 4.

$$pa_m(S, k, \tau) = \{t \mid t \in pa_u(X_v, k, \tau), X_v \in S\} \tag{3}$$

$$sa_m(S, k, \tau) = \{(b, e) \mid (b, e) \in sa_u(X_v, k, \tau), X_v \in S\} \tag{4}$$

3 Multivariate online anomaly detection

This section presents the most relevant strategies for online anomaly detection in multivariate time series. Figure 2 provides a taxonomy for the detection of anomalies in multivariate time series. Inspired by the work of Schmidl et al. [2022] and Ogasawara et al. [2021], it is divided into six categories based on the set of parameters found in most literature reviews. The categories cover various aspects of anomaly detection methodologies, encompassing data input, execution and main contribution of detection algorithms, areas of study and general strategies for detection, and learning modes.

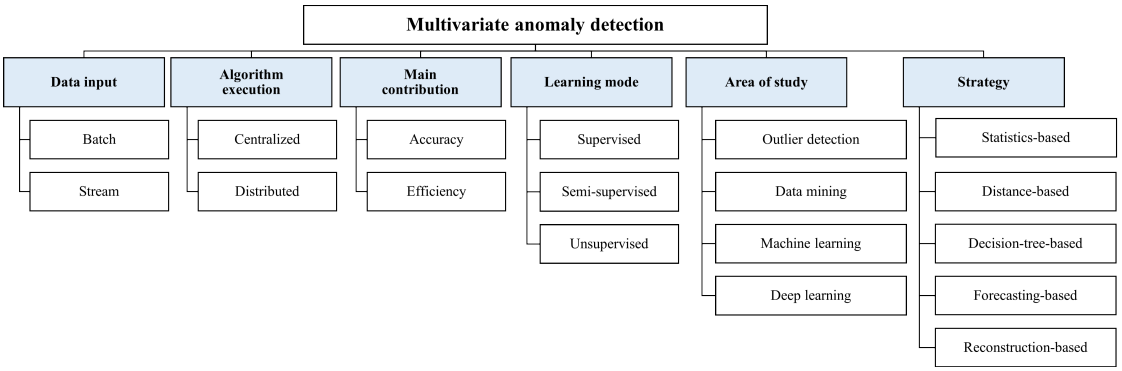


Figure 2: Taxonomy of multivariate anomaly detection techniques (inspired by Schmidl et al. [2022] and Ogasawara et al. [2021])

Data input Commonly, detection methods are designed to take a batch input containing the time series in its entirety and nearly all data instances can be compared with past and future behavior [Sylligardos et al., 2023, Schmidl et al., 2022, Fan et al., 2022, De Paepe et al., 2020, Choi et al., 2020]. However, recent online detection methods take streaming time series as input, where anomalies can be identified in real-time [Ntroumpogiannis et al., 2023a, Ang et al., 2023, Lu et al., 2023, He et al., 2023, Ahmed et al., 2022, Bäßler et al., 2022, Ahmed et al., 2021, Jacob et al., 2021, Boniol et al., 2021c,b, Toliopoulos et al., 2020, Tran et al., 2020, Munir et al., 2019b, Ariyaluran Habeeb et al., 2019]. Let X be a streaming time series $<\ldots, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}, \ldots>$. At each point in time t, a model trained on previous observations $<\ldots, x_{t-2}, x_{t-1}>$ is used to determine whether the current behavior of the system is unusual. This determination must be made in real-time by the arrival of the next input x_{t+1}. For this, the time series model must

be continuously updated [Ahmad et al., 2017]. The online anomaly detection problem is further discussed in Section 3.2.

Algorithm execution Detection algorithms are generally executed centralized. However, the increasing availability of large time series datasets demands the use of big data techniques [Gaspar et al., 2017], notably distributed and parallel computing [Ahmed et al., 2022, 2021, Jacob et al., 2021, Toliopoulos et al., 2020, Ariyaluran Habeeb et al., 2019, Özsu and Valduriez, 2019]. The distributed online anomaly detection problem is discussed in Section 4.

Main contribution of an algorithm Anomaly detection methods focus primarily on improving detection accuracy [Boniol et al., 2021c,b, De Paepe et al., 2020, Choi et al., 2020, Munir et al., 2019b], that is, the ability of a detection method to correctly label data instances as normal or anomalous. However, as time series datasets increase in scale and complexity [Özsu and Valduriez, 2019]. Recent methods improve detection efficiency as one of their main contributions.

Learning modes Based on the availability of the *normal* or *abnormal* labels, anomaly detection methods can be trained in three modes. In supervised mode, the methods assume the availability of a training dataset with labeled instances, and algorithms model both normal and abnormal behavior in the time series. In semi-supervised mode, algorithms try to learn only the normal behavior, assuming the training data contains only normal instances, or labeled instances only for the normal class. On the other hand, methods that operate in unsupervised mode do not require labels in training data, and thus are most widely applicable. In that case, algorithms separate anomalous instances from normal instances of the time series without prior knowledge, and no training step is required [Schmidl et al., 2022]. The methods in this category implicitly assume that normal instances are more frequent than abnormal [Chandola et al., 2009].

Area of study Current anomaly detection methods result from research in many different areas of study, including outlier detection (OD) in statistical analysis, data mining (DM) techniques, classic machine learning (ML) or deep learning (DL) [Schmidl et al., 2022]. In anomaly detection, parametric model-driven techniques are frequently used due to their lower computational costs and interpretability [Ahmad et al., 2017, Ariyaluran Habeeb et al., 2019]. In contrast, non-parametric data-driven methods, generally based on machine (deep) learning, present more flexibility and are not restricted to certain kinds of problem or function. DL approaches have received recent focus, frequently using a low dimensional latent representation for addressing issues derived from high-dimensionality in time series, as a form of intrinsic dimensionality recovery [Olteanu et al., 2023]. However, they can still require an infeasible computational cost for large-scale time series and are prone to overfitting [Salles et al., 2019].

3.1 Detection strategies

The anomaly detection methods found in the literature are usually based on five general strategies: statistics-based, distance-based, decision-tree-based, forecasting-based and reconstruction-based analysis [Schmidl et al., 2022, Han et al., 2022a]. This section provides an overview of the general strategies for multivariate anomaly detection and exemplifies their representative detection methods.

3.1.1 Statistics-based

Statistics-based anomaly detection methods generally identify deviations from the data distribution, with anomalies often found in the tails of the distributions. The anomaly scores are usually measured using probabilities, likelihoods, or distances of the observations or subsequences to the previously estimated distributions. Methods are generally unsupervised and distributions are calculated over the whole time series or sliding windows. Other probabilistic approaches involve an estimation of the density of the normal class [Alevizos et al., 2017].

Empirical-Cumulative-distribution-based Outlier Detection (ECOD) [Li et al., 2023] is a state-of-the-art representative statistics-based multivariate anomaly detection method. It assumes that anomalies tend to appear in the tails of a given data distribution. It computes the empirical cumulative distribution for each variable and estimates the tail probability for each observation. The aggregation of the estimated tail probabilities across all variables is used to compute anomaly scores. Figure 3 shows an example of an outlier score distribution computed by ECOD, where the outlier scores present a separate distribution and low density. Copula-Based Outlier Detection (COPOD) [Li et al., 2020] is another example of multivariate method based on the analysis of deviations from the data distribution. Inspired by copulas for modeling multivariate data distribution, it constructs an empirical copula and uses it to estimate the tail probabilities of each observation. Similarly to calculating an anomalous p-value, estimated tail probabilities are considered anomaly scores. Other representative histogram-based unsupervised methods include Histogram Based Outlier Scores (HBOS) [Goldstein and Dengel, 2012] and Light Online Detector of Anomalies (LODA) [Aguilera-Martos et al., 2023].

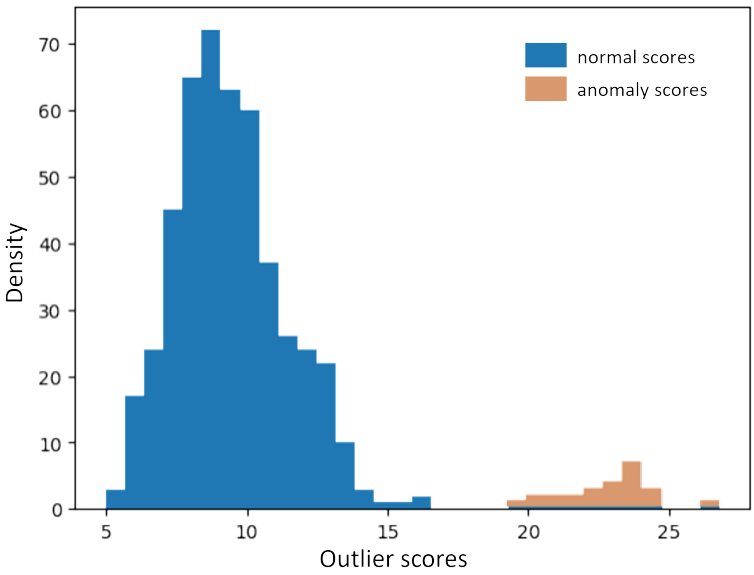


Figure 3: Example of outlier score distribution computed by ECOD (adapted from Kuo [2023])

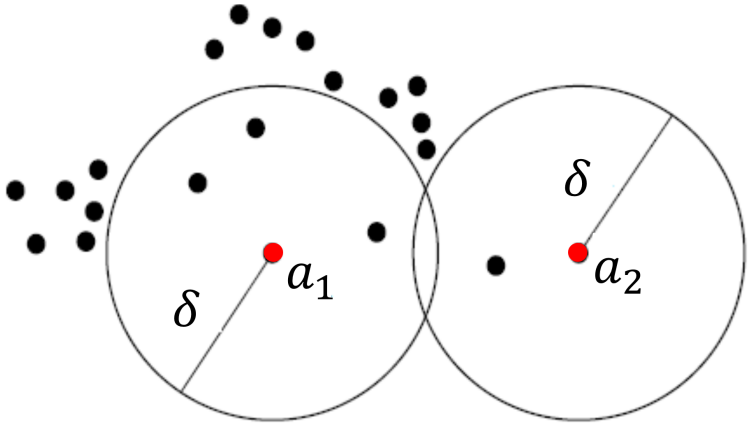


Figure 4: Example of KNN anomaly identification (adapted from Tran et al. [2020]

3.1.2 Distance-based

Distance-based anomaly detection methods use distance metrics to compare observations or subsequences of a time series [Gu et al., 2019]. In that case, anomalous subsequences are expected to have larger distances [Schmidl et al., 2022]. Most of the methods are unsupervised and take sliding windows as input. For computing the distances, the methods in this category adopt one of three main approaches: nearest-neighbor-based, cluster-based, or model-based.

For *nearest-neighbor-based* methods, anomaly scores are based on the distance of observations or subsequences to their nearest neighbors. A representative method is the k nearest neighbors (KNN) [Angiulli and Pizzuti, 2002]. In its simplest form, KNN assumes that an observation x_t is an anomaly considering parameters k and τ if no more than k time series observations are at a distance of τ or less from x_t. [Ramaswamy et al., 2000]. Figure 4 presents an example of a time series dataset that has two anomalies, a_1 and a_2, for $k = 4$. The Principal Component Analysis (PCA) method [Shyu et al., 2006, Aggarwal, 2013] is also based on the distances between time series observations, but the neighbors to which they are compared are projections [Ariyaluran Habeeb et al., 2019]. A covariance matrix of the data is decomposed into orthogonal vectors, called eigenvectors. Outliers are different from normal data points, which is more obvious on the hyperplane constructed by the eigenvectors with large eigenvalues. Outlier scores are obtained as the sum of the weighted distances between each observation and its projection on the hyperplane constructed by the selected eigenvectors.

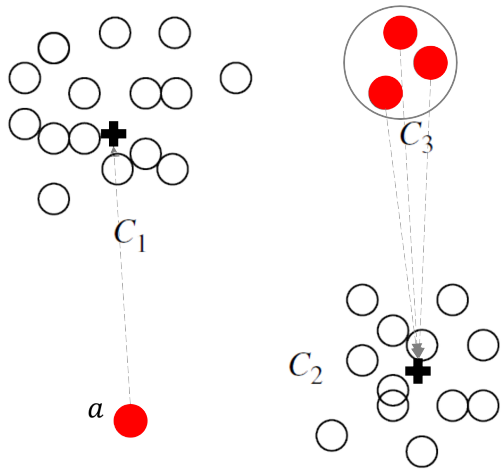


Figure 5: Example of CBLOF anomaly identification (adapted from Han et al. [2022a]

Another state-of-the-art technique that contributes to this category of detection methods is the Matrix profile [Yeh et al., 2016]. Matrix profile based methods efficiently compute the minimal distance between each time series subsequence and its nearest non-self neighbor subsequences [Audibert et al., 2020]. A low value in the matrix profile indicates that a subsequence has at least one similar subsequence in the original series, and a high value indicates a discord, which may suggest potential anomalous subsequences. Therefore, the matrix profile can be used as an anomaly score [Audibert et al., 2020].

Cluster-based methods produce clusters of similar multidimensional observations/subsequences. The anomaly scores are then based on the distances to the corresponding (closest) cluster centroids. Representative examples are the Cluster-Based Local Outlier Factor (CBLOF) [He et al., 2003] and Outlier Detection with Minimum Covariance Determinant (MCD) [Hardin and Rocke, 2004]. Figure 5 shows an example of the anomaly identification process of CBLOF. Three clusters are formed, C_1 to C_3, and 4 anomalies are identified. The anomaly a is too far from its closest large cluster, C_1. Also, C_3 is too small and its observations are too far from their closest large cluster, C_2 [Han et al., 2022a].

Distance-based methods that adopt a *model-based* approach build a reference model of normal behavior to which the subsequences are compared. It is one of the most commonly adopted and is based on analysis of model deviation. In this approach, first a statistical or machine learning model is fitted to the available data. The anomalies are then identified as the observations that most deviate from the fitted model [Han et al., 2022a]. For example, Liu et al. [2015] focus on multivariate anomaly detection based on the deviation from an ARIMA model above a threshold. An ensemble for all variables is computed and compared with Multivariate Euclidean distance (MED) and Linear prediction filters (LPF). Other works, on the other hand, rely on deep neural networks [Kieu et al., 2018] for multivariate anomaly detection. The Deep Support Vector Data Description (DeepSVDD) [Ruff et al., 2018b] constructs a deep one-class classification model to identify multivariate anomalies.

3.1.3 Decision-tree-based

Decision-tree-based anomaly detection methods build an ensemble of random trees that partition the data samples, either observations or subsequences. The methods select random features and random split values as tree nodes to eventually isolate the samples in the tree leaves. The number of splits required to isolate a sample is measured by the average path length on all random trees. The anomaly scores are based on the length of the path. The idea behind this strategy is that anomalous samples are easier to separate as they are closer to the root of the tree, having shorter paths [Schmidl et al., 2022]. Most of the methods in this category are based on the Isolation Forest (iForest) algorithm [Tony et al., 2012]. It is based on the assumption that anomalies are more easily separable from the majority of the data. Figure 6 illustrates the construction of a tree and the corresponding partitions that isolate the observations. The partitions are made on a two-feature map, where the red dot is the farthest from the others, followed by the orange and yellow dots. It takes only one "cut"/split to separate the red dot. The next "cuts"/splits isolate the orange and yellow dots. The number of "cuts" / splits corresponds to the depth of the tree. The inverse of the depth of each dot located in the tree is a proxy for its anomaly score [Kuo, 2023].

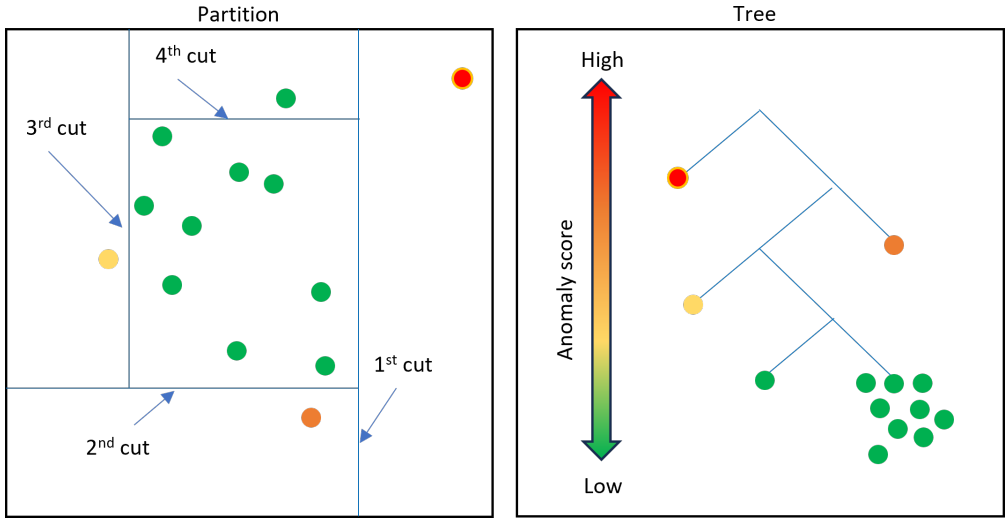


Figure 6: Example of iForest tree construction and anomaly identification (adapted from Kuo [2023]

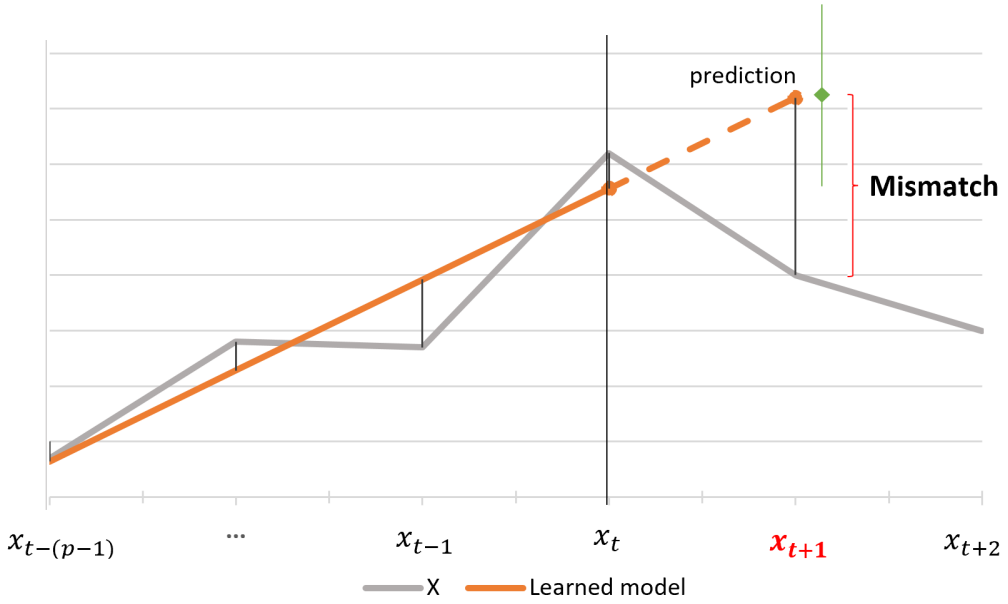


Figure 7: Forecasting-based anomaly detection strategy illustration

3.1.4 Forecasting-based

Forecasting-based anomaly detection methods predict future states of a time series to identify mismatches with expected behavior [Salfner et al., 2010, Pimentel et al., 2014]. Figure 7 illustrates this strategy that has been mainly studied in the context of the very recent area of event prediction [Zhao, 2021]. It is based on event detection from time series prediction, which involves two steps. The first step regards time series prediction. Past values of the time series are used to derive a predictive model (in orange), which can be statistical or machine learning based. Next, the derived model is applied for time series prediction and future values of the time series are returned. The second step concerns online event detection per se. Anomalies are identified based on the deviations from the predicted time series above an acceptable threshold [Mehrmolaei and Keyvanpourr, 2015] (in green). Most methods are semi-supervised and use sliding windows, performing one-step-ahead forecasts. Representative forecasting-based methods include the Long Short-Term Memory neural network for Anomaly Detection (LSTM-AD) [Malhotra et al., 2015], Telemanom [Hundman et al., 2018b] (LSTM-based) and DeepAnT [Munir et al., 2019c] (CNN-based). The latter is able to detect

