Odyssey: An Engine Enabling The Time-Series Clustering Journey

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

Sai Prasanna Teja Reddy
Exelon Utilities
teja.reddy@exeloncorp.com

ABSTRACT

Clustering is one of the most popular time-series tasks because it enables unsupervised data exploration and often serves as a subroutine or preprocessing step for other tasks. Despite being the subject of active research across disciplines for decades, only limited efforts focused on benchmarking clustering methods for time series. Unfortunately, these studies have (i) omitted popular methods and entire classes of methods; (ii) considered limited choices for underlying distance measures; (iii) performed evaluations on a small number of datasets; or (iv) avoided rigorous statistical validation of the findings. In addition, the sudden enthusiasm and recent slew of proposed deep learning methods underscore the vital need for a comprehensive study. Motivated by the aforementioned limitations, we present Odyssey, a modular and extensible web engine to comprehensively evaluate 80 time-series clustering methods spanning 9 different classes from the data mining, machine learning, and deep learning literature. Odyssey enables rigorous statistical analysis across 128 diverse time-series datasets. Through its interactive interface, Odyssey (i) reveals the best-performing method per class; (ii) identifies classes performing exceptionally well that were previously omitted; (iii) challenges claims about the use of elastic measures in clustering; (iv) highlights the effects of parameter tuning; and (v) debunks claims of superiority of deep learning methods. Odyssey does not only facilitate the most extensive study ever performed in this area but, importantly, reveals an illusion of progress while, in reality, none of the evaluated methods could outperform a traditional method, namely, *k*-Shape, with a statistically significant difference. Overall, Odyssey lays the foundations for advancing the state of the art in time-series clustering.

PVLDB Reference Format:
John Paparrizos and Sai Prasanna Teja Reddy. Odyssey: An Engine Enabling The Time-Series Clustering Journey. PVLDB, 16(12): 4066 - 4069, 2023. doi:10.14778/3611540.3611622

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://odyssey-engine.streamlit.app/.

1 INTRODUCTION

Time-series analysis has gained ample attention due to the increasing prevalence of time-varying measurements across industrial and scientific applications [17, 18, 21, 22, 29]. Among analytical tasks for time series [6, 7, 11, 12, 24, 25, 31], clustering is one of the most widely used as it does not require annotated data or human supervision [4, 5, 20]. Clustering does not only facilitate effective data exploration but often serves as a preprocessing step or subroutine for other tasks (e.g., anomaly detection [8, 23, 28]).

Table 1: Summary of the clustering classes evaluated across 128 datasets using Odyssey. Last columns show category cardinality and similarity measures (in parentheses) evaluated in previous studies.

Clustering Class	Category Cardinality	Distance Measures	[16]	[27]	[19]	[14]
Partitional	5	10	3 (3)	5 (5)	5 (3)	2 (9)
Kernel	2	4	✘	1 (3)	✘	✘
Hierarchical	2	10	1(1)	1 (3)	✘	✘
Density	3	10	1 (2)	2 (3)	✘	✘
Distribution	2	10	✘	✘	✘	✘
Model	5	-	✘	✘	✘	✘
Shapelet	3	-	✘	1	1	✘
Semi-Supervised	2	-	✘	✘	✘	✘
Deep Learning	32	-	✘	✘	26	✘

Despite decades of attention, only limited efforts have focused on comprehensively evaluating time-series clustering methods (notable examples [14, 16, 19, 27]). Unfortunately, existing benchmarking studies often overlooked popular methods and entire classes of methods, omitted state of the art underlying distance measures, or performed comparisons on a limited number of datasets. Importantly, some studies have avoided any form of statistical validation of the findings, resulting in incomplete assessments of the superiority of certain methods. In addition, the recent advances in deep learning technologies have sparked a surge of interest in using neural network architectures for time-series clustering [19].

Considering the sudden enthusiasm and recent slew of proposed methods, we believe it is critical to revisit this subject in more detail. Importantly, our effort is also motivated by the necessity to challenge misconceptions that have appeared in the literature. For example, we have observed misconceptions concerned with the (i) importance of distance measures; (ii) parameter tuning affected by supervised tasks; and (iii) inadequate comparisons among neural networks with many modular components (e.g., architectures, reconstruction losses, pretext losses, optimizers, etc.).

