Time-Series Clustering: A Comprehensive Study of Data Mining, Machine Learning, and Deep Learning Methods

John Paparrizos
The Ohio State University and
Aristotle University of Thessaloniki
paparrizos.1@osu.edu

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

ABSTRACT

Time-series clustering is a key task in time series analysis, enabling unsupervised data exploration and often serving as a subroutine for other tasks. Despite decades of active cross-disciplinary research, benchmarking of time-series clustering methods has received limited attention. Existing studies have (i) excluded popular methods and entire method classes; (ii) used a narrow range of distance measures; (iii) evaluated only a few datasets; (iv) lacked statistical validation; (v) had poor reproducibility; or (vi) relied on questionable evaluation setups. The rise of deep learning—especially foundation models claiming broad generalization—further emphasizes the need for comprehensive evaluation, as their role in time-series clustering remains largely untested. To address these gaps, we evaluate 84 time-series clustering methods across 10 method classes from data mining, machine learning, and deep learning. Our analysis spans 128 time-series datasets and uses rigorous statistical methods. Within a fair comparison framework, we (i) identify the top-performing method in each class; (ii) highlight previously overlooked, high-performing classes; (iii) challenge assumptions about elastic distance measures; (iv) refute the claimed superiority of deep learning methods, including foundation models; (v) expose reproducibility issues; (vi) analyze performance variation across dataset properties; and (vii) assess scalability. Our findings reveal an illusion of progress: no method significantly outperforms the decade-old k-Shape method. Still, we highlight a deep learning-based approach with notable promise. Our results provide a strong benchmark for advancing time-series clustering, and we have open-sourced our work to support future research.

PVLDB Reference Format:
John Paparrizos and Sai Prasanna Teja Reddy Bogireddy. Time-Series Clustering: A Comprehensive Study of Data Mining, Machine Learning, and Deep Learning Methods. PVLDB, 18(11): 4380 - 4395, 2025.
doi:10.14778/3749646.3749700

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at www.timeseries.org/tsclusteringeval.

1 INTRODUCTION

A time series is a temporal sequence of ordered, time-indexed measurements. Recent advances in data storage and processing allow us to capture and analyze large data volumes, including time series [69, 78, 79, 86, 92, 93, 98, 124]. Availability of large volumes of time-series data has led to an enormous interest in their analysis [61, 108, 115, 129, 130] utilizing tasks such as clustering [10, 13, 47, 59, 81, 120, 121, 127, 128, 132], classification [9, 49, 67, 119], anomaly detection [18–23, 23–25, 38, 94–97, 116, 117, 143], and similarity search [15, 28, 32, 45, 50, 91, 118, 123, 125, 126, 131, 153]. Applications of time-series analysis are prevalent across various domains in everyday life, such as astronomy [75, 148], biology [11, 12], economics [27, 106], energy sciences [8, 107], engineering [110, 111], environmental sciences [63, 70], medicine [35, 133], and social sciences [27, 109]. The remarkable growth, along with the widespread availability of time series data, has stimulated considerable interest in deriving insights from time series.

Table 1: Summary of our experimental evaluation across 128 datasets. The last four columns show category cardinality and distance measures (in parentheses) evaluated in previous studies.

Clustering Class	Category Cardinality	Distance Measures	[77]	[121]	[88]	[72]
Partitional	6	10	3(3)	5(3)	5(3)	2(9)
Kernel-Based	2	4	✘	1(3)	✘	✘
Hierarchical	2	10	1(1)	1(3)	✘	✘
Density-Based	3	10	1(2)	2(3)	✘	✘
Distribution-Based	2	10	✘	✘	✘	✘
Model-Based	5	-	✘	✘	✘	✘
Shapelet-Based	3	-	✘	1	1	✘
Semi-Supervised	2	-	✘	✘	✘	✘
Deep Learning	32	-	✘	✘	26	✘
Foundation	3	-	✘	✘	✘	✘

Clustering has emerged as a valuable technique in large-scale data analysis, allowing for effective summarization of dataset characteristics and serving as a crucial preprocessing step for various time-series analytical tasks. The goal is to partition data into multiple homogeneous groups where each group represents a characteristic pattern or structure in the data. However, applying traditional clustering methods to time-series data is challenging due to the interdependence of values across different time steps in a sequence. Consequently, the right choice of distance measure is crucial in accurately distinguishing similar and dissimilar time-series sequences. In recent decades, time-series clustering has received significant attention [10, 39, 120, 134, 154, 156]. Despite the abundance of clustering techniques and distance measures, factors such as domain diversity of datasets, distortions in sequences, and high dimensionality introduce challenges in developing robust algorithms. All these characteristics make time-series clustering a hard problem to formalize and solve. Therefore, it is imperative to conduct a systematic evaluation to compare various time-series clustering algorithms and their distance measures to gather a deeper understanding of the components that impact the efficacy of diverse models.

