# Theseus: Navigating the Labyrinth of Time-Series Anomaly Detection

Paul Boniol
Université Paris Cité
boniol.paul@gmail.com

John Paparrizos
The Ohio State University
paparrizos.1@osu.edu

Yuhao Kang
University of Chicago
yuhaok@uchicago.edu

Themis Palpanas
Université Paris Cité & IUF
themis@mi.parisdescartes.fr

Ruey S. Tsay
University of Chicago
ruey.tsay@chicagobooth.edu

Aaron J. Elmore
University of Chicago
aelmore@cs.uchicago.edu

Michael J. Franklin
University of Chicago
mjfranklin@uchicago.edu

## ABSTRACT

The detection of anomalies in time series has gained ample academic and industrial attention, yet, no comprehensive benchmark exists to evaluate time-series anomaly detection methods. Therefore, there is no final verdict on which method performs the best (and under what conditions). Consequently, we often observe methods performing exceptionally well on one dataset but surprisingly poorly on another, creating an illusion of progress. To address these issues, we thoroughly studied over one hundred papers, and summarized our effort in TSB-UAD, a new benchmark to evaluate univariate time series anomaly detection methods. In this paper, we describe Theseus, a modular and extensible web application that helps users navigate through the benchmark, and reason about the merits and drawbacks of both anomaly detection methods and accuracy measures, under different conditions. Overall, our system enables users to compare 12 anomaly detection methods on 1980 time series, using 13 accuracy measures, and decide on the most suitable method and measure for some application.

**PVLDB Reference Format:**
Paul Boniol, John Paparrizos, Yuhao Kang, Themis Palpanas, Ruey S. Tsay, Aaron J. Elmore, and Michael J. Franklin. Theseus: Navigating the Labyrinth of Time-Series Anomaly Detection. PVLDB, 15(12): XXX-XXX, 2022. doi:XX.XX/XXX.XX

## 1 INTRODUCTION

A wide range of technological advances in sensing solutions enables collecting enormous amounts of time-varying measurements commonly referred to as *time series*. In particular, analysts estimate that, shortly, billions of Internet-of-Things (IoT) devices will be responsible for generating zettabytes of time series [16]. This rapid growth of cost-effective IoT deployments already empowers diverse data science applications and has revolutionized the healthcare, manufacturing, transportation, agriculture, utilities, and automobile industries [22]. These time series collections then need to be analyzed in order to identify patterns and extract knowledge [2, 3, 24, 25, 28, 32]. Among analytical tasks for IoT data [17, 18, 20, 27, 29, 30, 33], *anomaly detection* (AD) focuses on identifying patterns that are different from the rest. Specifically, anomalies may either represent anomalous behavior exhibited by the process being monitored or correspond to imperfections in the monitoring and measurement systems used. In both cases, they need to be detected.

Despite over six decades of academic and industrial attention in time series AD [4–8, 15, 23, 37], only a few efforts have focused on establishing standard means of evaluating existing solutions (notable examples [19, 36, 38, 39]). Unfortunately, there is currently no consensus on using a single benchmark for assessing the performance of time series AD methods, and no comprehensive benchmark exists to evaluate time series AD methods. As a result, we observe two standard practices in the literature for benchmarking AD models by using (i) proprietary and synthetic data; or (ii) a limited collection of publicly available datasets. However, both of these practices are often flawed. In the former case, proprietary, or synthetic data may have been collected, or generated in a biased way so as to support particular claims, anomaly types, or methods. In the latter case, only a small fraction of datasets are publicly available, some of which suffer from several drawbacks (e.g., trivial anomalies, unrealistic anomaly density, or mislabeled ground truth [38]). Therefore, there is no final verdict on which method performs the best (and under what conditions).

In addition, the ambiguity and the startlingly different interpretation of anomalies across applications further hinders progress. It is not uncommon for methods to achieve high accuracy for some datasets, but surprisingly low accuracy for others. The lack of an established benchmark creates the illusion of progress, while the identification of robust approaches becomes unlikely. Notably, the recent advances in deep learning technologies have sparked a surge of interest in applying neural network architectures for time series tasks [11–14, 34], including for AD [9, 10, 21, 35]. This sudden enthusiasm and a slew of proposed methods in the preceding years underscore the vital need for a time series AD benchmark.