Motivated by the aforementioned issues and our curiosity to shed some light on these misconceptions, we present Odyssey [1], a modular and extensible system to assist in the navigation of the time-series clustering land. Odyssey aims to enhance the visualization and comprehension of a large-scale evaluation of time-series clustering methods using a wide variety of datasets, clustering methods, and quality assessment measures. In particular, Odyssey integrates 80 time-series clustering methods spanning 9 different classes from the data mining, machine learning, and deep learning literature [2]. Table 1 summarizes the characteristics of the evaluation enabled by Odyssey in comparison to earlier studies.

Odyssey facilitates on-the-fly, rigorous statistical analysis across 128 diverse time-series datasets [9], using 3 assessment measures.

Proceedings of the VLDB Endowment, Vol. 16, No. 12 ISSN 2150-8097.
doi:10.14778/3611540.3611622

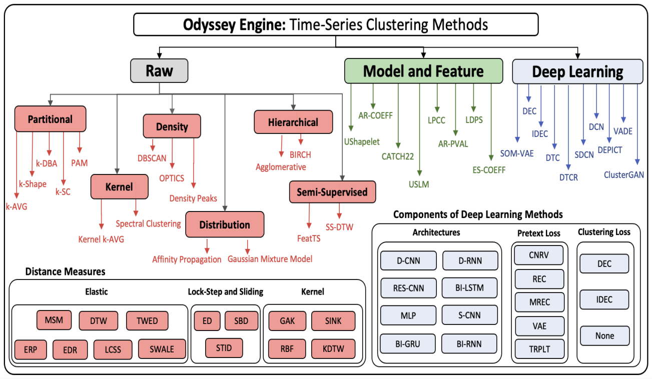


Figure 1: Taxonomy of time-series clustering methods in Odyssey.

Through its interactive interface, Odyssey not only reveals the best-performing methods or omitted classes but, importantly, challenges biases about the importance of distance measures and parameter tuning, and debunks claims of superiority of deep learning methods. Odyssey enables the most extensive study ever performed in this area and has revealed an illusion of progress: none of the evaluated methods was able to outperform with a statistically significant difference *k*-Shape, a scalable partitional method [26]. Overall, Odyssey alters the landscape of what is known about time-series clustering and lays the foundations for advancing state of the art.

2 EVALUATION FRAMEWORK

We now provide the background necessary to introduce Odyssey, including details for the methods, datasets, and quality measures.
Time-Series Clustering Methods: To enable a fair and reproducible study of time-series clustering methods, Odyssey requires all methods under the same framework. Specifically, Odyssey builds on top of our new (unpublished) library [2], which aims to hide all the complexity of benchmarking time-series clustering methods. Therefore, Odyssey integrates 80 methods that encompass a diverse range of algorithmic classes, including partitional, hierarchical, kernel, density, distribution, shapelet-based, semi-supervised, model-based, feature-based, and deep learning methods. For certain classes, Odyssey enables extensive in-depth analysis to derive critical insights and factors that influence the performance of methods. For example, for partitional and kernel methods, this analysis includes choices of the best 10 distance and best 4 kernel measures as identified before in [30]. For deep learning methods, the analysis dives into components of neural networks such as the architecture, the clustering losses, and the pretext losses, as described in detail at [19]. Figure 1 summarizes the classes of methods integrated at Odyssey, along with the distance measures and components of the deep learning methods (we provide all references online [1, 2]).
Datasets: We conduct our evaluation using the UCR Time-Series Archive [9], the largest collection of class-labeled time series datasets. The archive consists of 128 real and synthetic datasets, which span several different domains. Each dataset contains from 40 to 24000 sequences and their length vary from 15 to 2844.
Quality Assessment: To assess the clustering quality of each method, we use the following popular measures: Rand Index (RI) [32], Adjusted Rand Index (ARI) [15] and Normalized Mutual Information (NMI) [33]. Along with the above measures, Odyssey integrates the Friedman test [13] and the post-hoc Nemenyi test [10] to compare results for multiple methods across multiple datasets.

