

# A Review on Outlier/Anomaly Detection in Time Series Data

ANE BLÁZQUEZ-GARCÍA and ANGEL CONDE, Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Spain

USUE MORI, University of the Basque Country (UPV/EHU), Spain

JOSE A. LOZANO, University of the Basque Country (UPV/EHU) & Basque Center for Applied Mathematics (BCAM), Spain

Recent advances in technology have brought major breakthroughs in data collection, enabling a large amount of data to be gathered over time and thus generating time series. Mining this data has become an important task for researchers and practitioners in the past few years, including the detection of outliers or anomalies that may represent errors or events of interest. This review aims to provide a structured and comprehensive state-of-the-art on unsupervised outlier detection techniques in the context of time series. To this end, a taxonomy is presented based on the main aspects that characterize an outlier detection technique.

CCS Concepts: • **Mathematics of computing** → *Continuous functions*; • **General and reference** → *Surveys and overviews*; • **Information systems** → *Data mining*; • **Computing methodologies**;

Additional Key Words and Phrases: Outlier detection, anomaly detection, time series, data mining, taxonomy, software

## ACM Reference format:

Ane Blazquez-Garcia, Angel Conde, Usue Mori, and Jose A. Lozano. 2021. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Comput. Surv.* 54, 3, Article 56 (April 2021), 33 pages.

<https://doi.org/10.1145/3444690>

## 1 INTRODUCTION

Recent advances in technology allow us to collect a large amount of data over time in diverse research areas. Observations that have been recorded in an orderly fashion and that are correlated in time constitute a time series. Time series data mining aims to extract all meaningful knowledge

A. Blázquez-García and A. Conde are supported by Grant No. KK/2019-00095 from the Elkartek program under the DIGITAL project of the Basque Government. U. Mori is supported by the Basque Government through the research group Grant No. IT1244-19 and by the Spanish Ministry of Science, Innovation, and Universities: Grants No. TIN2016-78365-R, and No. PID2019-104966GB-I00; J. A. Lozano is supported by the Basque Government through the BERC 2018-2021 program and research group Grant No. IT1244-19 and by the Spanish Ministry of Science, Innovation, and Universities: BCAM Severo Ochoa accreditation Grants No. SEV-2017-0718, No. TIN2016-78365-R, and No. PID2019-104966GB-I00.

Authors' addresses: A. Blázquez-García and A. Conde, Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Pº J.M. Arizmendiarieta, 2, 20500, Arrasate, Spain; emails: {ablazquez, aconde}@ikerlan.es; U. Mori and J. A. Lozano, Intelligent Systems Group (ISG), Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Manuel de Lardizabal 1, 20018 Donostia, Spain; emails: {usue.mori, ja.lozano}@ehu.eus; J. A. Lozano, Basque Center for Applied Mathematics (BCAM), Mazarredo Zumarkalea, 14, 48009 Bilbao, Spain.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2021 Copyright held by the owner/author(s).

0360-0300/2021/04-ART56

<https://doi.org/10.1145/3444690>

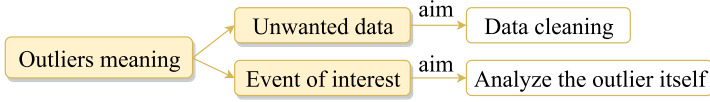


Fig. 1. Meaning of the outliers in time series data depending on the aim of the analyst.

from this data, and several mining tasks (e.g., classification, clustering, forecasting, and outlier detection) have been considered in the literature [Ratanamahatana et al. 2010; Fu 2011; Esling and Agon 2012].

Outlier detection has become a field of interest for many researchers and practitioners and is now one of the main tasks of time series data mining. Outlier detection has been studied in a variety of application domains such as credit card fraud detection, intrusion detection in cybersecurity, or fault diagnosis in industry. In particular, the analysis of outliers in time series data examines anomalous behaviors across time [Gupta et al. 2014a]. In the first study on this topic, which was conducted by Fox [1972], two types of outliers in univariate time series were defined: type I, which affects a single observation; and type II, which affects both a particular observation and the subsequent observations. This work was first extended to four outlier types [Tsay 1988], and then to the case of multivariate time series [Tsay et al. 2000]. Since then, many definitions of the term *outlier* and numerous detection methods have been proposed in the literature. However, to this day, there is still no consensus on the terms used [Carreño et al. 2019]; for example, outlier observations are often referred to as anomalies, discordant observations, discords, exceptions, aberrations, surprises, peculiarities or contaminants.

From a classical point of view, a widely used definition for the concept *outlier* has been provided by Hawkins [1980]:

“An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.”

Therefore, outliers can be thought of as observations that do not follow the expected behavior.

As shown in Figure 1, outliers in time series can have two different meanings, and the semantic distinction between them is mainly based on the interest of the analyst or the particular scenario considered. These observations have been related to noise, erroneous, or unwanted data, which by themselves are not interesting to the analyst [Aggarwal 2016]. In these cases, outliers should be deleted or corrected to improve the data quality and generate a cleaner dataset that can be used by other data mining algorithms. For example, sensor transmission errors are eliminated to obtain more accurate predictions, because the principal aim is to make predictions. Nevertheless, in recent years and, especially in the area of time series data, many researchers have aimed to detect and analyze unusual but interesting phenomena. Fraud detection is an example of this, because the main objective is to detect and analyze the outlier itself. These observations are often referred to as anomalies [Aggarwal 2016].

The purpose of this review is to present a structured and comprehensive state-of-the-art on outlier detection techniques in time series data and attempt to extract the essence of the concept *outlier*, focusing on the detection algorithms given by different authors. Despite the broad terminology that is used to refer to outliers, this review focuses on the identification of outliers in the unsupervised framework, regardless of the term used in the original papers.

Although a number of surveys on outlier detection methods have been presented in the literature [Hodge and Austin 2004; Chandola et al. 2007, 2009; Aguinis et al. 2013; Aggarwal 2016; Xu et al. 2019], very few focus on temporal data, including time series [Gupta et al. 2014b].

In this sense, the main contributions of this article are fourfold:

- We give a comprehensive review that focuses only on time series data. Thus, we provide an in-depth analysis, examining techniques that have hardly been explored in the literature so far. To the best of our knowledge, none of the existing surveys focus exclusively on unsupervised outlier detection in time series, and so, they do not provide enough details of this specific problem and the methods published to solve it.
- We propose a novel taxonomy for outlier detection methods in time series data by extracting the most relevant characteristics of the existing methodologies. This taxonomy provides a global understanding of the outliers and their detection in time series, and helps to choose the type of technique that best adapts to a given problem. We also provide details about other features that characterize each technique. As far as we know, the existing surveys do not propose a complete taxonomy and do not extract the characteristics of each method.
- We provide the publicly available software related to the analyzed methods. This is an important point nowadays, since it allows us to reproduce the methods. The existing related surveys do not present this information.
- We identify some future research directions on outlier detection in time series.

The rest of this article is organized as follows. In Section 1.1, the methodology followed in this study is described. In Section 2, a taxonomy for the classification of outlier detection techniques in time series data are proposed. Sections 3, 4, and 5 present different techniques used for point, subsequence, and time series outlier detection, respectively. The techniques are classified according to the taxonomy proposed in Section 2, and the intuition of the concept *outlier* on which the methods are based is provided. In Section 6, the publicly available software for some of the considered outlier detection methods is presented. Finally, Section 7 contains the concluding remarks and outlines some areas for further research.

## 1.1 METHODOLOGY

This study has been oriented and organized with the intention to answer the following research questions:

- (RQ1) What are the most important characteristics that define each outlier detection method? Based on this, how could the existing techniques be taxonomized?
- (RQ2) How do existing techniques detect point, subsequence and time series outliers in time series data? Which are the main differences between these methods?
- (RQ3) Are there publicly available software packages for outlier detection in time series? What type of methods do they implement?

The methodology used to provide answers to the proposed research questions is an *ad hoc* methodology that consists of three modules: Database Selection, Survey Search, and Literature Search.

*Database Selection.* The databases of scientific research used to do the literature search in this review are the following well-known repositories: Google Scholar, IEEE Xplore, ACM Digital Library, DBLP, Scopus, and ScienceDirect.

*Survey Search.* Since our research article is a literature review, we searched for related reviews that have been previously published. The keywords used are “outlier detection,” “anomaly detection,” “survey,” “review,” and “time series.”

*Literature Search.* To search for articles that propose outlier detection methods in time series, we mainly used the following keywords between 2000 and 2019: “time series,” “outlier detection,” and “anomaly detection.” Other useful keywords have been “univariate,” “multivariate,” “time series

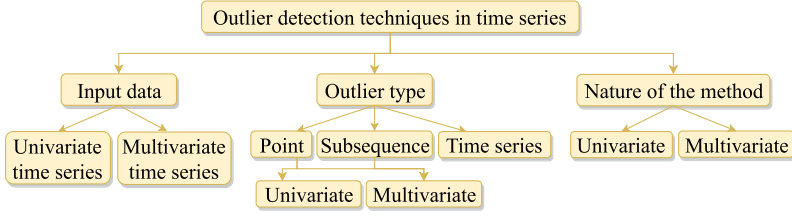


Fig. 2. Proposed taxonomy of outlier detection techniques in time series data.

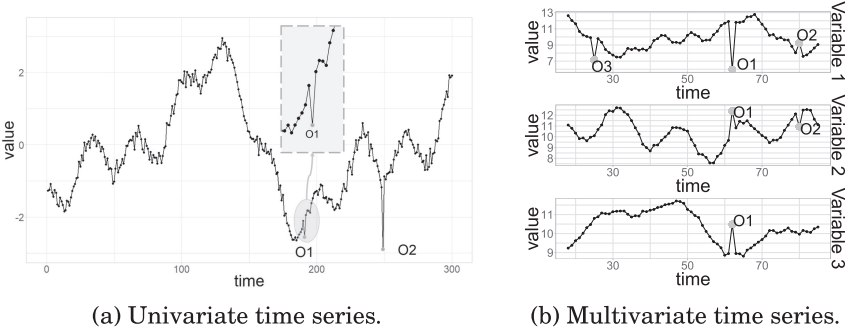


Fig. 3. Point outliers in time series data.

database,” and “subsequence.” We also analyzed additional papers that were referenced within papers already identified. Finally, we manually filtered the results by excluding irrelevant papers that have not been published in high-quality forums and that apply a previously developed method to a particular case rather than proposing a new methodology.

## 2 A TAXONOMY OF OUTLIER DETECTION TECHNIQUES IN THE TIME SERIES CONTEXT

Outlier detection techniques in time series data vary depending on the input data type, the outlier type, and the nature of the method. Therefore, a comprehensive taxonomy that encompasses these three aspects is proposed in this section. Figure 2 depicts an overview of the resulting taxonomy, and each axis is described in detail below.

### 2.1 Input Data

The first axis represents the type of input data that the detection method is able to deal with (i.e., a *univariate* or a *multivariate* time series).

*Definition 2.1 (Univariate time series).* A *univariate time series*  $X = \{x_t\}_{t \in T}$  is an ordered set of real-valued observations, where each observation is recorded at a specific time  $t \in T \subseteq \mathbb{Z}^+$ .

Then,  $x_t$  is the *point* or observation collected at time  $t$  and  $S = x_p, x_{p+1}, \dots, x_{p+n-1}$  the *subsequence* of length  $n \leq |T|$  starting at position  $p$  of the time series  $X$ , for  $p, t \in T$  and  $p \leq |T| - n + 1$ . It is assumed that each observation  $x_t$  is a realized value of a certain random variable  $X_t$ . Some examples of univariate time series can be found in Figures 3(a) and 4(a).

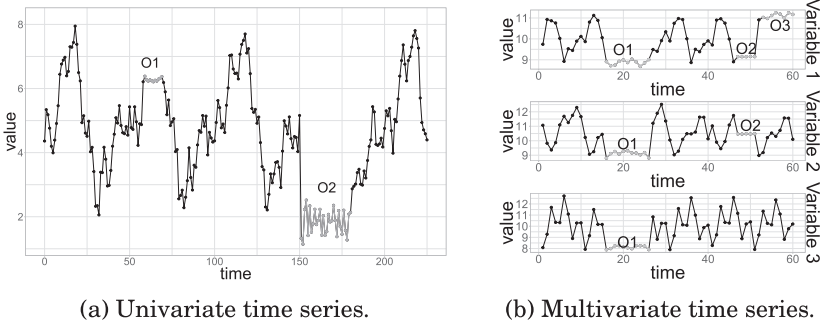


Fig. 4. Subsequence outliers in time series data.

**Definition 2.2 (Multivariate time series).** A *multivariate time series*  $X = \{\mathbf{x}_t\}_{t \in T}$  is defined as an ordered set of  $k$ -dimensional vectors, each of which is recorded at a specific time  $t \in T \subseteq \mathbb{Z}^+$  and consists of  $k$  real-valued observations,  $\mathbf{x}_t = (x_{1t}, \dots, x_{kt})$ .<sup>1</sup>

Then,  $\mathbf{x}_t$  is said to be a point and  $S = \mathbf{x}_p, \mathbf{x}_{p+1}, \dots, \mathbf{x}_{p+n-1}$  a subsequence of length  $n \leq |T|$  of the multivariate time series  $X$ , for  $p, t \in T$  and  $p \leq |T| - n + 1$ . For each dimension  $j \in \{1, \dots, k\}$ ,  $X_j = \{x_{jt}\}_{t \in T}$  is a univariate time series and each observation  $x_{jt}$  in the vector  $\mathbf{x}_t$  is a realized value of a random time-dependent variable  $X_{jt}$  in  $X_t = (X_{1t}, \dots, X_{kt})$ . In this case, each variable could depend not only on its past values but also on the other time-dependent variables. Some examples of multivariate time series can be seen in Figures 3(b) and 4(b).

## 2.2 Outlier Type

The second axis describes the outlier type that the method aims to detect (i.e., a point, a subsequence, or a time series).

