

Open Challenges in Time Series Anomaly Detection: An Industry Perspective

Andreas C. Müller
 amueller@microsoft.com
 Microsoft
 USA

Abstract

Current research in time-series anomaly detection is using definitions that miss critical aspects of how anomaly detection is commonly used in practice. We list several areas that are of practical relevance and that we believe are either under-investigated or missing entirely from the current discourse. Based on an investigation of systems deployed in a cloud environment, we motivate the areas of streaming algorithms, human-in-the-loop scenarios, point processes, conditional anomalies and populations analysis of time series. This paper serves as a motivation and call for action, including opportunities for theoretical and applied research, as well as for building new dataset and benchmarks.

CCS Concepts

- Computing methodologies → Online learning settings; Anomaly detection; Machine learning; • Human-centered computing → Human computer interaction (HCI).

Keywords

Time Series, Anomaly Detection, Outlier Detection, Applications

ACM Reference Format:

Andreas C. Müller. 2025. Open Challenges in Time Series Anomaly Detection: An Industry Perspective. In *Proceedings of Preprint*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Recent years have seen a renewed academic interest in time series anomaly detection (TAD), with several new methods being described, as well as broad discourse on metrics, benchmark datasets and categorization of anomaly types [7, 14, 19–21, 24, 27, 28, 37]. In this paper, we argue that while laudable and highly relevant, these efforts miss some fundamental aspects of time series anomaly detection in practical applications. This work explains these missing aspects, and why they are important for practical applications. Our perspective is informed by working with both internal and external customers at a cloud provider. In this paper we will argue for the following two core tenets that we believe should drive future research in time series anomaly detection:

Permission to make digital or hard copies of all or part 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 components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Preprint, Redmond, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
 ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
<https://doi.org/XXXXXXX.XXXXXXX>

ALERTING The main applications of time series anomaly detection are alerting and mitigation, i.e. either letting a human know that an anomalous event happened, or automatically reacting to an anomalous event.

APPLICATION SPECIFIC Whether a certain event is considered an anomaly in practice is not intrinsic to the data, but is application specific, i.e. an event that would be considered anomalous in one application could be considered normal in another.

We will elaborate on these two tenets in Section 3 and Section 4. The first tenet is an empirical observation from applications, which can also be found in the applied TAD literature, which is extremely biased towards alerting scenarios, including Space Flight [10], cloud operations [12, 15, 25, 34], building maintenance [39], smart grids [40], financial fraud [8] and cyber security [33]. In fact, it is hard to find practical applications of anomaly detection that do not require timely alerting. The second tenet is an assumption about the nature of anomalies relevant in practice. It is also motivated by the difficulties in evaluating current algorithms, and the absence of broadly applicable solutions, see the discussion in Section 1.1.1. We discuss alternative viewpoints in Section 6. From these two tenets, we derive the following requirements:

STREAMING Alerting requires streaming algorithms to efficiently update on newly arriving data (a computational optimization). More importantly, it also requires streaming evaluation, i.e. using only previously seen data for scoring a given datapoint, a fundamental requirement in algorithm comparison.

POPULATIONS Alerting is usually done on a population of semi-homogeneous time-series, i.e. monitoring the room temperature in all rooms in a building, or monitoring the traffic or disk usage on the nodes of a cluster, or app usage per user for a mobile app. From these populations, depending on the scenario, different levels of alerting, say on the whole population, on individuals, or on sub-populations could be desired.

CONDITIONAL ANOMALIES Alerting often requires external knowledge to determine whether a particular behavior is expected or anomalous. For example, treating sales as a purely univariate TAD problem, the 4th of July might show zero sales in a store, which would be interpreted as anomalous. Most domain experts would not consider this an anomaly, however, since the behavior could be easily explained away by the fact that it is a holiday, and the store is closed.

HUMAN IN THE LOOP If anomalies are not intrinsic to the data, but application specific, it's essential to have a human

in the loop, and detect relevant anomalies based on human feedback.

EXPLAINABILITY In any alerting system, users are likely to ask two “why” question for any given alert. The first is “Why was an alert raised?”, the second is “What caused the alert?”, corresponding to explainability and root cause analysis (RCA).

In addition to these requirements that follow from our core tenets, we have also found several requirements that are related to the low-level processing of signals. These requirements are critical to applying TAD, but are not usually discussed in the literature.

POINT PROCESSES Data is often presented as point processes (i.e. not observed at regular intervals) while TAD algorithms require timeseries data.

PERIODICITY Most TAD algorithms require a window size, which is often determined automatically from data. The method used for determining the window size are not standardized, often not even discussed, different implementation yield widely varying results, and the impact of these on TAD algorithms is not well-understood.

THRESHOLD SETTING Threshold setting is a critical component of any real-world TAD system, though usually treated as a separate issue. Given the usual unsupervised nature of the problem, even with human intervention, it’s unclear how to set a threshold effectively.

In the following sections, we will argue why these are requirements of real-world deployments of TAD, and why current research does not sufficiently address these requirements. We provide an motivating example based on a real use-case in Section 2. We discuss alerting requirements in Section 3 and propose explicit formulations of the TAD setup that satisfy some of these requirements, in particular cohort-level TAD and Conditional TAD in Section 3.2. We discuss the online setting with human feedback in Section 4 and propose a natural formulation resembling a censored bandit problem. We discuss pre-processing and normalization requirements in Section 5 and end with discussing several counter-arguments to our tenets and requirements in Section 6.

1.1 Related Work

1.1.1 Surveys of Time Series Anomaly Detection, Metrics and Definitions. In recent years, there has been a wealth of papers reviewing, classifying and benchmarking TAD algorithm and datasets, providing a wide array of insights into the characteristics of algorithms and dataset. This wave of reviews was started by the seminal work [37] that proclaimed “Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress” and showed how simple heuristics can solve most popular benchmark datasets used in the literature at that point. Lai et al. [14] provide a qualitative definition of different types of anomalies, and introduce formal characterizations of different types of anomalies in an elegant but limited framework. Given the criticism of existing benchmarks, Schmidl et al. [28] and Paparrizos et al. [24] concurrently proposed more expansive benchmarks and provided an extensive evaluation of existing methods, often finding classical and relatively simple methods to work well. Schmidl et al. [28] also provides a

taxonomy and survey of current methods, and evaluates their performance given several synthetic anomaly scenarios. Sarfraz et al. [27] showed that very simple baselines can not only match, but even outperform complex deep learning solutions, and that the commonly used adjusted F1 metric is flawed. Kim et al. [13] further disseminate this metric and propose alternatives, which is further built upon by Paparrizos et al. [23]. Liu and Paparrizos [20] updated the benchmark provided by Paparrizos et al. [24] and used human annotations and consistency rules to curate a diverse subset of the original benchmark that contains consistent labels. Liu et al. [19] provides another taxonomy of recently proposed algorithms. Mejri et al. [21] provide a meta-survey, comparing recent benchmarking papers, and introduce a novel experimental protocol. Sørøs and Ruocco [30] provide a thorough review of different evaluation metrics and their effects on algorithm ranking.