Proceedings of the VLDB Endowment, Vol. 18, No. 11 ISSN 2150-8097.
doi:10.14778/3749646.3749700

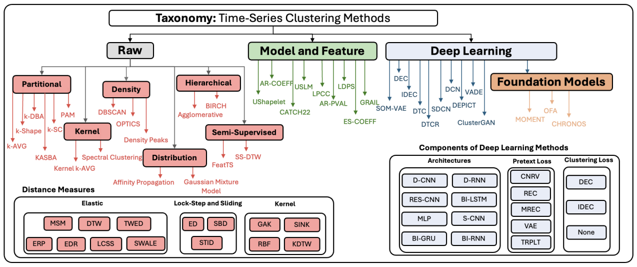


Figure 1: Taxonomy of time-series clustering methods.

Despite growing interest and numerous methods proposed, there has been a notable lack of thorough benchmarking analyses in the literature. Existing survey and benchmarking studies [2–4, 54, 77, 84, 88, 90, 121, 149] often suffer from limited scope regarding the diversity of clustering methods and categories. Furthermore, most clustering methods presented in these studies are evaluated on a limited number of datasets and arbitrary choice of assessment metrics. Reproducibility issues have emerged as a significant concern. The unavailability of original implementations for certain methods and the inadvertent introduction of potential bugs in popular third-party implementations hinder the ability to accurately replicate and validate previous findings. Table 1 compares and provides a quantitative summary of the existing benchmark studies and highlights the shortcomings in terms of missing classes and methods of time-series clustering algorithms and their distance measures. As a result, the conclusions from these studies are either incomplete or misleading and do not reflect actual progress in the field.

Given the gaps identified in previous research and the widespread interest in time-series clustering across various industries, conducting a thorough investigation is crucial. Our study is motivated by the need to address these shortcomings, aiming to provide research insights into time-series clustering. In addition, through our experimental evaluation of time-series clustering, we present several popular research questions (**RQ**) (see $\mathcal{RQ}1-\mathcal{RQ}3$ in Section 2) in the literature concerning (i) determining which time-series clustering methods, across the diverse classes of classical methods identified in the literature, demonstrate superior performance on a wide range of datasets; (ii) investigating the role of distance measures, including the effect of supervised tuning on parameter-dependent measures, in time-series clustering tasks; and (iii) exploring how deep learning and foundation models are leveraged in time-series clustering and assessing the impact they have on clustering performance. There are many misconceptions and biases in the answers to these questions, as they are derived from several popular prior studies on time-series clustering. Many of these studies employed buggy implementations, omitted methods, provided no or unclear parameter tuning instructions, omitted datasets, and arbitrarily selected metrics with little or no statistical testing. Therefore, it is imperative to address these enduring research questions.

In this study, we perform the most comprehensive analysis of time-series clustering to date, unveiling popular research questions in the current literature. In terms of **breadth** of this study, as depicted in Table 1, we incorporate 10 different clustering classes and consistently exceed the number of clustering techniques and appropriate distance measures implemented in each clustering class compared to previous influential benchmarking studies. In addition, we present a taxonomy of time-series clustering methods in Figure 1, which summarizes the research efforts in this field. Regarding the **depth**, we compare all the clustering techniques presented in the study by evaluating them on the UCR time-series archive [34], comprising 128 datasets. This evaluation extends beyond the norm established by most individual time-series clustering studies [52, 101, 103] and benchmarking studies [72, 77, 88, 121], as we assess the performance of each clustering technique using three widely recognized clustering evaluation metrics. Additionally, two statistical tests are employed to demonstrate the statistical significance of the performance of clustering assessment metrics for a clustering technique relative to others, thereby contributing to a more comprehensive understanding of time-series clustering.

In summary, this study makes several key contributions: (i) it demonstrates that none of the time-series clustering methods proposed in the literature significantly outperform a decade-old method, namely, k-Shape; (ii) it offers a comprehensive evaluation of clustering categories and techniques often omitted from baseline comparisons in individual studies and benchmarking papers on time-series clustering; (iii) it critically assesses recent claims concerning parameter tuning and issues of reproducibility; (iv) it shows that deep learning-based time-series clustering models, including foundation models, do not statistically outperform leading classical models; (v) it introduces a novel distance-based deep contrastive time-series clustering model that exhibits promising potential; (vi) it includes a systematic evaluation of leading clustering algorithms, assessing performance variability across key dataset characteristics, analyzing their accuracy–runtime trade-offs, and performing a scalability analysis with respect to sequence length and dataset size; (vii) to facilitate future research and address the challenges identified, we are releasing a comprehensive, open-source library for time-series clustering at [1].