- **Point outliers.** A point outlier is a datum that behaves unusually in a specific time instant when compared either to the other values in the time series (global outlier) or to its neighboring points (local outlier). Point outliers can be univariate or multivariate depending on whether they affect one or more time-dependent variables, respectively. For example, Figure 3(a) contains two univariate point outliers, O1 and O2, whereas the multivariate time series composed of three variables in Figure 3(b) has both univariate (O3) and multivariate (O1 and O2) point outliers.
- **Subsequence outliers.** This term refers to consecutive points in time whose joint behavior is unusual, although each observation individually is not necessarily a point outlier. Subsequence outliers can also be global or local and can affect one (univariate subsequence outlier) or more (multivariate subsequence outlier) time-dependent variables. Figure 4 provides an example of univariate (O1 and O2 in Figure 4(a), and O3 in Figure 4(b)) and multivariate (O1 and O2 in Figure 4(b)) subsequence outliers. Note that the latter do not necessarily affect all the variables (e.g., O2 in Figure 4(b)).
- **Outlier time series.** Entire time series can also be outliers, but they can only be detected when the input data are a multivariate time series. Figure 5 depicts an example of an outlier time series, *Variable 4*, in a multivariate time series composed of four variables. The behavior of *Variable 4* significantly differs from the rest.

<sup>1</sup>Note that this definition could include irregularly sampled time series or dimensions (variables) with different temporal granularities, assuming that some of the  $x_{it}$  values might be missing. However, the techniques discussed in this article only consider regularly sampled time series with the same temporal granularity in all dimensions.

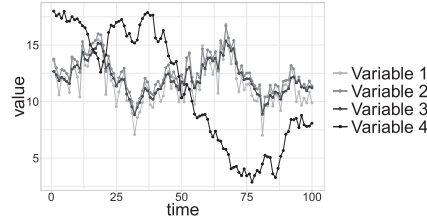


Fig. 5. Outlier time series (*Variable 4*) in a multivariate time series.



Fig. 6. Characteristics related to point outlier detection problems.

Observe that this axis is closely related to the input data type. If the method only allows univariate time series as input, then no multivariate point or subsequence outliers can be identified. In addition, outlier time series can only be found in multivariate time series. Finally, it should be noted that the outliers depend on the context. Thus, if the detection method uses the entire time series as contextual information, then the detected outliers are global. Otherwise, if the method only uses a segment of the series (a time window), then the detected outliers are local, because they are outliers within their neighborhood. Global outliers are also local, but not all local outliers are global. In other words, some local outliers may seem normal if all of the time series is observed but may be anomalous if we focus only on their neighborhood (e.g., O1 in Figure 3(a)).

### 2.3 Nature of the Method

The third axis analyzes the nature of the detection method employed (i.e., if the detection method is *univariate* or *multivariate*). A univariate detection method only considers a single time-dependent variable, whereas a multivariate detection method is able to simultaneously work with more than one time-dependent variable. Note that the detection method can be univariate, even if the input data are a multivariate time series, because an individual analysis can be performed on each time-dependent variable without considering the dependencies that may exist between the variables. In contrast, a multivariate technique cannot be used if the input data are a univariate time series. Thus, this axis will only be mentioned for multivariate time series data.

## 3 POINT OUTLIERS

Point outlier detection is the most common outlier detection task in the area of time series. This section presents the techniques used to detect this type of outlier, in both univariate (Section 3.1) and multivariate (Section 3.2) time series data.

Specifically, as shown in Figure 6, two key characteristics of these methods will be highlighted throughout their presentation. Concerning the first characteristic, or the treatment of temporality, some methods consider the inherent temporal order of the observations, while others completely ignore this information. The main difference between the methods that include temporal information and those that do not is that the latter methods produce the same results, even if they are applied to a shuffled version of the series. Within the methods that use temporality, a subgroup of methods use time windows. Consequently, the same results are obtained when shuffling the observations within the window, but not when shuffling the whole time series.

In relation to the second characteristic (see Figure 6), some techniques are able to detect outliers in streaming time series by determining whether or not a new incoming datum is an outlier as

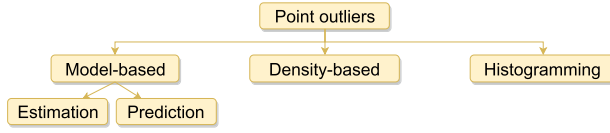


Fig. 7. Types of methods for detecting point outliers in univariate time series.

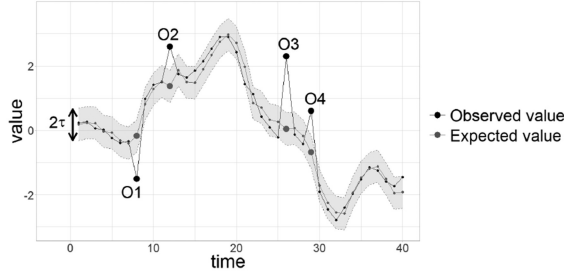


Fig. 8. Point outlier detection in univariate time series based on the comparison of expected and observed values.

soon as it arrives, without having to wait for more data. Within this group, some methods use a fixed model throughout the stream evolution, whereas others update the models used for detection with the new information received—either by retraining the whole model or by learning in an incremental manner. We consider that a technique does not apply to a streaming time series (i.e., non-streaming) if it is unable to make a decision at the arrival of the new datum.

Most of the analyzed point outlier detection techniques can be applied in a streaming context and they take the temporality of the data into account, either by considering the full time series as an ordered sequence or with the use of time windows. Therefore, we will only make reference to this axis for methods that cannot be applied in a streaming environment or that completely ignore the temporal information in the data. Finally, even though many techniques can theoretically deal with streaming time series, very few are able to adapt incrementally to the evolution of the stream. Consequently, we will also highlight these techniques.

### 3.1 Univariate Time Series

The techniques that will be discussed in this section intend to detect point outliers in a univariate time series and are organized based on the diagram shown in Figure 7. Since a single time-dependent variable is considered, recall that these outliers are univariate points and that only univariate detection techniques can be used for their detection.

The most popular and intuitive definition for the concept of *point outlier* is a point that significantly deviates from its expected value. Therefore, given a univariate time series, a point at time  $t$  can be declared an outlier if the distance to its expected value is higher than a predefined threshold  $\tau$ :

$$|x_t - \hat{x}_t| > \tau, \quad (1)$$

where  $x_t$  is the observed data point, and  $\hat{x}_t$  is its expected value. This problem is graphically depicted in Figure 8, where the observed values within the shadowed area are at most at distance  $\tau$  from their expected values.

The outlier detection methods based on the strategy described in Equation (1) are denominated *model-based* methods in this article and are the most common approaches in the literature. Even



Table 1. Data Used in Model-based Techniques in Univariate Time Series,  
for  $k \geq 1$  and  $k_1, k_2 \geq 0$  such that  $k_1 + k_2 > 0$

	Data used	→	Expected value	→	Point outliers
Estimation models	$\{x_{t-k_1}, \dots, x_t, \dots, x_{t+k_2}\}$	→	$\hat{x}_t$	→	$ x_t - \hat{x}_t  > \tau$
Prediction models	$\{x_{t-k}, \dots, x_{t-1}\}$	→	$\hat{x}_t$		

though each technique computes the expected value  $\hat{x}_t$  and the threshold  $\tau$  differently, they are all based on fitting a model (either explicitly or implicitly). As shown in Table 1, if  $\hat{x}_t$  is obtained using previous and subsequent observations to  $x_t$  (past, current, and future data), then the technique is within the *estimation model-based* methods. In contrast, if  $\hat{x}_t$  is obtained relying only on previous observations to  $x_t$  (past data), then the technique is within the *prediction model-based* methods. In practice, the main difference between using estimation or prediction methods is that techniques within this latter category can be used in streaming time series, because they can determine whether or not a new datum is an outlier as soon as it arrives. In the case of estimation methods, this can only be done if, besides some points preceding  $x_t$ , only the current point  $x_t$  is used to compute the estimated value ( $k_2 = 0$ ).

The most simple *estimation models* are based on constant or piecewise constant models, where basic statistics such as the median [Basu and Meckesheimer 2007] or the Median Absolute Deviation (MAD) [Mehrang et al. 2015] are used to obtain  $\hat{x}_t$ . These statistics are calculated using the full series or by grouping the data in equal-length segments and cannot be applied in streaming when future data are needed ( $k_2 > 0$ ). A more sophisticated approach is to utilize unequal-length segments obtained with some segmentation technique. Dani et al. [2015] use the mean of each segment to determine the expected value of the points within that segment, and an adaptive threshold  $\tau_i = \alpha \sigma_i$ , where  $\alpha$  is a fixed value and  $\sigma_i$  the standard deviation of segment  $i$ .

Other estimation-based techniques intend to identify data points that are unlikely if a certain fitted model or distribution is assumed to have generated the data. For instance, some authors model the structure of the data using smoothing methods such as B-splines or kernels [Chen et al. 2010; Holešovský et al. 2018; Čampulová et al. 2018] or variants of the Exponentially Weighted Moving Average (EWMA) method [Carter and Streilein 2012], Song et al. [2015] and Zhang et al. [2016] use slope constraints, Mehrang et al. [2015] assume that the data without outliers are approximately normal, and Reddy et al. [2017] use Gaussian Mixture Models (GMM).

Once a model or distribution is assumed and fitted, Chen et al. [2010], Carter and Streilein [2012], and Reddy et al. [2017] use Equation (1) directly to decide whether a point is an outlier or not. More elaborately, Holešovský et al. [2018] use extreme value theory to compute a confidence limit for the high values of the kernel smoothing residuals ( $|x_t - \hat{x}_t|$ ) and flag point outliers as those exceeding that limit. Within the estimation-methods that use slope constraints, Song et al. [2015] establish a maximum and a minimum possible slope between consecutive values, whereas Zhang et al. [2016] model the change in slopes before and after time  $t$ , assuming that the slopes should not change significantly at a time point. The Extreme Studentized Deviate (ESD) test has also been used to make the decision [Mehrang et al. 2015; Čampulová et al. 2018]: the null hypothesis considered is that there are no outliers, whereas the alternative is that there are up to  $k$ . Regardless of the temporal correlation, the algorithm computes  $k$  test statistics iteratively to detect  $k$  point outliers. At each iteration, it removes the most outlying observation (i.e., the furthest from the mean value). Finally, Čampulová et al. [2018] also propose defining the threshold in Equation (1) using Chebyshev's inequality or the quantiles of normal distribution, after dividing the residuals from kernel regression into homogeneous segments using change point analysis.



Some other univariate outlier detection methods analyze all of the residuals obtained from different models to identify the outliers. For example, Hochenbaum et al. [2017] use the STL decomposition, and Akouemo and Povinelli [2014, 2016, 2017] use ARIMA models with exogenous inputs, linear regression or Artificial Neural Networks (ANNs). Although most of these models can also be used in prediction, in this case, the outliers are detected in the residual set using past and future data. Specifically, once the selected model is learned, hypothesis testing is applied over the residuals to detect the outliers. In Hochenbaum et al. [2017], the ESD test is applied but using the median and MAD instead of the mean and standard deviation, for robustness. In Akouemo and Povinelli [2014, 2016, 2017], assuming that the underlying distribution of the residuals is known, the minimum and maximum values are examined simultaneously at each iteration of the algorithm. The hypothesis to be tested is whether an extremum is an outlier (alternative hypothesis) or not (null hypothesis). The detected outliers are corrected, and the process is repeated until no more outliers are detected.

In contrast to estimation models, techniques based on *prediction models* fit a model to the time series and obtain  $\hat{x}_t$  using only past data; that is, without using the current point  $x_t$  or any posterior observations. Points that are very different from their predicted values are identified as outliers. Recall that all of the techniques within this category can deal with streaming time series.

Within the prediction-based methods, some use a fixed model and thus are not able to adapt to the changes that occur in the data over time. For example, the DeepAnT outlier detection approach presented by Munir et al. [2019] applies a fixed Convolutional Neural Networks (CNNs) to predict values in the future. Other methods use an autoregressive model [Hill and Minsker 2010] or an ARIMA model [Zhang et al. 2012], which obtain confidence intervals for the predictions instead of only point estimates. Consequently, these methods implicitly define the value of  $\tau$ .

Other techniques adapt to the evolution of the time series by retraining the model. As the most basic approach, Basu and Meckesheimer [2007] describe a method that predicts the value  $\hat{x}_t$  with the median of its past data. More elaborately, Zhou et al. [2018a] fit an ARIMA model within a sliding window to compute the prediction interval, so the parameters are refitted each time that the window moves a step forward.

Extreme value theory has also been employed to detect point outliers in streaming univariate series, using past data and retraining. Given a fixed risk  $q$ , Siffer et al. [2017] use this theory to obtain a threshold value  $z_{q,t}$  that adapts itself to the evolution of the data such that  $P(X_t > z_{q,t}) < q$ , for any  $t \geq 0$ , assuming that the extreme values follow a Generalized Pareto Distribution. Incoming data are used to both detect anomalies ( $X_t > z_{q,t}$ ) and refine  $z_{q,t}$ . In particular, the authors propose two algorithms: SPOT, for data following any stationary distribution; and DSPOT, for data that can be subject to concept drift [Tsymbal 2004; Gama et al. 2014].

Some of these prediction-based methods retrain the underlying model periodically, or each time a new point arrives. Therefore, they can adapt to the evolution of the data. However, none of them applies incremental model learning approaches, where the model is not rebuilt from scratch each time but is updated incrementally using only the new information received. This avoids the cost associated with training the models more than once and permits a more gradual adaptation to changes that can occur in the data, which is of special interest in streaming contexts [Losing et al. 2018]. In this sense, and in contrast to the previous approaches, Xu et al. [2016, 2017] suggest modeling a univariate time series stream in an incremental fashion. This method uses Student-t processes to compute the prediction interval and updates the covariance matrix with the newly arrived data point. Ahmad et al. [2017] use the Hierarchical Temporal Memory (HTM) network, which is also a prediction model-based technique that updates incrementally as new observations arrive.

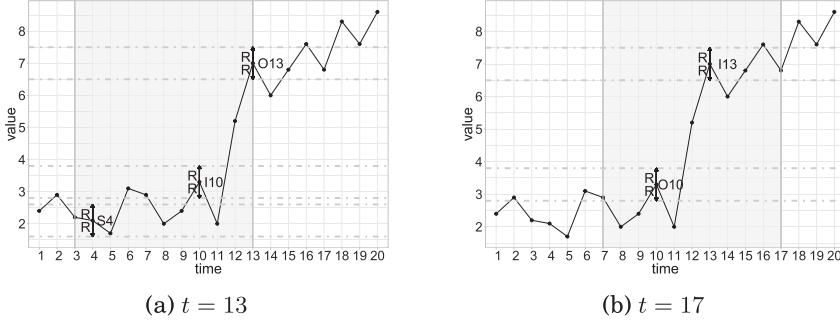


Fig. 9. Density-based outliers within a sliding window of length 11 at time step  $t$ .