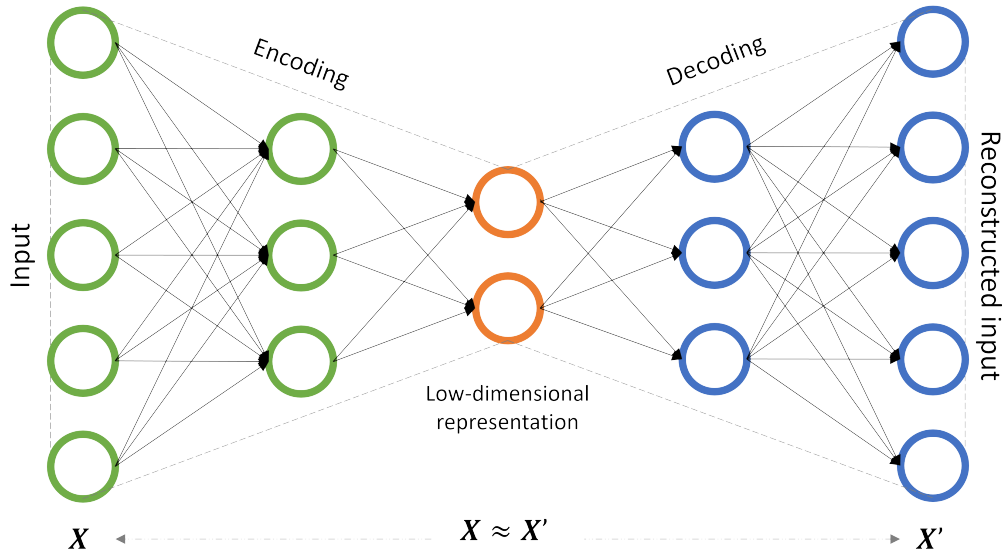


Figure 8: Network architecture of an Autoencoder model

pattern shifts and frequencies particularly well. Forecasting-based anomaly detection methods are typically motivated by specific application domains [Zhao, 2021].

3.1.5 Reconstruction-based

Reconstruction-based anomaly detection methods build a model of normal behavior by encoding time series subsequences in a (low dimensional) latent space. Most methods are semi-supervised, taking as input training sliding windows of assumed normal behavior. The subsequences of the test series are reconstructed from the latent space. Anomalous subsequences in the test series cannot be reconstructed by the model, and anomalies are identified by the difference between the original and reconstructed subsequences. Unsupervised methods, on the other hand, encode the input subsequences of the test series into a predefined latent space. As a result, there is deliberate loss of information, and not all details of the original subsequences can be recreated. The differences between the original and reconstructed subsequences are anomaly scores [Schmidl et al., 2022].

Clear representatives of reconstruction-based anomaly detection methods are based on autoencoders (AE) [Sakurada and Yairi, 2014], which are currently one of the best machine learning models for performing anomaly detection and diagnosis in multivariate time series data [Zhang et al., 2019]. Figure 8 illustrates the network architecture for an ordinary AE model. An autoencoder is a type of artificial neural network that is used to learn efficient data encodings in an unsupervised manner. First, it learns a representation (encoding) for a set of data, typically for dimensionality reduction. Next, the autoencoder tries to generate from the reduced encoding (decoding) a representation as close as possible to its original input [Provotar et al., 2019]. Other representative methods include Unsupervised Anomaly Detection (USAD) [Audibert et al., 2020], and OmniAnomaly [Su et al., 2019].

The latest work adopting this strategy is based on generative models [Correia et al., 2024], most commonly Variational Auto-Encoders (VAEs) [Kingma, 2013] and Generative Adversarial Network (GANs) [Goodfellow et al., 2014]. The former differs from traditional autoencoders, as the low-dimensionality representation is not mapped to a vector z but a distribution (μ_z,σ_z). In this context, the encoder is also referenced as the recognition model and the decoder as the generative model. While autoencoders are made up of encoder-decoder architectures, GANs, on the other hand, are composed of generator and discriminator networks. First, a generator network is trained to generate a plausible time series window sampled from a normal distribution. Next, a discriminator network tries to classify both the generated window and the training example. After iterations, the generator becomes better at generating samples, and discriminating between the generated window and the real data becomes increasingly harder [Correia et al., 2024]. For anomaly detection, the generator generates expected time series behavior, while the discriminator is responsible for distinguishing between normal and anomalous observations. Representative generative model-based methods include Long Short-Term Memory Variational Auto-Encoders (LSTM-VAE) [Audibert et al., 2021], and BiGAN [Schlegl et al., 2017]. In addition, recently, MTAD_RF has combined the problems of reconstruction and forecasting, leveraging VAE and single-step forecast by MLP to jointly detect multivariate time series anomalies [Qin et al., 2023].

3.2 Online multivariate anomaly detection

Online anomaly detection entails analyzing streaming data to identify the occurrence of anomalies in real-time. In contrast to offline detection (batch processing), the full dataset is not available, and the system needs to observe each data record as it arrives [Ahmad et al., 2017].

3.2.1 Online detection methods

In this section, we discuss some of the most relevant online anomaly detection methods for multivariate streaming time series. Table 1 presents 33 methods gathered from systematic review of the literature and categorized according to our taxonomy. The methods are organized based on their characteristic detection strategy and the first year of publication of their application to online anomaly detection. Information on their area of study, their algorithm execution (distributed or not), their possible contribution to detection efficiency, and their characteristic learning mode (unsupervised or not) is also included. We also provide information on the availability of their implementation and corresponding programming language.

Table 1: Some of most relevant online anomaly detection methods for multivariate streaming time series

Strat. base	Year	Method	Area	Distr. exec.	Effcy. contr.	Unsup.	Impl. avl.	Lang./ lib.
Stats.	2022	ECOD [Li et al., 2022, Ang et al., 2023]	OD	√	√	√	√	Python
	2018	RS-Hash [Sathe and Aggarwal, 2018, Ntroumpogiannis et al., 2023b]	ML		√	√	√	Python
	2016	LODA [Pevnỳ, 2016, Ntroumpogiannis et al., 2023b]	DM		√	√	√	Matlab
Distance	2023	CAD [Ang et al., 2023]	ML		√	√	√	C++
	2023	DAMP [Lu et al., 2023]	DM		√	√	√	Matlab
	2022	SparX [Zhang et al., 2022a]	ML	√	√	√	√	Python
	2022	RCAD [Ahmed et al., 2022]	ML	√	√	√	√	Python
	2022	OeSNN-UAD [Bäßler et al., 2022]	ML			√	√	Cython
	2021	NormA [Boniol et al., 2021a, Lu et al., 2023]	DM		√	√	√	Python
	2021	NSIBF [Feng and Tian, 2021, Ahmed et al., 2022]	ML			√	√	Python
	2020	CPOD [Tran et al., 2020, Ntroumpogiannis et al., 2023b]	ML		√	√	√	Java
	2020	STARE [Yoon et al., 2020, Ntroumpogiannis et al., 2023b]	ML		√	√	√	Java
	2018	SCRIMP [Zhu et al., 2018b, Lu et al., 2023]	DM		√	√	√	Python
	2018	XSTREAM [Manzoor et al., 2018, Ntroumpogiannis et al., 2023b]	ML		√	√	√	C++
	2018	DeepSVDD [Ruff et al., 2018a]	ML		√	√	√	Python
	2017	HTM [Ahmad et al., 2017, Ariyaluran Habeeb et al., 2019]	ML	√	√	√	√	Python
	2014	LEAP [Cao et al., 2014, Ntroumpogiannis et al., 2023b]	ML		√	√	√	Java
	2011	MCOD [Kontaki et al., 2011, Ntroumpogiannis et al., 2023b]	ML			√	√	Java
	2006	PCA [Shyu et al., 2006, Ariyaluran Habeeb et al., 2019, Munir et al., 2019a]	ML			√	√	Python
	2005	SPIRIT [Čulić Gambiroža et al., 2023, Bäßler et al., 2022]	ML			√	√	Cython
	2000	LOF [Breunig et al., 2000, Ang et al., 2023]	OD			√	√	Python
Decis.-tree	2016	RRCF [Guha et al., 2016, Ntroumpogiannis et al., 2023b]	ML			√	√	Java
	2012	iForest [Liu et al., 2012, Ang et al., 2023]	OD			√	√	Python
	2011	HST [Tan et al., 2011, Ntroumpogiannis et al., 2023b]	ML		√	√	√	Java
Fcst.	2018	Telemanom [Hundman et al., 2018a, Lu et al., 2023]	DL				√	Tensorflow
	2015	LSTM-AD [Malhotra et al., 2015, Jacob et al., 2021]	DL				√	Pytorch
Reconstruction	2021	Rcoders [Abdulaal and Lancewicki, 2021, Ang et al., 2023]	DL			√		N/A
	2021	PMUNET [Ahmed et al., 2021]	DL	√	√	√		N/A
	2020	USAD [Audibert et al., 2020, Ang et al., 2023, Lu et al., 2023, Ahmed et al., 2022]	DL			√	√	Pytorch
	2019	AE [Schreyer et al., 2017, Lu et al., 2023, Jacob et al., 2021, Berahmand et al., 2024]	DL				√	Tensorflow
	2019	OmniAnomaly [Su et al., 2019, Ahmed et al., 2022]	DL				√	Tensorflow
	2018	LSTM-VAE [Park et al., 2018, Lu et al., 2023]	DL				√	Tensorflow
	2017	BiGAN [Schlegl et al., 2017, Jacob et al., 2021]	DL			√	√	Python

The selected methods, applied to anomaly detection, were published from 2000 to 2023. Most multivariate online anomaly detection methods are based on distance. As it happens, the high velocity of time series data streams leaves little opportunity for experts to label the observations. Moreover, a process of hyperparameter optimization is especially challenging when data properties evolve over time [Ntroumpogiannis et al., 2023b]. For these reasons, the relatively simple, unsupervised and nonparametric nature of distance-based methods makes them particularly suitable

for the scenario of online anomaly detection [Giannoulidis et al., 2024]. The same applies to statistics-based and decision-tree-based methods, which are also mostly unsupervised.

On the other hand, forecasting-based or reconstruction-based methods are generally semi-supervised, demanding training on time series data that do not contain anomalies, which in turn is a condition with low guarantees in streaming data. Furthermore, the detection process for forecasting-based methods involves continually analyzing the current state to predict expectations for the next future streaming observations [Schmidl et al., 2022], which can be slow and costly depending on the predictive model adopted.

Reconstruction-based methods, such as Autoencoder-based, involve offline training and online inference. Approaches with both online training and inference have not yet been developed [Correia et al., 2024]. Moreover, for such methods, anomaly detection is not the direct goal, but dimensionality reduction. In that case, the main challenge is to choose the right degree of compression for the latent space. With no compression, an autoencoder represents the identity function, and with extreme high compression (one single value), it represents the mean. Choosing the right balance, as a hyperparameter, is hard in an unsupervised scenario [Ruff et al., 2018b]. In addition, the performance of the autoencoder anomaly detection can be sensitive to noise or outliers in the training data, which can make the resulting model overly tolerant to anomalies [Berahmand et al., 2024]. Ultimately, the models used by both forecasting-based and reconstruction-based methods may need to be periodically rebuilt to adapt to concept drifts.

The majority of the overall methods are derived from studies in ML, which have been the topic of attention in the last decade, followed by deep learning [Wagner et al., 2023]. Except for Rcoders [Abdulaal and Lancewicki, 2021, Ang et al., 2023] and PMUNET [Ahmed et al., 2021], all methods made their implementation publicly available via code scripts or generic frameworks (mainly Github) written using the corresponding listed programming language (mostly Python and Java) or library (Tensorflow and Pytorch).

Next, we discuss representative methods for all detection strategies.

Representative detection methods A representative example of a distance-based method (clustering-based) adapted to an online scenario is given by the Micro cluster outlier detection (MCOD) [Kontaki et al., 2011]. Figure 9 illustrates the execution over two consecutive sliding windows of a time series with two variables, namely $F1$ and $F2$. In the first window, the microcluster $MC1$ is constructed. Observation $p4$ is normal as it has $K = 3$ neighbors within distance R. Similarly, observations $p5$ and $p7$ are anomalies. For the next sliding window, the process continues with the construction of the microcluster $MC2$. Now $p7$ is a normal observation, while $p6$ becomes an anomaly because its previous neighbors have been removed from the sliding window. The observations $p10$ and $p12$ are also anomalies, since they have less than $K = 3$ neighbors within distance R [Ntroumpogiannis et al., 2023b].

Windowed approaches are also commonly adopted for the adaptation of methods from the remaining strategies to an online scenario. In that case, it is important to aggregate the results obtained for the same observation in different data windows. This is also the case for the methods presented in Table 1. Among them, ECOD [Li et al., 2022, Ang et al., 2023], RS-Hash [Sathe and Aggarwal, 2018, Ntroumpogiannis et al., 2023b], and LODA [Pevnỳ, 2016, Ntroumpogiannis et al., 2023b] are statistics-based. Both RS-Hash and LODA are histogram-based, while RS-Hash is composed of an ensemble.

In addition to MCOD, among the remaining distance-based methods, DAMP [Lu et al., 2023], CAD[Ang et al., 2023], NormA [Boniol et al., 2021a, Lu et al., 2023], LEAP [Cao et al., 2014, Ntroumpogiannis et al., 2023b], SCRIMP [Zhu et al., 2018b, Lu et al., 2023] and LOF [Breunig et al., 2000, Ang et al., 2023] are nearest-neighbor-based. LOF focuses on identifying local anomalies, while CAD uses graph representation to monitor correlation anomalies. It monitors communities of correlated variables and also allows anomaly localization among different variables. Furthermore, DAMP, NormA, and SCRIMP are Matrix-profile-based.

The methods CPOD [Tran et al., 2020, Ntroumpogiannis et al., 2023b], STARE [Yoon et al., 2020, Ntroumpogiannis et al., 2023b], XSTREAM [Manzoor et al., 2018, Ntroumpogiannis et al., 2023b], and SparX [Zhang et al., 2022a] are clustering-based. In particular, CPOD outperforms MCOD in terms of runtime and memory usage [Ntroumpogiannis et al., 2023b]. XSTREAM [Manzoor et al., 2018] is composed of an ensemble capable of handling feature evolving streams and concept-drifts. In addition, XSTREAM performance was validated to outperform baselines [Ntroumpogiannis et al., 2023b] and to exhibit robustness to hyperparameters and a high-dimensional feature space, which admits mixed-type data [Zhang et al., 2022a]. SparX builds on XSTREAM adapting it mainly for massive-scale cloud-resident data, although by design it can also handle distributed evolving streaming input.

The methods RCAD [Ahmed et al., 2022], OeSNN-UAD [Bäßler et al., 2022], NSIBF [Feng and Tian, 2021, Ahmed et al., 2022], DeepSVDD [Ruff et al., 2018a], HTM [Ahmad et al., 2017, Ariyaluran Habeeb et al., 2019], SPIRIT [Čulić Gambiroža et al., 2023, Bäßler et al., 2022], PCA [Shyu et al., 2006, Ariyaluran Habeeb et al., 2019, Munir et al., 2019a] are classification model-based, whose model describes normal behavior. NSIBF uses a dynamical state-space

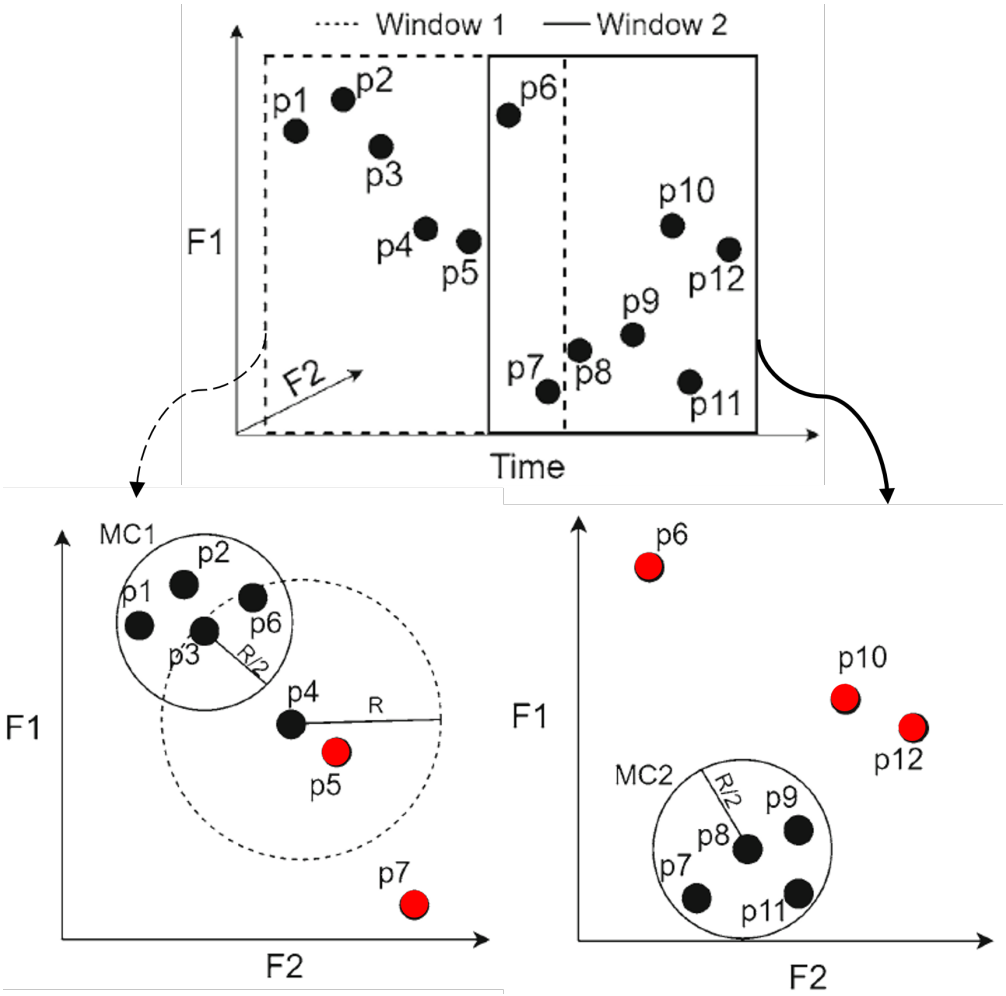


Figure 9: Example of MCOD running on sliding windows (adapted from Ntroumpogiannis et al. [2023b])

model and a Bayesian filtering algorithm. Both HTM and RCAD are based on the Hierarchical Temporal Memory (HTM) model, while RCAD is composed of an ensemble of models with transfer learning.