To address the issues mentioned above and provide an objective means of quantifying the performance of univariate time series AD methods, we propose *Theseus*, a system that aims to (i) easily navigate and compare several anomaly detection methods on a very large collection of time series; and (ii) showcase the differences between a large selection of accuracy measures. Theseus is based on TSB-UAD[1] (Time Series Benchmark for Univariate Anomaly Detection) [31], an open end-to-end benchmark suite, and on a recent, extensive study of accuracy evaluation measures for AD methods in time series, which resulted in the development of two new families of measures, namely, Range-AUC and Volume-Under-the-Surface (VUS) measures[2] [26]. Overall, the objective of Theseus is to facilitate the visualization and understanding of a large-scale evaluation of AD methods and accuracy measures, and guide users in the selection of the appropriate method and measure.

## 2 PRELIMINARIES

We now provide the background necessary for the rest of the paper.

[1] www.timeseries.org/tsb-uad
[2] www.timeseries.org/vus

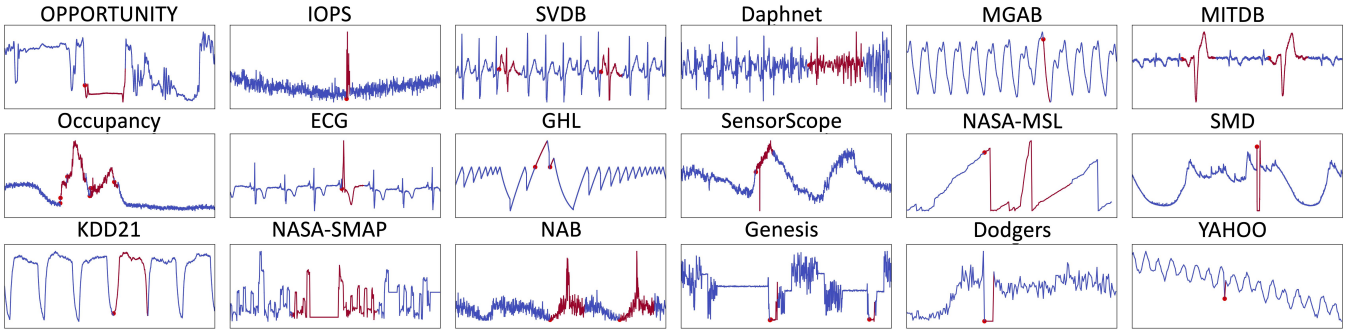


**Figure 1: Representative examples from the public, highly diverse (in anomaly type, length, size, density) datasets included in TSB-UAD. The ground truth anomalies are annotated in *red*.**

**[Datasets]** We use 18 different datasets introduced in the TSB-UAD benchmark [31], and summarized in Table 1. In total, this corresponds to 1980 time series, in which each point is labeled as normal or abnormal. These datasets contain either long or short time series (e.g., SVDB contains 230400 points on average, while YAHOO contains 1561 points), and with single or multiple anomalies (e.g., KDD21 is composed of time series with only one anomaly, while MITDB contains 210 anomalies on average). Examples of TSB-UAD time series and anomalies are shown in Figure 1.

**[AD methods]** We select 12 different AD methods, summarized in Table 1. Out of these, 8 are fully unsupervised (i.e., they require no prior information on the anomalies to be detected): IForest, IForest1, LOF, MP, NormA, PCA, HBOS, and POLY. The remaining 4 methods are semi-supervised (i.e., they require some information related to the normal behavior): OCSVM, AE, LSTM-AD, and CNN.

**[Accuracy measures]** Several measures have been proposed to quantify the quality of AD methods. We use 13 evaluation measures, listed in Table 1. Overall, we use 9 measures already proposed in the literature, and 4 recent measures developed specifically for the time series anomaly detection task [26].

## 3 THESEUS: SYSTEM OVERVIEW

In this section, we describe Theseus, the system we have developed to help analysts navigate through the datasets, methods, and results of the benchmark. The GUI is a stand-alone web application, developed using Python 3.6 and the Dash framework [1].

In total, the GUI is composed of 6 frames: (1) Home, (2) Overview (see Figure 3(A)), (3) Methods comparison (see Figure 3(B)), (4) Measures comparison (see Figure 3(C)), (5) Background, and (6) References. The Home frame contains a brief description of the objectives of the system, while the Background and References frames contain the notations, definitions and related works relevant to our system. (The other three frames are described below.)