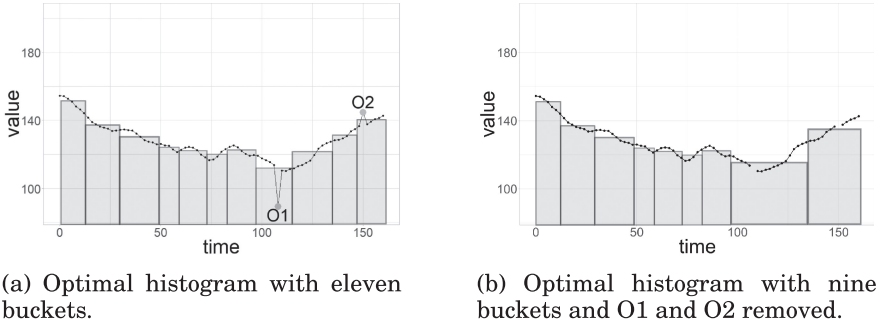


Fig. 10. Example of a deviant set  $D = \{O1, O2\}$  in a univariate time series.

The techniques mentioned so far are based on Equation (1). However, not all the existing point outlier detection methods rely on that idea, such as the *density-based* methods, which belong to the second category depicted in Figure 7. Techniques within this group consider that points with less than  $\tau$  neighbors are outliers; i.e., when less than  $\tau$  objects lie within distance  $R$  from those points. This could be denoted as

$$x_t \text{ is an outlier} \iff |\{x \in X | d(x, x_t) \leq R\}| < \tau, \quad (2)$$

where  $d$  is most commonly the Euclidean distance,  $x_t$  is the data point at time stamp  $t$  to be analyzed,  $X$  is the set of data points, and  $R \in \mathbb{R}^+$ .

The detection of density-based outliers has been widely handled in non-temporal data, but the concept of neighborhood is more complex in time series, because the data are ordered. To take temporality into account, Angiulli and Fassetto [2007, 2010] and Ishimtsev et al. [2017] apply this method within a sliding window, which allows us to determine whether or not a new value of a streaming time series is an outlier upon arrival. An illustration of density-based outliers is provided in Figure 9 at two different time steps with  $R = 0.5$ ,  $\tau = 3$ , and a sliding window of length 11. When using sliding windows, a point can be an outlier for a window (e.g., O13 at  $t = 13$ ) but not for another (e.g., I13 at  $t = 17$ ). However, if a data point has at least  $\tau$  succeeding neighbors within a window, then it cannot be an outlier for any future evolution (e.g., S4 at  $t = 13$ ).

*Histogramming* is the last group analyzed in this section. This type of method is based on detecting the points whose removal from the univariate time series results in a histogram representation with lower error than the original, even after the number of buckets has been reduced to account for the separate storage of these points (see Figure 10). The histogram is built by computing the average of the values within each bucket, in which the order of the observations is preserved.

Then, given a sequence of points  $X$  and a number of buckets  $B$ ,  $D \subset X$  is a deviant set if

$$E_X(H_B^*) > E_{X-D}(H_{B-|D|}^*), \quad (3)$$

where  $H_B^*$  is the optimal histogram (histogram with lowest approximation error) on  $X$  with  $B$  buckets,  $E_X()$  is the total error in the approximation and  $H_{B-|D|}^*$  is the optimal histogram on  $X - D$  with  $B - |D|$  buckets. Jagadish et al. [1999] introduced the term *deviant* to refer to these point outliers. They proposed a dynamic programming mechanism to produce a histogram consisting of  $B - |D|$  buckets and  $|D|$  deviants, minimizing its total error. Some years later Muthukrishnan et al. [2004] observed that for any bucket, the optimal set of  $k = |D|$  deviants always consists of the  $l$  highest and remaining  $k - l$  lowest values within the bucket, for some  $l \leq k$ . Moreover, they presented not only an optimal algorithm for non-streaming time series but also a closely approximate algorithm and a heuristic approach for the streaming case.

To conclude, the techniques that detect point outliers in univariate time series can be grouped mainly in three categories based on the intuition they have of the concept of outlier, and basically what differentiates one technique from another are the characteristics gathered in Table 2. In particular, we consider that a method is iterative if it is repeatedly applied on the same set to find the outliers. Concerning the parametricity of the methods, we consider that parametric methods assume that the underlying distribution of the phenomenon under analysis belongs to a parametric family, and estimate its parameters (a fixed number) from the given data. Contrarily, in non-parametric techniques, the model or the family of distributions is not defined a priori but instead it is determined from the given data and has a flexible number of parameters. Also, we refer to semi-parametric methods as those that combine both parametric and non-parametric approaches.

In general, the authors use the term *outlier*, and the described techniques consider temporality and can be applied in a streaming context. Also, the few iterative methods are related to detecting unwanted data and improving the quality of the time series. Most of the model-based techniques are parametric or semi-parametric. Semi-parametric techniques usually assume a distribution over the residuals when defining a threshold value, even if the residuals are obtained with a non-parametric approach.

Finally, caution must be taken with the estimation methods that can theoretically be applied in a streaming time series; that is, those that do not use subsequent observations to the last arrived data point  $x_t$  ( $k_2 = 0$  in Table 1). Although, it may in theory be possible to apply these techniques in streaming contexts, these methods use the last observation received ( $x_t$ ) and some other past information to calculate its expected value ( $\hat{x}_t$ ) and they then decide whether or not it is an outlier. Consequently, they must perform some calculations after the new point has arrived. The cost of this computation depends on the complexity of each method (which has not been analyzed in this article or in other original works) but may not always guarantee a fast enough response. Thus, in practice, some of these techniques may not be applicable in streaming contexts and prediction methods are more recommendable for these situations.

### 3.2 Multivariate Time Series

The input time series is sometimes a multivariate time series with possibly correlated variables rather than a univariate time series. As opposed to the univariate time series case, the detection method used to identify point outliers in multivariate time series can deal not only with a single variable (Section 3.2.1) but also with more than one variable simultaneously (Section 3.2.2). Additionally, a point outlier in a multivariate time series can affect one (univariate point) or more than one (multivariate point, a vector at time  $t$ ) variables (see Figure 3(b)). As will be seen in the following sections, some multivariate techniques are able to detect univariate point outliers

Table 2. Summary of Point Outlier Detection Techniques in Univariate Time Series

Paper	Technique	Iterative	Temporality	Streaming	Meaning		Parametricity			Term
					I	U	P	SP	NP	
Jagadish et al. [1999]	Histogram	✗	✓	✓	✓				✓	D
Muthukrishnan et al. [2004]	Histogram	✗	✓	✓*	✓				✓	D
Basu and Meckesheimer [2007]	Estimation	✗	W	✗		✓			✓	O
	Prediction	✗	W	✓		✓			✓	O
Angiulli and Fassetto [2007, 2010]	Density	✗	W	✓*	✓				✓	O
Chen et al. [2010]	Estimation	✗	✓	✓		✓		✓		O/C
Hill and Minsker [2010]	Prediction	✗	✓	✓		✓		✓		A
Zhang et al. [2012]	Prediction	✗	✓	✓	✓	✓	✓			O
Carter and Streilein [2012]	Estimation	✗	✓	✓*	✓		✓			A
Akouemo and Povinelli [2014]	Estimation	✓	✓	✓		✓	✓			O
Mehrang et al. [2015]	Estimation <sup>1</sup>	✗	W	✗		✓		✓		O
	Estimation <sup>2</sup>	✓	✗	✓		✓	✓			O
Dani et al. [2015]	Estimation	✗	✓	✓	✓			✓		O/A
Song et al. [2015]	Estimation	✗	✓	✓		✓			✓	Dirty
Zhang et al. [2016]	Estimation	✗	✓	✓		✓			✓	Dirty
Akouemo and Povinelli [2016]	Estimation	✓	✓	✓		✓	✓			O/A
Xu et al. [2016, 2017]	Prediction	✗	✓	✓*	✓				✓	O/A
Ishimtsev et al. [2017]	Density	✗	✓	✓	✓				✓	A
Akouemo and Povinelli [2017]	Estimation	✓	✓	✓		✓	✓			O/A
Reddy et al. [2017]	Estimation	✗	✓	✓	✓		✓			O
Siffer et al. [2017]	Prediction	✗	✗	✓	✓				✓	O/A
Ahmad et al. [2017]	Prediction	✗	✓	✓*	✓			✓		A
Hochenbaum et al. [2017]	Estimation	✓	✗	✓	✗		✓			O/A
Zhou et al. [2018a]	Prediction	✗	✓	✓		✓	✓			O
Holešovský et al. [2018]	Estimation	✗	✓	✓	✓			✓		O
Čampulová et al. [2018]	Estimation	✗	✓	✓	✓				✓	O
Munir et al. [2019]	Prediction	✗	✓	✓	✓		✓			O/A

I: Event of interest; U: Unwanted data // W: Window // P: Parametric; SP: Semi-parametric; NP: Non-parametric // D: Deviant; O: Outlier; A: Anomaly; C: Corrupted.

<sup>1</sup>MAD; <sup>2</sup>Hypothesis testing; \*Incremental updating.

and (similarly) some univariate techniques can be used to detect multivariate point outliers. The characteristics mentioned in Figure 6 will also be highlighted.

**3.2.1 Univariate Techniques.** Given that a multivariate time series is composed of more than one time-dependent variable, a univariate analysis can be performed for each variable to detect univariate point outliers, without considering dependencies that may exist between the variables. Although the literature barely provides examples of this type of approach, in essence, all of the univariate techniques discussed in Section 3.1 could be applied to each time-dependent variable of the input multivariate time series. As one of the few examples, Hundman et al. [2018] propose using the Long Short-Term Memory (LSTM) prediction model-based method to predict spacecraft telemetry and find point outliers within each variable in a multivariate time series, following the idea of Equation (1). The authors also present a dynamic thresholding approach in which some smoothed residuals of the model obtained from past data are used to determine the threshold at each time step.

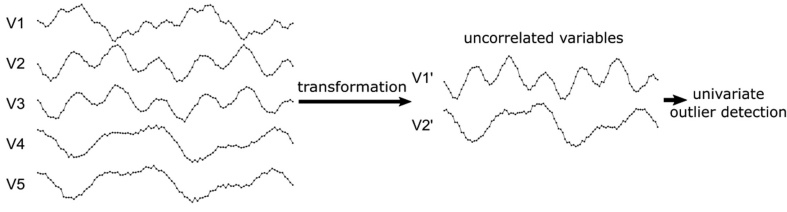


Fig. 11. Simplification of point outlier detection in multivariate time series.

Correlation dependencies between the variables are not considered when applying univariate techniques to each time-dependent variable, leading to a loss of information. To overcome this problem, and at the same time to leverage that univariate detection techniques are highly developed, some researchers apply a preprocessing method to the multivariate time series to find a new set of uncorrelated variables where univariate techniques can be applied. These methods are based on *dimensionality reduction* techniques, and as depicted in Figure 11, the multivariate series is simplified into a representation of lower dimension before applying univariate detection techniques. Since the new series are combinations of the initial input variables, the identified outliers are multivariate; that is, they affect more than one variable.

Some of those dimensionality reduction techniques are based on finding the new set of uncorrelated variables by calculating linear combinations of the initial variables. For example, Papadimitriou et al. [2005] propose an incremental Principal Component Analysis (PCA) algorithm to determine the new independent variables. In this case, the posterior univariate point outlier detection technique that is applied is based on the autoregressive (AR) prediction model. Alternatively, Galeano et al. [2006] suggest reducing the dimensionality with projection pursuit, which aims to find the best projections to identify outliers. The authors mathematically prove that the optimal directions are those that maximize or minimize the kurtosis coefficient of the projected time series. Univariate statistical tests [Fox 1972; Chen and Liu 1993] are then applied iteratively to each projected univariate series for multivariate point outlier detection. Similarly, Baragona and Battaglia [2007] propose using Independent Component Analysis (ICA) to obtain a set of unobservable independent nonGaussian variables. Outliers are identified in each new series independently if Equation (1) is satisfied for  $\hat{x}_{it} = \mu_i$  and  $\tau_i = 4.47\sigma_i$ , where  $\mu_i$  is the mean and  $\sigma_i$  the standard deviation of the  $i$ th new variable.

Other techniques reduce the input multivariate time series into a single time-dependent variable rather than into a set of uncorrelated variables. Lu et al. [2018] define the transformed univariate series using the cross-correlation function between adjacent vectors in time; that is,  $\mathbf{x}_{t-1}$  and  $\mathbf{x}_t$ . Point outliers are iteratively identified in this new series as those that have a low correlation with their adjacent multivariate points. The threshold value  $\tau$  is determined at each iteration by the multilevel Otsu's method. Shahriar et al. [2016] also transform the multivariate time series into a univariate series using a transformation specifically designed for the application domain considered. Point outliers are identified using Equation (1) and the 3-sigma rule.

The main characteristics of all of the techniques analyzed in this section are described in Table 3 in chronological order. Most of the methods reduce the dimensionality of the multivariate time series before applying a univariate detection technique. Particularly, note that the transformation methods proposed by Galeano et al. [2006] and Lu et al. [2018] are specific for outlier detection and not general dimensionality reduction techniques. Also, most of the techniques are non-iterative and would be non-parametric if it were not for the thresholding approach, which typically assumes normality. Finally, point outliers are events of interest for almost all the researchers and are mainly referred to as *outliers*.