Decision-tree-based methods include RRCF [Guha et al., 2016, Ntroumpogiannis et al., 2023b], HST [Liu et al., 2012, Ang et al., 2023], and iForest [Tan et al., 2011, Ntroumpogiannis et al., 2023b]. RRCF and HST are able to handle sequence anomalies. However, for forecasting-based methods, Telemanom [Hundman et al., 2018a, Lu et al., 2023] and LSTM-AD [Malhotra et al., 2015, Jacob et al., 2021], the adopted predictive model is the LSTM. Finally, representative online reconstruction-based methods include Rcoders[Abdulaal and Lancewicki, 2021, Ang et al., 2023], PMUNET [Ahmed et al., 2021], USAD [Audibert et al., 2020, Ang et al., 2023, Lu et al., 2023, Ahmed et al., 2022], AE [Schreyer et al., 2017, Lu et al., 2023, Jacob et al., 2021, Berahmand et al., 2024], OmniAnomaly [Su et al., 2019, Ahmed et al., 2022], LSTM-VAE [Park et al., 2018, Lu et al., 2023], and BiGAN [Schlegl et al., 2017, Jacob et al., 2021]. PMUNET is deep learning-based and data-driven, and is able to detect concept drifts. A comprehensive review of the state-of-the-art works adopting a reconstruction-based strategy is provided by Correia et al. [2024].

Efficient detection methods An important property of online detection methods is their contribution to execution performance, especially in a scenario of high-velocity streams of increasingly large high-dimensional time series. In this context, among the mapped online detection methods 16 of them claimed to reduce the use of computational resources, such as runtime and memory usage, and others claimed to be scalable in terms of data volume and / or dimensions. These include RS-Hash [Sathe and Aggarwal, 2018, Ntroumpogiannis et al., 2023b], and LODA [Pevnỳ, 2016, Ntroumpogiannis et al., 2023b] for statistics-based. For distance-based that would be CAD [Ang et al., 2023], DAMP [Lu et al., 2023], NormA [Boniol et al., 2021a, Lu et al., 2023], SCRIMP [Zhu et al., 2018b, Lu et al., 2023],

XSTREAM [Manzoor et al., 2018], SparX [Zhang et al., 2022a], RCAD [Ahmed et al., 2022], CPOD[Tran et al., 2020, Ntroumpogiannis et al., 2023b], STARE [Yoon et al., 2020, Ntroumpogiannis et al., 2023b], DeepSVDD [Ruff et al., 2018a], HTM [Ahmad et al., 2017, Ariyaluran Habeeb et al., 2019], and LEAP [Cao et al., 2014, Ntroumpogiannis et al., 2023b]. Furthermore, there is only one claimed efficient method for the decision-tree-based strategy, HST [Liu et al., 2012, Ang et al., 2023], and one for reconstruction-based, PMUNET [Ahmed et al., 2021]. Moreover, among these methods only SparX, RCAD, HTM, and PMUNET addressed distributed computing for anomaly detection. ECOD also proposes a distributed execution of its algorithm, despite its main contribution being focused on detection accuracy. The efficiency contribution of each of these methods is further discussed in the next section.

4 Scalable and distributed online anomaly detection

Recent advances in sensors and the thriving of the Internet of Things (IoT) caused a massive increase in the volume of time series data produced worldwide [Gaspar et al., 2017]. As the number of dimensions increases exponentially, a corresponding increase in data is required to build accurate models, making the anomaly detection process more complex and computationally expensive [Zhai et al., 2014]. Traditional methods often struggle with algorithmic instability and high-dimensionality, leading to greater resource consumption. However, the development of big data technologies, such as parallel and distributed processing frameworks, can minimize such computational overhead. Using cloud infrastructure and modern processors, systems can now handle massive datasets in real-time, reducing the overall computational expense and improving the accuracy of anomaly detection in high-dimensional environments [Fan et al., 2014].

In addition, online anomaly detection algorithms, such as sliding window-based methods, are viable solutions to efficiently manage high-volume data streams [Chen et al., 2016]. These approaches enable real-time anomaly detection by processing data as it arrives, improving detection latency, and adapting to dynamic data flows.

In this paper, we provide a taxonomy for the main solutions adopted in the literature to address the problem of online and scalable multivariate time series anomaly detection (Figure 10).

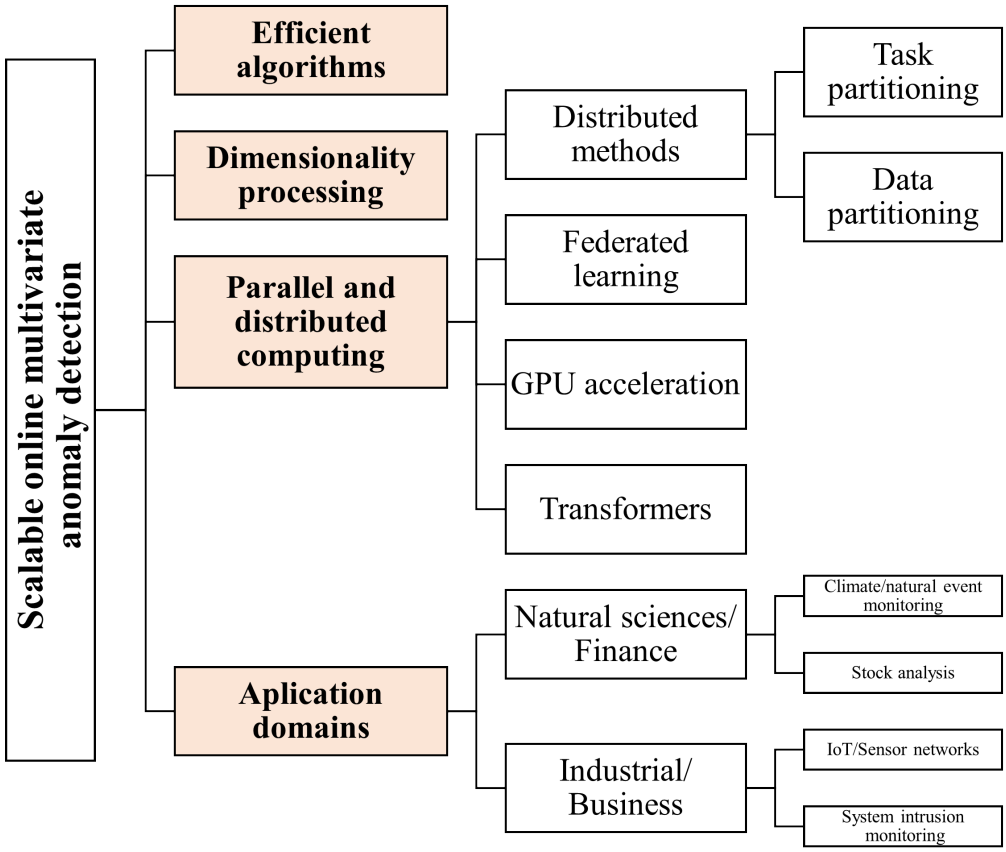


Figure 10: Taxonomy of scalable online multivariate anomaly detection techniques

Figure 10 illustrates the main approaches to improve scalability for multivariate anomaly detection, namely, (i) efficient anomaly detection algorithms, (ii) efficient data dimensionality processing, and (iii) parallel and distributed computing. The first approaches, (i) and (ii), encompass the problem of complexity management. Efficient algorithms focus on minimizing computational complexity and reducing the use of computational resources, such as CPU and memory. Dimensionality processing approaches focus either on dimensionality reduction or efficient techniques to deal with similarity search on high-dimensional time series.

In addition to the development of scalable methods, the latest advancements in the field of parallel and distributed computing include the adoption of: federated learning, a promising edge computing solution [Abreha et al., 2022], GPU acceleration [Zhu et al., 2021], and transformers, a prominent robust AI machine learning architecture that uses parallelism as a key property [Merrill and Sabharwal, 2023, Sanford et al., 2024].

The next subsections discuss and give several examples of recent work on each topic of our taxonomy presented in Figure 10. Finally, Subsection 4.4 presents some of the latest relevant applications and use cases of scalable multivariate anomaly detection in an online scenario.

4.1 Efficient detection algorithms

A fundamental property of online detection methods, especially in the case of large, high-velocity and high-dimensional time series streams, is their contribution to execution efficiency. This section presents some of the most relevant efficient online multivariate time series anomaly detection methods that are designed to either reduce the use of computational resources (runtime and memory usage) or to scale effectively with the volume and/or dimensionality of time series data. These methods were previously listed in Table 1 of Section 3.2.1.

Statistics-based anomaly detection methods are well-suited for handling large-scale data with efficiency, particularly in multivariate time series. For example, RS-Hash constructs an ensemble of histograms on feature subspaces, leveraging hash functions to achieve linear time complexity during training and maintaining efficiency during updates and scoring [Sathe and Aggarwal, 2018, Ntroumpogiannis et al., 2023b]. Similarly, LODA, which builds an ensemble of one-dimensional histogram density estimators, operates effectively in both batch and streaming modes without requiring hyper-parameter tuning, making it highly adaptable for continuous anomaly detection in evolving data streams [Pevnỳ, 2016, Ntroumpogiannis et al., 2023b]. The Correlation Anomaly Detection (CAD) method, although distance-based, focuses on statistical correlation. Particularly effective in sensor-based multivariate time series, it transforms time series data into graphs to monitor sensor correlations, identifying anomalies through variations in these correlations. CAD demonstrates scalability and stability, though it is limited to environments where sensor correlation is significant [Ang et al., 2023].

Distance-based methods offer another approach to online anomaly detection by efficiently modeling the relationships between data points in a dataset. For instance, CPOD addresses efficiency issues encountered by MCOD when handling sparse data regions by using core points for neighbor searches, leading to significantly faster performance and lower memory consumption [Tran et al., 2020, Ntroumpogiannis et al., 2023b]. Methods like STARE, which identify local anomalies in sliding windows, excel in terms of execution time while maintaining high detection accuracy, which makes them suitable for real-time applications [Yoon et al., 2020, Ntroumpogiannis et al., 2023b]. Additionally, neural network-based models such as DeepSVDD utilize stochastic gradient descent (SGD) to optimize their performance, scaling well with large datasets through parallel processing (e.g. by processing on multiple GPUs). This ability to handle large-scale data efficiently makes it particularly attractive for online learning environments [Ruff et al., 2018a].

Other distance-based anomaly detection methods, such as LEAP and HST, have significantly improved the efficiency of detecting anomalies. LEAP addresses the computational cost of range queries using efficient indexing, making it up to three orders of magnitude faster than traditional k-nearest neighbors (KNN) methods [Cao et al., 2014, Ntroumpogiannis et al., 2023b]. Similarly, HST employs a balanced tree structure to efficiently sketch data streams, outperforming other tree-based detectors in terms of speed [Liu et al., 2012, Ang et al., 2023]. For high-dimensional data, XSTREAM excels by creating on-the-fly data sketches and using histogram-based density estimation in random subspaces, allowing it to continuously update detection models with a superior balance of efficiency and effectiveness. This method has proven particularly robust in practical settings, outperforming other detectors in high-dimensional datasets [Zhang et al., 2022a, Ntroumpogiannis et al., 2023b].

Matrix-based methods have also been developed to improve scalability in real-time anomaly detection. DAMP, a matrix profile-based technique, is designed for the detection of discord anomalies in fast-arriving streams, reaching processing speeds of up to 300,000 Hz on commodity hardware [Lu et al., 2023]. Although DAMP does not yet handle multivariate data or distributed computing environments, its scalability for single-stream data is significant. NormA, another matrix profile-based method, incorporates a clustering preprocessing step that reduces the size of the reference dataset, thereby improving search speed during anomaly detection [Boniol et al., 2021a, Lu et al., 2023].

Surprisingly, the solutions to the problem of sequence anomaly detection that have been proposed in the literature are not as effective and scalable as real-world applications require. Linardi et al. [2020] extended their framework to deal with variable-length discord discovery. Their approach enables the detection of a large number of anomalous patterns of different lengths. By itself, the Matrix Profile can also be used to find the top discords in a series and detect anomalies as unique behaviors. Similarly, SCRIMP accelerates the computation of the classic matrix profile, offering efficiency for large-scale time series data [Zhu et al., 2018b, Lu et al., 2023]. Several other efficient algorithms have been proposed for computing the Matrix Profile of a times series, such as the STAMP, STOMP [Yeh et al., 2018] and SCRIMP++ [Zhu et al., 2018a]. Other works have also derived adaptations, such as the KNN Matrix Profile, which looks for the kth nearest neighbor for each subsequence, thus allowing the detection of clusters of anomalies/discords [Mondal et al., 2023].

Additional techniques that help in scaling time series anomaly detection include preprocessing algorithms such as OneShotSTL [He et al., 2023]. This method performs online seasonal-trend decomposition with constant time complexity, making it particularly effective for real-time applications. Experiments demonstrate that OneShotSTL is significantly faster—by factors of 10 to over 1,000—compared to other methods, while maintaining comparable or superior accuracy for anomaly detection tasks in streaming time series [He et al., 2023].

4.2 Dimensionality processing

This section presents relevant works that address time series dimensionality processing focusing either on dimensionality reduction or efficient techniques to deal with similarity search on high-dimensional time series.

Generally, anomaly detection is addressed by search techniques applied directly to the multivariate time series, however, a common limitation of these techniques is the high computational cost [Wang et al., 2010]. A common approach to deal with this problem is to transform the time series into a single variable and represent the problem as a univariate anomaly detection task. However, it is crucial to minimize the loss of information. For this purpose, PCA has been extensively used [Tanaka and Uehara, 2003, Tanaka et al., 2005]. However, this method is limited to the data that can be accurately represented by the first principal component of PCA alone [Minnen et al., 2006, Wang et al., 2010].

When dealing with large time series, it is also important to adopt size reduction techniques such as the Piecewise Aggregate Approximation (PAA) [Lin et al., 2003] and the Symbolic Aggregate Approximation (SAX) [Lin et al., 2002]. Using PAA, a time series X of length n can be represented in a reduced w-dimensional space as another time series $X' = (\mu_1; \ldots; \mu_w)$ by segmenting X into w equally sized segments and replacing each segment by its mean value μ_i [Xuan and Anh, 2018]. The SAX technique, on the other hand, transforms the time series X into a symbolic sequence $A = a_1 \ldots a_w$ in which each real value μ_i is mapped to a symbol a_i from an alphabet of size a. Thus, SAX allows a time series to be reduced to a string of arbitrary length w, ($w < n$, typically $w << n$). The size of the alphabet is also an arbitrary integer a, where $a > 2$ [Lin et al., 2003]. SAX has been successful in data mining [Tanaka et al., 2005].

With complex and massive time series datasets, fast and accurate similarity search is crucial to perform many data mining tasks like classification, clustering, motif/discord discovery, and anomaly detection. To this end, indexing has been established as one of the main techniques to improve the performance of similarity queries [Esling and Agon, 2012]. In this context, the works of Yagoubi et al. [2018] and Levchenko et al. [2021] employed indexing to derive parallel solutions to query large-scale time series datasets. Yagoubi et al. [2018] proposed a novel parallel indexing solution that scales to billions of time series, as well as a parallel query algorithm that efficiently exploits this index. Levchenko et al. [2021] presented several parallel solutions to evaluate k-nearest neighbor queries on large time series databases. They also offered a recommendation tool for their selection and parameterization. The solutions presented were developed on the basis of two state-of-the-art algorithms (iSAX and sketch), which allow the use of distributed data processing frameworks, such as Spark.

High-dimensional spaces often contain redundant or irrelevant features that hide meaningful patterns. Feature subspace search has emerged as a promising approach to tackle this challenge to outlier detection in high-dimensional data streams. Traditional methods focus on identifying outliers within subspaces of features that hold significant information. However, this approach remains underexplored in the context of data streams, where memory, processing time, and adaptability to data changes are critical. To address these limitations, Souiden et al. [2022] proposed a metaheuristic-based approach using the Adapted Binary Gravitational Search algorithm, which efficiently identifies high-contrast subspaces for outlier detection while accommodating the constraints of data streams. Also, recent methods focus on finding the best K time series in any N-dimensional time series that are most representative of anomalies based on Matrix Profile [Tafazoli and Keogh, 2023].

The similarity measure is also an important design choice and a key part of anomaly detection algorithms. Recently, d'Hondt et al. [2024] proposed a taxonomy and a structured evaluation of multivariate time series distance measures.

Typical similarity measures include the Euclidean distance (EDist), correlation coefficient, and dynamic time warping (DTW) [Mueen, 2014, Akbarinia and Cloez, 2019]. Among them, the classic DTW has proven to be the best measure in many domains [Ding et al., 2008]. However, the problem is that DTW has a quadratic time complexity [Torkamani and Lohweg, 2017]. Nevertheless, recent works make use of bounding techniques [Keogh and Ratanamahatana, 2005, Torkamani and Lohweg, 2017], and have developed scalable methods for anomaly detection under DTW [Alaee et al., 2020, 2021]. Other current state-of-the-art methods include DTW-based anomaly detection [Alaee et al., 2020], and exact anomaly detection on fast arriving streams [Lu et al., 2022, 2023].

4.3 Parallel and distributed computing

As time series datasets become larger and more complex, traditional single-node computing approaches struggle to handle processing demands. By using parallel and distributed computing, time series data can be partitioned and processed concurrently across multiple computers, enabling the analysis of large-scale datasets [Özsu and Valduriez, 2019]. This concurrency speeds up computations, improves scalability, and enables efficient use of computing resources.