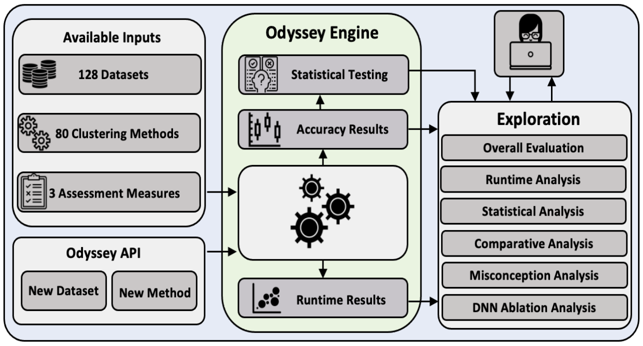


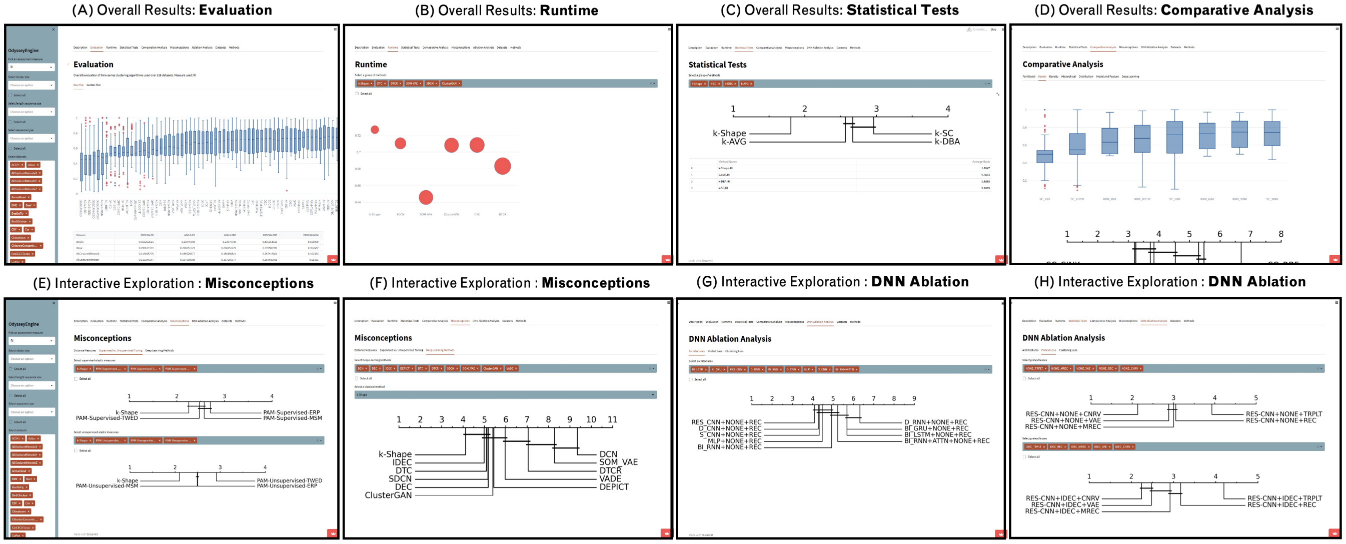
Figure 2: Overview of Odyssey's architecture.

3 ODYSSEY ENGINE: SYSTEM OVERVIEW

In this section, we present Odyssey, a modular and extensible engine designed to assist analysts in navigating large-scale benchmark evaluations containing results from hundreds of datasets and methods. Odyssey features a stand-alone web application with a GUI, developed using Python 3.9 and the Streamlit framework [3].

Figure 2 shows the architecture of the Odyssey engine, including the inputs and outputs. Odyssey exploits 128 different datasets, 80 time-series clustering algorithms spanning 9 classes, and 3 quality assessment measures. The GUI permits interactions with these inputs (selection of dataset, dataset characteristics, clustering classes, clustering methods, and assessment measures). The user interacts with Odyssey to visualize and compare the performance of various classes and methods under different desired settings. Odyssey enables an interactive, on-the-fly, rigorous statistical analysis, which is critical for navigating thousands of possible options of the underlying large-scale study of time-series clustering methods.