Table 3. Summary of the Univariate Techniques Used in Multivariate Time Series for Point Outlier Detection

Paper	Technique	Iterative	Temporality	Meaning		Parametricity			Term
				I	U	P	SP	NP	
Papadimitriou et al. [2005]	Dim. reduction	✗	✓	✓			✓		O
Galeano et al. [2006]	Dim. reduction	✓	✓		✓	✓			O
Baragona and Battaglia [2007]	Dim. reduction	✗	✗	✓			✓		O
Shahriar et al. [2016]	Dim. reduction	✗	✓	✓			✓		O/E
Lu et al. [2018]	Dim. reduction	✓	✓	✓				✓	O
Hundman et al. [2018]	Prediction	✗	✓	✓				✓	A

I: Event of interest; U: Unwanted data // O: Outlier; A: Anomaly; E: Event. // P: Parametric; SP: Semi-parametric; NP: Non-parametric.

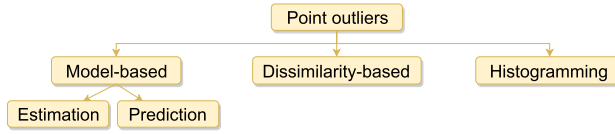


Fig. 12. Types of methods for detecting point outliers in multivariate time series.

As far as the characteristics mentioned in Figure 6 are concerned, in dimensionality reduction techniques, the temporality depends on both the transformation and the outlier detection method applied. If at least one of these considers temporality, then so does the complete method, because the same results are not obtained when the observations of the input multivariate time series are shuffled. In the case of PCA, projection pursuit, and ICA, these transformation methods do not consider temporality, so the temporality depends on the univariate detection method. Conversely, Shahriar et al. [2016] and Lu et al. [2018] use methods that include temporality in the transformation phase. In the other approach where the dimensionality is not reduced, the temporality directly depends on the applied detection method.

Finally, all of the methods that we have reviewed in this section can theoretically detect point outliers in a streaming context using sliding windows, because none of them need future data to provide an output. In this context, the prediction-based approach proposed by Hundman et al. [2018] would work the best, because, in contrast to techniques based on dimensionality reduction, the newly arrived point to be analyzed is not used on the model construction. However, no incremental versions have been proposed.

**3.2.2 Multivariate Techniques.** In contrast to the univariate techniques discussed previously, this section analyzes the multivariate methods that deal simultaneously with multiple time-dependent variables, without applying previous transformations. These approaches perform the detection directly using all the original input data variables and can be divided into three groups, as described in Figure 12.

As in univariate time series, *model-based* techniques can also be used to detect point outliers in multivariate time series. Methods within this group are based on fitting a model that captures the dynamics of the series to obtain expected values in the original input time series. Then, for a predefined threshold  $\tau$ , outliers are identified if

$$\|\mathbf{x}_t - \hat{\mathbf{x}}_t\| > \tau, \quad (4)$$

where  $\mathbf{x}_t$  is the actual  $k$ -dimensional data point, and  $\hat{\mathbf{x}}_t$  its expected value. Note that this is a generalization of the definition given for the model-based techniques in univariate time series and



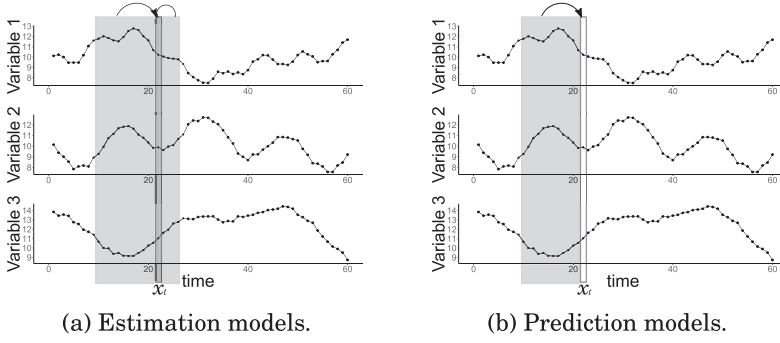


Fig. 13. Example of the data used in model-based techniques.

that the intuition is repeated (refer to Equation (1) and Table 1). In fact,  $\hat{\mathbf{x}}_t$  can be obtained using *estimation models*—which use past, current, and future values—or *prediction models*—which only use past values (see Figure 13).

Within the *estimation model-based* category, autoencoders are one of the most commonly used methods. Autoencoders are a type of neural network that learns only the most significant features of a training set used as the reference of normality. Since outliers often correspond to non-representative features, autoencoders fail to reconstruct them, providing large errors in Equation (4). Sakurada and Yairi [2014] use this method, where the input of the autoencoder is a single multivariate point of the time series. The temporal correlation between the observations within each variable is not considered in this approximation. Therefore, to account for temporal dependencies, Kieu et al. [2018] propose extracting features within overlapping sliding windows (e.g., statistical features) before applying the autoencoder. Su et al. [2019] suggest a more complex approach based on a Variational AutoEncoder (VAE) with a Gated Recurrent Unit (GRU). The input of the model is a sequence of observations containing  $\mathbf{x}_t$  and  $l$  preceding observations to it. The output is the reconstructed  $\mathbf{x}_t$  ( $\hat{\mathbf{x}}_t$ ). Additionally, they apply extreme value theory [Siffer et al. 2017] in the reference of normality to automatically select the threshold value.

Apart from using autoencoders,  $\hat{\mathbf{x}}_t$  can also be derived from the general trend estimation of multiple co-evolving time series. Zhou et al. [2018b, 2019] use a non-parametric model that considers both the temporal correlation between the values within each variable and inter-series relatedness in a unified manner to estimate that trend. In this case, even though the trend of each variable is estimated using the multivariate time series, univariate point outliers are identified within each variable using Equation (1) instead of  $k$ -dimensional data points.

*Prediction model-based* techniques can also be used to obtain  $\hat{\mathbf{x}}_t$  in Equation (4) (see Figure 13(b)). These techniques also fit a model to a multivariate time series, but the expected values are the predictions for the future made on the basis of past values. For example, the Contextual Hidden Markov Model (CHMM) incrementally captures both the temporal dependencies and the correlations between the variables in a multivariate time series [Zhou et al. 2016]. The temporal dependence is modeled by a basic HMM, and the correlation between the variables is included into the model by adding an extra layer to the HMM network. The DeepAnt algorithm [Munir et al. 2019] mentioned in Section 3.1 is also capable of detecting point outliers in multivariate time series using the CNN prediction model. Once the model is learned, the next timestamp is predicted using a window of previous observations as input.

All of these estimation and prediction-based methods can theoretically be employed in a streaming context using sliding windows, because no subsequent points to  $\mathbf{x}_t$  are needed. As with univariate time series, the estimation-based methods need to consider at least the newly arrived point

Table 4. Summary of the Multivariate Techniques Used in Multivariate Time Series for Point Outlier Detection

Paper	Technique	Point	Temporality	Incremental	Parametricity			Term
					P	SP	NP	
Muthukrishnan et al. [2004]	Histogramming	Multivariate	✓	✓			✓	D
Cheng et al. [2008, 2009]	Estimation	Multivariate	✗	✗			✓	A
Li et al. [2009]	Dissimilarity	Univariate	✓	✓			✓	O
Sakurada and Yairi [2014]	Estimation	Multivariate	✗	✗	✓			A
Zhou et al. [2016]	Prediction	Multivariate	✓	✓	✓			N/A'
Kieu et al. [2018]	Estimation	Multivariate	✓	✗	✓			O
Munir et al. [2019]	Prediction	Multivariate	✓	✗	✓			A/O
Zhou et al. [2018b, 2019]	Estimation	Univariate	✓	✗			✓	O/N/E
Su et al. [2019]	Estimation	Multivariate	✓	✗		✓		A

D: Deviant; O: Outlier; A: Anomaly; A': Abnormality; E: Event; N: Novelty // P: Parametric; SP: Semi-parametric; NP: Non-parametric.

$\mathbf{x}_t$  ( $k_2 = 0$  in Table 1), so prediction-based techniques are more appropriate for detecting outliers in a streaming fashion. Moreover, these model-based techniques all use a fixed model, and they do not adapt to changes over time, except for the proposal of Zhou et al. [2016], which is incrementally updated.

The *dissimilarity-based* methods will be discussed next. These techniques are based on computing the pairwise dissimilarity between multivariate points or their representations, without the need for fitting a model. Therefore, for a predefined threshold  $\tau$ ,  $\mathbf{x}_t$  is a point outlier if

$$s(\mathbf{x}_t, \hat{\mathbf{x}}_t) > \tau, \quad (5)$$

where  $\mathbf{x}_t$  is the actual  $k$ -dimensional point,  $\hat{\mathbf{x}}_t$  its expected value, and  $s$  measures the dissimilarity between two multivariate points. These methods do not usually use the raw data directly, but instead use different representation methods. For example, Cheng et al. [2008, 2009] represent the data using a graph where nodes are the multivariate points of the series and the edges the similarity values between them computed with the Radial Basis Function (RBF). The idea is to apply a random walk model in the graph to detect the nodes that are dissimilar to the others (i.e., hardly accessible in the graph). By contrast, Li et al. [2009] propose recording the historical similarity and dissimilarity values between the variables in a vector. The aim is to analyze the dissimilarity of consecutive points over time and detect changes using  $\|\cdot\|_1$ .

The last group depicted in Figure 12 refers to the *histogramming* approach, where the term *deviant* has also been used in the context of multivariate series (see the definition given in Equation (3)). Muthukrishnan et al. [2004] extend the technique for deviant detection explained in Section 3.1 to multivariate time series by treating the measurements collected at the same timestamp as a vector. The authors propose an algorithm for both streaming and non-streaming series to find approximate optimal deviants.

To conclude, the multivariate techniques that detect point outliers in multivariate time series can be grouped mainly in three categories based on the intuition of the concept of outlier that they employ. The main characteristics of these multivariate techniques are depicted in Table 4 in chronological order. Even if most of them find multivariate point outliers, some use the multivariate information to identify point outliers that only affect a single variable (i.e., univariate point outliers). All of the analyzed techniques are non-iterative, and outliers represent events of interest for the researchers. In addition, most of them obtain different results if they are applied to a shuf-

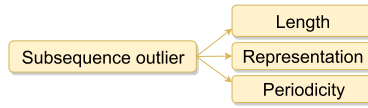


Fig. 14. Characteristics related to subsequence outlier detection problems.

fled version of the time series, i.e., most methods consider the temporal information. Although all of these methods can detect outliers in a streaming context, few are incremental or updated as new data arrives. Finally, most of the model-based techniques are parametric or semi-parametric, while all the techniques based on histograms or dissimilarity are non-parametric.

#### 4 SUBSEQUENCE OUTLIERS

As shown in Figure 2, subsequence outliers are the second type of outliers that can be detected in time series data. In this case, the aim is to identify a set of consecutive points that jointly behave unusually. To this end, subsequence outlier detection methods need to consider some key aspects, which are shown in Figure 14, and which make the detection of subsequence outliers more challenging than point outlier detection.

To begin with, subsequences are composed of a set of points and not of a single point, so they have a certain length. Typically, methods consider fixed-length subsequences, although some techniques do allow us to detect subsequences of different lengths (variable-length) simultaneously (e.g., Senin et al. [2015] and Wang et al. [2018]).<sup>2</sup> Methods based on fixed-length subsequences need the user to predefine the length, and they commonly obtain the subsequences using a sliding window over the time series. In contrast, methods that allow finding variable-length subsequences do not prespecify this length. A final aspect to consider regarding the length of the subsequence is that the number of subsequences that the method will consider and analyze depends on the chosen length (i.e., the shorter the length, the higher the number of subsequences).

The second characteristic that subsequence outlier detection methods need to consider is the representation of the data. Since the comparison between subsequences is more challenging and costly than the comparison between points, many techniques rely on a representation of the subsequences rather than using the original raw values. Discretization is a widely used representation method and can be obtained using approaches such as equal-frequency binning (e.g., Keogh et al. [2002]), equal-width binning (e.g., Chen and Cook [2011], Chen et al. [2013], and Rasheed and Alhadjj [2014]), or SAX (e.g., Senin et al. [2015] and Wang et al. [2018]). These discretization techniques can also be used as a starting point to obtain other representations, such as bitmaps (e.g., Wei et al. [2005] and Kumar et al. [2005]). A detailed overview of the existing research regarding outlier detection in discrete sequences can be found in Chandola et al. [2012], which highlights the applicability of those techniques to time series data. Additionally, apart from discretization, raw data has also been used directly to obtain representations based on dictionaries (e.g., Carrera et al. [2016]), exemplars (e.g., Jones et al. [2016]), or connectivity values (e.g., Ren et al. [2017]).

Another issue that has been barely considered in the literature but that makes the detection of subsequence outliers more challenging is that they can be periodic. Periodic subsequence outliers are unusual subsequences that repeat over time [Yang et al. 2001; Rasheed and Alhadjj 2014]. Unlike point outliers where periodicity is not relevant, periodic subsequence outliers are interesting in areas such as fraud detection to discover certain periodic anomalous transactions over time.

<sup>2</sup>Note that methods that find fixed-length subsequence outliers could also find subsequences of different lengths by applying the method repeatedly for each possible length.

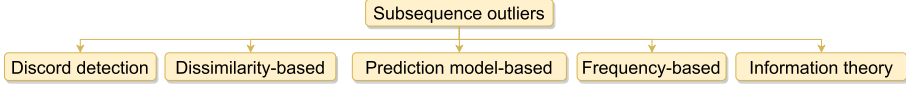


Fig. 15. Types of methods for detecting subsequence outliers in univariate time series.

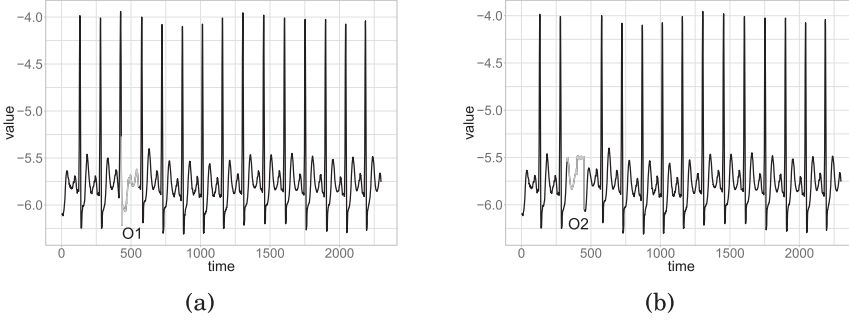


Fig. 16. Discord examples using jmotif package [Senin 2018].