We want to highlight some common themes in these works; first, there is no accepted definition of anomaly, though several definitions have been proposed. However, most definitions are rather qualitative and not formally specified, with the exception of Lai et al. [14]. At the same time, datasets have been criticized either for being overly simple, unrealistic, or inconsistently annotated. We posit that these issues are interrelated, as the vague definition of the problem makes it difficult to define what an appropriate benchmark should look like. Several synthetic benchmarks have been proposed [14, 28] but without consistent definitions, it’s unclear whether the assumptions made there are realistic for real-world scenarios. A possible explanation for the continued disagreement about evaluation and datasets could be our tenet **APPLICATION SPECIFIC**, which states that without additional information or assumptions, the problem is ill-specified and potentially unsolvable.

We also want to point out that despite this flood of benchmarks and surveys, drawing conclusions is still extremely difficult [30]; two of the most comprehensive benchmarks, Schmidl et al. [28] and Liu and Paparrizos [20] come to very different conclusions: Schmidl et al. [28] finds LSTM-AD to be one of the highest ranked algorithms, while Liu and Paparrizos [20] finds it to be ranked in the middle, outperformed by simple methods like k -means (and Paparrizos et al. [24] ranks it near the bottom), while Liu and Paparrizos [20] showed good success with MCD [26], which was excluded from the analysis in Schmidl et al. [28] for common failure.

1.1.2 Streaming. Despite the extreme prevalence of streaming applications of TAD, we are aware of only two recent works addressing the streaming setting, Ren et al. [25], who propose a simple score based on spectral residuals, which are then optionally re-calibrated using a CNN trained on synthetic data, and Boniol et al. [2], who propose a clustering based algorithm using shape-based distances. Schmidl et al. [28] discuss the ability to operate on streams in their taxonomy, but do not evaluate the streaming setting. Unfortunately, Ren et al. [25] and Boniol et al. [2] only provide limited benchmarks, both in terms of datasets considered and in terms of baseline algorithms.

1.1.3 Human In The Loop. Somewhat surprisingly, there is little work on using human annotations in TAD in the academic literature. Liu et al. [17] describe a system that is used in a production setting and uses human annotations. However, the system is purely supervised, and requires a significant amount of human annotation.

Evaluation is limited to the KPI dataset; however, the work includes many aspects of TAD that are extremely relevant in practical applications, but have been neglected by the academic community. A more recent approach focusing on the user interface, is described in Deng et al. [3].

1.1.4 Root Cause Analysis. Traditionally, root cause analysis (RCA), i.e. finding the cause of an anomaly, has been treated as a separate problem from anomaly detection [29]. However, Yang et al. [38] proposed an integrated causal framework for TAD and RCA, which allows for addressing CONDITIONAL ANOMALIES. This work is likely motivated by real applications, given the industry affiliation; however it has received little attention. While there is a large array of work in the operations and BI community, as reviewed by Soldani and Brogi [29], the area seems not well studied within the ML community.

1.1.5 Signal processing for TAD. There are two critical signal processing components involved in most TAD applications: resampling or aggregating to a time series, and detecting periodicity lengths. While resampling signals is a well-studied problem in signal processing, the usual goal is to remove artifacts and obtain smooth signals [22] which at odds with the goal of anomaly detection, that usually seeks extreme and unusual values. To the best of our knowledge, this interaction has not been studied. Finding the dominant periodicity of a signal has been studied in the data mining community [35] and the database community [4, 36]. However, despite many TAD algorithms requiring a window length, the impact of this choice has not been systematically studied in any of the surveys on TAD 1.1.1, and this length has been either manually specified, or set using simple heuristics. Recently, [6] evaluated several methods for window size selection; however, the benchmark considers only a small subset of the algorithms considered in the other review articles, and uses the discredited adjusted F1-score.

1.2 Definition and Notation

The notion of anomaly in a time series is difficult to formalize, as discussed in Section 1.1.1. Without a clear mathematical characterization of what an anomaly is, however, formulating different problem settings and requirements is difficult, if not impossible. We attempt to introduce a relatively general formal notation here to have a more precise language to describe problem settings, not for the purpose of a theoretical analysis of the assumptions, nor as a practical tool. We assume that any time-series under consideration $\mathbf{x} = (x_t)_{t=1,\dots,n}$ is drawn from a distribution

$$\bar{p}(x_t | \mathbf{x}_{<t}) = (1 - \epsilon)p(x_t | \mathbf{x}_{<t}) + \epsilon q(x_t | \mathbf{x}_{<t})$$

where here and in the following,

$$\mathbf{x}_{<t} = (x_1, \dots, x_{t-1}).$$

Here ϵ is the probability of any point x_t being an anomaly, q is an arbitrary distribution, and p conforms to a Markov process of unknown order $k \ll n$, i.e.

$$p(x_t | \mathbf{x}_{<t}) = p(x_t | x_{t-k}, \dots, x_{t-1}),$$

The intuition is that \bar{p} is a mixture of a regular distribution p and a noise distribution q . We call x_t an anomaly when $p(x_t | \mathbf{x}_{<t}) < \eta$ for some threshold $\eta > 0$, i.e. if x_t is unlikely under the noise-free

distribution. The definition of \bar{p} only captures *point anomalies* as described in Lai et al. [14] as each point is independently drawn from regular or noise distribution; modeling pattern anomalies is possible by making ϵ a function of $\mathbf{x}_{<t}$; however, for the arguments in this paper, limiting ourselves to point anomalies is sufficient, even though pattern anomalies are also extremely relevant in practice.