In total, the GUI consists of 9 frames: (1) Description, (2) Evaluation, (3) Runtime, (4) Statistical Test, (5) Comparative Analysis, (6) Misconceptions, (7) DNN Ablation Analysis, (8) Datasets, and (9) Methods. The Description frame introduces the objectives of the system along with some additional resources, while the Datasets and Methods frames contain the summary information about datasets and methods. Next, we provide details for the remaining frames.
Evaluation Frame: This frame has two sub-frames. The first sub-frame, shows the overall performance and compares our results using a table and boxplot (see Figure 3 (A)). The table displays one clustering assessment value for each dataset and method, while the boxplots illustrate the distribution of clustering assessment values for each method. The second sub-frame compares the performance of any pair of methods using a scatterplot. By hovering the mouse over these plots, the user obtains dataset details.
Runtime Frame: This frame uses a bubbleplot (as shown in Figure 3 (B) to illustrate the runtime performance of clustering methods on 128 datasets. The runtime performance is the total time required by a clustering method to fit and infer clusters for the selected datasets. Deep learning methods utilize a GPU and, therefore, some of them may appear faster than traditional CPU-bound methods. The interface marks the GPU-accelerated methods for clarity.


(A) Overall Results: **Evaluation**

(B) Overall Results: **Runtime**

(C) Overall Results: **Statistical Tests**

(D) Overall Results: **Comparative Analysis**

(E) Interactive Exploration : **Misconceptions**

(F) Interactive Exploration : **Misconceptions**

(G) Interactive Exploration : **DNN Ablation**

(H) Interactive Exploration : **DNN Ablation**

Figure 3: The main frames of the Odyssey engine.

Statistical Testing Frame: This frame in Figure 3 (C) allows the user to choose a subset of methods to conduct statistical testing and produce a critical difference diagram. The critical difference diagram shows the statistical significance of the performance differences among various methods across multiple datasets. By interacting with the left sidebar, the user can statistically validate any potential hypothesis using the underlying large-scale results.

Comparative Analysis Frame: Each sub-frame shows the comparative analysis of time-series clustering methods within a class of methods. Figure 3 (D) consists of a table that reports the clustering assessment measure, a boxplot that summarizes the performance, and a critical difference diagram that indicates the statistical significance in performance among the methods in a class.

Misconceptions Frame: This frame, shown in Figures 3 (E & F), is composed of three sub-frames, where each sub-frame addresses an important misconception in the time-series clustering literature. Specifically, in these subframes, the user explores misconceptions about the impact of (i) choosing a distance measure in clustering methods, (ii) tuning parameters using supervised and unsupervised settings, and (iii) evaluating standalone deep learning methods as proposed in the literature without an effort of bringing them in the exact same evaluation framework (i.e., same optimizers, loss functions, etc.). These results help debunk several misconceptions about the superiority of deep learning methods [19] or about the fact that elastic distance measures do not improve the performance of clustering measures compared to Euclidean distance [14].

DNN Ablation Analysis Frame: Finally, this frame provides a deeper analysis of the contributions of the individual components of deep learning clustering methods. As shown in Figures 3 (G & H), each sub-frame evaluates the contributions of the selected architecture, pretext loss, and clustering loss on model performance by visualizing critical difference diagrams. This is of great importance because Odyssey enables users to perform large-scale ablation analysis for neural networks and understand trade-offs on different datasets. This ablation analysis essentially permits keeping two characteristics static (e.g., clustering and pretext loss) and varying the third characteristic (e.g., architecture), to understand the impact of different architectures in certain datasets or the benchmark.