Finally, as opposed to point outlier detection, where some methods did not take into account the temporal dependencies, subsequences consider the temporality by default. In addition, when analyzing subsequence outliers in a streaming context, three cases can occur: (i) a single data point arrives, and an answer (i.e., outlier or non-outlier) must be given for a subsequence containing this new point; (ii) a subsequence arrives, and it needs to be marked as an outlier or non-outlier; and (iii) a batch of data arrives and subsequence outliers need to be found within it. In either case, the literature provides methods that can give an answer in a streaming fashion using sliding windows. However, most of them keep the model fixed and do not adapt to changes in the streaming sequence. We will focus on these incremental techniques, because they are more suitable for processing streaming time series.

#### 4.1 Univariate Time Series

The detection methods used to detect univariate subsequence outliers in univariate time series will be analyzed in this section. We have grouped these techniques according to the different ideas or definitions on which they are based (see Figure 15).

The first and most straightforward idea consists of detecting the most unusual subsequence in a time series (denominated time series *discord*) [Keogh et al. 2005; Lin et al. 2005], by comparing each subsequence with the others; that is,  $D$  is a discord of time series  $X$  if

$$\forall S \in A, \min_{D' \in A, D \cap D' = \emptyset} (d(D, D')) > \min_{S' \in A, S \cap S' = \emptyset} (d(S, S')), \quad (6)$$

where  $A$  is the set of all subsequences of  $X$  extracted by a sliding window,  $D'$  is a subsequence in  $A$  that does not overlap with  $D$  (non-overlapping subsequences),  $S'$  in  $A$  does not overlap with  $S$  (non-overlapping subsequences), and  $d$  is the Euclidean distance between two subsequences of equal length. Typically, discord discovery techniques require the user to specify the length of the discord. Two examples are given in Figure 16, in which the most unusual subsequences (O1 and O2) are shown.

The simplest way of finding discords is using brute-force, which requires a quadratic pairwise comparison. Even so, the process can be sped-up by reordering the search using heuristics and pruning off some fruitless calculations with the HOT-SAX algorithm [Keogh et al. 2005, 2007; Lin et al. 2005], which is based on the SAX discrete representation.

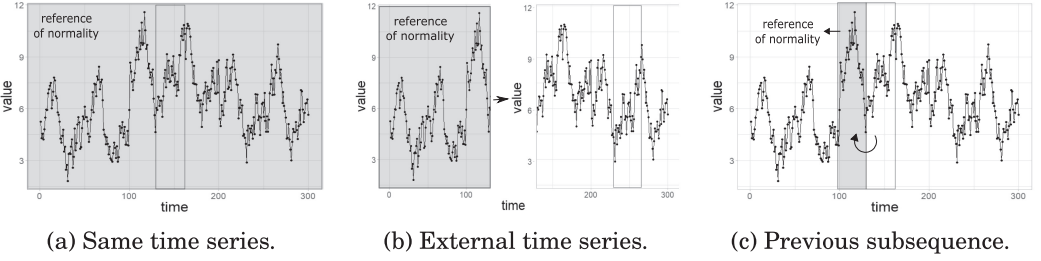


Fig. 17. Reference of normality used by dissimilarity-based approaches.

Many variants of the HOT-SAX algorithm aim to reduce its time complexity by enhancing both the heuristic reordering and the pruning for discord detection, which may be based on the Haar wavelet transformation and augmented trie [Fu et al. 2006; Bu et al. 2007], on bit representation clustering [Li et al. 2013], on bounding boxes [Sanchez and Bustos 2014], on clustering categorical data [Chau et al. 2018], and on the iSAX representation [Buu and Anh 2011]. Additionally, in Liu et al. [2009], not only is the HOT-SAX algorithm applied in a streaming context using a sliding window but an incremental algorithm has also been suggested to simplify this computation. The method proposed by Chau et al. [2018] is also based on this incremental algorithm.

Those discord discovery techniques require the user to prespecify the length of the discord, which in many cases may not be known. Consequently, Senin et al. [2015] present two approaches to detect variable-length discords applying grammar-induction procedures in the time series discretized with SAX. Since symbols that are rarely used in grammar rules are non-repetitive and thus most likely to be unusual, subsequence outliers correspond to rare grammar rules, which naturally vary in length. Given that the lengths of the subsequences vary, the Euclidean distance between them is calculated by shrinking the longest subsequence with the Piecewise Aggregate Approximation [Keogh et al. 2001] to obtain subsequences of the same length.

The abovementioned discord detection techniques are limited to finding the most unusual subsequence within a time series. However, since they do not have a reference of normality or a threshold, they cannot specify whether it is indeed an outlier or not. Therefore, this decision must be made by the user. Conversely, the other categories in Figure 15 consider a criterium of normality and contain specific rules to decide whether or not the identified subsequences are outliers.

For example, *dissimilarity-based* methods are based on the direct comparison of subsequences or their representations, using a reference of normality. In this category, the reference of normality, as well as the representations used to describe the subsequences, vary widely, in contrast to the dissimilarity-based techniques analyzed in Section 3.2.2. Then, for a predefined threshold  $\tau$ , subsequence outliers are those that are dissimilar to normal subsequences; that is,

$$s(S, \hat{S}) > \tau, \quad (7)$$

where  $S$  is the subsequence being analyzed or its representation,  $\hat{S}$  is the expected value of  $S$  obtained based on the reference of normality, and  $s$  measures the dissimilarity between two subsequences. Typically,  $S$  is of fixed-length, non-periodic, and extracted via a sliding window. Some dissimilarity-based approaches are described below ordered based on the considered reference of normality (see Figure 17), which will be used to obtain the expected value  $\hat{S}$ .

*Same time series.* The most straightforward approach is to consider the same time series object of the analysis as the reference of normality (Figure 17(a)). Clustering techniques commonly use this reference of normality to create clusters by grouping subsequences that are similar to each other and by separating dissimilar subsequences into different clusters (see Figure 18). Then,  $\hat{S}$

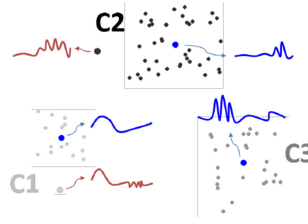


Fig. 18. Clustering of the subsequences in a univariate time series. Cluster centroids are highlighted, and C1 and C2 contain subsequence outliers.

in Equation (7) can be defined by the centroid or center of the cluster to which  $S$  belongs. For example, Chen and Cook [2011] and Chen et al. [2013] cluster discretized subsequences of the same length and flag subsequences that are far from the nearest centroid ( $\hat{S}$ ) as outliers. In this case, the distance used is the Euclidean norm of three specific distances between the discretized subsequences. Alternatively, Izakian and Pedrycz [2013] employ Fuzzy C-Means clustering on the raw data, allowing each subsequence to belong to more than one cluster. In particular, the authors use the Euclidean distance in Equation (7). Note that these two clustering approaches are not applied to streaming time series in the original papers but could be extended using stream clustering algorithms [Silva et al. 2013].

As opposed to the clustering methods that detect fixed-length subsequence outliers, Wang et al. [2018] propose dynamic clustering to identify variable-length outliers in a streaming context. The method works directly with raw data and assumes that the observations arrive in fixed-length batches ( $B_i$  is the  $i$ th batch). Given  $L$  initial batches, grammar-induction is used to find a rule for dividing the incoming batches of data into  $M$  non-overlapping variable-length subsequences; that is,  $B_i = S_{i1} \cup S_{i2} \cup \dots \cup S_{iM}$  (all batches are split in the same points). Clustering is then applied to the set of subsequences  $S_{1j}, S_{2j}, \dots, S_{Lj}$  for each  $j$  separately. A new incoming subsequence is flagged as an outlier using dynamic clustering and observing when it is either far from its closest cluster centroid or belongs to a sparsely populated cluster.

There are other techniques besides clustering that also use the same time series as the reference of normality. For instance, Ren et al. [2017] use the same time series to represent each subsequence with a value that indicates how dissimilar it is with the rest. To this end, a Markov chain random walk model is applied in a graph where each node is a subsequence and each edge the pairwise similarity value between the nodes [Moonesinghe and Tan 2006]. These pairwise similarity values are computed based on the Piecewise Aggregate Pattern Representation, which is a matrix representation that captures the statistical information covered in the subsequence.

*External time series.* The methods that rely on an external time series as a reference of normality assume that it has been generated by the same underlying process but with no outliers. For example, Jones et al. [2016] use an external time series to find a set of exemplars that summarizes all of its subsequences within it and that detects outliers as those that are far from their nearest exemplar ( $\hat{S}$ ). In this case, subsequences are represented by a feature vector of two components that captures both the shape and the stochastic variations of the subsequences in a time series, using smoothing and some statistics (e.g., the mean and the standard deviation). The authors use a weighted Euclidean distance in Equation (7), taking into account the components of both a subsequence and its nearest exemplar.

In this category, we have also included techniques that use past non-outlier subsequences as the reference of normality. Even if these subsequences belong to the same time series, this set is considered as an external set where outliers are not detected, unlike the previous category. For



instance, Carrera et al. [2016] collect the most relevant subsequences within a series of past subsequences in a dictionary. Subsequence outliers are those that cannot be accurately approximated by a linear combination of some subsequences of the dictionary ( $\hat{S}$ ), resulting in a large error in Equation (7) using the Euclidean distance [Longoni et al. 2018; Carrera et al. 2019].

*Previous subsequence.* Some techniques only use the previous adjacent non-overlapping window to the subsequence being analyzed as the reference of normality, which means that they have a much more local perspective than the others. Wei et al. [2005] and Kumar et al. [2005] use this reference of normality to obtain  $\hat{S}$ . Specifically, the authors represent each subsequence by a bitmap, a matrix in which each cell represents the frequency of a local region within a subsequence. A subsequence whose local regions differ from the regions of its previous adjacent subsequence ( $\hat{S}$ ) is flagged as an outlier, using the squared Euclidean distance between each pair of elements of the bitmaps. In this case, bitmaps are incrementally updated at each time step.

Returning to the general classification in Figure 15, the third group of methods belongs to the *prediction model-based* category, which assumes that normality is reflected by a time series composed of past subsequences. These methods intend to build a prediction model that captures the dynamics of the series using past data and thus make predictions of the future. Subsequences that are far from those predictions are flagged as outliers, because, although they may resemble subsequences that have appeared previously, they do not follow the dynamics captured by the model:

$$\sum_{i=p}^{p+n-1} |x_i - \hat{x}_i| > \tau, \quad (8)$$

where  $S = x_p, \dots, x_{p+n-1}$  is the subsequence being analyzed,  $\hat{S}$  is its predicted value, and  $\tau$  is the selected threshold. Predictions can be made in two different manners: point-wise or subsequence-wise. Models that use point-wise prediction make predictions for as many points as the length of the subsequence iteratively. With this in mind, any method within Section 3.1 could be used for this purpose. However, since each predicted value is used to predict the subsequent point, the errors accumulate as predictions are made farther into the future. In contrast, the subsequence-wise prediction calculates predictions for whole subsequences at once (not iteratively). Within this category, Munir et al. [2019] use CNN to predict whole subsequences and detect outliers using a model built with past subsequences.

The next subsequence outlier detection methods are the *frequency-based*, as shown in Figure 15, which also use one of the reference of normality mentioned in Figure 17. A subsequence  $S$  is an outlier if it does not appear as frequently as expected:

$$|f(S) - \hat{f}(S)| > \tau, \quad (9)$$

where  $f(S)$  is the frequency of occurrence of  $S$ ,  $\hat{f}(S)$  its expected frequency, and  $\tau$  a predefined threshold. Due to the difficulty of finding two exact real-valued subsequences in a time series when counting the frequencies, these methods all work with a discretized time series.

A paradigmatic example can be found in Keogh et al. [2002]. Given an external univariate time series as reference, a subsequence extracted via a sliding window in a new univariate time series is an outlier relative to that reference if the frequency of its occurrence in the new series is very different to that expected. Rasheed and Alhajj [2014] also propose an algorithm based on the frequency of the subsequences but the aim is to detect periodic subsequence outliers together with their periodicity [Rasheed et al. 2011]. In this case, the reference of normality is the same time series. The intuition behind this method is that a periodic subsequence outlier repeats less frequently in comparison to the more frequent subsequences of same length.

As shown in Figure 15, the last group of subsequence outlier detection methods correspond to *information theory*-based techniques, which are closely related to the frequency-based methods. In particular, Yang et al. [2001, 2004] focus on detecting periodic subsequence outliers in discretized univariate time series using this theory. They assume that a subsequence that occurs frequently is less surprising and thus carries less information than a rare subsequence. Therefore, the aim is to find infrequent but still repetitive subsequences with rare symbols, using the same time series as the reference of normality; that is, Yang et al. [2001, 2004] mark  $S$  as an outlier if

$$I(S) \times f(S) > \tau, \quad (10)$$

where  $I(S) \geq 0$  is the information carried by  $S$  and  $f(S) \geq 1$  is the number of occurrences of  $S$ .  $I(S)$  is computed taking into account the number of times the symbols within  $S$  are repeated through the series, so a discretized subsequence  $S$  that has symbols that do not commonly appear in the time series has a large  $I(S)$ . Conversely, if  $f(S)$  is large ( $S$  occurs frequently), then the information  $I(S)$  will be lower, closer to 0.

To conclude, the techniques that detect subsequence outliers in univariate time series can be divided into five categories according to the different ideas on which they are based. These techniques usually consider a reference of normality, which can be of three types, when detecting the outliers.

A summary of all of these methods is presented in Table 5. Most of them detect non-periodic outliers of fixed-length, assuming the length is known in advance, and use a discretized version of the time series to compare real-valued subsequences effectively or to speed up the search process of outliers, this one specifically in the discord discovery category. In contrast to the many methods that discover point outliers, the analyzed methods for subsequences are not iterative, and there are very few fully parametric methods. In addition, for all of the authors, the outlier represents an event of interest, and they often use the term *discords* to refer to the subsequence outliers.

## 4.2 Multivariate Time Series

This section presents some of the detection techniques that have been used in the literature to detect subsequence outliers in multivariate time series data. The nature of these methods can be either univariate (Section 4.2.1) or multivariate (Section 4.2.2), and they can detect subsequence outliers that affect either one variable (univariate outliers) or multiple variables, which are all aligned (multivariate outliers).