2 An Illustrative Example: Temperature Sensors

We start our discussion of our requirements with a representative example of a slightly simplified real-world use-case in which TAD was applied to illustrate these requirement in the context of an application.

Consider a nation-wide retail chain selling fresh groceries. This requires a network of warehouses with refrigerators and cooling trucks. Each cooling unit has a thermometer and is monitored for being in the allowable range given its contents. This means, depending on the scale of the company, hundreds, thousands or tens of thousands of time series that are of the same nature. If a certain unit is above the allowable temperature for a certain time, the food can no longer be sold. Any abnormality in the data that could indicate an imminent incident is therefore business critical. Clearly, STREAMING is required, as notification of anomalies should be as fast as possible, and future data is not available. Also, new data in the form of current temperature measurements needs to be constantly incorporated. When using the data for alerting, there are several possible consumers of the alert. Most importantly, there might be an operator or team that can investigate failures. Constantly monitoring tens of thousands of series is impractical, and alerting on individual series might be overwhelming. Often multiple anomalies are correlated, for example if the cooling units are co-located. If a truck with several units crashes, several might open and change their temperature, or they could short-circuit and no longer send signals. Ideally, the system would alert the operator about a subpopulation of anomalies associated with this truck. Depending on the scenario, this could either fall in POPULATIONS, if the assignment of units to locations is fixed (i.e. a unit is always installed in the same truck, warehouse, or store, so the assignment independent of time), or into CONDITIONAL ANOMALIES if the units travel between trucks, warehouses and stores, leading to a time series of locations associated with each unit. In either case, alerting should likely be at the most general level possible, say, alerting about a regional power-outage as something happening in a region, not on hundreds of individual units.

While the main consumer of anomaly reports on the cooling units might be an operations team, another usage is device maintenance. For example, a temperature sensor that shows extreme spikes for short periods of time is likely malfunctioning, and needs to be replaced. This is of no consequence to the logistics operations, since short high temperature spikes are physically implausible, and therefore measurement errors when it comes to the actual state of the cooling unit. However, from the perspective of a maintenance team, this malfunctioning sensor clearly needs attention and might need replacement. Here, the notion of anomaly does not only depend on the signal, but also on the downstream usage, exhibiting APPLICATION SPECIFIC. A real-live example of this is shown in Figure 1 using data of ultra-low temperature freezers, commonly

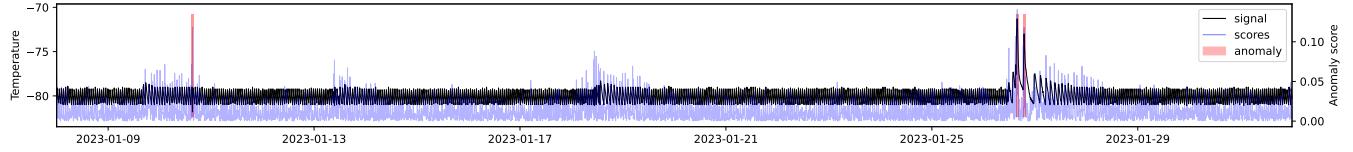


Figure 1: Temperature data for an ultra-low temperature freezer (Unit Haier 810545 from Huang et al. [9]. Anomaly scores are computed with spectral residual [25] and thresholded at the 99.9th percentile. The two detected anomalies have very different patterns and real-life consequences.

used for pharmaceutical applications [9]. The first anomaly marked in the figure shows an extreme outlier: a dip below the standard temperature of -81°C . The second anomaly consists of two large spikes of temperatures of between -72°C and -74°C . From a statistical point of view, both are significant anomalies, even using very simple measures. The low temperature is caused by a temporary power outage, the high temperature by the door to the unit being left open. From an operational perspective, only the second event is likely to be relevant.

3 Alerting Requirements

3.1 Streaming

As discussed in Section 1.1.2, there have been several efforts in streaming anomaly detection [2, 25]. However, even though some work discuss the streaming capabilities of certain algorithms [28], there exist few actual implementations of streaming algorithms. More importantly, review work discussed in Section 1.1.1 use the batch setting for evaluation, even though many algorithms under consideration, such as k -Means and CNNs, two basic and well studied ML algorithms that perform well in some TAD scenarios [20], are algorithmically capable of addressing the streaming setting. As the evaluation strategy appropriate for streaming is different from the strategies employed in all the papers discussed in 1.1.1, none of these actually provide guidance for practitioners. The two works that explicitly discuss this setting deploy rather niche algorithms, and don't evaluate on the wide array of current benchmark datasets, and don't consider common baselines.

To be more explicit, the evaluation in the streaming scenario for an algorithm \mathcal{A} (producing a binary sequence of labels a) on a time series x with ground truth labels l and a loss ℓ , as discussed in Boniol et al. [2], Ren et al. [25]¹ can be formulated as

$$R_o(\mathcal{A}, x, l) = \sum_t \ell(l_t, \mathcal{A}(x_{\leq t})_t), \quad (1)$$

i.e. computing the predicted anomaly label for each point t using the algorithm on all data up to the point t . This is in contrast to the batch evaluation used in all of the benchmarks described in Section 1.1.1, which use batch evaluation:

$$R_b(\mathcal{A}, x, l) = \sum_t \ell(l_t, \mathcal{A}(x)_t), \quad (2)$$

which allows fitting the algorithm once on the whole data set, and producing one set of predictions.

¹Boniol et al. [2] use minibatches, not individual points, and Ren et al. [25] use an adjustment with a pre-specified delay tolerance.

3.1.1 Detection Delay. While there has been much debate about metrics for time-series anomaly detection, with Paparrizos et al. [23] providing a comprehensive review as well as novel metrics, many of the recent proposals, such as Range-based Precision and Recall [32] and VUS [23] take detection delay and detection window as a property to be invariant to. For the streaming setting, in particular in time critical applications, the detection delay is actually an important aspect of algorithm performance, as described in Lavin and Ahmad [16]. This differentiation has been noted in Sørø and Ruocco [30], but has been mostly ignored by other recent benchmarks.

3.2 Side Information

The pattern described in Section 2, where there is a population of time series of interest, and additional side-information attached to each time series is extremely common in practice. There are two cases we want to distinguish: time-independent and time-varying side information.