Figure 2 illustrates the inputs and features of Theseus. The system can incorporate any number of datasets (18 in our demo), AD methods (12 in our demo), and accuracy evaluation measures (13 in our demo). The GUI permits interactions with these inputs. First, the user can visualize the time series, the positions of the anomalies, and the anomaly score. Then, the user can measure the performance of all methods or pairs of methods on one or more datasets. Finally, the user can measure the robustness and behavior of different evaluation measures. We now dive into each of the aforementioned actions and describe the three main frames of the GUI.

| Datasets | Description |
|---|---|
| Dodgers | unusual traffic after a Dodgers game |
| ECG | standard electrocardiogram dataset |
| IOPS | performance indicators of a machine |
| KDD21 | composite dataset released in a recent SIGKDD 2021 |
| MGAB | Mackey-Glass time series with non-trivial anomalies |
| NAB | Web-related real-world and artificial time series |
| SensorScope | environmental data |
| YAHOO | real and synthetic time series based on Yahoo production systems |
| NASA-MSL | Curiosity rover telemetry |
| NASA-SMAP | Soil-Moisture-Active-Passive spacecraft telemetry |
| Daphnet | acceleration sensors on Parkinson's disease patients |
| GHL | Gasoil Heating Loop telemetry |
| Genesis | portable pick-and-place demonstrator |
| MITDB | ambulatory ECG recordings |
| OPPORTUNITY | motion sensors for human activity recognition |
| Occupancy | temperature, humidity, light, and CO2 of a room |
| SMD | Server Machine telemetry |
| SVDB | ECG recordings |
| **AD Methods** | **Description** |
| IForest | tree-based method using subsequences as input |
| IForest1 | tree-based method using points as input |
| LOF | density-based method |
| MP | matrix profile method |
| NormA | cluster-based method |
| PCA | principle components analysis |
| AE | autoencoder model |
| LSTM-AD | recurrent neural network |
| POLY | polynomial approximation |
| CNN | convolutional neural network |
| OCSVM | one-class support vector machine |
| HBOS | histogram-based method |
| **Evaluation Measures** | **Description** |
| Precision@k | fraction of real anomalies among the $k$ most important detected sequences |
| Recall | fraction of detected anomalies among all anomalies |
| Precision | fraction of real anomalies among the detected sequences |
| F score | measure computed as $F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$ |
| Rrecall | Range-based version of Recall |
| Rprecision | Range-based version of Precision |
| RF score | measure computed as $RF = \frac{2 \cdot Rprecision \cdot Rrecall}{Rprecision + Rrecall}$ |
| AUC-PR | Area under the Precision-Recall curve |
| AUC-ROC | Area under the Receiver operating characteristic curves |
| R-AUC-PR | Range-based version of AUC-PR |
| R-AUC-ROC | Range-based version of AUC-ROC |
| VUS-PR | Volume Under the Surface, extension of R-AUC-PR |
| VUS-ROC | Volume Under the Surface, extension of R-AUC-ROC |

**Table 1: Summary of datasets, methods, and measures (for details and references to original papers/sources for the methods and datasets see [31], and for the measures see [26]).**

**[Overview Frame]** This frame depicts the accuracy of all anomaly detection methods for all datasets. Figure 3(A.a) depicts a table containing the row accuracy values, which are summarized in the