**4.2.1 Univariate Detection Techniques.** Each variable of the multivariate time series can contain subsequence outliers that are independent of other variables. The identification of those subsequences may be carried out by applying the univariate techniques discussed in Section 4.1 to each time-dependent variable. In particular, Jones et al. [2014, 2016] apply the exemplar-based method to each variable of the multivariate input time series, using an external time series as the reference of normality and without considering the correlation that may exist between variables. Recall that omitting this correlation can lead to a loss of information.

Intending to simplify the multivariate task but without completely ignoring the correlation between the variables, some methods reduce the dimensionality of the input series before applying a univariate detection technique. For example, Wang et al. [2018] extend their method for univariate subsequences by applying clustering to a simplified series to detect variable-length subsequence outliers. The simplified series is obtained by computing the distance to some representative subsequences, which have been obtained by applying their univariate technique to each of the variables independently (see Section 4.1). Each new multivariate batch of data are then represented by a vector of distances,  $(d_1, d_2, \dots, d_l)$ , where  $d_j$  represents the Euclidean distance between the  $j^{th}$

Table 5. Summary of the Characteristics of Subsequence Outlier Detection Approaches in Univariate Time Series

Paper	Technique	Periodic outliers	Length		Discretization	Parametricity			Term
			F	V		P	SP	NP	
Yang et al. [2001, 2004]	Inf. theory	✓	✓		✓			✓	S
Keogh et al. [2002]	Frequency	✗	✓		✓	✓			S
Wei et al. [2005]	Similarity	✗	✓		✓			✓	A
Kumar et al. [2005]									
Keogh et al. [2005, 2007]	Discord	✗	✓		✓			✓	D
Lin et al. [2005]									
Fu et al. [2006]	Discord	✗	✓		✓			✓	D/A
Bu et al. [2007]	Discord	✗	✓		✓			✓	D
Liu et al. [2009]	Discord	✗	✓		✓			✓	D
Buu and Anh [2011]	Discord	✗	✓		✓			✓	D
Chen and Cook [2011]	Dissimilarity	✗	✓		✓		✓		O/A
Chen et al. [2013]									
Li et al. [2013]	Discord	✗	✓		✓			✓	D
Izakian and Pedrycz [2013]	Dissimilarity	✗	✓		✗			✓	A
Sanchez and Bustos [2014]	Discord	✗	✓		✗			✓	D/A
Rasheed and Alhajj [2014]	Frequency	✓	✓		✓			✓	O/S
Senin et al. [2015]	Discord	✗		✓	✓			✓	D/A
Carrera et al. [2016, 2019]	Dissimilarity	✗	✓		✗		✓		O/A
Jones et al. [2016]	Dissimilarity	✗	✓		✗			✓	A
Ren et al. [2017]	Dissimilarity	✗	✓		✗	✓			A
Chau et al. [2018]	Discord	✗	✓		✓			✓	D/O/A
Munir et al. [2019]	Model	✗	✓		✗	✓			D/O/A
Wang et al. [2018]	Dissimilarity	✗		✓	✓		✓		A

F: Fixed; V: Variable // S: Surprise; D: Discord; O: Outlier; A: Anomaly // P: Parametric; SP: Semi-parametric; NP: Non-parametric.

variable-length subsequence of the new batch and its corresponding representative subsequence. As with their univariate technique, the reference of normality that is considered by this method is the same time series.

The technique proposed by Hu et al. [2019] is also based on reducing the dimensionality of the time series and allows us to detect variable-length discords, while using the same time series as the reference of normality. This is based on the fact that the most unusual subsequences tend to have local regions with significantly different densities (points that are similar) in comparison to the other subsequences in the series. Each point in the new built univariate time series describes the density of a local region of the input multivariate time series obtained by a sliding window. This series is also used to obtain the variable-length subsequences. Discords are identified using the Euclidean and Bhattacharyya distances simultaneously.

To conclude, Table 6 provides a summary of all these univariate techniques together with their characteristics. Most of the discussed techniques reduce the dimensionality of the multivariate time series before employing a univariate detection technique to detect subsequence outliers. If the univariate technique is applied to each variable, then the detected subsequence outliers will affect a single variable. Otherwise, if the technique is employed in a reduced time series, then the outliers commonly affect multiple variables, because the new series contains multivariate infor-

Table 6. Summary of the Univariate Techniques Used in Multivariate Time Series for Subsequence Outlier Detection

Paper	Technique	Length		Subsequence	Discretization	Parametricity			Term
		Fixed	Variable			P	SP	NP	
Jones et al. [2014]	Dissimilarity	✓		Univariate	✗			✓	A
Wang et al. [2018]	Dim.red.		✓	Univariate	✓		✓		A
Hu et al. [2019]	Dim.red.		✓	Multivariate	✗			✓	A/D

A: Anomaly; D: Discord // P: Parametric; SP: Semi-parametric; NP: Non-parametric.

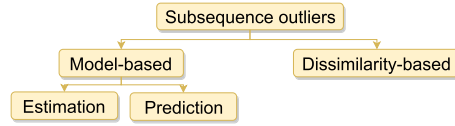


Fig. 19. Types of subsequence outlier detection methods in multivariate time series.

mation. It should be noted that although Wang et al. [2018] identify multivariate batch outliers, the variable-length subsequence outliers that are then identified within those batches affect a single variable. None of them detects periodic subsequence outliers nor are any of them fully parametric. Subsequence outliers are mainly referred to as *anomalies*.

**4.2.2 Multivariate Detection Techniques.** The techniques for multivariate subsequence outlier detection that will be reviewed in this section deal with multiple time-dependent variables simultaneously and typically detect temporally aligned outliers that affect multiple variables. As shown in Figure 19, these techniques are divided into two main groups. However, the philosophy behind some of them is repeated, because they are an extension of simpler techniques introduced in previous sections.

The first type of method to be discussed is the *model-based* method. As mentioned in previous sections, the aim is to detect the subsequences that are far from their expected value, which can be obtained using an *estimation* or a *prediction model*. A subsequence  $S = \mathbf{x}_p, \dots, \mathbf{x}_{p+n-1}$  of length  $n$  is an outlier for a predefined threshold  $\tau$  if

$$\sum_{i=p}^{p+n-1} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| > \tau. \quad (11)$$

Within the techniques based on *estimation models*, the approach proposed by Jones et al. [2014] intends to find pairs of related variables using a set of nonlinear functions. These functions are learned using an external multivariate time series as the reference of normality (see Figure 17). Then, a subsequence within a variable of a new time series is estimated using the learned functions and data from another variable of the new series (past and future observations with respect to  $\mathbf{x}_t$ ). In contrast, the CNN method described by Munir et al. [2019] is a *prediction model-based* technique, which is a direct extension of the method explained in Section 4.1 to detect time-aligned outliers that affect multiple variables.

The second group corresponds to the *dissimilarity-based* techniques, a generalization of Equation (7) that finds unusual subsequences in a multivariate time series based on the pairwise dissimilarity calculation between subsequences or their representations. Unlike in the univariate subsequence outlier detection, this type of technique has barely been used in multivariate series. In the

Table 7. Summary of the Multivariate Techniques Used in Multivariate Time Series for Subsequence Outlier Detection

Paper	Technique	Subsequence	Parametricity	Term
Cheng et al. [2008, 2009]	Dissimilarity	Multivariate	Non-parametric	Anomaly
Jones et al. [2014]	Estimation	Univariate	Parametric	Anomaly
Munir et al. [2019]	Prediction	Multivariate	Parametric	Anomaly/Outlier

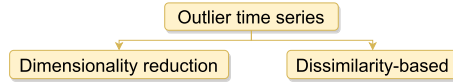


Fig. 20. Types of methods for detecting outlier time series in multivariate time series.

only example, Cheng et al. [2008, 2009] extend their method for point outliers (Section 3.2.2) to obtain how dissimilar is a node representing a fixed-length subsequence with the others. This dissimilarity value is obtained by applying a random walk model in the graph, and the computation of the pairwise similarity between those nodes is also based on the RBF.

To summarize this section, the multivariate techniques that detect subsequence outliers in multivariate time series can be grouped into two different categories according to the intuition behind the methods. A summary of these techniques is given in Table 7. In all of the cases, the subsequence outliers represent an event of interest and are denominated as *anomalies*. In addition, the techniques find non-periodic outliers of fixed-length without using a discretization technique. Finally, the model-based techniques are parametric, while the only dissimilarity-based method is non-parametric.

## 5 OUTLIER TIME SERIES

The analyzed task has so far been to identify point and subsequence outliers within a time series, either univariate or multivariate. However, there are some cases where it may also be interesting to find entire unusual variables in a multivariate time series. This could lead, for example, to the identification of malicious users or mail servers behaving unusually. Recall that outlier time series can only be detected in multivariate time series and using a multivariate detection technique.

Some of the key aspects presented for the subsequence outliers (see Figure 14) can also appear when attempting to detect outlier time series. First, each time-dependent variable in a multivariate time series can have a different length. Second, representation methods such as discretization can also be used to facilitate the comparisons between variables. In addition, all of the outlier time series detection methods can theoretically be applied in a streaming context using sliding windows. However, in contrast to subsequences, the property of temporality is not necessarily considered by these methods.

This section will examine the techniques that detect outlier time series, following the diagram given in Figure 20.

The first type of technique to be discussed is based on *dimensionality reduction*. As mentioned in the previous sections, the aim of these techniques is to reduce the dimensionality of the input multivariate time series into a set of uncorrelated variables. As shown in Figure 21, Hyndman et al. [2015] propose reducing the dimensionality by extracting some representative statistical features from each time series (e.g., mean and first order of autocorrelation) and then applying PCA. Outlier time series are detected by their deviation from the highest density region in the PCA space, which

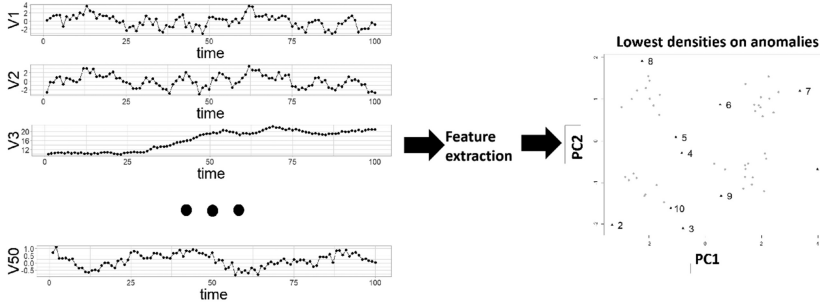


Fig. 21. Outlier time series detection in a multivariate time series composed of 50 variables using PCA in the extracted features with the *anomalous* package in R.

is defined by the first two principal components. Laptev et al. [2015] use clustering in that PCA space to detect outlier time series by measuring the deviation between the cluster centroids and the points that represent the time series.

Instead of applying a detection technique in a lower space, original raw data can also be used directly. Indeed, the *dissimilarity-based* techniques in Figure 20 directly analyze the pairwise dissimilarity between the time series. The most common approach within this category is clustering. The intuition is similar to that depicted in Figure 18, but whole time series are considered instead of subsequences. In this case, the reference of normality is the same multivariate time series.

Within the dissimilarity-based techniques that use clustering, Rebbapragada et al. [2009] propose applying Phased *k*-means in unsynchronized periodic multivariate time series to obtain a set of representative time series centroids. This technique is a modification of *k*-means so that the phase of each time-dependent variable to its closest centroid is adjusted prior to dissimilarity calculation at every iteration. The outliers are identified using Equation (7) and cross-correlation as the similarity measure between time series. Benkabou et al. [2018] also use clustering but, in this case, based on the Dynamic Time Warping (DTW) measure, which allows the series to be warped and shifted in time and have different lengths. The method optimizes an objective function by both minimizing the within-cluster distances using the DTW measure and maximizing the negative entropy of some weights that are assigned to each time series. Time series that increase the within-cluster distances to their closest cluster have smaller weights. Specifically, time series with small weights are considered to be outliers. A different approach that also relies on clustering time series has been proposed by Lara et al. [2019]. This method uses agglomerative hierarchical clustering with single linkage and identifies time series outliers based on four criteria where basically the size of the cluster is considered. Clustering is conducted using a time series similarity measure that has the same underlying idea as the Jaccard similarity coefficient.

Other dissimilarity-based techniques are based on shapelets, which are representative time series subsequences [Ye and Keogh 2009]. Beggel et al. [2019] use them to describe the shape of normal variables and detect outlier time-dependent variables in a multivariate time series. The shapelets are learned using an external time series as the reference of normality. The idea is to measure the dissimilarity that each learned shapelet has with a variable using the Euclidean distance. Subsequences of outlier variables are dissimilar to the learned shapelets.

To conclude, outlier time series can be grouped into two categories based on the intuition behind the detection method. A summary of these techniques is provided in Table 8. The works reviewed in this section are non-parametric and non-iterative techniques that intend to find events of interest by directly using raw data and considering temporality. Unlike the subsequences, the length



Table 8. Summary of the Characteristics of Outlier Time Series Detection in Multivariate Time Series

Paper	Technique	Variable-length	Term
Rebbapragada et al. [2009]	Dissimilarity-based	✗	Anomaly/Outlier
Hyndman et al. [2015]	Dimensionality reduction	✗	Anomaly/Outlier/Unusual
Benkabou et al. [2018]	Dissimilarity-based	✓	Outlier
Beggel et al. [2019]	Dissimilarity-based	✗	Anomaly
Lara et al. [2019]	Dissimilarity-based	✗	Outlier

of the series is always specified. The approach proposed by Benkabou et al. [2018] is the only one that can deal with time series with variables of different lengths.

## 6 PUBLICLY AVAILABLE SOFTWARE

In this section, the publicly available software for outlier detection in time series data according to the techniques mentioned in previous sections is provided. A summary of this software can be found in Table 9, in which the technical descriptions (*Related research* column) and the link to access the code (*Code* column) are presented. The table is organized based on the outlier type the technique detects.

## 7 CONCLUDING REMARKS AND FUTURE WORK

In this article, an organized overview of outlier detection techniques in time series data has been proposed. Moreover, a taxonomy that categorizes these methods depending on the input data type, the outlier type, and the nature of the detection method has been included. This section first discusses some general remarks about the analyzed techniques, and it then introduces the conclusions regarding the axes of the taxonomy.