3.2.1 Time-independent Side Information: Subpopulation Mining.

Time-independent side-information is often high-dimensional or categorical; for example, monitoring app-usage, each individual user might be associated with a country, a network provider, an operating system and a manufacturer. It could happen that suddenly traffic is altered because a patch rolled out by a phone manufacturer interacts badly with the routing policies of a national network provider². As a streaming movie provider, you might find that suddenly interest in a certain actor or director spikes or decreases based on location and demographic factors, maybe because a news outlet has featured a story. In the cooling scenario described in 2, we might have the manufacturer, years since installation, and state and warehouse ID as time-independent characteristics.

Formally, given a family of d time series of length n , $x_{i,t}$, $t = 1, \dots, n$, $i = 1, \dots, d$, each associated with some time-independent features θ_i , we want to find a rule or classifier f_t , such that $x_{i,t}$ is anomalous iff $f_t(\theta_i) > 0$. Note that we want f_t to be simple as a function of θ , often just a conjunction of univariate terms, i.e. “time series i belongs to device A in region B”, while the rules f_t could vary arbitrarily over time t . A widely used approach in practice is to apply an anomaly detection algorithm \mathcal{A} to obtain anomaly labels $a_{i,t}$ and then run a generic rule-mining approach \mathcal{F} to find $f_t = \mathcal{F}((a_i, \theta_i)_{i=1, \dots, n})$ for each t independently. However, treating the anomaly detection problem jointly over instances, taking

²This is a sketch of a real example, however similar incidents are common in any large-scale IT infrastructure and often have dedicated teams to investigate

sub-populations into account, could inform the difficult univariate unsupervised TAD problem.

Modeling each individual time series could lead to problems, though; very fine grained modeling of a process can lead to very sparse signals that might need to be aggregated, see Section 5.2, and to potentially very high-dimensional rule-mining spaces.

3.2.2 Time-dependent Side Information: Conditional Anomalies.

The other form of side information is including time-varying anomaly detection, which is closely related, but separate, from multivariate anomaly detection. In any application, no time-series stands alone, and sales, traffic or electricity usage depend on external factors like day of the week, holidays, temperature, interest rates etc. Given multiple time-series $\mathbf{x}, \mathbf{y}, \mathbf{z}$, the question most commonly asked in the TAD literature is:

$$\min_t p(x_t, y_t, z_t | \mathbf{x}_{<t}, \mathbf{y}_{<t}, \mathbf{z}_{<t}) \quad (3)$$

that is to find anomalies in the joint distribution of all variables at time t , given previous values, i.e. multivariate anomaly detection. However the most practically relevant question is usually:

$$\min_t p(x_t | \mathbf{x}_{<t}, \mathbf{y}_{<t+1}, \mathbf{z}_{<t+1}) \quad (4)$$

i.e. find anomalies in x_t given the values of the other series at that time and all previous timesteps. We call these *conditional* anomalies.

To illustrate the difference between the notions, consider the classic mock example of the interaction of temperature x_t and ice cream sales y_t . Let's assume they are highly related, and that heat-waves (which extremely high ice-cream sales) are rare. Under these assumptions, a heat-wave would be an anomaly w.r.t (3) (as the likelihood of a heatwave is low), and not an anomaly w.r.t. (4) as the relationship between ice-cream sales and temperature is maintained. Now consider an event in which a very high ice-cream sale happens for another reason (a circus came to town, but we don't have corresponding data). This scenario is an anomaly according to both (3) and (4). Both of these formulations could be relevant, depending on the practical setting, while the literature only discusses (3). From our experience, in most scenarios, the two quantities of interest are anomalies in $p(x_t)$ and $p(x_t | \mathbf{x}_{<t}, \mathbf{y}_{<t+1}, \mathbf{z}_{<t+1})$, which detect univariate anomalies, and univariate anomalies that are not explained by other factors, respectively. As another illustrative example, consider monitoring flu cases as x_t , which are highly correlated with temperature y_t . We might be interested in whether there are abnormally many flu cases, i.e. low $p(x_t | \mathbf{x}_{<t})$, or we might be interested whether there are more flu cases than explained by cold temperatures, i.e. low $p(x_t | \mathbf{x}_{<t}, \mathbf{y}_{<t+1})$. But we are likely not interested in a sudden heatwave or arctic vortexes, i.e. low $p(y_t | \mathbf{y}_{<t})$. While there are many ways to address these problem of finding anomalies using (4), say using residuals of a regression model or a quantile regression model, we advocate for a broader study of this area. The challenge here is both in the availability of datasets as well as methods; however, it might be possible to reuse data from existing multivariate benchmarks and relabel them to reflect this different setting. We have not found a reference to the definition in Equation (4) in the literature, which is somewhat surprising, as this seems to be the most commonly encountered scenario in practice, together with the time-independent information considered in Section 3.2.1.

3.3 Explainability and Root-Cause Analysis

There are two layers of explainability that are commonly required in alerting. The first is "Why was an alert triggered?" and the second is "What is the cause of the alert?". These two questions require different tools and different levels of analysis. The first question can be posed and possibly answered even in the simplest case of a single univariate time series; it is a question of interpreting the anomaly-score provided by the algorithm. Essentially, if the score can be explained to the user, the alert is explainable. A simple threshold, say "two standard deviations from the mean" is easy to explain and can easily be depicted in a chart. Using an explainable forecasting method as the basis of anomaly detection is also relatively straight-forward: using exponential smoothing or possibly a simple autoregressive model, we can show the value that was forecast, explain why this forecast was made, and show that the difference to the forecast crossed a threshold. Some discord-based methods can also be explained: "The closest window we saw so far is this, and the distance crosses a threshold." Unfortunately none of the works described in Section 1.1.1 consider this important axis of interpretability. Interpretability might not be absolute, and measuring the interpretability of methods would be a worthwhile survey; black-box forecasting methods like deep learning methods might or might not be considered explainable, depending on whether the forecast makes intuitive sense to users. A k-Means based approach might possibly be explainable, depending on the scenario. Anomaly scores directly produced by methods such as IForest [18] or Spectral Residual [25] offer limited explainability.

The second level of interpretability is usually related to external factors, ideally with a causal relationship, and can not usually be answered based on a single univariate series. Both time-independent and time-dependent side information, as described in Section 3.2 could be considered to answer such a "why" question. Given the broad scope of this question, we will not discuss it in detail here, but refer the interested reader to [38] for one possible formulation of the problem. Solving this why question in a general sense could be seen as the holy grail of alerting; however, it is likely hard to solve robustly and in general, and we expect many of the more fundamental problems described in this paper are more likely to yield direct applications.