Large-scale scientific time series analysis applications are commonly modeled by workflows [Ogasawara et al., 2011]. Some of the main approaches developed to enable distributed and parallel execution of such workflows include Pegasus [Deelman et al., 2005], MapReduce [Dean and Ghemawat, 2008], Hadoop [Shafer et al., 2010], and Spark [Zaharia et al., 2010]. These approaches exploit shared-nothing clusters of computers, in which large volumes of scientific data may be stored and processed in parallel using common machines. However, most current approaches, based on MapReduce or Hadoop, do not handle memory and data locality appropriately, as they do not analyze the entire workflow [Gaspar et al., 2017].

The Apache Spark parallel programming framework, on the other hand, correctly chains activities that should be executed in a specific node of the cluster. It focuses on the efficient processing of large datasets, performing data analytics with in-memory storage of intermediate data in RDDs (Resilient Distributed Datasets). RDD is an efficient and fault-tolerant abstraction for distributing data in a cluster [Yagoubi et al., 2018]. A Spark cluster consists of a master node to coordinate the job execution and a set of worker nodes to execute the parallel operations (like Map/Reduce of RDDs). Spark provides a significant performance increase over Hadoop or other MapReduce implementations [Zaharia et al., 2012].

The rest of this section presents state-of-the-art scalable distributed multivariate time series anomaly detection methods. Furthermore, the latest advances in parallel and distributed computing for multivariate anomaly detection are discussed, including the adoption of: federated learning [Abreha et al., 2022], GPU acceleration [Zhu et al., 2021], and transformer architectures [Merrill and Sabharwal, 2023].

4.3.1 Distributed online detection methods

There are two main approaches to distributed processing of streaming time series data, namely task partitioning and data partitioning [Chen et al., 2016]. For task partitioning, a detection algorithm is partitioned into many sub-tasks assigned to multiple nodes for parallel processing of the incoming stream. On the other hand, for data partitioning (see Figure 12), n instances of a window-based detection algorithm are deployed on different nodes. For parallel processing, first, the current data window is partitioned into sub-windows by a dedicated split node, responsible for splitting the data stream into many sub-streams and routing them to the detection algorithm instances. Each instance accesses its corresponding data partition (sub-window), used to compute local detection results. Finally, local results are combined by a dedicated merge node that generates final anomaly detection results.

Recently, significant advances have been made in distributed online multivariate anomaly detection, with methods in this category playing a crucial role in managing large-scale, high-dimensional data streams. The first examples are set by methods ECOD and RCAD. ECOD has linear complexity and can be parallelized across dimensions to efficiently handle datasets with millions of observations, while avoiding the overhead of hyperparameter tuning [Li et al., 2022]. In contrast, RCAD employs a real-time collaborative anomaly detection system for network data, utilizing Hierarchical Temporal Memory (HTM) for unsupervised detection. PMUNET [Ahmed et al., 2021] is a novel device-level deep learning-based data-driven approach for online anomaly detection, localization, and classification of multivariate streaming data. It adapts a deep learning algorithm to data drift and enables online learning to detect anomalies over data drifting synchrophasor data streams. However, it still lacks efficiency comparison with state-of-the-art and its code is not available.

When it comes to distributed outlier detection, current state-of-the-art methods are few and often limited in performance. For example, DDLOF [Yan et al., 2017], SPIF [Tao et al., 2018], and DBSCOUT [Corain et al., 2021] represent some of the prominent approaches, yet each has critical shortcomings. DDLOF, a distributed version of the LOF al-

gorithm, is hampered by its reliance on Hadoop, making it significantly slower than Spark-based alternatives [Zhang et al., 2022a]. SPIF and DBSCOUT, despite using Spark, either suffer from poor scalability with large datasets or fail to perform well on data with varying-density support. These methods do not fully meet the desired properties, such as linear time and space complexity, robustness to hyperparameter choices, and efficient handling of high-dimensional data.

To address these limitations, SparkX [Zhang et al., 2022a] emerged as a leading distributed outlier detection algorithm designed to scale efficiently across massive datasets on cloud platforms. Based on the XSTREAM algorithm, SparkX integrates with Apache Spark's data-parallel capabilities, making it suitable for shared-nothing infrastructures. By inheriting XSTREAM's desirable properties, such as handling high-dimensional data streams, it outperforms existing methods in both detection performance and scalability.

Other frameworks such as GDSW and PROUD provide generalizable solutions for distributed anomaly detection over data streams. GDSW offers a versatile framework for distributed sliding window-based operations, accommodating both data-independent and data-dependent operators [Chen et al., 2016]. PROUD is an open-source engine that supports continuous parallel and distributed distance-based outlier detection for big data streams, implemented on Apache Flink for scalability and extensibility [Toliopoulos et al., 2020]. These frameworks, along with approaches that utilize Kafka queues and Spark Streaming, ensure that distributed anomaly detection can meet the demands of high processing capacity and low latency, essential for real-time applications [Rettig et al., 2019].

4.3.2 Federated learning

In federated learning-based anomaly detection [Agrawal et al., 2022], local models are run on individual nodes to detect anomalies within their data partitions. Global models periodically average model parameters, combine local results (anomaly scores) or train on processed data for a holistic view. Federated learning has emerged as a critical solution to address privacy and bandwidth concerns in edge computing environments [Abreha et al., 2022]. It enables distributed devices, such as mobile phones and IoT devices, to collaboratively train machine learning models without the need to transmit sensitive data to a centralized server. This paradigm shift not only preserves data privacy but also reduces legal complexities and bandwidth requirements. Abreha et al. [2022] highlights the challenges and advanced solutions associated with implementing Federated Learning in edge computing, identifying key open problems. Similarly, Agrawal et al. [2022] explores the use of federated learning in Intrusion Detection Systems (IDS), where it provides a decentralized privacy-preserving framework that helps to secure complex and heterogeneous network infrastructures while maintaining high detection accuracy. Karras et al. [2023] also use federated learning combined with Apache Spark and Federated AI Technology Enabler (FATE) to provide a novel strategy for managing IoT-based Big Data.

In the context of multivariate time series anomaly detection, federated learning has shown significant promise in decentralized anomaly detection. Zhu et al. [2022] investigate this problem in environments where data is heterogeneously distributed across various IoT edge devices. Their Federated Exemplar-based Deep Neural Network (Fed-ExDNN) enables collaborative detection of anomalies while ensuring that sensitive data remains localized. Zhang et al. [2022b] propose FedGroup, a federated approach tailored to detect anomalies in IoT devices. de Cámara et al. [2023] further extend this work by introducing a clustered federated learning architecture for large-scale heterogeneous IoT networks, demonstrating its effectiveness in reducing network overhead and improving the scalability of anomaly detection.

The combination of federated learning and autoencoders has also been explored to enhance anomaly detection in multivariate time series data. Vucovich et al. [2023] present an federated learning-based anomaly detector that uses an autoencoder with a classifier to identify malicious network activities. Their framework allows each client to improve its defense against cyber-attacks while maintaining data privacy. Zhang et al. [2021] propose an unsupervised anomaly detection framework for Cyber-Physical Systems (CPS) that leverages a Variational Autoencoder (VAE) and Convolutional Gated Recurrent Unit (ConvGRU). This approach captures both feature and temporal dependencies in high-dimensional time series data, making it particularly suited for detecting anomalies in networked sensor data without requiring centralized data storage or extensive labeled datasets.

4.3.3 GPU acceleration

Recent methods designed for data-intensive and/or time-critical tasks [Zhao et al., 2021a] address the challenge of improving execution efficiency by developing distributed algorithms either using CPUs [Bhaduri et al., 2011, Lozano and Acufia, 2005, Oku et al., 2014, Toliopoulos et al., 2020, Yan et al., 2017, Zhang et al., 2022a, Zhao et al., 2021b] or accelerating algorithms by using GPUs [Zhao et al., 2021a]. Recent advances in GPU acceleration and parallel computing have significantly improved the efficiency and scalability of anomaly discovery tasks. Zhao et al. [2021a] introduce TOD, the first tensor-based system for outlier detection on distributed multi-GPU machines, which decom-

poses complex detection tasks into basic tensor algebra operators, leveraging modern deep learning infrastructure for accelerated computations. Similarly, Zymbler and Kraeva [2023] present a novel GPU parallelization scheme, called PALMAD, which enhances the performance of the MERLIN algorithm for time series anomaly discovery, enabling efficient detection of discords of varying lengths. Zhu et al. [2021] further contribute by developing a GPU acceleration framework for pattern mining, which breaks down subsequence-based computations into fine-grained patterns for efficient parallel processing, facilitating both motif and discord discovery under different distance metrics.

4.3.4 Transformers

Transformer architectures have increasingly been explored for scalable anomaly detection, particularly due to their ability to model complex temporal dependencies while benefiting from parallel processing [Arslan et al., 2023]. First introduced by Vaswani [2017], transformers leverage feed-forward layers and multi-headed attention mechanisms, allowing efficient parallelization, which accelerates training compared to recurrent models like LSTMs or Gated Recurrent Units (GRU). This inherent parallelism introduces both opportunities for scalability and potential limitations tied to the trade-offs in parallelizable model architectures [Merrill and Sabharwal, 2023]. Despite their growing popularity, transformers have not yet been widely applied to time-series anomaly detection compared to other model families. However, they hold significant promise, with studies such as Sanford et al. [2024] focusing on the computational efficiency that transformers can achieve through logarithmic depth, a property that further distinguishes them from traditional neural sequence models. Maintaining the temporal order of input data through positional encoding and utilizing multi-headed attention, transformers are poised to improve detection capabilities, though an online, fully scalable version for real-time anomaly detection has yet to be developed [Correia et al., 2024].

Transformers have also been adopted for multivariate time series anomaly detection, leveraging their capacity to manage both temporal dependencies and inter-sensor relationships. Wang et al. [2023] propose the Disentangled Dynamic Deviation Transformer Network (D^3TN), which effectively handles multiscale sensor dependencies, while Fan et al. [2022] introduce Sepformer, a model that combines discrete wavelet transforms with transformer networks to capture different frequency components in data streams, although its applicability to anomaly detection remains untested. Fu et al. [2024] also propose a two-stream multivariate time series anomaly detection method that uses separation, decomposition, and dual transformer-based autoencoder. Continuous and discrete features are separated before decomposition. Furthermore, Song et al. [2023] present MEMTO, a memory-guided transformer designed for reconstruction-based anomaly detection, incorporating a memory module that updates dynamically based on input data. These approaches highlight the growing trend of integrating transformers with advanced techniques like wavelet transforms and memory modules to enhance anomaly detection performance in multivariate time series.

4.4 Applications and Case Studies

Scalable multivariate time series anomaly detection is crucial for monitoring complex, real-time systems across various fields. Ariyaluran Habeeb et al. [2019] discuss extensive applications, ranging from operational monitoring, web analytics, and smart cities to biometric devices and social media platforms. As real-time data processing becomes increasingly vital, Olteanu et al. [2023] highlight key domains such as financial fraud detection, network security, sensor networks, and IoT, where timely and accurate anomaly detection is essential to maintain operational stability and security in rapidly evolving environments. This section presents other relevant applications of scalable multivariate time series.

In *Natural sciences/Finance*, anomaly detection addresses critical challenges like climate monitoring, astrophysical event detection, and extreme weather forecasting. Climate trends and weather indicators, often used in anomaly detection, are crucial in fields such as agriculture, public health, and economics, as discussed by Saldanha et al. [2024]. Porto et al. [2022] underscore the growing importance of machine learning approaches in predicting extreme weather events in urban areas, particularly in Rio de Janeiro, where traditional numerical weather prediction models struggle to capture the complexity of such events. Meanwhile, in the astrophysical realm, Zhu and Shasha [2003] present the application of burst detection algorithms to detect gamma-ray bursts, while demonstrating their utility in high-frequency stock market trading activities. Time series anomaly detection plays also an important role in automatic network-stream monitoring [Mason et al., 2019].

In *Industrial/Business*, multivariate time series anomaly detection is adopted to monitor critical infrastructure and systems. Lu et al. [2023] and Ahmed et al. [2021] discuss its use in energy grids and machinery failures, while anomaly detection in electrical grid data is being developed for smarter systems. Smart meters record energy use in real-time, with scalable online anomaly detection applied to this data [Liu and Nielsen, 2018, Aligholian et al., 2019].

Particularly in the IoT domain, time series anomaly detection has been extensively used. Sgueglia et al. [2022] provide a systematic literature review of IoT time series anomaly detection solutions including a taxonomy. Similarly,

DeMedeiros et al. [2023] and Zhou et al. [2023] emphasize the critical role of anomaly detection in IoT and Industrial IoT (IIoT), where detecting anomalies in vast streams of sensor data is crucial for the success of smart manufacturing and agriculture. Ahmed et al. [2022] presents an application for monitoring business systems based on mobile broadband networks.

5 Efficiency and accuracy trade-off in distributed online anomaly detection

Distributed methods for online multivariate anomaly detection are still not fully explored. Centralized systems are easier and work well for small data sets but struggle with larger ones. In financial markets like Nasdaq[2], with tens of millions of daily transactions, detecting anomalies in 24-hour window data is challenging. Centralized methods can't handle such scale and speed, and the effect of distributed computing on detection accuracy is uncertain.

The *efficiency-accuracy trade-off* of distributed online anomaly detection algorithms, especially for multivariate time series, has not yet been publicly analyzed. This analysis is crucial for indicating the potential of distributed online detection over high-throughput and high-dimensional time series and motivating future endeavors in the area.

In this section, we compare the accuracy and efficiency of established anomaly detection algorithms over multivariate streaming time series data with both centralized and distributed executions. We have devised a novel methodology for performing and benchmarking window-based online distributed time series anomaly detection, described in the next section. In the remainder of this section, we present adopted experimental settings, multivariate datasets, and discuss the results of this analysis.

5.1 2OD: Online Distributed Outlier Detection

We have devised a novel systematic methodology, called Online Distributed Outlier Detection (2OD), that allows the user to benchmark both computational efficiency and detection accuracy of the window-based online and distributed execution of any outlier detection method currently available, even if it is originally offline or centralized.

To enable systematic performance benchmarking in different scenarios, 2OD is composed of three simple, yet challenging anomaly detection approaches, namely *offline* (*OffA*), *online* (*OnA*), and *online and distributed* (*OnDA*). Figure 11 illustrates their steps and application scenarios. In particular, the goal of OffA is to establish a baseline for the experimental analysis of the global efficiency/accuracy performances of both OnA and OnDA, while OnA also serves as a reference for assessing OnDA regarding its benefits as a distributed approach in an online context.

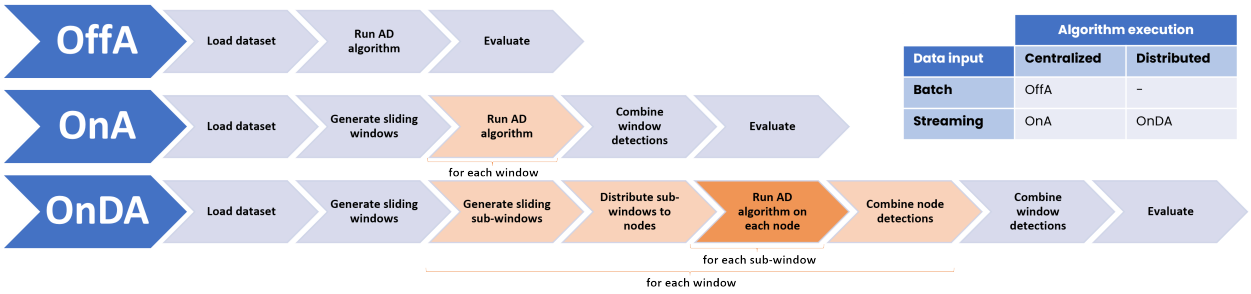


Figure 11: Illustration of 2OD experimental methodology

OffA (Offline approach) Most anomaly detection (AD) methods are designed for *offline* detection using batch inputs of the entire time series, prioritizing accuracy over efficiency [Ariyaluran Habeeb et al., 2019]. OffA addresses this scenario with a straightforward process: load the dataset, execute a multivariate AD algorithm *centralized*, and calculate accuracy and efficiency metrics.

OnA (Online approach) OnA follows a *online* approach to AD, using a sliding window to simulate data streams. Each latest window is input to the algorithm for a *centralized* execution, following a no-memory approach.

In overlapping sliding windows of size p, each observation x_t is subject to AD p/l times, where l is the window step, consequently receiving an equal number of different anomaly scores. The set of scores for each observation x_t must be combined to obtain a final classification [Lima et al., 2024]. Accuracy and efficiency performance metrics can then be computed.

[2] `https://www.nasdaqtrader.com/Trader.aspx?id=DailyMarketSummary`

OnDA (Online & Distributed approach) The last approach, defined by OnDA, represents an *online and distributed* approach to AD. OnDA concerns a scenario of high-throughput and high-dimensional data streams, and the execution of a given AD algorithm with a distributed and parallel setup based on data partitioning, as illustrated in Figure 12.

Distributed sliding window analysis starts by dividing each window into sub-windows. A single node partitions the data stream and routes sub-windows to processing instances, where an AD algorithm runs on multiple cluster nodes for parallel processing. Each instance computes local results for its sub-window [Chen et al., 2016].

Similarly to OnA, each observation x_t is subject to AD multiple times, receiving multiple local anomaly scores from the processed overlapping sliding sub-windows. Local scores are centralized and combined at the merging node to generate final window results. This process is repeated to obtain final classifications, and then accuracy and efficiency metrics are computed for each window and the entire time series.