As seen in previous sections, a broad terminology has been used to refer to the same concept: the outlier in unlabeled time series data. These terms are usually related to the objective of the detection such that *outlier* is mostly used when detecting unwanted data, whereas *anomaly* has been used when detecting events of interest. In most of the analyzed works, the concept outlier represents an event of interest; that is, the authors mainly focus on extracting the outlier information considered as useful data rather than on cleaning the useless or unwanted data to improve the data quality for further analysis. Thus, it might be interesting to extend this type of methods by developing techniques that improve the data quality, mainly of multivariate time series data.

Despite the variety in terminology and purpose, all of the methods that we have reviewed are based on the same idea: detecting those parts in a time series (point, subsequence, or whole time series) that significantly differ from their expected value. Each author uses different means to obtain this expected value and compute how far it is from the actual observation to decide whether or not it is an outlier. Although some techniques obtain the expected value based on an external or reference set, caution must be taken, because it can itself contain outliers and distort the detection. In fact, they are in the limit of the unsupervised framework, because even if the time series has no labels, it is usually assumed that all the external or reference data are non-outliers.

Once the expected value is obtained, a threshold is often needed to decide whether or not we have found an outlier. Given that the results directly vary depending on that selection and few techniques give an automatic way to determine the threshold [Siffer et al. 2017; Lu et al. 2018; Hundman et al. 2018; Su et al. 2019], an interesting future line of research would be to deepen on the dynamic and adaptive selection of thresholds in both univariate and multivariate time series.

Table 9. Summary of the Publicly Available Software in Chronological Order

Name	Language	Related research	Code
Point outliers			
tsoutliers	R	Chen and Liu [1993]	<a href="https://cran.r-project.org/web/packages/tsoutliers">https://cran.r-project.org/web/packages/tsoutliers</a>
spirit	Matlab	Papadimitriou et al. [2005]	<a href="http://www.cs.cmu.edu/afs/cs/project/spirit-1/www">http://www.cs.cmu.edu/afs/cs/project/spirit-1/www</a>
STORM	Java	Angiulli and Fasseti [2007, 2010]	<a href="https://github.com/Waikato/moa/tree/master/moa/src/main/java/moa/clusterers/outliers/Angiulli">https://github.com/Waikato/moa/tree/master/moa/src/main/java/moa/clusterers/outliers/Angiulli</a>
SCREEN	Java	Song et al. [2015]	<a href="https://github.com/zaqthss/sigmod15-screen">https://github.com/zaqthss/sigmod15-screen</a>
EGADS	Java	Laptev et al. [2015]	<a href="https://github.com/yahoo/egads">https://github.com/yahoo/egads</a>
SCR	Java	Zhang et al. [2016]	<a href="https://github.com/zaqthss/sigmod16-scr">https://github.com/zaqthss/sigmod16-scr</a>
libspot	C++	Siffer et al. [2017]	<a href="https://github.com/asiffer/libspot">https://github.com/asiffer/libspot</a>
AnomalyDetection	R	Hochbaum et al. [2017]	<a href="https://github.com/twitter/AnomalyDetection">https://github.com/twitter/AnomalyDetection</a>
Nupic	Python	Ahmad et al. [2017]	<a href="https://github.com/numenta/nupic">https://github.com/numenta/nupic</a>
teleanom	Python	Hundman et al. [2018]	<a href="https://github.com/khundman/teleanom">https://github.com/khundman/teleanom</a>
OmniAnomaly	Python	Su et al. [2019]	<a href="https://github.com/smallcowbaby/OmniAnomaly">https://github.com/smallcowbaby/OmniAnomaly</a>
OTSAD	R	Carter and Streilein [2012]; Ishimtsev et al. [2017]; Iturria et al. [2020]	<a href="https://cran.r-project.org/package=otsad">https://cran.r-project.org/package=otsad</a>
envoutliers	R	Holešovský et al. [2018]; Čampulová et al. [2018]	<a href="https://cran.r-project.org/web/packages/envoutliers/index.html">https://cran.r-project.org/web/packages/envoutliers/index.html</a>
Subsequence outliers			
tsbitmaps	Python	Wei et al. [2005]; Kumar et al. [2005]	<a href="https://github.com/binhmop/tsbitmaps">https://github.com/binhmop/tsbitmaps</a>
jmotif	R	Keogh et al. [2005, 2007] Senin et al. [2015, 2018]	<a href="https://github.com/jMotif/jmotif-R">https://github.com/jMotif/jmotif-R</a>
jmotif	Java	Keogh et al. [2005, 2007] Senin et al. [2015]	<a href="https://github.com/jMotif/SAX">https://github.com/jMotif/SAX</a>
saxpy	Python	Keogh et al. [2005, 2007]	<a href="https://pypi.org/project/saxpy">https://pypi.org/project/saxpy</a> <a href="https://github.com/seninp/saxpy">https://github.com/seninp/saxpy</a>
EBAD	C	Jones et al. [2016]	<a href="http://www.merl.com/research/license">http://www.merl.com/research/license</a>
GrammarViz	Java	Senin et al. [2015, 2018]	<a href="https://github.com/GrammarViz2/grammarviz2_src">https://github.com/GrammarViz2/grammarviz2_src</a>
Outlier time series			
anomalous	R	Hyndman et al. [2015]	<a href="http://github.com/robjhyndman/anomalous-acm">http://github.com/robjhyndman/anomalous-acm</a>

Indeed, to the best of our knowledge, there are no methods that include this type of threshold in the subsequence and entire time series outlier analysis.

As a final general remark and before proceeding with the conclusions regarding the axes of the taxonomy, the time elapsed from one observation to the subsequent is also an important aspect to consider. The vast majority of methods assume that the time series are regularly sampled. However, real life often does not provide this type of data, and converting it into such type is not always the best option. Therefore, outlier detection in an irregularly sampled time series is an interesting future direction line.

Having provided some general conclusions, we will now focus on each of the axes. Starting from the first axis, the most remarkable conclusion is that even if most of the analysis has been performed on univariate input time series data, in recent years special emphasis has been placed on multivariate time series.

For the second axis, point outlier detection is the most researched problem due to its simplicity in comparison to the rest. Within this category, it is remarkable that some techniques do not consider the temporal information at all. This can lead to not detecting outliers that violate the temporal correlation of the series. Thus, possible future work may include temporal information to techniques that do not consider it (e.g., Cheng et al. [2008, 2009] and Siffer et al. [2017]). Additionally, even if theoretically many techniques can determine if a new data point is an outlier upon arrival, no study has been conducted to analyze whether these methods can really give an immediate response in real time or not. Consequently, an in-depth investigation could be done to analyze the computational cost of outlier detection techniques and to determine whether these methods can be practically used in real-time scenarios. There is also a chance for further incremental algorithm developments to ensure quick responses that adapt to the evolution of the stream.

Subsequence outliers and outlier time series have been handled less frequently. Most of the subsequence outlier detection methods find non-periodic and fixed-length outliers. Indeed, there are no techniques that identify periodic subsequence outliers in multivariate time series. This can be a promising area for further investigation, especially in fields such as cybersecurity to detect periodic network attacks, or in credit-fraud detection to detect periodic thefts. Also, within techniques that detect subsequence outliers, care must be taken with the way in which clustering is performed, because it has been shown that clustering all of the subsequences extracted from a time series by a sliding window produces meaningless results [Keogh and Lin 2005].

Not much work has been carried out on the detection of outlier time series. Other research directions in this line include using different distance measures and creating more effective methods for dealing with time-dependent variables of different lengths.

Within the dissimilarity-based techniques that detect either subsequence or whole time series outliers, the dissimilarity measure that is used influences the results obtained. The Euclidean distance accounts for the majority of all published work mentioned in this review due to its computation efficiency. However, other measures, such as DTW, could lead to an improvement in the detection of outliers, because they include temporal information. An interesting research direction would be to observe how different dissimilarity measures influence the outlier detection algorithms in time series data to see if any of them improve the results of the Euclidean distance. In particular, meta-learning approaches such as that proposed by Mori et al. [2016] could be used as they provide an automatic selection of the most suitable distance measure.

In addition to these types of outliers, there could be other types of unexplored outliers. For example, it may be interesting to detect outliers that propagate over time within different variables of a multivariate time series; that is, an outlier may begin in a unique variable and then propagate through other variables in later time steps. As far as we know, this problem has not been addressed yet in the literature, or at least it has not been done under the name of outlier/anomaly detection.

Finally, with regard to the nature of the detection method, it should be noted that some univariate techniques (e.g., the density-based) can easily be generalized to the multivariate case by considering the distance between vectors instead of points. However, complex correlations between the variables in a multivariate time series are not taken into account. This may lead us to not identify observations that look normal in each variable individually but that violate the correlation structure between variables. In addition, when applying a univariate technique separately to each variable in a multivariate time series, the correlation dependencies are ignored. This can be computationally expensive when the number of variables is large. Hence, an extension of the univariate detection techniques applied to multivariate time series should be studied (e.g., Xu et al. [2017] and Hundman et al. [2018]) so that the correlations between variables representing complex system behaviors are included.

## REFERENCES

- C. Aggarwal. 2016. *Outlier Analysis* (2nd ed.). Springer.
- H. Aguinis, R. K. Gottfredson, and H. Joo. 2013. Best-practice recommendations for defining, identifying, and handling outliers. *Organiz. Res. Methods* 16, 2 (2013), 270–301.
- S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147.
- H. N. Akouemo and R. J. Povinelli. 2014. Time series outlier detection and imputation. In *Proceedings of the IEEE PES General Meeting—Conference Exposition*, 1–5.
- H. N. Akouemo and R. J. Povinelli. 2016. Probabilistic anomaly detection in natural gas time series data. *Int. J. Forecast.* 32, 3 (2016), 948–956.
- H. N. Akouemo and R. J. Povinelli. 2017. Data improving in time series using ARX and ANN models. *IEEE Trans. Power Syst.* 32, 5 (2017), 3352–3359.
- F. Angiulli and F. Fassetti. 2007. Detecting distance-based outliers in streams of data. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. ACM, 811–820.
- F. Angiulli and F. Fassetti. 2010. Distance-based outlier queries in data streams: The novel task and algorithms. *Data Min. Knowl. Discov.* 20, 2 (2010), 290–324.
- R. Baragona and F. Battaglia. 2007. Outliers detection in multivariate time series by independent component analysis. *Neural Comput.* 19, 7 (2007), 1962–1984.
- S. Basu and M. Meckesheimer. 2007. Automatic outlier detection for time series: An application to sensor data. *Knowl. Info. Syst.* 11, 2 (2007), 137–154.
- L. Beggel, B. X. Kausler, M. Schiegg, M. Pfeiffer, and B. Bischl. 2019. Time series anomaly detection based on shapelet learning. *Comput. Stat.* 34, 3 (2019), 945–976.
- S. E. Benkabou, K. Benabdeslem, and B. Canitia. 2018. Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowl. Info. Syst.* 54, 2 (2018), 463–486.
- Y. Bu, T. W. Leung, A. W. C. Fu, E. Keogh, J. Pei, and S. Meshkin. 2007. Wat: Finding top-k discords in time series database. In *Proceedings of the 7th SIAM International Conference on Data Mining*. SIAM, 449–454.
- H. T. Q. Buu and D. T. Anh. 2011. Time series discord discovery based on iSAX symbolic representation. In *Proceedings of the 3rd International Conference on Knowledge and Systems Engineering*. IEEE, 11–18.
- M. Čampulová, J. Michálek, P. Mikuška, and D. Bokál. 2018. Nonparametric algorithm for identification of outliers in environmental data. *J. Chemom.* 32, 5 (2018), 1–17.
- A. Carreño, I. Inza, and J. A. Lozano. 2020. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artific. Intell. Rev.* 53, 5 (2020), 3575–3594.
- D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi. 2019. Online anomaly detection for long-term ECG monitoring using wearable devices. *Pattern Recogn.* 88 (2019), 482–492.
- D. Carrera, B. Rossi, D. Zambon, P. Fragneto, and G. Boracchi. 2016. ECG monitoring in wearable devices by sparse models. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer, 145–160.
- K. M. Carter and W. W. Streilein. 2012. Probabilistic reasoning for streaming anomaly detection. In *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP'12)*, Vol. 1. IEEE, 1–4.
- V. Chandola, A. Banerjee, and V. Kumar. 2007. Outlier Detection: A Survey. Technical Report. Technical Report TR 07-017. Department of Computer Science and Engineering, University of Minnesota.
- V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3 (2009), 1–72.
- V. Chandola, A. Banerjee, and V. Kumar. 2012. Anomaly detection for discrete sequences: A survey. *IEEE Trans. Knowl. Data Eng.* 24, 5 (2012), 823–839.
- P. M. Chau, B. M. Duc, and D. T. Anh. 2018. Discord discovery in streaming time series based on an improved HOT SAX algorithm. In *Proceedings of the 9th International Symposium on Information and Communication Technology*. ACM, 24–30.
- C. Chen and D. J. Cook. 2011. Energy outlier detection in smart environments. In *Proceedings of the Workshops at the 25th AAAI Conference on Artificial Intelligence*. AAAI Press, 9–14.
- C. Chen, D. J. Cook, and A. S. Crandall. 2013. The user side of sustainability: Modeling behavior and energy usage in the home. *Pervas. Mobile Comput.* 9, 1 (2013), 161–175.
- C. Chen and L. M. Liu. 1993. Joint estimation of model parameters and outlier effects in time series. *J. Amer. Statist. Assoc.* 88, 421 (1993), 284–297.
- J. Chen, W. Li, A. Lau, J. Cao, and K. Wang. 2010. Automated load curve data cleansing in power systems. *IEEE Trans. Smart Grid* 1, 2 (2010), 213–221.
- H. Cheng, P. N. Tan, C. Potter, and S. Klooster. 2008. A robust graph-based algorithm for detection and characterization of anomalies in noisy multivariate time series. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*. IEEE, 349–358.