4 Contextualizing and Human in the Loop

As described in Section 2, there might be different kinds of anomalies present in the data, corresponding to different causes or physical effects, that might be relevant to different usage scenarios. More generally, there could be an arbitrary number of ways that a series could deviate from their usual pattern, only a sub-set of which might be relevant. In many applications, suppressing spurious alerts is critical for user trust and satisfaction, and efficient automation. Imagine a system in which false alerts about irrelevant behavior are sent to the fridge operators, who are sent to rescue food that is perfectly safe. Any alerting feature that has high false-positive rate is likely to be ignored soon. This could be addressed from a machine learning perspective by using a supervised system of online or active learning on top of unsupervised anomaly detection to decouple these two tasks. However, given the difficulty of

unsupervised AD, it is likely that an integration of the two learning systems can yield substantial benefits. Liu and Paparrizos [20] and Sørboe and Ruocco [30] argue that inconsistent annotation in existing benchmarks creates a problem for evaluating algorithms; and while several benchmarks might have human error, another explanation is that not only that side-information required to determine anomalies is not included in the benchmark as time-varying features, but that side-information is present as context specific properties of the signal, only present in the operators mental model. While other mechanisms might be possible, the most obvious way to solve this problem is by using human annotation in the TAD process.

This problem could be cast as a classifying each point in a time series; however, given the extremely imbalanced labeling and sparse annotations, we argue that modeling this class of problems as its own task is more appropriate. Unfortunately, there are few datasets available for this task. The most highly curated benchmark, introduced in Wu and Keogh [37], only contains a single anomaly per time series, preventing any supervised or weakly supervised learning. While modeling the problem of including human annotations using a batch setting with training and test set is possible, and was investigated in Liu et al. [17] we think that practical applications most closely resemble the online learning or active learning settings. One of the differences compared to standard online or active learning is that in the TAD steaming setting, the feedback is usually one-sided, i.e. humans would annotate false positives, but not provide labels for false negatives, leading to partial feedback resembling a Bandit setting. Given that alerting usually includes a human recipient that will investigate each alert, we focus on the online setting. The online setting can be formalized as finding an algorithm \mathcal{A} that, given a sequence x and a set of labels $L_{<t}$ produces a binary outcome $a_t = \mathcal{A}(x_{<t}, L_{<t}) \in \{0, 1\}$ predicting whether x_t is anomalous or not³, with

$$L_{<t} = \{(i, l_i) \mid \mathcal{A}(x_{<i+1}, L_{<i}) = 1, i \in \{1, \dots, t-1\}\}.$$

In other words, the algorithm \mathcal{A} has as input the time series x up to time t , together with the true labels l_t for all time points for which $\mathcal{A}(x_t) = 1$, i.e. that were flagged as anomalous before. Then the regret to be minimized is

$$R_{\text{hil}}(\mathcal{A}, x, l) = \sum_{i=1}^n \lambda_{\text{fn}} \mathbf{1}_{a_t=0 \wedge l_t=1} + \lambda_{\text{fp}} \mathbf{1}_{a_t=1 \wedge l_t=0} \quad (5)$$

where λ_{fn} and λ_{fp} are application-specific costs for false negatives and false positives. A related problem of censored feedback in multi-armed bandits was studied in Abernethy et al. [1], but we are not aware of a formulation with deterministic censorship based on the selected arm. Alternatively, this could be seen as a reinforcement learning setting in an augmented state space in which the state is the pair $(x_{<t+1}, L_{<t})$.

While the above seems the most straight-forward formulation of univariate anomaly detection as it is encountered in practice, we are not aware of prior work using this formulation, and invite the research community to consider this approach.

³While thresholding continuous anomaly scores is an important topic, see 5.3, we assume binary labels here for simplicity.

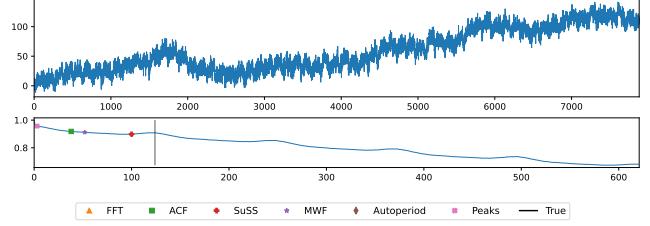


Figure 2: A synthetic time series (top), on which the periodicity can easily be read from the autocorrelation function (bottom), but on which current methods fail, see markers on the autocorrelation function.

5 Preprocessing and Normalization

5.1 Periodicity Detection and Window Size

Algorithms for periodicity detection, in particular for multiple periodicities, still fail on cases that can be solved by manual inspection, leaving much room for improvement, see Figure 2 for an example comparing the methods investigated in Ermshaus et al. [6] on a simple synthetic benchmark; see Appendix A for details. On the other hand, it is not entirely clear whether the periodicity of a signal is the right choice for the window size of TAD algorithms; while this choice is intuitive, benchmarks have shown fixed values of 4 for PCA window size [28] and 125 (used for many sequences by default in Liu and Paparrizos [20], Paparrizos et al. [24]) to be very effective.

5.2 Point Processes and Resampling

In many IOT and other sensor scenarios, data does not form a time series, but instead is more naturally represented as a point process, for example measuring a sale or a person walking through a turnstile, or a visit to a website. Essentially all algorithms studied in the literature require a regular time series, however, and so resampling is required. Unfortunately, all current AD benchmarks contain regular time series, assuming this preprocessing has already been done. However, it's not obvious how to resample a signal in a way that does not smooth over existing anomalies or introduce new anomalies through sampling artifacts. An example is shown in Figure 3 which shows citibike rental from fall 2015 for two different stations⁴. The data is stored as eventstream; however the number of events is very different for very busy stations (top) and slow stations (bottom). The figure shows three different resample rates: days, hours, and 10 minutes. For the busy station, all three give relatively clear signals, while for the slow station, only the daily resampling seems appropriate. However, using the daily rate for both stations would have a clear downside; on the top, an outlier is marked in red, showing two days with abnormal traffic patterns. Throughout the year, this station is most busy on workday evenings, though on these two days, the station is busy on a weekend morning. This is a strong anomaly on the hourly data, but can no longer be detected if aggregated to the daily level. We are not aware of the TAD literature discussing the issue of sampling rate determination at all, despite it being prominent in applications. This issue is partially tied to