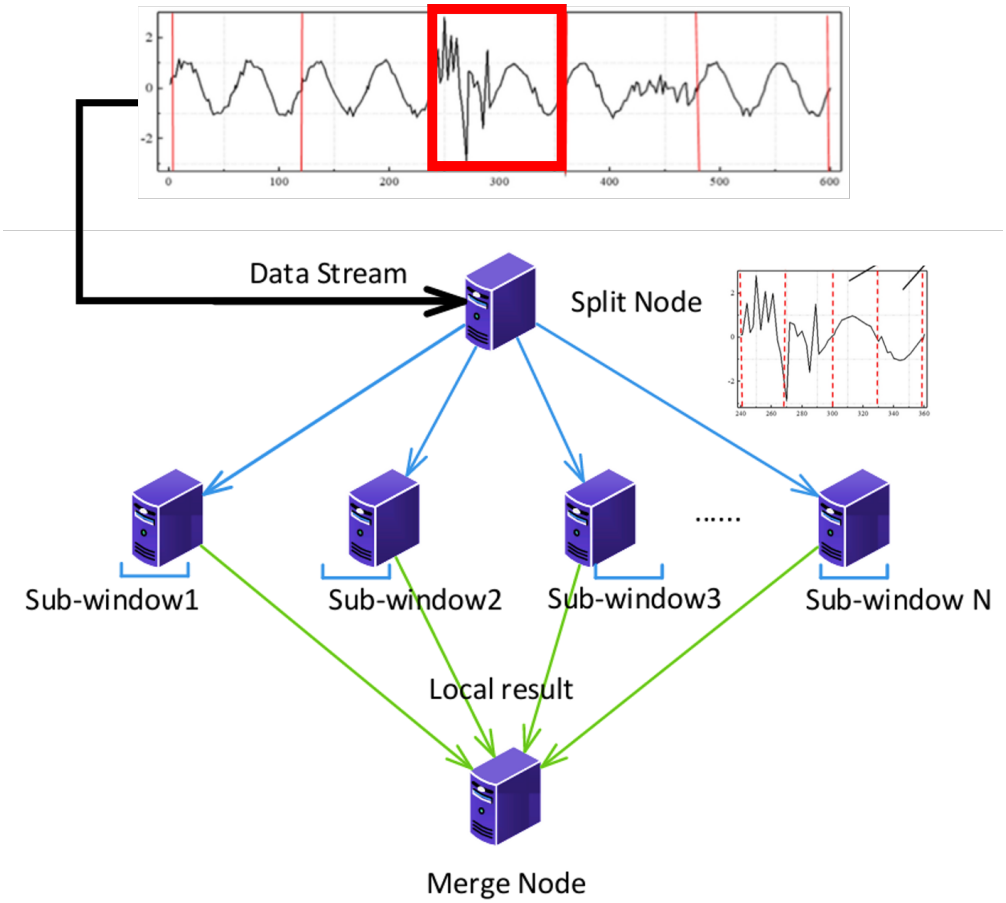


Figure 12: Distributed online anomaly detection process performed by OnDA (adapted from Chen et al. [2016])

Each approach outputs anomaly detections (binary classifications) and performance metrics (accuracy and efficiency) for comparison and assessment in anomaly detection. Required inputs: an unsupervised AD algorithm, experimental settings with hyperparameters, and a high-dimensional multivariate time series dataset with anomalies. Experimental inputs for OffA, OnA, and OnDA are detailed in the following subsections. The implementation of 2OD is publicly available on Github[3].

5.2 Experimental settings

Most anomaly detection methods originate from outlier detection (OD) research, focusing on identifying punctual anomalies [Blázquez-García et al., 2021]. This study evaluates OD algorithms for accuracy and efficiency in streaming data and distributed execution, excluding sequence anomaly detection.

[3] `https://github.com/RebeccaSalles/2OD`

For this comparison, 10 of the most commonly adopted unsupervised state-of-the-art OD algorithms were selected [Kuo, 2023, Ntroumpogiannis et al., 2023b, Schmidl et al., 2022, Han et al., 2022b]. Listed in Table 2, they cover the three main multivariate AD strategies for the online scenario, namely: Distance-based, Statistics-based and Decision-tree-based. Following the discussion of Section 3.2.1, forecasting-based or reconstruction-based methods are generally semi-supervised, requiring "uncontaminated" training data drawn from streams [Berahmand et al., 2024], hyperparameter optimization [Ruff et al., 2018b], and are potentially computationally costly [Schmidl et al., 2022]. For this reason, our experimental analysis leaves out-of-scope (semi-)supervised methods for anomaly detection. While most algorithms are distance-based, the latest algorithms are based on statistics computation, which can be less resource demanding for big data streams. Not all of the selected algorithms were originally designed for streaming data. However, due to our experimental design, this was not required. Our proposed methodology is prepared to adapt the execution of OD algorithms to a distributed online scenario.

Table 2: Selected algorithms

Algorithm	Publication Year	Detection strategy
ECOD	2022	Statistics-based
COPOD	2020	Statistics-based
DeepSVDD	2018	Distance-based
LODA	2016	Statistics-based
HBOS	2012	Distance-based
IForest	2012	Decision-tree-based
MCD	2004	Distance-based
PCA	2003	Distance-based
CBLOF	2003	Distance-based
KNN	2002	Distance-based

Algorithm implementations are publicly available in the PyOD framework [Zhao et al., 2019, Han et al., 2022b] with thehyperparameters defined in PyOD by default.

All executions were performed in 5 nodes equipped with Dell C6220 dual-Xeon E5-2680 v2 @ 2.80GHz (20 cores), with a RAM capacity of 192 GB. Each node runs on CentOS Linux 7.4.1708. One node is dedicated to scheduling Spark tasks, and the others are in charge of running the Spark load. Each node can simultaneously run 20 tasks. The entire cluster can perform 80 detection tasks in parallel.

All approaches begin by loading a given time series dataset into the master Spark node, which is responsible for distributing data partitions and routing the execution to worker nodes, for OnDA, or executing the process itself, for OffA and OnA. This setup helps level up data availability for all AD approaches, taking into account the overhead of using task operations [Lange and Fortin, 2014]. Performance metrics adopted to measure the detection accuracy of all algorithms based on each AD approach include: the Area Under the Curve (AUC) of the Operating Characteristic Curve (ROC) curve (AUC-ROC), and F1. For measuring efficiency, runtime (in seconds) and speedup metrics are analyzed.

The experimental settings for the online approaches, OnA and OnDA, are presented in Table 3. Both approaches rely on data partitioning, making the partition size or window size p a crucial parameter for defining data stream throughput. In high-throughput time series scenarios, p ranges from 300,000 (a very fast rate [Lu et al., 2023]) to 3 million observations. The window step l is defined as $p/2$.

For OnA, the OD algorithm execution is performed in one single node and thread (th). In turn, OnDA explores distributed execution over 4 nodes and over a number of threads going from 2 to 80 as processing instances (maximum, as each node allows the allocation of 20 threads for simultaneous processing). In order to take maximum advantage of parallelism, the sub-window sizes sp for OnDA are set such that the number of generated sub-windows $(p-sp)/sl+sl$ is equal to the number of distributed processing instances, th. The adopted sub-window step $sl = sp/2$ is analogous to OnA. Finally, the functions used to combine results for both windows and sub-windows are the maximum, or average of detection scores.

5.3 Datasets

This experimental analysis is based on four different multivariate time series datasets containing anomalous observations. They encompass both synthetic and real-world data, containing up to hundreds of millions of observations. They allow the simulation of high-throughput and high-dimensional data streams, ranging up to 10 million time points and

Table 3: Experimental settings for OnA and OnDA

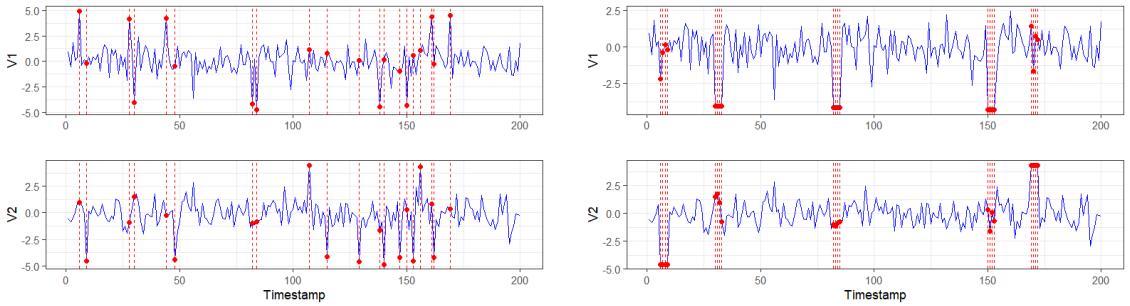
Appr.	Settings	Values
OnA	Window size (p)	300k, 500k, 1M, 3M, 5M, 10M
	Window step (l)	$p/2$
	# of nodes	1
	# of threads (th)	1
	Score comb. func.	maximum, average
OnDA	Window size (p)	300k, 500k, 1M, 3M, 5M, 10M
	Window step (l)	$p/2$
	# of nodes	4
	# of threads (th)	2, 5, 10, 20, 30, 40, ..., 80
	Sub-window size (sp)	$sp = p - (th - sl) * sl$
	Sub-window step (sl))	$sp/2$
	Score comb. func.	maximum, average

more than one hundred variables, being among the biggest multivariate time series AD benchmark datasets currently available in the literature [Schmidl et al., 2022]. The metadata corresponding to the selected time series datasets are summarized in Table 4.

Table 4: Selected datasets metadata

Dataset name	Origin	Dim.	Learn.	# time series (total)	Avg. length	# of variables	Avg. # of anomalies
3M (GutenTAG)	synthetic	multi	u	14	3000000	2, 3, 5, 10, 20, 50, 100	300000 (10%)
10M (GutenTAG)	synthetic	multi	u	10	10000000	2, 3, 5, 10, 20	1000000 (10%)
Kitsune	real	multi	u	9	2335288	116 (avg.)	110673 (5%)
LTDB	real	multi	u	7	9706422	3 (avg.)	8733 (0.1%)

For the sake of comparing the approaches specifically to the problem of multivariate OD, in a scenario of increasing dimensionality, we produced two synthetic datasets, inspired by the data generation process implemented by the GutenTAG tool[Wenig et al., 2022]. They contain time series ranging up to 3 million (*3M* dataset) and up to 10 million (*10M* dataset) points in time, respectively, and varying from 2 to 100 variables. All time series follow the same stationary generating function (N(0,1)), but contain a 10% contamination of punctual anomalies that considerably exceed the mean of the observations in a 10% neighboring range (context), called extremum anomalies. To simulate the real scenario in which anomalies may appear consecutively, we reproduce all time series including contiguous extremum anomalies, called platform anomalies. Synthetic multivariate time series dataset examples are given in Figure 13.



(a) Extremum anomalies

(b) Platform anomalies

Figure 13: Synthetic multivariate time series datasets examples

We also selected two real-world multivariate time series datasets preprocessed and used as a benchmark by the work of Schmidl et al. [2022], currently the most comprehensive experimental review of AD algorithms publicly available. In particular, we selected the biggest datasets in both size and dimensions, namely Kitsune [Mirsky et al., 2018] and

LTDB [Goldberger et al., 2000, Moody and Mark, 2001]. Kitsune is the most dimensional dataset containing 9 time series with over one hundred variables on average and over 2 million observations. On the other hand, LTDB contains 7 time series, which are less dimensional (3 variables on average), but have lengths of around 10 million observations. In terms of anomaly contamination, Kitsune time series have, on average, 5% anomalous observations, whereas LTDB series anomalies are more scarce, summing on average 0. 1% of the observations.

5.4 Experimental results

This section presents and discusses the results of the comparison between the efficiency and accuracy of the conducted multivariate AD approaches, based on the 2OD methodology. The experimental questions we aim to answer are as follows:

1. Can a distributed approach to a problem of online multivariate OD improve execution efficiency when compared to a centralized approach?
2. How is the distributed execution efficiency affected by increasingly high throughput streams and time series dimensionality?
3. Are efficiency improvements observable for both synthetic and real data?
4. Are efficiency improvements consistent over all sliding windows and over all different time series and datasets?
5. How is the detection accuracy performance of OD algorithms affected by distributed execution?
6. How does the global detection accuracy performance of online and distributed-online OD approaches compare to an offline centralized OD approach over a non-streaming multivariate time series?

5.4.1 Online vs Distributed online OD

To answer the first two questions, we compare the speedup provided by OnDA with that provided by OnA for all selected OD algorithms. Moreover, we observe how the speedup is affected in different scenarios of increasingly throughput and dimensionality of multivariate time series streams, as well as, increasingly number of parallel processing units (threads). Figure 14 presents speedup results for the *3M* dataset, containing simulated data. In the matrix of plots, each row represents the results for a given OD algorithm (on the right), and each column represents the results for a time series with a given number of variables (on the top). Each line curve on the plots represents a different streaming throughput. The darker the color, the higher the data throughput, going from 300 thousand (300k) to 3 million (3M) observations. The increasing number of parallel threads is represented in the x-axis of each plot. The dashed horizontal line marks the threshold for efficiency improvement (speedup greater than one). Missing values are the result of either execution timeout (over 8 hours) or failure.

Scenario results based on simulated data Figure 14 shows that OnDA's distributed approach improves execution efficiency over OnA. Speedup curves rise with increasing threads due to parallel processing but decline beyond a peak due to task management overhead [Lange and Fortin, 2014]. Thus, balancing parallel processing with task management trade-offs is recommended to find optimal setups for maximum speedup.

KNN shows exceptionally high speedup, reaching over 700, partly due to distributed execution. However, the speedup results also benefit from a loss in OnA efficiency due to memory resource management of large data window matrices (with 1 million by 100 variables). And among the algorithms, KNN is particularly susceptible to this problem.

The behavior represented by the curves is similar for almost all selected OD algorithms in Figure 14. Clear exceptions are posed by PCA and HBOS, which do not seem to benefit from a distributed approach. These algorithms are particularly simple and fast, and in this case, the task management workload is slower than the actual algorithm execution, making a distributed approach not advisable in terms of efficiency. Other algorithms, such as COPOD and ECOD, most benefit from distribution depending on the time series throughput and dimensionality. In general, a higher number of variables results in an increase in speedup, indicating the benefit of adopting a distributed OD approach for streams of increasingly high dimensions. Exceptions include DeepSVDD, which decreases speedup for higher dimensions, and CBLOF, which could not be computed for more than three variables. This particular efficiency behavior is further studied on the basis of Figure 15.

Generally, higher throughput yields higher curves, indicating an advantage of distributed OD approaches for multivariate streams of increasingly high throughput. Exceptions indicate the presence of an overhead from data management of large sub-windows, affecting the overall OD execution runtime. Figure 15a presents the runtime results (in seconds) for *3M* dataset by varying the time series throughput. Different curves now represent the number of time series

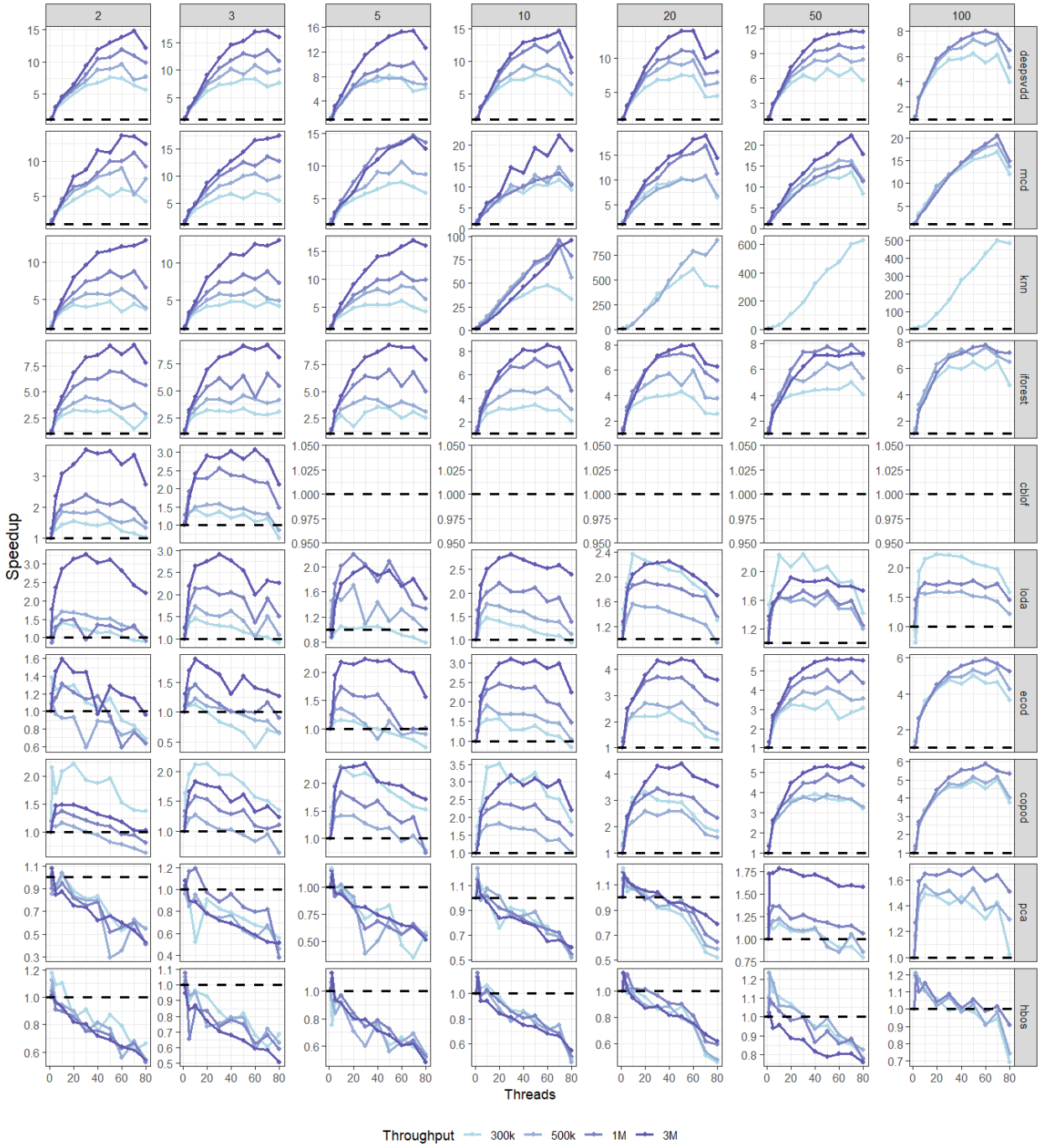
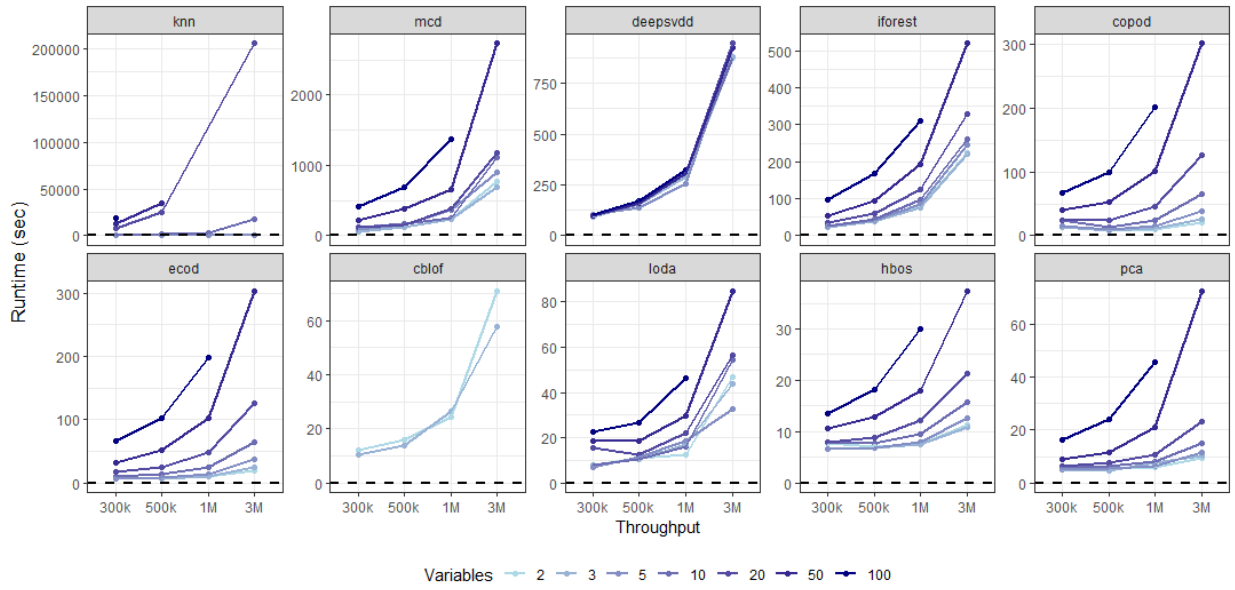