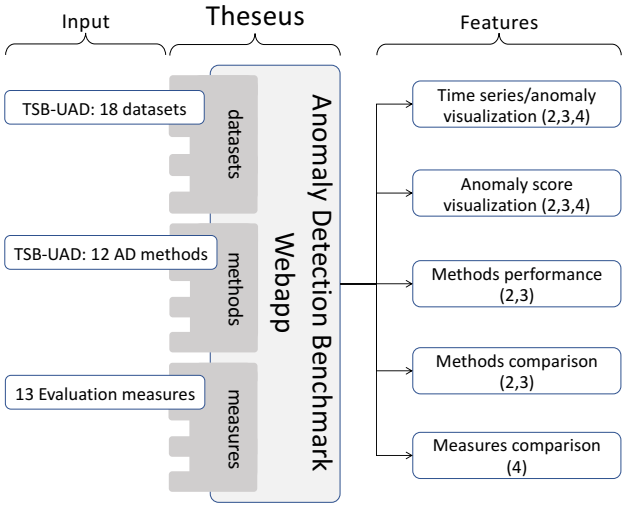


**Figure 2: Summary of the webapp inputs and features (with indicated related frames of the GUI).**

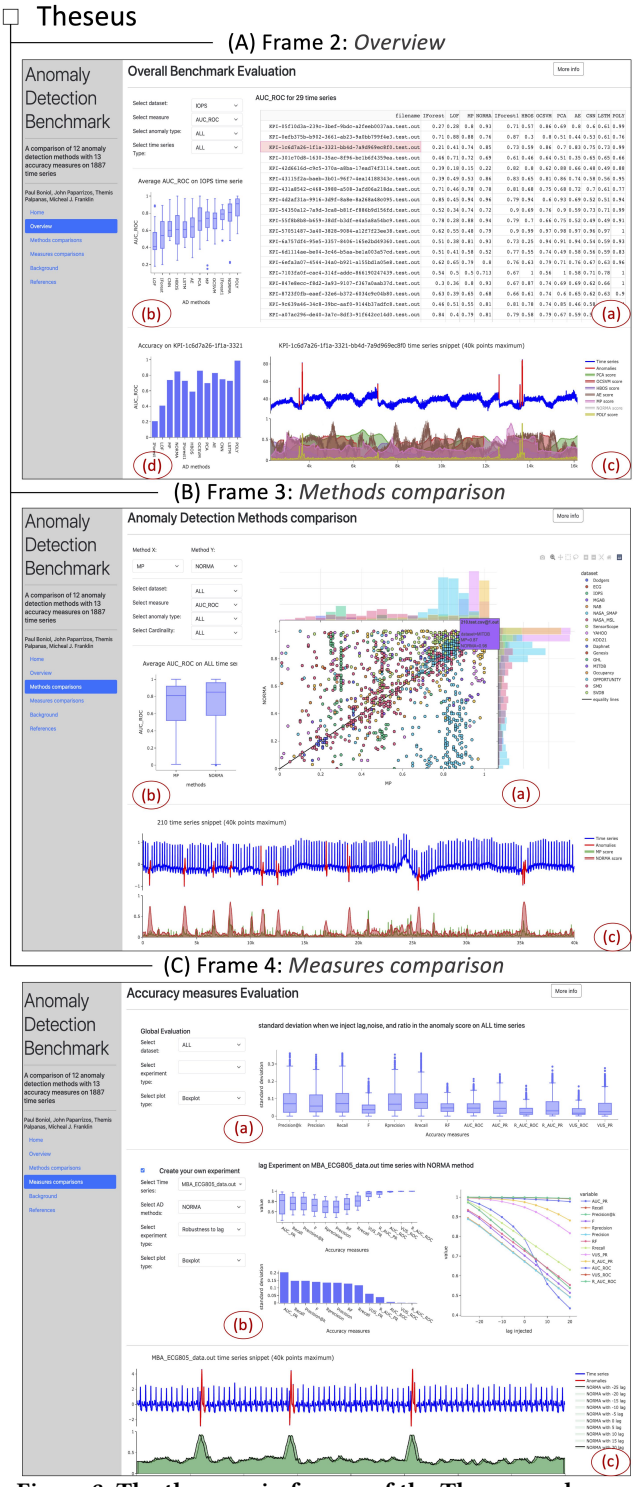


**Figure 3: The three main frames of the Theseus webapp.**

boxplot shown in Figure 3(A.b). The user can change the accuracy evaluation measure form the dropdown menu, and both the table and the boxplot are automatically updated. The user can also filter the table by selecting a specific dataset, a type of anomaly (i.e., point, or subsequence), and the cardinality of anomalies (i.e., single, or multiple). Finally, the user can click on any row of the table: this action will display the chosen time series as illustrated in Figure 3(A.c). The GUI also displays the anomaly scores and the annotated anomalies (highlighted in red). When one time series is selected, the GUI shows a bar plot that depicts the accuracy of each anomaly detection method on the selected time series.

**[Methods Comparison Frame]** In the previous frame, the user has a global overview of the methods' performance overthe datasets. In this frame (Figure 3(B)), the user can select any pair of methods and perfomr a detailed comparison. After choosing two methods, the GUI displays a scatter plot (Figure 3(B.a)), in which each point corresponds to a time series. The x- and y-axes correspond to the accuracy of the two selected methods. The color of the scatter points depends on the dataset to which the corresponding time series belongs. Moreover, the GUI displays a box plot (Figure 3(B.b)) that corresponds to the overall performance of the two methods. The user can then filter the scatter plot by selecting a specific dataset, a specific type of anomaly, or the anomaly cardinally; the scatter plot and the box plot are updated automatically. Finally, the user can click on any scatter point, and the corresponding time series will be displayed, along with the anomaly score of the two selected methods (Figure 3(B.c)).