⁴Available at <https://citibikenyc.com/system-data>

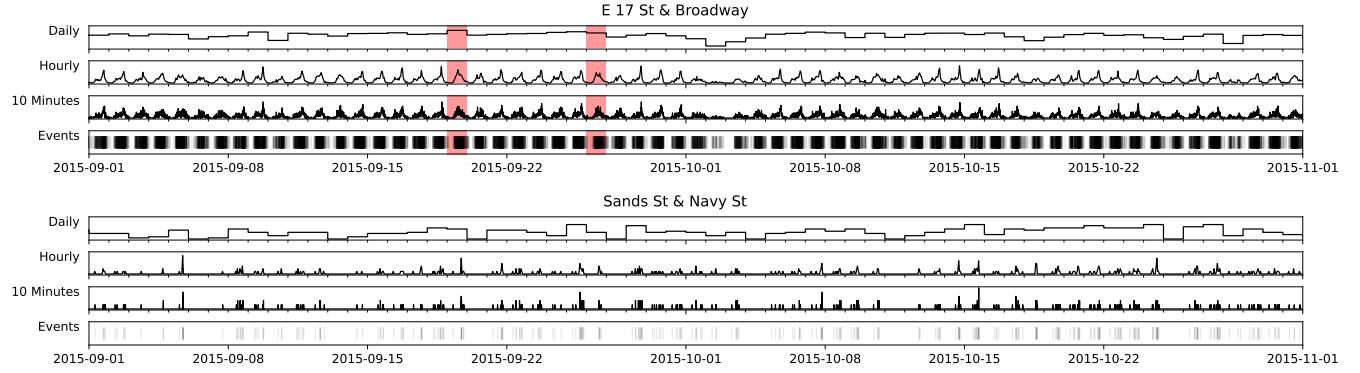


Figure 3: New York City citibike rental data from fall of 2015 for two different stations. The data is present as a point process and needs to be resampled to a time series for anomaly detection. Different downsampling rates might be appropriate for different stations, and it's non-obvious how to determine them automatically.

the population issue discussed in Section 3.2.1; looking at the sales for an individual product in a particular store might create an extremely sparse point process that could be hard to resample, while aggregating sales either across stores or products, or even both, would result in a much denser signal, that might be easier to process. Even among different products, sales might differ by orders of magnitude, requiring different resampling strategies depending on the popularity of a product, or aggregating some products into sub-categories to allow effective modeling. Many real-world processes follow long-tail distributions, such as the popularity of products, songs, movies and websites, which potentially requires different strategies for members of the same population. Another hurdle that might seem trivial but can lead to complications in practice is that the aggregation required on a point process is not obvious from the signal. A temperature measurement reading of 18° C at 10:10h and 19° C at 10:20h resampled to hourly data should lead to 18.5° C at 10:00h, but a \$18 sale at 10:10h and a \$19 sale at 10:20h means a sale of \$37 if resampled to the hour.

5.3 Threshold Setting

In an alerting setting, the choice of threshold for converting an anomaly score into an anomaly prediction is critical, since whether or not to send an alert is a binary decision. In a streaming setting, defining a single threshold might be difficult, and full information of the whole signal is not available. In contrast to these limitations, evaluation in benchmarks often uses an ideal threshold that not only requires knowledge of the full time series, but also of the ground truth [30]. In the streaming setting, Hundman et al. [10] and Ren et al. [25] both define ad-hoc thresholding strategies, but unfortunately neither has been further investigate in the literature.

6 Alternative Views

In this section we inspect possible counter-arguments and alternative points of view on our assumptions.

6.1 Streaming Algorithms are not Necessary

There is a potential caveat to the importance of streaming for TAD. While alerting is usually done on a live data stream, the frequency of

the data is relatively low, often daily or hourly, and rarely less than several minutes. Given these time frames it can be feasible to refit models from scratch at each time step. Depending on the scale of the application, this might or might not lead to infeasible computational costs (say refitting models for 1 million time series with three years of hourly data vs refitting a single model with several weeks of hourly data). However, the coarser the resolution, the more important it is to have small alerting delays; if one algorithm alerts on the day of an anomaly, and another requires three more days of data to fill a window, this difference can be extremely consequential for applications.

6.2 Supervision is not Required

While we propose the viewpoint that a purely unsupervised approach to anomaly detection, in particular in the highly popular alerting setting, is sub-optimal, that does not imply it is entirely impractical. Sylligardos et al. [31] show that it is possible to select a TAD algorithm for a particular series without any additional annotation; however, the results are not outperforming a naive ensemble of all algorithms in an out-of-distribution setting. Several surveys have found that different TAD algorithms are best suited for different datasets [20, 28]. The approach of Sylligardos et al. [31] can only be effective if the information about which algorithm is best suited to a time series is intrinsic to the signal encoded in the time series itself. Only then can a priori decisions about the algorithm choice be made. Our second tenet contradicts this assumption. Whether our requirement holds true on existing academic benchmarks is yet to be determined; the difference between in-distribution and out-of-distribution performance observed in Sylligardos et al. [31] might be a clue that the nature of the annotation within a dataset, and not only the signal characteristics, are required to choose an algorithm. On the other hand, better algorithms or better features could yield better algorithm selection, which would lead to the conclusion that most of the information required to determine anomalies is indeed contained in the signal itself. We are convinced that determining the application-relevant anomalies just from the signal is impossible; however, assuming a reasonable superset can be identified, this might be enough for some applications. If an anomaly occurs every

100,000 observations, but only half of those are relevant for a specific application, this would still reduce manual labor by orders of magnitude, even if precision could not exceed a certain level (0.5 in this example). Rigorously proving our tenet might be difficult; however, given appropriate datasets, it seems worthwhile to investigate how much we can gain by incorporating human knowledge.