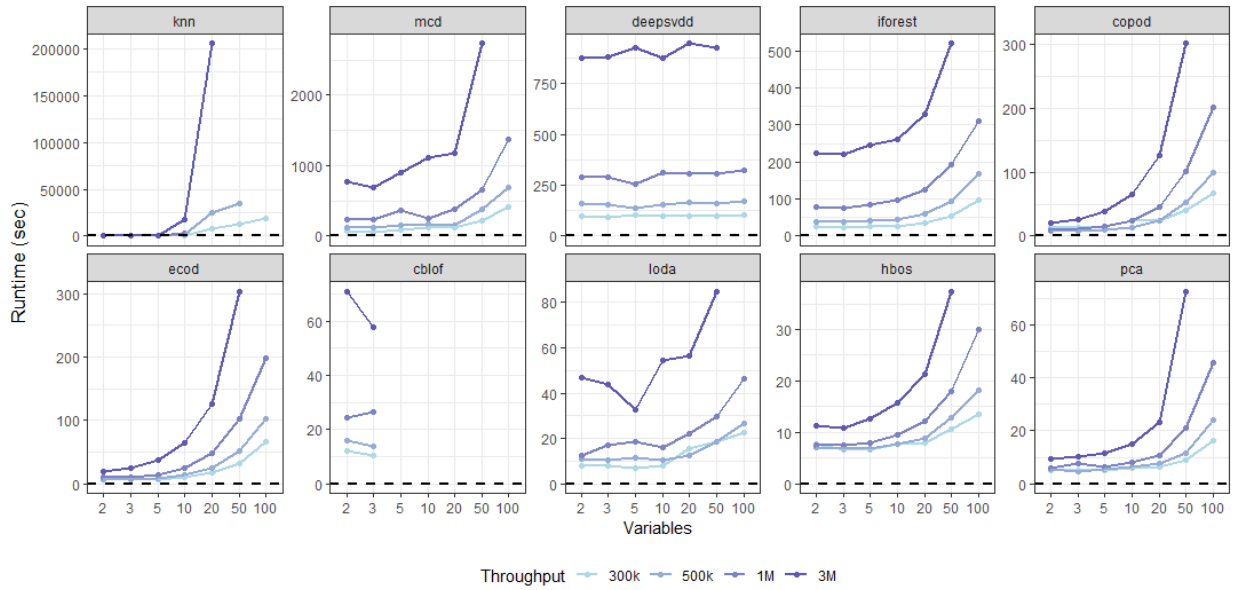
Figure 14: Different scenario speedup results for 3M (simulated). Rows represent results for different OD algorithms (on the right), and columns represent results for time series with increasing number of variables (on the top).

variables. They show that the execution runtime increases exponentially based on time series throughput for all algorithms. Among them, KNN, MCD, and DeepSVDD are particularly unsuited for data scaling, presenting high slope curves. However, it is possible to analyze variable scalability by observing the closeness of the curves. In that case, DeepSVDD seems reasonably unaffected by the number of variables.

This is confirmed in Figure 15b that presents the runtime results for *3M* dataset by varying the number of time series variables. Similarly, different curves represent different stream throughput. DeepSVDD presents an almost continuous



(a) Runtime by time series throughput



(b) Runtime by number of time series variables

Figure 15: Runtime results for 3M dataset (simulated)

runtime with regard to the increasing time series dimensionality. The remaining algorithm runtimes increase exponentially based on the number of variables, while KNN and MCD remain the least scalable algorithms, followed by COPOD and ECOD. Still, most of the algorithms could not run with a time series of 100 variables, resulting in timeout or failure.

Scenario results based on real data To help us answer the third experimental question, Figure 16 shows analogous speedup curves for real multivariate time series drawn from the LTDB dataset. It corresponds to a representative time series of LTDB (*14149*), containing around 11M observations. Although this time series is not high-dimensional, having a fixed number of 2 variables, it allows the speedup analysis over high-throughput real data streams. For this, we generated sliding windows of varying sizes, representing increasing stream throughputs ranging from 300k to 10 million (10M) observations.

Results are consistent with those from the 3M dataset, particularly for low-dimensional streams. Algorithms like KNN, DeepSVDD, MCD, and iForest benefit most from distributed execution. PCA and HBOS, however, continue to suffer from distribution task management overhead. High throughput scenarios yield higher speedup for most algorithms, demonstrating OnDA's robustness across synthetic and real data.

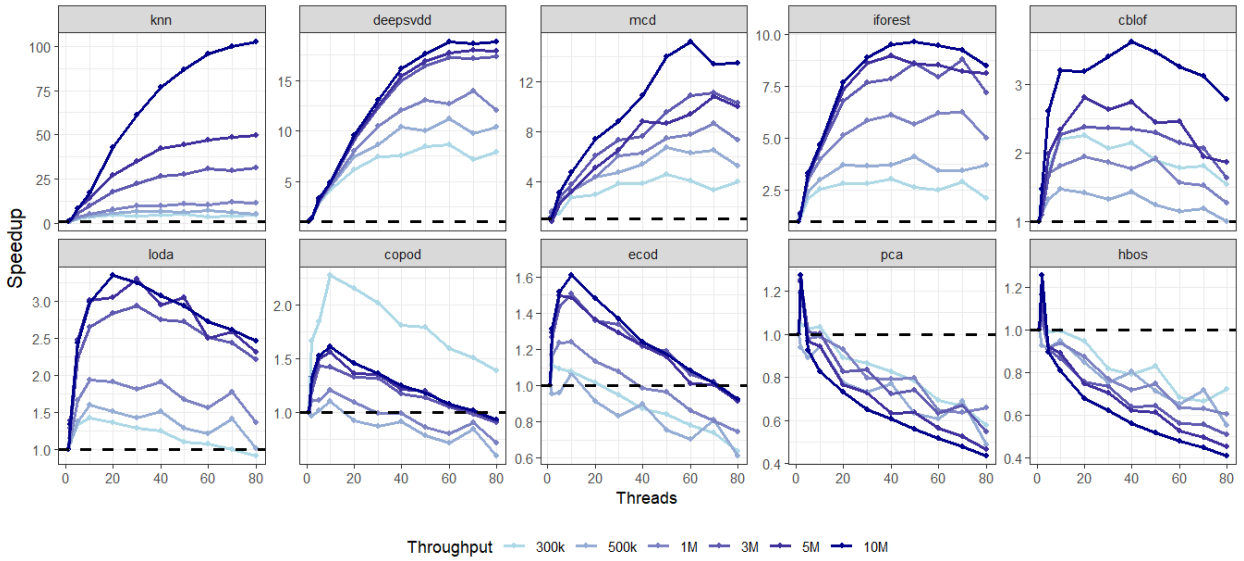


Figure 16: Speedup results for LTDB dataset (real)

Overall efficiency comparison The fourth experimental question examines the consistency of execution efficiency improvements across datasets. We compare the average runtime and speedup of OnDA over OnA for all sliding windows in the time series of the adopted datasets. Results are shown in Figure 17. For comparison, we define a fixed scenario for both online and distributed approaches, focusing on streams with 1M observation throughput. Algorithms like CBLOF and KNN could not handle high-dimensional settings due to parameter and memory constraints. For this reason, CBLOF and KNN are not compared for datasets *3M*, *10M*, and *Kitsune*, which contain high-dimensional time series. Also, for the distributed approach of OnDA, we fix the number of threads *th* to 50. For this definition, based on Figure 14, we took the 1M throughput curves for all algorithms and listed the number of threads that yield the maximum speedup results. Then *th* is the average number of the threads listed.

Figure 17 shows that OnDA significantly reduces runtime compared to OnA for most algorithms and datasets, with statistically significant improvements except for HBOS in dataset *3M*. The bottom plots display the corresponding average speedup. This improvement is pronounced for top-ranked algorithms iForest, MCD, and DeepSVDD across datasets with 2 to 100 variables, indicating the benefits of a distributed approach for low- and high-dimensional time series at 1M throughput. KNN also ranks highly for the low-dimensional LTDB dataset.

Overall accuracy comparison We investigate if data-partitioning leads to changes in OD algorithm detection accuracy by comparing OnDA and OnA performance across all sliding windows of time series datasets. A positive difference indicates increased accuracy with a distributed approach, while a negative difference indicates a decrease. No difference means the accuracy remains unaffected. Figure 18 presents the distributions of the overall differences in accuracy performance based on F1 and AUC-ROC for all algorithms and time series datasets, while Table 5 shows the mean and standard deviations. Results show that OnDA has minimal impact on detection accuracy, with mean differences close to zero and low standard deviations across datasets. Table 5 reinforces this, with most differences falling within $[-0.05, 0.05]$. This indicates that distributed execution maintains accuracy while improving efficiency.

This is a surprising result for a distributed execution based on data partitioning. The OD algorithms can only analyze sub-windows of data, therefore assuming the risk of overlooking global anomalies. Nevertheless, this limitation can become an advantage in a scenario of streaming time series containing mostly local punctual anomalies, making this distributed approach particularly suitable for detecting such anomalies while improving execution efficiency.

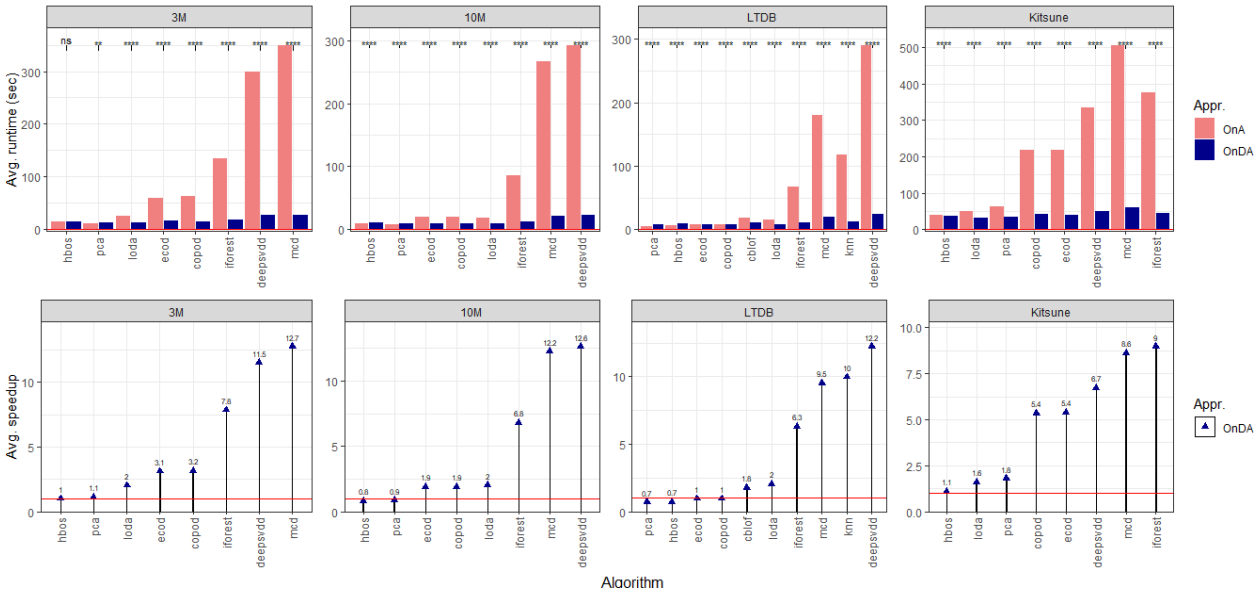


Figure 17: Overall execution efficiency comparison for all OD algorithms over all possible sliding windows of the adopted time series datasets.

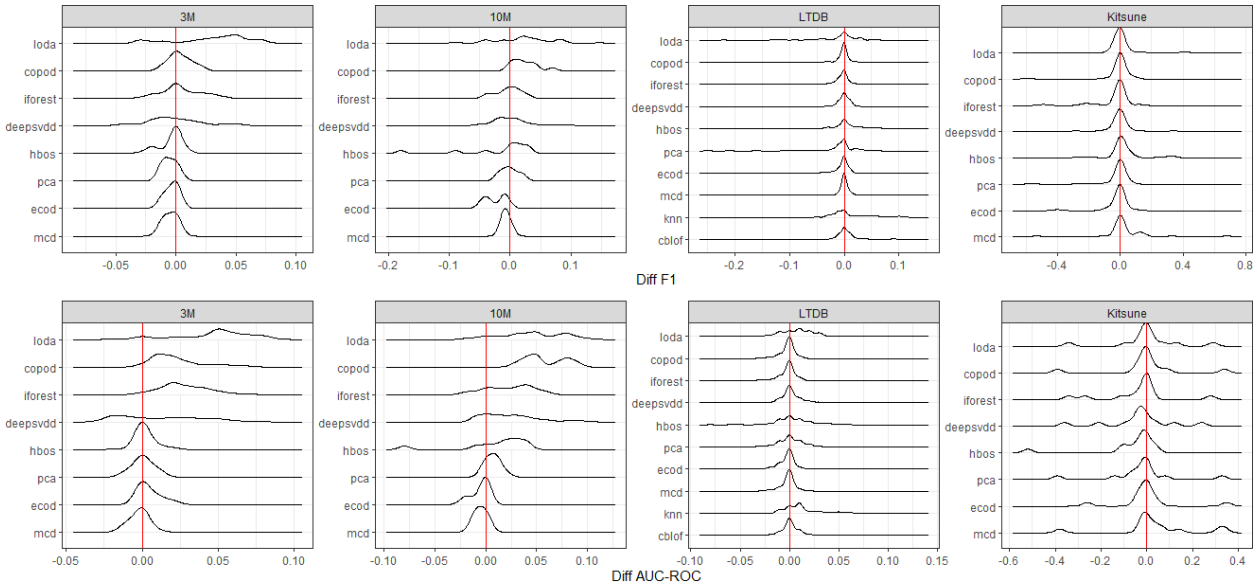


Figure 18: Overall comparison of accuracy for all datasets based on F1 and AUC-ROC. Lines represent the distribution of the differences between accuracy measures from OnDA compared to OnA

5.4.2 Offline vs Online vs Distributed online

Finally, our last experimental question refers to the impact of online and distributed-online OD approaches on the global detection accuracy performances of the algorithms, when compared to an offline centralized OD approach. To help answer this question, Figure 19 compares OnA and OnDA with OffA and shows the mean and standard deviation of their global accuracy differences. As expected, OnA and OnDA generally show reduced global accuracy compared to OffA, particularly for datasets with global anomalies, such as Kitsune. Exceptions are posed by the LODA and MCD algorithms, for which OnA and OnDA approaches seem to result in relevant global accuracy improvements over OffA, especially regarding AUC-ROC measures. Unfortunately, this is actually not a merit of OnA and OnDA, but a demerit of the offline detection of the Kitsune time series based on these algorithms, due to their high-dimensionality.

Table 5: Mean and standard deviation of AUC-ROC and F1 differences between accuracy measures from OnDA compared to OnA

Algorithm	3M		10M		LTDB		Kitsune	
	Diff F1	Diff AUC-ROC	Diff F1	Diff AUC-ROC	Diff F1	Diff AUC-ROC	Diff F1	Diff AUC-ROC
DeepSVDD	0.00 ±0.03	0.01 ±0.03	0.00 ±0.04	0.03 ±0.03	0.00 ±0.01	0.00 ±0.01	0.00 ±0.09	-0.03 ±0.14
MCD	0.00 ±0.01	0.00 ±0.01	-0.01 ±0.00	0.00 ±0.01	0.00 ±0.01	0.00 ±0.01	0.04 ±0.18	0.04 ±0.18
ECOD	0.00 ±0.00	0.00 ±0.01	-0.02 ±0.02	-0.01 ±0.01	0.00 ±0.01	0.00 ±0.01	-0.02 ±0.08	0.00 ±0.13
iForest	0.00 ±0.02	0.03 ±0.02	0.00 ±0.02	0.02 ±0.02	-0.01 ±0.02	0.00 ±0.01	-0.03 ±0.11	-0.04 ±0.15
HBOS	0.00 ±0.01	0.00 ±0.01	-0.02 ±0.06	0.01 ±0.03	0.01 ±0.03	0.00 ±0.02	0.03 ±0.12	-0.06 ±0.14
LODA	0.03 ±0.03	0.05 ±0.03	0.02 ±0.05	0.04 ±0.03	-0.01 ±0.05	0.01 ±0.02	0.01 ±0.08	0.00 ±0.14
COPOD	0.00 ±0.01	0.02 ±0.02	0.02 ±0.02	0.06 ±0.02	0.00 ±0.01	0.00 ±0.01	-0.02 ±0.11	0.00 ±0.15
PCA	-0.01 ±0.01	0.00 ±0.01	0.00 ±0.01	0.01 ±0.01	-0.01 ±0.04	0.00 ±0.01	-0.03 ±0.12	-0.02 ±0.16
KNN	-	-	-	-	0.01 ±0.04	0.01 ±0.03	-	-
CBLOF	-	-	-	-	0.00 ±0.02	0.00 ±0.01	-	-

However, datasets with local anomalies, like 3M and LTDB, show comparable results across all approaches. This indicates OnDA's potential for efficient, scalable anomaly detection with minimal accuracy trade-offs.