**[Measures Comparison Frame]** In this third frame (shown in Figure 3(C)), the user is invited to focus on the accuracy measures evaluation. The GUI presents a robustness analysis of the evaluation measures. The user can learn more about this robustness analysis by clicking on the "more info" button: the user can visualize the effect of lag (i.e., injecting lag in the anomaly annotation), noise (i.e., injecting noise in the anomaly score), and normal/abnormal ratio variation (i.e., varying the ratio between anomalous and normal points) on the accuracy measures values. The system computes the accuracy values for all AD methods on all time series, and reports the average accuracy values along with the standard deviation. Next, the user can select one of the three experiments (i.e., varying lag, noise, or normal/abnormal ratio) and visualize the corresponding box plot for each measure (Figure 3(C.a)). The user may also

execute these experiments for each time series independently (see Figure 3(C.b)). First, the user selects a time series (which is automatically displayed in Figure 3(C.c)), an AD method, and a type of experiment (lag, noise, or ratio), and the system computes the accuracy values. Finally, the user can see the results in three plots (Figure 3(C.b)): (i) a box plot depicting the different values of each accuracy measure, (ii) a bar plot corresponding to the standard deviation for each measure, and (iii) a line plot for each accuracy measure for the different lag, noise or ratio values.

## 4 DEMONSTRATION SCENARIOS

This demo has three goals: (i) showcase the importance of tools to organize relevant benchmarks, and help users navigate the search space and reason about the results; (ii) enable the user to visualize, interact, and conduct statistical analysis using the benchmark; and (iii) challenge the user to understand the benefits and limitations of AD methods, as well as of accuracy measures.

**[Scenario 1: Finding the best method for a use case]**: This scenario begins in frame 1. We will ask the user to select a specific type of time series (based on their use case). To guide their choice, they can navigate through the table, click on different time series, and visualize the types of anomalies for specific datasets. First, the user chooses a specific dataset, e.g., medicine, environmental, or engineering (like the IOPS dataset selected in Figure 3(A.a)). The GUI will then indicate which AD methods are the most accurate for this use case. For instance, in Figure 3(A.b), the user discovers that POLY is outperforming the other AD methods on IOPS.

**[Scenario 2: Understanding the weak points of a method]**: In this scenario, we will let the user select two specific AD methods. The objective is to discover the strong and weak points of each method. For instance, as illustrated in Figure 3(B.a), the user may choose NormA and MP. Then, using the scatter plot that compares in detail the two selected methods, the user can identify clusters of time series for which these methods are more, or less accurate. By clicking on these time series and examining the relevant annotations (raw data, method results and anomaly scores, ground truth, etc.), the user can gain insight on which types of anomalies an AD method is more accurate than another.

**[Scenario 3: Selecting the correct evaluation measures]** Finally, the third scenario focuses on the evaluation and comparison of AD accuracy measures. In this scenario, we will ask the user to describe the behavior they expect that an accuracy measure should have (which could vary depending on the domain and application). For example, whether the accuracy measure should penalize an anomaly annotation with a small lag (when compared to the ground truth, with which it may still overlap). The user will select an AD method and a time-series, and he will analyze the behavior of different accuracy measures when we vary the parameter of interest (e.g., the lag of the annotation). For instance, in Figure 3(C.c), if lagged anomaly scoring cannot be tolerated, then, based on the line plot in Figure 3(C.b), the user should select the AUC-PR measure; otherwise, the user should choose R-AUC-ROC.

## 5 CONCLUSIONS

We demonstrate Theseus, a system that enables users to navigate through an extensive search space of datasets, methods, and accuracy measures for AD. It helps users discover which methods are better for specific use cases, identify weak and strong points of AD methods, and gain insight in the sensitivity of accuracy measures.