6.3 Anomaly Semantics Could be Made Explicit

An alternative approach to manual annotation with a human in the loop as described in Section 4 would be to include expert knowledge without annotating the data, for example via hyper-parameter or algorithm choice. Schmidl et al. [28] show that properties of the anomaly type determine the accuracy of different algorithms, at least to a certain degree, and it is possible that with a better understanding, we can make qualitative recommendations to users based on their understanding of the data. Executing this strategy, however, requires communicating the relevant categories of anomalies to the user to make the correct determination; whether it is possible is an open question.

6.4 Metrics and Annotations are to Blame

Another possible explanation of the diversity of results, and the difficulties in ranking algorithms and providing consistent results is given by Sørbo and Ruocco [30]; they conclude “many metrics are not appropriate for certain kinds of labelling strategies”, which implies that labeling strategy and metric need to be matched. Given the many different labeling strategies employed for different datasets [20], and the patterns rewarded or penalized by different metrics, this rings true. However, this stance distributes the burden of providing accurate results between annotations, metrics and algorithms, making it extremely difficult for a practitioner to know what combination is most appropriate for their specific application. Liu and Paparrizos [20] made an attempt in unifying annotations; however, our own preliminary experiments have shown rankings to still be very metric dependent.

7 Conclusions

This paper discussed two core tenets, ALERTING and APPLICATION SPECIFIC, that we believe are the foundation of applying time series anomaly detection in practice. These lead us to a wide variety of problems that appear in practice that we believe are understudied in the ML community. While this work contains a long list of challenges, these can be summarized in three different categories: defining problems, solving problems and integrating solutions. While we attempted to define some of the problems, there is much work to be done to crisply define anomalies, RCA, or the different TAD settings we described. Based on these, there are clear theoretical and practical challenges to address these highly practically relevant but understudied settings. Finally, many aspects of the problem have been treated separately or ad-hoc, such as human feedback, signal processing and explanations, even though they are deeply interconnected, and more integrated approaches could improve the overall performance of the systems. We hope that this paper will motivate others to investigate these problem settings.

References

- [1] Jacob D Abernethy, Kareem Amin, and Ruihao Zhu. 2016. Threshold bandits, with and without censored feedback. *Advances In Neural Information Processing Systems* 29 (2016).
- [2] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND: streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1717–1729.
- [3] Ziquan Deng, Xiwei Xuan, Kwan-Liu Ma, and Zhaodan Kong. 2024. A Reliable Framework for Human-in-the-Loop Anomaly Detection in Time Series. *arXiv preprint arXiv:2405.03234* (2024).
- [4] Mohamed G Elfeky, Walid G Aref, and Ahmed K Elmagarmid. 2005. Periodicity detection in time series databases. *IEEE Transactions on Knowledge and Data Engineering* 17, 7 (2005), 875–887.
- [5] Arik Ermshaus, Patrick Schäfer, and Ulf Leser. 2023. ClaSP: parameter-free time series segmentation. *Data Mining and Knowledge Discovery* 37, 3 (2023), 1262–1300.
- [6] Arik Ermshaus, Patrick Schäfer, and Ulf Leser. 2023. Window size selection in unsupervised time series analytics: A review and benchmark. In *International Workshop on Advanced Analytics and Learning on Temporal Data*. Springer, 83–101.
- [7] Onat Gungor, Amanda Rios, Priyanka Mudgal, Nilesh Ahuja, and Tajana Rosing. 2025. A Robust Framework for Evaluation of Unsupervised Time-Series Anomaly Detection. In *International Conference on Pattern Recognition*. Springer, 48–64.
- [8] Waleed Hilal, S Andrew Gadsden, and John Yawney. 2022. Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications* 193 (2022), 116429.
- [9] Tao Huang, Silas Nøstvik, Peder Bacher, Jonas Kjær Jensen, Wiebke Brix Markussen, and Jan Kloppenborg Møller. 2023. Labelled dataset for Ultra-Low Temperature Freezer to aid dynamic modelling & fault detection and diagnostics. *Scientific Data* 10, 1 (2023), 888.
- [10] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using ltsms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 387–395.
- [11] Shima Imani and Eamonn Keogh. 2021. Multi-window-finder: domain agnostic window size for time series data. *Proceedings of the MileTS* 21 (2021).
- [12] Nicholas A James, Arun Kejriwal, and David S Matteson. 2016. Leveraging cloud data to mitigate user experience from ‘breaking bad’. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 3499–3508.
- [13] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. 2022. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, 7194–7201.
- [14] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting time series outlier detection: Definitions and benchmarks. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*.
- [15] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. 2015. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1939–1947.
- [16] Alexander Lavin and Subtai Ahmad. 2015. Evaluating real-time anomaly detection algorithms—the Numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, 38–44.
- [17] Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, and Mei Feng. 2015. Opprentice: Towards practical and automatic anomaly detection through machine learning. In *Proceedings of the 2015 internet measurement conference*, 211–224.
- [18] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. IEEE, 413–422.
- [19] Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. 2024. Time-Series Anomaly Detection: Overview and New Trends. *Proceedings of the VLDB Endowment (PVLDB)* 17, 12 (2024), 4229–4232.
- [20] Qinghua Liu and John Paparrizos. 2024. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [21] Nesryne Mejri, Laura Lopez-Fuentes, Kankana Roy, Pavel Chernakov, Enjie Ghorbel, and Djamilia Aouada. 2024. Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods. *Expert Systems with Applications* (2024), 124922.
- [22] Alan V Oppenheim. 1999. *Discrete-time signal processing*. Pearson Education India.
- [23] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. 2022. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2774–2787.
- [24] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. 2022. TSB-UAD: an end-to-end benchmark suite for univariate