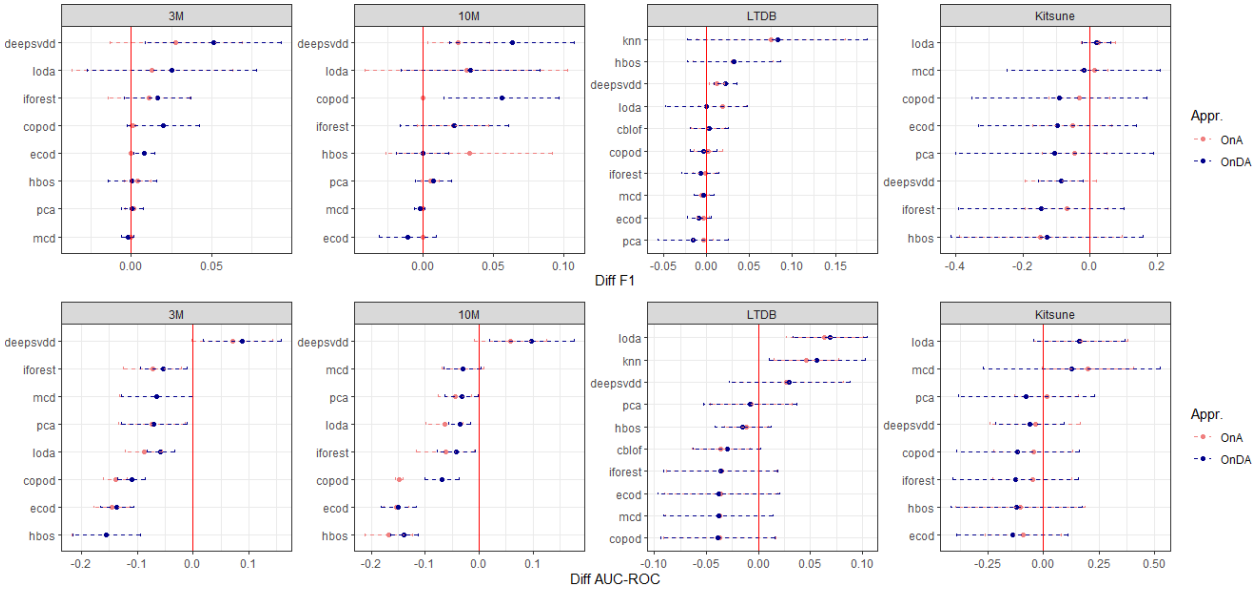


Figure 19: Overall accuracy comparison for all datasets based on F1 and AUC-ROC. Solid lines represent the mean and standard deviations of the differences between accuracy measures from OnA and ONDA compared to OffA

Our results show that online anomaly detection algorithms, like OnA, are comparable to offline ones, like OffA, as supported by prior studies [Ntroumpogiannis et al., 2023b]. This depends on the anomaly type (global or local) and the algorithm's ability to handle high-dimensional time series. The same is true for distributed online methods like OnDA. Overall, OnDA is promising for efficient, scalable online multivariate anomaly detection without losing accuracy.

6 Related Work

A large number of comparative reviews and surveys have been published on outlier and anomaly detection. Olteanu et al. [2023] provides a systematic meta-survey of reviews, analyzing the evolution of outlier detection over 20 years and emphasizing the underexplored area of high-dimensional anomaly detection [Thudumu et al., 2020]. Table 6 lists key papers from the last 8 years, categorized by their focus on anomaly detection in (i) multivariate time series, (ii) online streaming data, and (iii) scalable detection in large, high-dimensional time series.

Multivariate anomaly detection has been a prominent area of study for the last decade. Despite most current surveys focusing on univariate aspects [Munir et al., 2019b], several works review solutions for multivariate time series.

Table 6: Most relevant surveys/comparative reviews published in the last 8 years on anomaly detection. They are categorized based on their main scope focus concerning the problem of anomaly detection over (i) multivariate time series, (ii) online detection over streaming data, and (iii) large and high-dimensional time series data. They are also categorized according to their experimental contribution.

Reference	Year	Main focus			Experimental contribution	
		Multivariate	Online	Scalable	Benchmarking	Methodology
Zamanzadeh Darban et al. [2024]	2024	√				
Li and Jung [2023]	2023	√				
Schmidl et al. [2022]	2022	√			√	
Ntroumpogiannis et al. [2023b]	2023		√		√	
Munir et al. [2019b]	2019		√		√	
Ahmad et al. [2017]	2017		√		√	
Correia et al. [2024]	2024	√	√			
Aggarwal [2017]	2017	√	√			
Ariyaluran Habeeb et al. [2019]	2019		√	√		
Mason et al. [2019]	2019		√	√		
Our work	2025	√	√	√	√	√

Schmidl et al. [2022] provides a comprehensive survey of 33 state-of-the-art methods for multivariate detection (also reviewing 38 methods for univariate time series). Zamanzadeh Darban et al. [2024] surveys 56 deep anomaly detection models in multivariate time series, focusing on forecasting or reconstruction-based strategies. Similarly, Li and Jung [2023] reviews deep learning techniques for anomaly detection in multivariate time series. These works, however, do not address online detection challenges 3.2.1.

Not many works focus on contextualizing online anomaly detection methods. Among them, Ntroumpogiannis et al. [2023b] surveys and experimentally compares 9 online anomaly detection methods from different strategies (i.e., statistics-based, distance-based, and decision-tree-based) evaluated along with their offline counterparts. Munir et al. [2019b], compares 13 online methods, being either traditional or deep learning-based. Also, Ahmad et al. [2017] evaluates 7 unsupervised online anomaly detection methods. However, these works mainly focus on univariate time series anomalies.

The work of Correia et al. [2024], on the other hand, unifies multivariate and online anomaly detection problems, providing a taxonomy and reviewing methods, datasets, and metrics, with a focus on model-based and reconstruction-based strategies. Aggarwal [2017] reviews solutions for outlier detection in multidimensional streaming time series. However, neither of these works prioritize efficiency and scalability.

Ariyaluran Habeeb et al. [2019] and Mason et al. [2019] address online big data processing for anomaly detection. They summarize tools and use cases such as network traffic monitoring, healthcare, and autonomous systems. However, their scope does not address multivariate time series data and their particular demands.

To the best of our knowledge, no surveys or comparative reviews in the literature focus on scalable and online anomaly detection over multivariate time series streaming data. Our work addresses this issue by:

- extending previous surveys on online multivariate anomaly detection methods, considering all detection strategies;
- surveying and contextualizing the state-of-the-art on scalable and distributed online multivariate time series anomaly detection;
- providing a unified taxonomy of multivariate anomaly detection solutions.

Additionally, few surveys propose experimental contributions, generally limited to benchmarking [Ntroumpogiannis et al., 2023b, Schmidl et al., 2022, Munir et al., 2019b, Ahmad et al., 2017]. Only a subset analyze efficiency [Ntroumpogiannis et al., 2023b, Schmidl et al., 2022, Ahmad et al., 2017]. Our study fills this gap by introducing 2OD, a unified methodology for anomaly detection that applies to all reviewed methods, demonstrating the potential of distributed online anomaly detection over high-throughput and high-dimensional time series.

7 Conclusion

This survey presents a comprehensive review of scalable, online anomaly detection methods for multivariate time series, providing a unified taxonomy and analysis of state-of-the-art techniques, especially addressing the unique challenges of high-dimensional, real-time data streams. This survey, fills a gap in the literature, providing researchers and practitioners with a detailed, structured reference for scalable online multivariate anomaly detection.

We also propose 2OD, an innovative methodology for evaluating the efficiency and accuracy of distributed approaches for a broad array of outlier detection methods in high-throughput scenarios, thereby extending the applicability of distributed analysis to previously offline or centralized algorithms. Experimental results demonstrate substantial computational efficiency improvements without compromising detection accuracy, showcasing the potential of distributed approaches across diverse datasets and scenarios.

Acknowledgements

The authors express their gratitude to the OPAL infrastructure from Université Côte d'Azur for providing resources and support. The author(s) used OpenAI ChatGPT 4.0 in order to improve readability and language.

References

A. Abdulaal and T. Lancewicki. Real-time synchronization in neural networks for multivariate time series anomaly detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3570–3574. IEEE, 2021.

H. G. Abreha, M. Hayajneh, and M. A. Serhani. Federated learning in edge computing: a systematic survey. *Sensors*, 22(2):450, 2022.

C. C. Aggarwal. *Outlier Analysis*. Springer Science & Business Media, jan 2013. ISBN 978-1-4614-6396-2.

C. C. Aggarwal. Time series and multidimensional streaming outlier detection. *Outlier analysis*, pages 273–310, 2017.

S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*, 195: 346–361, 2022.

I. Aguilera-Martos, M. García-Barzana, D. García-Gil, J. Carrasco, D. López, J. Luengo, and F. Herrera. Multi-step histogram based outlier scores for unsupervised anomaly detection: Arcelormittal engineering dataset case of study. *Neurocomputing*, 544: 126228, 2023. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.126228.

S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262: 134–147, 2017. doi: 10.1016/j.neucom.2017.04.070.

A. Ahmed, K. Sajan, A. Srivastava, and Y. Wu. Anomaly detection, localization and classification using drifting synchrophasor data streams. *IEEE Transactions on Smart Grid*, 12(4):3570 – 3580, 2021. doi: 10.1109/TSG.2021.3054375.

A. H. Ahmed, M. A. Riegler, S. A. Hicks, and A. Elmokashfi. Rcad: Real-time collaborative anomaly detection system for mobile broadband networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2682 – 2691, 2022. doi: 10.1145/3534678.3539097.

M. Ahmed, A. Mahmood, and M. Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016. doi: 10.1016/j.future.2015.01.001.

R. Akbarinia and B. Cloez. Efficient Matrix Profile Computation Using Different Distance Functions, jan 2019.

S. Alaee, K. Kamgar, and E. Keogh. Matrix profile XXII: Exact discovery of time series motifs under DTW. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, volume 2020-November, pages 900–905, 2020. doi: 10.1109/ICDM50108.2020.00099.

S. Alaee, R. Mercer, K. Kamgar, and E. Keogh. Time series motifs discovery under DTW allows more robust discovery of conserved structure. *Data Mining and Knowledge Discovery*, 35(3):863–910, 2021. doi: 10.1007/s10618-021-00740-0.

E. Alevizos, A. Skarlatidis, A. Artikis, and G. Paliouras. Probabilistic complex event recognition: A survey. *ACM Computing Surveys*, 50(5), 2017. doi: 10.1145/3117809.

A. Aligholian, M. Farajollahi, and H. Mohsenian-Rad. Unsupervised learning for online abnormality detection in smart meter data. In *2019 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5, 2019. doi: 10.1109/PESGM40551.2019.8973564.

Y. Ang, Q. Huang, A. K. H. Tung, and Z. Huang. A stitch in time saves nine: Enabling early anomaly detection with correlation analysis. In *Proceedings - International Conference on Data Engineering*, volume 2023-April, page 1832 – 1845, 2023. doi: 10.1109/ICDE55515.2023.00143.

F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

R. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, I. Targio Hashem, E. Ahmed, and M. Imran. Real-time big data processing for anomaly detection: A Survey. *International Journal of Information Management*, 45:289–307, 2019. doi: 10.1016/j.ijinfomgt.2018.08.006.

F. Arslan, A. Javaid, M. D. Z. Awan, Ebad-ur-Rehman, F. Arslan, A. Javaid, M. D. Z. Awan, and Ebad-ur-Rehman. Anomaly Detection in Time Series: Current Focus and Future Challenges. In *Anomaly Detection - Recent Advances, AI and ML Perspectives and Applications*, pages 1–10. IntechOpen, jul 2023. ISBN 978-1-83769-027-5. doi: 10.5772/intechopen.111886.

J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3395–3404, 2020.

J. Audibert, S. Marti, F. Guyard, and M. A. Zuluaga. From univariate to multivariate time series anomaly detection with non-local information. In *Advanced Analytics and Learning on Temporal Data: 6th ECML PKDD Workshop, AALTD 2021, Bilbao, Spain, September 13, 2021, Revised Selected Papers 6*, pages 186–194. Springer, 2021.

D. Bäßler, T. Kortus, and G. Gühring. Unsupervised anomaly detection in multivariate time series with online evolving spiking neural networks. *Machine Learning*, 111(4):1377 – 1408, 2022. doi: 10.1007/s10994-022-06129-4.

K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu. Autoencoders and their applications in machine learning: a survey. *Artificial Intelligence Review*, 57(2):28, 2024.

K. Bhaduri, B. L. Matthews, and C. R. Giannella. Algorithms for speeding up distance-based outlier detection. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 859–867, 2011.

A. Blázquez-García, A. Conde, U. Mori, and J. Lozano. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys*, 54(3), 2021. doi: 10.1145/3444690.

P. Boniol, M. Linardi, F. Roncallo, T. Palpanas, M. Meftah, and E. Remy. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal*, 30(6):909–931, 2021a.

P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin. Sand: Streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment*, 14(10):1717 – 1729, 2021b. doi: 10.14778/3467861.3467863.

P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin. Sand in action: Subsequence anomaly detection for streams. *Proceedings of the VLDB Endowment*, 14(12):2867 – 2870, 2021c. doi: 10.14778/3476311.3476365.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner. Scalable distance-based outlier detection over high-volume data streams. In *2014 IEEE 30th international conference on data engineering*, pages 76–87. IEEE, 2014.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 2009. doi: 10.1145/1541880.1541882.

H. Chen, Y. Wang, Y. Wang, and X. Ma. Gdsw: a general framework for distributed sliding window over data streams. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 729–736. IEEE, 2016.

Y. Choi, H. Lim, H. Choi, and I.-J. Kim. Gan-based anomaly detection and localization of multivariate time series data for power plant. In *Proceedings - 2020 IEEE International Conference on Big Data and Smart Computing, BigComp 2020*, page 71 – 74, 2020. doi: 10.1109/BigComp48618.2020.00-97.

M. Corain, P. Garza, and A. Asudeh. Dbscout: A density-based method for scalable outlier detection in very large datasets. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 37–48. IEEE, 2021.

L. Correia, J.-C. Goos, P. Klein, T. Bäck, and A. V. Kononova. Online model-based anomaly detection in multivariate time series: Taxonomy, survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence*, 138:109323, 2024.

X. S. de Cámara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza. Clustered federated learning architecture for network anomaly detection in large scale heterogeneous iot networks. *Computers & Security*, 131:103299, 2023.

D. De Paepe, S. Vanden Hautte, B. Steenwinckel, F. De Turck, F. Ongenae, O. Janssens, and S. Van Hoecke. A generalized matrix profile framework with support for contextual series analysis. *Engineering Applications of Artificial Intelligence*, 90, 2020. doi: 10.1016/j.engappai.2020.103487.

J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, and others. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237, 2005.

K. DeMedeiros, A. Hendawi, and M. Alvarez. A survey of ai-based anomaly detection in iot and sensor networks. *Sensors*, 23(3):1352, 2023.

J. E. d'Hondt, O. Papapetrou, and J. Paparrizos. Beyond the dimensions: A structured evaluation of multivariate time series distance measures. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 107–112. IEEE, 2024.

H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. In *Proceedings of the VLDB Endowment*, volume 1, pages 1542–1552, 2008. doi: 10.14778/1454159.1454226.

X. Ding, Y. Li, A. Belatreche, and L. Maguire. An experimental evaluation of novelty detection methods. *Neurocomputing*, 135: 313–327, 2014. doi: 10.1016/j.neucom.2013.12.002.

P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys*, 45(1), 2012. doi: 10.1145/2379776.2379788.

J. Fan, F. Han, and H. Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.

J. Fan, Z. Wang, D. Sun, and H. Wu. Sepformer-based models: More efficient models for long sequence time-series forecasting. *IEEE Transactions on Emerging Topics in Computing*, page 1–12, 2022. doi: 10.1109/TETC.2022.3230920.

C. Feng and P. Tian. Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2858–2867, 2021.

S. Fu, X. Gao, B. Li, F. Zhai, J. Lu, B. Xue, J. Yu, and C. Xiao. Multivariate time series anomaly detection via separation, decomposition, and dual transformer-based autoencoder. *Applied Soft Computing*, 159:111671, 2024.

J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 2014. doi: 10.1145/2523813.

D. Gaspar, F. Porto, R. Akbarinia, and E. Pacitti. Tardis: Optimal execution of scientific workflows in apache spark. In L. Bellatreche and S. Chakravarthy, editors, *Big Data Analytics and Knowledge Discovery*, pages 74–87, Cham, 2017. Springer International Publishing. ISBN 978-3-319-64283-3.

A. Giannoulidis, N. Nikolaidis, and A. Gounaris. Parameter-free streaming distance-based outlier detection. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 102–106. IEEE, 2024.

F. Gmati, S. Chakhar, W. Chaari, and M. Xu. A taxonomy of event prediction methods. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11606 LNAI:12–26, 2019. doi: 10.1007/978-3-030-22999-3_2.

A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 1:59–63, 2012.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

X. Gu, L. Akoglu, and A. Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. *Advances in Neural Information Processing Systems*, 32, 2019.

S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust random cut forest based anomaly detection on streams. In *International conference on machine learning*, pages 2712–2721. PMLR, 2016.

D. N. Gujarati. *Essentials of Econometrics*. SAGE, sep 2021. ISBN 978-1-07-185039-8.

V. Guralnik and J. Srivastava. Event Detection from Time Series Data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 33–42, New York, NY, USA, 1999. ACM. ISBN 978-1-58113-143-7. doi: 10.1145/312129.312190.

J. Han, J. Pei, and H. Tong. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Cambridge, MA, 4th edition edition, oct 2022a. ISBN 978-0-12-811760-6.

S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao. Adbench: Anomaly detection benchmark. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 32142–32159. Curran Associates, Inc., 2022b.

J. Hardin and D. M. Rocke. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis*, 44(4):625–638, 2004.

X. He, B. Wu, Y. Li, F. Li, and J. Tan. Oneshotstl: One-shot seasonal-trend decomposition for online time series anomaly detection and forecasting. *Proceedings of the VLDB Endowment*, 16(6):1399 – 1412, 2023. doi: 10.14778/3583140.3583155.

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

K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018a.

K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 387–395, New York, NY, USA, jul 2018b. Association for Computing Machinery. ISBN 978-1-4503-5552-0. doi: 10.1145/3219819.3219845.

V. Jacob, F. Song, A. Stiegler, B. Rad, Y. Diao, and N. Tatbul. Exathlon: A benchmark for explainable anomaly detection over time series. *Proceedings of the VLDB Endowment*, 14(11):2613 – 2626, 2021. doi: 10.14778/3476249.3476307.

A. Karras, A. Giannaros, L. Theodorakopoulos, G. A. Krimpas, G. Kalogeratos, C. Karras, and S. Sioutas. Flibd: A federated learning-based iot big data management approach for privacy-preserving over apache spark with fate. *Electronics*, 12(22):4633, 2023.

E. Keogh and C. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005. doi: 10.1007/s10115-004-0154-9.

T. Kieu, B. Yang, and C. Jensen. Outlier detection for multidimensional time series using deep neural networks. In *Proceedings - IEEE International Conference on Mobile Data Management*, volume 2018-June, pages 125–134, 2018. doi: 10.1109/MDM.2018.00029.

D. P. Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos. Continuous monitoring of distance-based outliers over data streams. In *2011 IEEE 27th International Conference on Data Engineering*, pages 135–146. IEEE, 2011.

C. Kuo. *Handbook of Anomaly Detection: With Python Outlier Detection: Build and modernize your anomaly detection models with examples*. Independently published, jan 2023. ISBN 9798372339767.

K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu. Revisiting time series outlier detection: Definitions and benchmarks. In J. Vanschoren and S. Yeung, editors, *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, volume 1, pages 1–13, 2021.