4 DEMONSTRATION SCENARIOS

This demo has five goals: (i) emphasize the significance of utilizing a Web engine to derive valuable insights regarding the disparities in the performances of an extensive set of clustering methods when assessed on a diverse range of datasets; (ii) facilitate the user with the ability to visualize and contrast the trade-off between runtime and performance among various clustering methods; (iii) empower the user to perform and visualize statistical testing to enhance their comprehension of the significance of differences in performance among various methods and classes; (iv) enable users to explore prevalent misconceptions within the time-series clustering literature; and (v) derive deeper insights on the contribution of different components towards the performance of deep learning methods.

Scenario 1: Finding the best clustering method: This scenario is depicted in the evaluation frame 3 (A). First, utilizing the sidebar, users have the option to either select all available datasets or manually choose a specific set based on characteristics such as cluster size, sequence length, or sequence type. Then, users may select from all implemented time-series clustering methods or choose a subset of methods from specific classes. The first sub-frame displays the time-series clustering methods in increasing order of performance using a boxplot and provides raw performance scores in the table below. The second sub-frame offers an interactive user interface for comparing pairs of methods using a scatterplot. The scatterplot enables the user to identify datasets where the compared methods are more or less accurate. This enables users to determine the most suitable time-series clustering methods per dataset.

Scenario 2: Understanding the accuracy to runtime performance trade-off of different methods: In this scenario, the user is asked to select a clustering assessment measure and a set of time-series clustering methods to understand the tradeoffs between accuracy and runtime. Figure 3 (B) displays a bubble plot with time-series clustering methods on the x-axis and their average accuracy on the y-axis. The size of each bubble indicates the magnitude of runtime. The runtime and accuracy scores displayed in this frame are aggregated values for selected datasets, and the plot is ordered with methods consuming the least to most runtime. For instance, Figure 3 (B) highlights the significance of scalable methods like *k*-Shape, which is faster and more accurate than DNN methods.

Scenario 3: Highlighting the current state of each time-series clustering class: In this scenario, the methods for each time-series clustering class are preset, as shown in Figure 3 (D). Each sub-frame in this frame summarizes the results for a clustering class. Users can change clustering assessment measures and have the option to either select all available datasets or manually choose a specific set based on characteristics such as cluster size, sequence length, or sequence type. The results for each class are summarized using a boxplot, a critical difference diagram, and a table. This information helps users understand the current state of each time-series clustering class. For example, in the partitional clustering class, we observe that *k*-Shape statistically outperforms other methods, while in the deep learning class, we observe that no method statistically outperforms other methods in the class.

Scenario 4: Uncovering challenging claims and misconceptions in time-series clustering literature: The fourth scenario focuses on uncovering challenging claims and misconceptions in the literature. The misconceptions frame, as shown in Figure 3 (E & F), is composed of three sub-frames. Each sub-frame is preset to a set of methods, with its results highlighting a misconception or challenging an existing claim. In the first sub-frame, contrary to the claims from [14] that DTW fails to outperform the ED similarity measure, our results show that under both supervised and unsupervised settings, DTW outperforms the ED similarity measure with a statistically significant difference. The choice of parameters for distance measures such as MSM, TWED, SWALE, DTW, EDR, and LCSS are often arbitrary in the literature. The results from the second sub-frame show that in the unsupervised setting, *k*-Shape for example outperforms the top elastic measure with statistical significance. However, in the supervised setting, there is no significant difference in performance. Likewise, in the supervised setting, measures like TWED outrank all other elastic measures. In contrast, in the unsupervised setting, it is outranked by non-parametric measures like ERP. Finally, using the results from the third sub-frame, we gather that many deep learning-based time-series clustering models have no significant difference in their performances, and none of the best-performing deep learning-based time-series clustering models can outperform *k*-Shape. It is crucial to identify such observations to debunk the illusion of progress in this area.

Scenario 5: Evaluating contributions of components for deep learning models: As shown in Figure 3 (G & H), this final scenario evaluates how neural network components, such as the architecture, pretext loss, and clustering loss affects the performance of these methods. Despite previous extensive evaluations [19], simultaneous evaluation of numerous components makes rigorous conclusions difficult. In the first sub-frame, convolution-based models (RES_CNN, D_CNN, S_CNN) outrank recurrent (BI_RNN, BI_LSTM, BI_GRU, D_RNN) and fully-connected (MLP) models with no statistically significant difference. Similarly, in the second and third sub-frames, our implementation of contrastive pretext loss (CNRV) outperforms the previously evaluated pretext losses (REC, MREC, VAE, TRPLT). Results are summarized with critical diagrams.