- time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1697–1711.
- [25] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 3009–3017.
- [26] Peter J Rousseeuw and Katrien Van Driesssen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 3 (1999), 212–223.
- [27] M Saquib Sarfraz, Mei-Yen Chen, Lukas Layer, Kunyu Peng, and Marios Koulikakis. 2024. Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?. In *Forty-first International Conference on Machine Learning*.
- [28] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment* 15, 9 (2022), 1779–1797.
- [29] Jacopo Soldani and Antonio Brogi. 2022. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–39.
- [30] Sondre Sørøe and Massimiliano Ruocco. 2024. Navigating the metric maze: A taxonomy of evaluation metrics for anomaly detection in time series. *Data Mining and Knowledge Discovery* 38, 3 (2024), 1027–1068.
- [31] Emmanouil Sylligardos, Paul Boniol, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2023. Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3418–3432.
- [32] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. 2018. Precision and Recall for Time Series. 31 (2018). https://proceedings.neurips.cc/paper_files/paper/2018/file/8f468c873a32bb0619eae2050ba45d1-Paper.pdf
- [33] Chee-Wooi Ten, Junho Hong, and Chen-Ching Liu. 2011. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid* 2, 4 (2011), 865–873.
- [34] Owen Vallis, Jordan Hochenbaum, and Arun Kejariwal. 2014. A novel technique for {Long-Term} anomaly detection in the cloud. In *6th USENIX workshop on hot topics in cloud computing (HotCloud 14)*.
- [35] Michail Vlachos, Philip Yu, and Vittorio Castelli. 2005. On periodicity detection and structural periodic similarity. In *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 449–460.
- [36] Qingsong Wen, Kai He, Liang Sun, Yingying Zhang, Min Ke, and Huan Xu. 2021. RobustPeriod: Robust time-frequency mining for multiple periodicity detection. In *Proceedings of the 2021 international conference on management of data*, 2328–2337.
- [37] Renjie Wu and Eamonn J Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering* 35, 3 (2021), 2421–2429.
- [38] Wenzhuo Yang, Kun Zhang, and Steven CH Hoi. 2022. A causal approach to detecting multivariate time-series anomalies and root causes. *arXiv preprint arXiv:2206.15033* (2022).
- [39] Zhun Jerry Yu, Fariborz Haghhighat, and Benjamin CM Fung. 2016. Advances and challenges in building engineering and data mining applications for energy-efficient communities. *Sustainable Cities and Society* 25 (2016), 33–38.
- [40] Jiuqi Elise Zhang, Di Wu, and Benoit Boulet. 2021. Time series anomaly detection for smart grids: A survey. In *2021 IEEE electrical power and energy conference (EPEC)*. IEEE, 125–130.

Algorithm	Accuracy	Avg Runtime (s)
Random	0.007	-
SuSS [5]	0.008	0.0113
MWF [11]	0.074	0.0441
FFT [6]	0.169	0.0026
Autoperiod [35]	0.512	2.6729
ACF [6]	0.687	0.0006
Peaks	0.754	0.0007

Table 1: Accuracy and runtime of period detection methods on synthetic datasets.

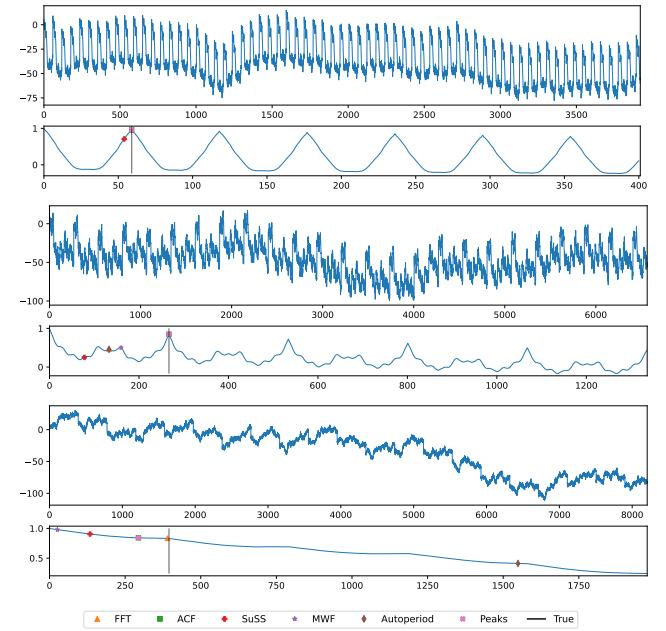


Figure 4: Three synthetic time series, together with their autocorrelation function. Detection results and ground truth are labeled in the ACF.

A Periodicity Detection Experiments

We conduct experiments to evaluate periodicity selection on a simple synthetic benchmark. Our signal is composed of three components, a random walk, noise, and a periodic random walk. First we draw a series length $n \sim \text{uniform}(60, 10000)$ and then an integer period $3 < k < \frac{n}{10}$. We draw the noise strength and periodicity strength $\eta_n, \eta_p \sim \text{uniform}(0, 10)$ and then sample a series as

$$x_t = w(t) + \eta_n \epsilon(t) + \eta_p w'(t \bmod k)$$

with w and w' random walks and ϵ independent noise. This very simple mechanism provides us with a ground truth periodicity, sequences of different difficulties (depending on the noise strength, periodicity strength and potential autocorrelation in w'). Given our experience with real data, this seems a realistic (if simplified) family of signals, signals that resemble commonly observed patterns,

such as heartbeats or weekly traffic patterns. In practice, there is often one dominating periodicity, say daily or weekly patterns in human behavior, where each individual daily or weekly pattern can be quite complex (see Figure 3), but the repetition is quite clear. We also experimented with having more than one periodicity, either with one being a multiple of the other (such as weeks and days) or them being independent. Given the difficulties we encountered with determining a single periodicity, we did not attempt to measure success for multiple periodicities. We limited ourselves to additive seasonal patterns, which we found more challenging than multiplicative patterns; several obvious extensions are possible. Figure 4 shows typical examples of various difficulties; the top shows a simple example, for which all but SuSS find the correct periodicity. The middle figure shows a more challenging series, for which periodicity is easily distinguished by eye on the original series, but on which several methods fail. The bottom shows a very noisy example in which an expert can find the solution based on

the ACF, but on which most approaches fail. Table 1 shows results from the methods investigated in Ermshaus et al. [6] using their code on 1000 synthetic series. Unfortunately, we were unable to run the RobustPeriod [36] code provided by Ermshaus et al. [6], despite efforts to fix version mismatches. Here, Peaks denotes a heuristic based on the autocorrelation function (ACF) that we investigated that returns the first non-dominated peak i.e. a peak that is higher than any following ACF values. This is only a slight modification of the ACF heuristic proposed by Ermshaus et al. [6], who propose to return the first peak. The Random score is calculated by permuting the known responses, i.e. it is random given the true distribution of periodicities. We conclude that on this synthetic, but realistic and challenging benchmark, current methods are outperformed by simple heuristics, which take much lower runtime. However, room for improvement remains for more sophisticated methods, as all methods still fail on cases where the truth can be determined by visual inspection of the ACF, see Figure 2.