- H. Cheng, P. N. Tan, C. Potter, and S. Klooster. 2009. Detection and characterization of anomalies in multivariate time series. In *Proceedings of the SIAM International Conference on Data Mining*, SIAM, 413–424.
- M. C. Dani, F. X. Jollois, M. Nadif, and C. Freixo. 2015. Adaptive threshold for anomaly detection using time series segmentation. In *Proceedings of the 22nd International Conference on Neural Information Processing (ICONIP'15)*. Springer, Cham, 82–89.
- P. Esling and C. Agon. 2012. Time-series data mining. *ACM Comput. Surv.* 45, 1 (2012), 1–34.
- A. J. Fox. 1972. Outliers in time series. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* 34, 3 (1972), 350–363.
- A. W. C. Fu, O. T. W. Leung, E. Keogh, and J. Lin. 2006. Finding time series discords based on Haar transform. In *Proceedings of the 2nd International Conference on Advanced Data Mining and Applications (ADMA'06)*. Springer, Berlin, 31–41.
- T. C. Fu. 2011. A review on time series data mining. *Eng. Appl. Artif. Intell.* 24, 1 (2011), 164–181.
- P. Galeano, D. Peña, and R. S. Tsay. 2006. Outlier detection in multivariate time series by projection pursuit. *J. Amer. Statist. Assoc.* 101, 474 (2006), 654–669.
- J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.* 46, 4 (2014), 1–35.
- M. Gupta, J. Gao, C. Aggarwal, and J. Han. 2014a. *Outlier Detection for Temporal Data*. Morgan & Claypool Publishers. 1–129 pages.
- M. Gupta, J. Gao, C. Aggarwal, and J. Han. 2014b. Outlier detection for temporal data: A survey. *IEEE Trans. Knowl. Data Eng.* 26, 9 (2014), 2250–2267.
- D. M. Hawkins. 1980. *Identification of Outliers*. Springer Netherlands.
- D. J. Hill and B. S. Minsker. 2010. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environ. Model. Softw.* 25, 9 (2010), 1014–1022.
- J. Hochenbaum, O. S. Vallis, and A. Kejariwal. 2017. Automatic anomaly detection in the cloud via statistical learning. Retrieved from <https://arXiv:1704.07706>.
- V. Hodge and J. Austin. 2004. A survey of outlier detection methodologies. *Artific. Intell. Rev.* 22, 2 (2004), 85–126.
- J. Holešovský, M. Čampulová, and J. Michálek. 2018. Semiparametric outlier detection in nonstationary times series: Case study for atmospheric pollution in Brno, Czech Republic. *Atmosph. Pollut. Res.* 9, 1 (2018), 27–36.
- M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou. 2019. A novel computational approach for discord search with local recurrence rates in multivariate time series. *Info. Sci.* 477 (2019), 220–233.
- K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. 2018. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 387–395.
- R. J. Hyndman, E. Wang, and N. Laptev. 2015. Large-scale unusual time series detection. In *Proceedings of the 15th IEEE International Conference on Data Mining Workshop (ICDMW'15)*. IEEE, 1616–1619.
- V. Ishimtsev, A. Bernstein, E. Burnaev, and I. Nazarov. 2017. Conformal k-NN anomaly detector for univariate data streams. In *Proceedings of the 6th Workshop on Conformal and Probabilistic Prediction and Applications*, Vol. 60. PMLR, 213–227.
- A. Iturria, J. Carrasco, S. Charramendieta, A. Conde, and F. Herrera. 2020. OTSAD: A package for online time-series anomaly detectors. *Neurocomputing* 374 (2020), 49–53.
- H. Izakian and W. Pedrycz. 2013. Anomaly detection in time series data using a fuzzy c-means clustering. In *Proceedings of the Joint IFSA World Congress and NAFIPS Annual Meeting*. IEEE, 1513–1518.
- H.V. Jagadish, N. Koudas, and S. Muthukrishnan. 1999. Mining deviants in a time series database. In *Proceedings of the 25th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers, 102–113.
- M. Jones, D. Nikovski, M. Imamura, and T. Hirata. 2014. Anomaly detection in real-valued multidimensional time series. In *Proceedings of the International Conference on Bigdata/Socialcom/Cybersecurity*. Citeseer.
- M. Jones, D. Nikovski, M. Imamura, and T. Hirata. 2016. Exemplar learning for extremely efficient anomaly detection in real-valued time series. *Data Min. Knowl. Discov.* 30, 6 (2016), 1427–1454.
- E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowl. Info. Syst.* 3, 3 (2001), 263–286.
- E. Keogh and J. Lin. 2005. Clustering of time-series subsequences is meaningless: Implications for previous and future research. *Knowl. Info. Syst.* 8, 2 (2005), 154–177.
- E. Keogh, J. Lin, and A. Fu. 2005. HOT SAX: Efficiently finding the most unusual time series subsequence. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 226–233.
- E. Keogh, J. Lin, S. H. Lee, and H. van Herle. 2007. Finding the most unusual time series subsequence: Algorithms and applications. *Knowl. Info. Syst.* 11, 1 (2007), 1–27.
- E. Keogh, S. Lonardi, and B. Y. C. Chiu. 2002. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*. ACM, 550–556.



- T. Kieu, B. Yang, and C. S. Jensen. 2018. Outlier detection for multidimensional time series using deep neural networks. In *Proceedings of the 19th IEEE International Conference on Mobile Data Management*. IEEE, 125–134.
- N. Kumar, N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana. 2005. Time-series bitmaps: A practical visualization tool for working with large time series databases. In *Proceedings of the 5th SIAM International Conference on Data Mining*. SIAM, 531–535.
- N. Laptev, S. Amizadeh, and I. Flint. 2015. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, 1939–1947.
- J. A. Lara, D. Lizcano, V. Rampérez, and J. Soriano. 2020. A method for outlier detection based on cluster analysis and visual expert criteria. *Expert Syst.* 37, 5 (2020), 1–23.
- G. Li, O. Bräysy, L. Jiang, Z. Wu, and Y. Wang. 2013. Finding time series discord based on bit representation clustering. *Knowl.-Based Syst.* 54 (2013), 243–254.
- X. Li, Z. Li, J. Han, and J. G. Lee. 2009. Temporal outlier detection in vehicle traffic data. In *Proceedings of the IEEE 25th International Conference on Data Engineering*. IEEE, 1319–1322.
- J. Lin, E. Keogh, A. Fu, and H. Van Herle. 2005. Approximations to magic: Finding unusual medical time series. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*. IEEE, 329–334.
- Y. Liu, X. Chen, F. Wang, and J. Yin. 2009. Efficient detection of discords for time series stream. In *Proceedings of the Joint International Conferences on Advances in Data and Web Management*. Springer, Berlin, 629–634.
- M. Longoni, D. Carrera, B. Rossi, P. Fragneto, M. Pession, and G. Boracchi. 2018. A wearable device for online and long-term ECG monitoring. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. International Joint Conferences on Artificial Intelligence, 5838–5840.
- V. Losing, B. Hammer, and H. Wersing. 2018. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing* 275 (2018), 1261–1274.
- H. Lu, Y. Liu, Z. Fei, and C. Guan. 2018. An outlier detection algorithm based on cross-correlation analysis for time series dataset. *IEEE Access* 6 (2018), 53593–53610.
- S. Mehrang, E. Helander, M. Pavel, A. Chieh, and I. Korhonen. 2015. Outlier detection in weight time series of connected scales. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM'15)*. IEEE, 1489–1496.
- H. D. K. Moonesinghe and P. N. Tan. 2006. Outlier detection using random walks. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. IEEE, 532–539.
- U. Mori, A. Mendiburu, and J. A. Lozano. 2016. Similarity measure selection for clustering time series databases. *IEEE Trans. Knowl. Data Eng.* 28, 1 (2016), 181–195.
- M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. 2019. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access* 7 (2019), 1991–2005.
- S. Muthukrishnan, R. Shah, and J. S. Vitter. 2004. Mining deviants in time series data streams. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*. IEEE, 41–50.
- S. Papadimitriou, J. Sun, and C. Faloutsos. 2005. Streaming pattern discovery in multiple time-series. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05)*. ACM, 697–708.
- F. Rasheed and R. Alhajj. 2014. A framework for periodic outlier pattern detection in time-series sequences. *IEEE Trans. Cybernet.* 44, 5 (2014), 569–582.
- F. Rasheed, M. Alshalalfa, and R. Alhajj. 2011. Efficient periodicity mining in time series databases using suffix trees. *IEEE Trans. Knowl. Data Eng.* 23, 1 (2011), 79–94.
- C. A. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das. 2010. Mining time series data. In *Data Mining and Knowledge Discovery Handbook* (2nd ed.), Oded Maimon and Lior Rokach (Eds.). Springer US, 1049–1077.
- U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock. 2009. Finding anomalous periodic time series: An application to catalogs of periodic variable stars. *Mach. Learn.* 74, 3 (2009), 281–313.
- A. Reddy, M. Ordway-West, M. Lee, M. Dugan, J. Whitney, R. Kahana, B. Ford, J. Muedsam, A. Henslee, and M. Rao. 2017. Using Gaussian mixture models to detect outliers in seasonal univariate network traffic. In *Proceedings of the IEEE Security and Privacy Workshops (SPW'17)*. IEEE, 229–234.
- H. Ren, M. Liu, Z. Li, and W. Pedrycz. 2017. A piecewise aggregate pattern representation approach for anomaly detection in time series. *Knowl.-Based Syst.* 135 (2017), 29–39.
- M. Sakurada and T. Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA'14)*. ACM, 4–12.
- H. Sanchez and B. Bustos. 2014. Anomaly detection in streaming time series based on bounding boxes. In *Proceedings of the 7th International Conference on Similarity Search and Applications (SISAP'14)*. Springer, Cham, 201–213.
- P. Senin. 2018. jmotif: Time Series Analysis Toolkit Based on Symbolic Aggregate Discretization, i.e., SAX. Retrieved from <https://cran.r-project.org/package=jmotif>.



- P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein. 2015. Time series anomaly discovery with grammar-based compression. In *Proceedings of 18th International Conference on Extending Database Technology (EDBT'15)*. OpenProceedings.org, 481–492.
- P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein. 2018. GrammarViz 3.0: Interactive discovery of variable-length time series patterns. *ACM Trans. Knowl. Discov. Data* 12, 1 (2018), 1–28.
- M. S. Shahriar, D. Smith, A. Rahman, M. Freeman, J. Hills, R. Rawnsley, D. Henry, and G. Bishop-Hurley. 2016. Detecting heat events in dairy cows using accelerometers and unsupervised learning. *Comput. Electron. Agric.* 128 (2016), 20–26.
- A. Siffer, P. A. Fouque, A. Termier, and C. Largouët. 2017. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*. ACM, 1067–1075.
- J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. De Carvalho, and J. Gama. 2013. Data stream clustering: A survey. *Comput. Surveys* 46, 1 (2013), 1–37.
- S. Song, A. Zhang, J. Wang, and P. S. Yu. 2015. SCREEN: Stream data cleaning under speed constraints. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 827–841.
- Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19)*. 2828–2837.
- R. S. Tsay. 1988. Outliers, level shifts, and variance changes in time series. *J. Forecast.* 7 (1988), 1–20.
- R. S. Tsay, D. Peña, and A. E. Pankratz. 2000. Outliers in multivariate time series. *Biometrika* 87, 4 (2000), 789–804.
- A. Tsybmal. 2004. *The Problem of Concept Drift: Definitions and Related Work*. Technical Report. Department of Computer Science, Trinity College: Dublin.
- X. Wang, J. Lin, N. Patel, and M. Braun. 2018. Exact variable-length anomaly detection algorithm for univariate and multivariate time series. *Data Min. Knowl. Discov.* 32, 6 (2018), 1806–1844.
- L. Wei, N. Kumar, V. Lolla, E. Keogh, S. Lonardi, and C. Ann. 2005. Assumption-free anomaly detection in time series. In *Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM'05)*. Lawrence Berkeley Laboratory, 237–240.
- X. Xu, H. Liu, and M. Yao. 2019. Recent progress of anomaly detection. *Complexity* 2019 (2019), 1–11.
- Z. Xu, K. Kersting, and L. von Ritter. 2016. Adaptive streaming anomaly analysis. In *Proceedings of NIPS Workshop on Artificial Intelligence for Data Science*.
- Z. Xu, K. Kersting, and L. von Ritter. 2017. Stochastic online anomaly analysis for streaming time series. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. International Joint Conferences on Artificial Intelligence, 3189–3195.
- J. Yang, W. Wang, and P. S. Yu. 2001. InfoMiner: Mining surprising periodic patterns. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*. ACM, 395–400.
- J. Yang, W. Wang, and P. S. Yu. 2004. Mining surprising periodic patterns. *Data Min. Knowl. Discov.* 9 (2004), 189–216.
- L. Ye and E. Keogh. 2009. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, 947–956.
- A. Zhang, S. Song, and J. Wang. 2016. Sequential data cleaning: A statistical approach. In *Proceedings of the International Conference on Management of Data (SIGMOD'16)*. ACM, 909–924.
- Y. Zhang, N. A. S. Hamm, N. Meratnia, A. Stein, M. van de Voort, and P. J. M. Havinga. 2012. Statistics-based outlier detection for wireless sensor networks. *Int. J. Geogr. Info. Sci.* 26, 8 (2012), 1373–1392.
- Y. Zhou, R. Arghandeh, and C. J. Spanos. 2016. Online learning of Contextual Hidden Markov Models for temporal-spatial data analysis. In *Proceedings of the IEEE 55th Conference on Decision and Control (CDC)*. IEEE, Las Vegas, NV, USA, 6335–6341.
- Y. Zhou, R. Arghandeh, H. Zou, and C. J. Spanos. 2019. Nonparametric event detection in multiple time series for power distribution networks. *IEEE Trans. Industr. Electron.* 66, 2 (2019), 1619–1628.
- Y. Zhou, R. Qin, H. Xu, S. Sadiq, and Y. Yu. 2018a. A data quality control method for seafloor observatories: The application of observed time series data in the East China Sea. *Sensors* 18, 8 (2018), 2628.
- Y. Zhou, H. Zou, R. Arghandeh, W. Gu, and C. J. Spanos. 2018b. Non-parametric outliers detection in multiple time series a case study: Power grid data analysis. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, 4605–4612.

Received January 2020; revised July 2020; accepted December 2020