5 CONCLUSIONS

We described Odyssey, a system that allows users to navigate the vast search space of datasets, methods, and evaluation techniques for time-series clustering. Odyssey revealed best-performing methods per class, challenged biases concerned with the impact of distance measures and parameter tuning, and debunked claims of the superiority of deep learning methods. Odyssey revealed an illusion of progress and altered the landscape in this area.

REFERENCES

[1] Odyssey Engine available online. https://odyssey-engine.streamlit.app/.

[2] Our new (unpublished) clustering library. https://www.timeseries.org/tsclusteringeval.

[3] Streamlit documentation. https://dash.plotly.com/.

[4] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time-series clustering–a decade review. *Information systems*, 53:16–38, 2015.

[5] 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.

[6] P. Boniol, J. Paparrizos, Y. Kang, T. Palpanas, R. S. Tsay, A. J. Elmore, and M. J. Franklin. Theseus: navigating the labyrinth of time-series anomaly detection. *VLDB*, 15(12):3702–3705, 2022.

[7] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin. Sand in action: subsequence anomaly detection for streams. *VLDB*, 14(12):2867–2870, 2021.

[8] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin. Sand: streaming subsequence anomaly detection. *VLDB*, 14(10):1717–1729, 2021.

[9] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[10] C. E. Douglas and F. A. Michael. On distribution-free multiple comparisons in the one-way analysis of variance. *Communications in Statistics-Theory and Methods*, 20(1):127–139, 1991.

[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] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34, 2012.

[13] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[14] C. Holder, M. Middlehurst, and A. Bagnall. A review and evaluation of elastic distance functions for time series clustering. *arXiv preprint arXiv:2205.15181*, 2022.

[15] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[16] A. Javed, B. S. Lee, and D. M. Rizzo. A benchmark study on time series clustering. *Machine Learning with Applications*, 1:100001, 2020.

[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. *VLDB*, 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] B. Lafabregue, J. Weber, P. Gançarski, and G. Forestier. End-to-end deep representation learning for time series clustering: a comparative study. *Data Mining and Knowledge Discovery*, pages 1–53, 2021.

[20] T. W. Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[21] C. Liu, H. Jiang, J. Paparrizos, and A. J. Elmore. Decomposed bounded floats for fast compression and queries. *VLDB*, 14(11):2586–2598, 2021.

[22] S. Liu, T. Mangla, T. Shaowang, J. Zhao, J. Paparrizos, S. Krishnan, and N. Feamster. Amir: Active multimodal interaction recognition from video and network traffic in connected environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(1):1–26, 2023.

[23] 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. *VLDB*, 15(11):2774–2787, 2022.

[24] J. Paparrizos, I. Edian, C. Liu, A. J. Elmore, and M. J. Franklin. Fast adaptive similarity search through variance-aware quantization. In *ICDE*, pages 2969–2983. IEEE, 2022.

[25] J. Paparrizos and M. J. Franklin. Grail: efficient time-series representation learning. *VLDB*, 12(11):1762–1777, 2019.

[26] J. Paparrizos and L. Gravano. k-shape: Efficient and accurate clustering of time series. In *SIGMOD*, pages 1855–1870, 2015.

[27] J. Paparrizos and L. Gravano. Fast and accurate time-series clustering. *TODS*, 42(2):1–49, 2017.

[28] 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. *VLDB*, 15(8):1697–1711, 2022.

[29] 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.

[30] 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.

[31] J. Paparrizos, K. Wu, A. Elmore, C. Faloutsos, and M. J. Franklin. Accelerating similarity search for elastic measures: A study and new generalization of lower bounding distances. *VLDB*, 16(8):2019–2032, 2023.

[32] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[33] H. Zhang, T. B. Ho, Y. Zhang, and M.-S. Lin. Unsupervised feature extraction for time series clustering using orthogonal wavelet transform. *Informatica*, 30(3), 2006.