## REFERENCES


[1] Dash documentation. https://dash.plotly.com/.
[2] A. J. Bagnall, R. L. Cole, T. Palpanas, and K. Zoumpatianos. Data series management (dagstuhl seminar 19282). *Dagstuhl Reports*, 9(7), 2019.
[3] M. Bariya, A. von Meier, J. Paparrizos, and M. J. Franklin. k-shapestream: Probabilistic streaming clustering for electric grid events. In *2021 IEEE Madrid PowerTech*, pages 1–6. IEEE, 2021.
[4] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3), apr 2021.
[5] P. Boniol, M. Linardi, F. Roncallo, T. Palpanas, M. Meftah, and E. Remy. Unsupervised and scalable subsequence anomaly detection in large data series. *VLDBJ*, 2021.
[6] P. Boniol and T. Palpanas. Series2graph: Graph-based subsequence anomaly detection for time series. *PVLDB*, 13(11), 2020.
[7] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin. Sand in action: subsequence anomaly detection for streams. *PVLDB*, 14(12):2867–2870, 2021.
[8] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin. Sand: streaming subsequence anomaly detection. *PVLDB*, 14(10):1717–1729, 2021.
[9] L. Bontemps, J. McDermott, N.-A. Le-Khac, et al. Collective anomaly detection based on long short-term memory recurrent neural networks. In *FDSE*, 2016.
[10] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.
[11] A. Dziedzic, J. Paparrizos, S. Krishnan, A. Elmore, and M. Franklin. Band-limited training and inference for convolutional neural networks. In *ICML*, pages 1745–1754. PMLR, 2019.
[12] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.
[13] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean. Inceptiontime: Finding alexnet for time series classification. *DMKD*, 34(6), 2020.
[14] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch. Som-vae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199*, 2018.
[15] A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(3):350–363, 1972.
[16] S. George. *IoT Signals report: IoT's promise will be unlocked by addressing skills shortage, complexity and security.*, 2019. https://blogs.microsoft.com/blog/2019/07/30/.
[17] H. Jiang, C. Liu, Q. Jin, J. Paparrizos, and A. J. Elmore. Pids: attribute decomposition for improved compression and query performance in columnar storage. *PVLDB*, 13(6):925–938, 2020.
[18] H. Jiang, C. Liu, J. Paparrizos, A. A. Chien, J. Ma, and A. J. Elmore. Good to the last bit: Data-driven encoding with codecdb. In *SIGMOD*, pages 843–856, 2021.
[19] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *NeurIPS Track on Datasets and Benchmarks*, 2021.
[20] C. Liu, H. Jiang, J. Paparrizos, and A. J. Elmore. Decomposed bounded floats for fast compression and queries. *PVLDB*, 14(11):2586–2598, 2021.
[21] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN*, 2015.
[22] I. C. Ng and S. Y. Wakenshaw. The internet-of-things: Review and research directions. *International Journal of Research in Marketing*, 34(1):3–21, 2017.
[23] E. Page. On problems in which a change in a parameter occurs at an unknown point. *Biometrika*, 44(1/2):248–252, 1957.
[24] T. Palpanas and V. Beckmann. Report on the first and second interdisciplinary time series analysis workshop (ITISA). *SIGMOD Rec.*, 48(3), 2019.
[25] I. Paparrizos. *Fast, Scalable, and Accurate Algorithms for Time-Series Analysis.* PhD thesis, Columbia University, 2018.
[26] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. Elmore, and M. J. Franklin. Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *PVLDB*, 2022.
[27] J. Paparrizos, I. Edian, C. Liu, A. J. Elmore, and M. J. Franklin. Fast adaptive similarity search through variance-aware quantization. In *ICDE*, 2022.
[28] J. Paparrizos and M. J. Franklin. Grail: efficient time-series representation learning. *PVLDB*, 12(11):1762–1777, 2019.
[29] J. Paparrizos and L. Gravano. k-shape: Efficient and accurate clustering of time series. In *SIGMOD*, pages 1855–1870, 2015.
[30] J. Paparrizos and L. Gravano. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)*, 42(2):1–49, 2017.
[31] J. Paparrizos, Y. Kang, P. Boniol, R. S. Tsay, T. Palpanas, and M. J. Franklin. TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly Detection. *PVLDB*, 15(8):1697–1711, 2022.
[32] J. Paparrizos, C. Liu, B. Barbarioli, J. Hwang, I. Edian, A. J. Elmore, M. J. Franklin, and S. Krishnan. Vergedb: A database for iot analytics on edge devices. In *CIDR*, 2021.
[33] J. Paparrizos, C. Liu, A. J. Elmore, and M. J. Franklin. Debunking four long-standing misconceptions of time-series distance measures. In *SIGMOD*, pages 1887–1905, 2020.
[34] C. Pelletier, G. I. Webb, and F. Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11(5):523, 2019.
[35] M. Sakurada and T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *MLSDA*, 2014.
[36] N. Tatbul, T. J. Lee, S. Zdonik, M. Alam, and J. Gottschlich. Precision and recall for time series. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1924–1934, 2018.
[37] R. S. Tsay. Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, 7(1):1–20, 1988.
[38] R. Wu and E. J. Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *arXiv preprint arXiv:2009.13807*, 2020.
[39] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI*, volume 33, 2019.