B. Lange and P. Fortin. Parallel dual tree traversal on multi-core and many-core architectures for astrophysical N-body simulations. In *20th International Conference Euro-Par 2014 Parallel Processing*, volume 8632 of *Lecture Notes in Computer Science*, pages 716–727, Porto, Portugal, Aug 2014. Springer. doi: 10.1007/978-3-319-09873-9_60.

O. Levchenko, B. Kolev, D.-E. Yagoubi, R. Akbarinia, F. Masseglia, T. Palpanas, D. Shasha, and P. Valduriez. Bestneighbor: efficient evaluation of knn queries on large time series databases. *Knowledge and Information Systems*, 63:349–378, 2021.

G. Li and J. J. Jung. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, 91:93–102, 2023.

Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. Copod: Copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123, 2020. doi: 10.1109/ICDM50108.2020.00135.

Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193, 2022.

Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193, 2023. doi: 10.1109/TKDE.2022.3159580.

J. Lima, R. Salles, F. Porto, R. Coutinho, P. Alpis, L. Escobar, E. Pacitti, and E. Ogasawara. Forward and Backward Inertial Anomaly Detector: A Novel Time Series Event Detection Method. In *2022 International Joint Conference on Neural Networks (IJCNN)*, volume 2022-July, pages 1–8, jul 2022. doi: 10.1109/IJCNN55064.2022.9892088.

J. Lima, L. G. Tavares, E. Pacitti, J. ao Eduardo Ferreira, I. Santos, I. G. aes Siqueira, D. Carvalho, F. Porto, R. Coutinho, and E. Ogasawara. Online event detection in streaming time series: Novel metrics and practical insights. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding Motifs in Time Series. *Proceedings of the Second Workshop on Temporal Data Mining*, 2002.

J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, pages 2–11, 2003. doi: 10.1145/882082.882086.

M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh. Matrix profile goes MAD: variable-length motif and discord discovery in data series. *Data Mining and Knowledge Discovery*, 34(4):1022–1071, 2020. doi: 10.1007/s10618-020-00685-w.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39, 2012.

S. Liu, K. Smith, and H. Che. A multivariate based event detection method and performance comparison with two baseline methods. *Water Research*, 80:109–118, 2015. doi: 10.1016/j.watres.2015.05.013.

X. Liu and P. S. Nielsen. Scalable prediction-based online anomaly detection for smart meter data. *Information Systems*, 77:34–47, 2018. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2018.05.007.

E. Lozano and E. Acufia. Parallel algorithms for distance-based and density-based outliers. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.

Y. Lu, R. Wu, A. Mueen, M. A. Zuluaga, and E. Keogh. Matrix profile xxiv: Scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1173–1182, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539271.

Y. Lu, R. Wu, A. Mueen, M. A. Zuluaga, and E. Keogh. Damp: accurate time series anomaly detection on trillions of datapoints and ultra-fast arriving data streams. *Data Mining and Knowledge Discovery*, 37(2):627 – 669, 2023. doi: 10.1007/s10618-022-00911-7.

P. Malhotra, L. Vig, G. Shroff, P. Agarwal, and others. Long short term memory networks for anomaly detection in time series. In *Esann*, volume 2015, page 89, 2015.

E. Manzoor, H. Lamba, and L. Akoglu. xstream: Outlier detection in feature-evolving data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1963–1972, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220107.

A. Mason, Y. Zhao, H. He, R. Gompelman, and S. Mandava. Online anomaly detection of time series at scale. In *2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, pages 1–8, 2019. doi: 10.1109/CyberSA.2019.8899398.

S. Mehrmolaei and M. Keyvanpourr. A brief survey on event prediction methods in time series. *Advances in Intelligent Systems and Computing*, 347:235–246, 2015. doi: 10.1007/978-3-319-18476-0_24.

W. Merrill and A. Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023.

D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. In *2006 10th IEEE International Symposium on Wearable Computers*, pages 11–18, 2006. doi: 10.1109/ISWC.2006.286337.

Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018. doi: 10.14722/ndss.2018.23204.

S. M. Molaei and M. R. Keyvanpour. An analytical review for event prediction system on time series. In *2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 1–6, mar 2015. doi: 10.1109/PRIA.2015.7161635.

T. Mondal, R. Akbarinia, and F. Masseglia. knn matrix profile for knowledge discovery from time series. *Data Mining and Knowledge Discovery*, 37(3):1055–1089, 2023.

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

A. Mueen. Time series motif discovery: Dimensions and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):152–159, 2014. doi: 10.1002/widm.1119.

M. Munir, M. Chattha, A. Dengel, and S. Ahmed. A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming data. In *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pages 561–566, 2019a. doi: 10.1109/ICMLA.2019.00105.

M. Munir, M. A. Chattha, A. Dengel, and S. Ahmed. A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming data. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 561–566. IEEE, 2019b.

M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2019c. doi: 10.1109/ACCESS.2018.2886457.

A. Ntroumpogiannis, M. Giannoulis, N. Myrtakis, V. Christophides, E. Simon, and I. Tsamardinos. A meta-level analysis of online anomaly detectors. *VLDB Journal*, 32(4):845 – 886, 2023a. doi: 10.1007/s00778-022-00773-x.

A. Ntroumpogiannis, M. Giannoulis, N. Myrtakis, V. Christophides, E. Simon, and I. Tsamardinos. A meta-level analysis of online anomaly detectors. *The VLDB Journal*, 32(4):845–886, 2023b.

E. Ogasawara, D. de Oliveira, P. Valduriez, J. Dias, F. Porto, and M. Mattoso. An algebraic approach for data-centric scientific workflows. *Proceedings of the VLDB Endowment*, 4(12):1328–1339, 2011.

E. Ogasawara, R. Salles, L. Escobar, L. Baroni, J. Lima, and F. Porto. Online event detection for sensor data. In *XLII Ibero-Latin American Congress on Computational Methods in Engineering*, volume 3, pages 1–7, Rio de Janeiro, RJ, 2021.

E. Ogasawara, R. Salles, F. Porto, and E. Pacitti. *Event Detection in Time Series*. Springer Nature, 2024.

J. Oku, K. Tamura, and H. Kitakami. Parallel processing for distance-based outlier detection on a multi-core cpu. In *2014 IEEE 7th International Workshop on Computational Intelligence and Applications (IWCIA)*, pages 65–70. IEEE, 2014.

M. Olteanu, F. Rossi, and F. Yger. Meta-survey on outlier and anomaly detection. *Neurocomputing*, 555:126634, 2023.

M. T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Springer Nature, dec 2019. ISBN 978-3-030-26253-2.

G. Pang, C. Shen, L. Cao, and A. Hengel. Deep Learning for Anomaly Detection: A Review. *ACM Computing Surveys*, 54(2), 2021. doi: 10.1145/3439950.

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

T. Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2016.

M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014. doi: 10.1016/j.sigpro.2013.12.026.

F. Porto, M. Ferro, E. Ogasawara, T. Moeda, C. D. T. de Barros, A. C. Silva, R. Zorrilla, R. S. Pereira, R. N. Castro, J. V. Silva, and others. Machine learning approaches to extreme weather events forecast in urban areas: Challenges and initial results. *Supercomputing Frontiers and Innovations*, 9(1):49–73, 2022.

O. Provotar, Y. Linder, and M. Veres. Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders. In *2019 IEEE International Conference on Advanced Trends in Information Theory, ATIT 2019 - Proceedings*, pages 513–517, 2019. doi: 10.1109/ATIT49449.2019.9030505.

K. Qin, M. Xu, B. A. Muhammad, and J. Han. Mtad rf: Multivariate time-series anomaly detection based on reconstruction and forecast. *Journal of Networking and Network Applications*, 3(1):45–57, 2023.

S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000.

L. Rettig, M. Khayati, P. Cudré-Mauroux, and M. Piórkowski. Online anomaly detection over big data streams. *Applied Data Science: Lessons Learned for the Data-Driven Business*, pages 289–312, 2019.

L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018a.

L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018b.

M. Sakurada and T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11, 2014.

R. Saldanha, R. Akbarinia, M. Pedroso, V. Ribeiro, C. Cardoso, E. H. Pena, P. Valduriez, and F. Porto. Zonal statistics datasets of climate indicators for brazilian municipalities. *Environmental Data Science*, 3:e2, 2024.

F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. *ACM Computing Surveys*, 42(3), 2010. doi: 10.1145/1670679.1670680.

R. Salles, K. Belloze, F. Porto, P. Gonzalez, and E. Ogasawara. Nonstationary time series transformation methods: An experimental review. *Knowledge-Based Systems*, 164:274–291, 2019. doi: 10.1016/j.knosys.2018.10.041.

R. Salles, J. Lima, M. Reis, R. Coutinho, E. Pacitti, F. Masseglia, R. Akbarinia, C. Chen, J. Garibaldi, F. Porto, et al. Softed: Metrics for soft evaluation of time series event detection. *Computers & Industrial Engineering*, 198:110728, 2024.

C. Sanford, D. Hsu, and M. Telgarsky. Transformers, parallel computation, and logarithmic depth. *arXiv preprint arXiv:2402.09268*, 2024.

S. Sathe and C. C. Aggarwal. Subspace histograms for outlier detection in linear time. *Knowledge and Information Systems*, 56: 691–715, 2018.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

S. Schmidl, P. Wenig, and T. Papenbrock. Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779 – 1797, 2022. doi: 10.14778/3538598.3538602.

M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer. Detection of anomalies in large scale accounting data using deep autoencoder networks. *arXiv preprint arXiv:1709.05254*, 2017.

G. Sebestyen, A. Hangan, Z. Czako, and G. Kovacs. A taxonomy and platform for anomaly detection. In *2018 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2018 - THETA 21st Edition, Proceedings*, pages 1–6, 2018. doi: 10.1109/AQTR.2018.8402710.

A. Sgueglia, A. Di Sorbo, C. A. Visaggio, and G. Canfora. A systematic literature review of iot time series anomaly detection solutions. *Future Generation Computer Systems*, 134:170–186, 2022.

J. Shafer, S. Rixner, and A. L. Cox. The hadoop distributed filesystem: Balancing portability and performance. In *2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS)*, pages 122–133. IEEE, 2010.

R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, apr 2017. ISBN 978-3-319-52452-8.

M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. Principal component-based anomaly detection scheme. *Foundations and novel approaches in data mining*, pages 311–329, 2006.

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

I. Souiden, Z. Brahmi, and M. N. Omri. A metaheuristic-based subspace search approach for outlier detection in high-dimensional data streams. In *International Conference on Disruptive Technologies: Innovations & Interdisciplinary Considerations*, pages 29–41. Springer, 2022.

Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837, 2019.

E. Sylligardos, P. Boniol, J. Paparrizos, P. Trahanias, and T. Palpanas. Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. *Proceedings of the VLDB Endowment*, 16(11):3418 – 3432, 2023. doi: 10.14778/3611479.3611536.

S. Tafazoli and E. Keogh. Matrix profile xxviii: Discovering multi-dimensional time series anomalies with k of n anomaly detection. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 685–693. SIAM, 2023.

S. C. Tan, K. M. Ting, and T. F. Liu. Fast anomaly detection for streaming data. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1511–1516, Barcelona, Catalonia, Spain, jul 2011. AAAI Press. ISBN 978-1-57735-514-4.

Y. Tanaka and K. Uehara. Discover motifs in multi-dimensional time-series using the principal component analysis and the MDL principle. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 2734, pages 252–265, 2003. doi: 10.1007/3-540-45065-3_22.

Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*, 58(2-3):269–300, 2005. doi: 10.1007/s10994-005-5829-2.

X. Tao, Y. Peng, F. Zhao, P. Zhao, and Y. Wang. A parallel algorithm for network traffic anomaly detection based on isolation forest. *International Journal of Distributed Sensor Networks*, 14(11):1550147718814471, 2018.

S. Thudumu, P. Branch, J. Jin, and J. Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7:1–30, 2020.

T. Toliopoulos, C. Bellas, A. Gounaris, and A. Papadopoulos. Proud: Parallel outlier detection for streams. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, page 2717 – 2720, 2020. doi: 10.1145/3318464.3384688.

L. F. Tony, T. K. Ming, and Z. Zhi-Hua. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3, 2012.

S. Torkamani and V. Lohweg. Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(2), 2017. doi: 10.1002/widm.1199.

L. Tran, M. Mun, and C. Shahabi. Real-time distance-based outlier detection in data streams. *Proceedings of the VLDB Endowment*, 14(2):141–153, 2020. doi: 10.14778/3425879.3425885.

J. Čulić Gambiroža, T. Mastelić, I. Nižetić Kosović, and M. Čagalj. Lost in data: recognizing type of time series sensor data using signal pattern classification. *International Journal of Data Science and Analytics*, pages 1–12, 2023.

A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

M. Vucovich, A. Tarcar, P. Rebelo, A. Rahman, D. Nandakumar, C. Redino, K. Choi, R. Schiller, S. Bhattacharya, B. Veeramani, and others. Anomaly detection via federated learning. In *2023 33rd International Telecommunication Networks and Applications Conference*, pages 259–266. IEEE, 2023.

D. Wagner, T. Michels, F. C. Schulz, A. Nair, M. Rudolph, and M. Kloft. Timesead: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research*, 2023.

C. Wang, S. Xing, R. Gao, L. Yan, N. Xiong, and R. Wang. Disentangled dynamic deviation transformer networks for multivariate time series anomaly detection. *Sensors*, 23(3):1104, 2023.

H. Wang, M. Bah, and M. Hammad. Progress in Outlier Detection Techniques: A Survey. *IEEE Access*, 7:107964–108000, 2019. doi: 10.1109/ACCESS.2019.2932769.

L. Wang, E. Chng, and H. Li. A tree-construction search approach for multivariate time series motifs discovery. *Pattern Recognition Letters*, 31(9):869–875, 2010. doi: 10.1016/j.patrec.2010.01.005.

P. Wenig, S. Schmidl, and T. Papenbrock. Timeeval: A benchmarking toolkit for time series anomaly detection algorithms. *Proceedings of the VLDB Endowment (PVLDB)*, 15(12):3678 – 3681, 2022. doi: 10.14778/3554821.3554873.

P. Wenig, S. Schmidl, and T. Papenbrock. Anomaly detectors for multivariate time series: The proof of the pudding is in the eating. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 96–101. IEEE, 2024.

P. Xuan and D. Anh. An efficient hash-based method for time series motif discovery. In *Multi-disciplinary Trends in Artificial Intelligence*, volume 11248 LNAI, pages 205–211, 2018. doi: 10.1007/978-3-030-03014-8_17.

D.-E. Yagoubi, R. Akbarinia, F. Masseglia, and T. Palpanas. Massively distributed time series indexing and querying. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):108–120, 2018.

X.-B. Yan, L. Tao, Y.-J. Li, and G.-B. Cui. Research on event prediction in time-series data. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, volume 5, pages 2874–2878, 2004.

Y. Yan, L. Cao, C. Kulhman, and E. Rundensteiner. Distributed local outlier detection in big data. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1225–1234, 2017.

C.-C. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. Dau, Z. Zimmerman, D. Silva, A. Mueen, and E. Keogh. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery*, 32(1): 83–123, 2018. doi: 10.1007/s10618-017-0519-9.

C.-C. M. Yeh, H. Van Herle, and E. Keogh. Matrix Profile III: The Matrix Profile Allows Visualization of Salient Subsequences in Massive Time Series. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 579–588, dec 2016. doi: 10.1109/ICDM.2016.0069.

S. Yoon, J.-G. Lee, and B. S. Lee. Ultrafast local outlier detection from a data stream with stationary region skipping. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1181–1191, 2020.

M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, HotCloud'10, page 10, USA, jun 2010. USENIX Association.

M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauly, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, pages 15–28, 2012.

Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1), oct 2024. ISSN 0360-0300. doi: 10.1145/3691338.

Y. Zhai, Y.-S. Ong, and I. W. Tsang. The emerging" big dimensionality. *IEEE Computational Intelligence Magazine*, 9(3):14–26, 2014.

C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1409–1416, Jul. 2019. doi: 10.1609/aaai.v33i01.33011409.

K. Zhang, Y. Jiang, L. Seversky, C. Xu, D. Liu, and H. Song. Federated variational learning for anomaly detection in multivariate time series. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 1–9. IEEE, 2021.

S. Zhang, V. Ursekar, and L. Akoglu. Sparx: Distributed outlier detection at scale. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4530–4540, 2022a.

Y. Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 12(2):159–170, 2010. doi: 10.1109/SURV.2010.021510.00088.

Y. Zhang, B. Suleiman, and M. J. Alibasa. Fedgroup: a federated learning approach for anomaly detection in iot environments. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 121–132. Springer, 2022b.

L. Zhao. Event Prediction in the Big Data Era: A Systematic Survey. *ACM Computing Surveys*, 54(5), 2021. doi: 10.1145/3450287.

Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019.

Y. Zhao, G. H. Chen, and Z. Jia. Tod: Gpu-accelerated outlier detection via tensor operations. *arXiv preprint arXiv:2110.14007*, 2021a.

Y. Zhao, X. Hu, C. Cheng, C. Wang, C. Wan, W. Wang, J. Yang, H. Bai, Z. Li, C. Xiao, and others. Suod: Accelerating large-scale unsupervised heterogeneous outlier detection. *Proceedings of Machine Learning and Systems*, 3:463–478, 2021b.

X. Zhou, C. Dai, W. Wang, and T. Qiu. Global-local association discrepancy for multivariate time series anomaly detection in iiot. *IEEE Internet of Things Journal*, 2023.

Y. Zhou, R. Arghandeh, H. Zou, and C. Spanos. Nonparametric Event Detection in Multiple Time Series for Power Distribution Networks. *IEEE Transactions on Industrial Electronics*, 66(2):1619–1628, 2019. doi: 10.1109/TIE.2018.2840508.

B. Zhu, Y. Jiang, M. Gu, and Y. Deng. A gpu acceleration framework for motif and discord based pattern mining. *IEEE Transactions on Parallel and Distributed Systems*, 32(8):1987–2004, 2021.

W. Zhu, D. Song, Y. Chen, W. Cheng, B. Zong, T. Mizoguchi, C. Lumezanu, H. Chen, and J. Luo. Deep federated anomaly detection for multivariate time series data. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1–10. IEEE, 2022.

Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 336–345, 2003.

Y. Zhu, C.-C. Yeh, Z. Zimmerman, K. Kamgar, and E. Keogh. Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, volume 2018-November, pages 837–846, 2018a. doi: 10.1109/ICDM.2018.00099.

Y. Zhu, C.-C. M. Yeh, Z. Zimmerman, K. Kamgar, and E. Keogh. Matrix profile xi: Scrimp++: time series motif discovery at interactive speeds. In *2018 IEEE international conference on data mining (ICDM)*, pages 837–846. IEEE, 2018b.

M. Zymbler and Y. Kraeva. High-performance time series anomaly discovery on graphics processors. *Mathematics*, 11(14):3193, 2023.