We start with the discussion on three popular research questions (Section 2). Then, we present our contributions:

- We provide an end-to-end and open-source benchmark containing 84 clustering methods spanning 10 different classes to ensure the reproducibility of our results (Section 3).
- We conduct a comprehensive evaluation of 8 well-known classes of classical time-series clustering methods (Section 4).
- We review and evaluate popular deep learning time-series clustering methods from the literature (Section 5.1).
- We further decompose the components of deep learning-based models into different design choices and analyze the impact of each component (Section 5.2 and 5.3).
- Finally, we present a comprehensive evaluation of clustering methods, assessing performance in terms of accuracy–runtime trade-offs, data distribution, and scalability (Section 6).

We conclude with the implications of our work and a discussion of new directions and challenges (Section 7).

2 THE THREE RESEARCH QUESTIONS

In this section, we outline three primary research questions arising from misconceptions and biases in time-series clustering. These questions reflect broader misunderstandings in the field that have, over time, been reinforced by subsequent research. Specifically, these questions address: (1) dataset selection, (2) evaluation metrics and statistical tests, and (3) baseline comparisons and reproducibility. One major issue is the arbitrary selection of datasets for evaluation, as observed in [52, 77, 101, 103]. Often, researchers offer insufficient rationale for dataset choices, which may compromise claims of generalizability. Similarly, the choice of evaluation metrics and statistical tests can appear discretionary; applying these consistently would improve research objectivity and transparency. A further concern is the selection of baselines for comparisons. Some studies [52, 72, 77, 88, 101, 103] include a limited set of baselines, potentially omitting key methods that would provide a more comprehensive benchmark. Reproducibility also emerges as a significant challenge: the unavailability of original implementations complicates efforts to replicate and validate results in many published studies. Moreover, certain methods conceal critical evaluation parameters. For example, [72] evaluated popular parameter-dependent measures using default parameter values, thus obscuring their true potential. Similarly, [71] tested baseline models under parameter settings that appear suboptimal compared to those of the proposed approach. In particular, some baselines were restricted to 50 iterations, whereas the proposed model ran for 300 iterations; baseline centroids were initialized randomly, while the proposed model used k-Means++ initialization, a well-established enhancement that improves clustering quality. This reliance on additional iterations and k-Means++ initialization provided a performance advantage that baseline models did not leverage, raising concerns about fairness. Such disparities can mask the baseline methods' capabilities and complicate assessment of individual contributions in the proposed model. These issues continue to hinder the ability to draw definitive conclusions in the field, underscoring the persistent nature of these misconceptions across the literature.

Before framing our research questions, we must clarify that we do not suggest the discussed misconceptions were deliberately created or maliciously fabricated. Instead, we acknowledge they might arise from resource limitations, honest misinterpretations, or oversight. To address these misconceptions, we will use a question-and-answer format in our discussion to provide clarity and insight.

$\mathcal{RQ}$ 1: **Which classical time-series clustering methods exhibit superior performance across datasets?**

Discussion: Literature lacks consensus on the most effective classical time-series clustering technique, as studies present conflicting results. For example, [77] evaluated nine partitional, hierarchical, and density-based methods, concluding that no method consistently outperforms others, since results depend on datasets and evaluation metrics. Similarly, [88] compared partitional methods and found no significant differences among k-AVG, k-Shape, and k-DBA. This ambiguity arises from two factors. First, many studies omit entire categories of methods, yielding incomplete evaluations. Second, methodological inconsistencies, such as variations in algorithm implementations (e.g., the tslearn implementation of k-Shape differs from the original authors) and differences in dataset selection or evaluation metrics, hinder reliable comparison. These limitations preclude definitive conclusions regarding the relative efficacy of classical clustering approaches.

$\mathcal{RQ}$ 2: **What is the role of distance measures in time-series clustering tasks, and how does supervised tuning of parameter-dependent measures affect their clustering performance?**

Discussion: Distance measures are pivotal in time-series clustering, as they quantify the degree of similarity or dissimilarity between sequences, influencing clustering outcomes and the algorithm's ability to discern patterns. Given the sequential nature of time-series data, often exhibiting variations in timing, amplitude, and shape, selecting an appropriate measure is essential. Traditionally, Euclidean Distance (ED) has been deemed inadequate for capturing shifts within sequences, leading to the belief in Dynamic Time Warping (DTW) as the superior approach due to its capacity to handle temporal distortions. However, Holder et al. [72] challenged this view by claiming ED's superiority in clustering tasks and reporting that parameter-free measures outperform many parameter-dependent measures. We contend that their study did not adequately explore parameter tuning for parameter-dependent measures, potentially overlooking their full capability, since the performance of parameter-dependent measures is highly sensitive to parameter settings. These conflicting findings underscore the need for a comprehensive evaluation of distance measures and the impact of supervised tuning. The lack of consensus on optimal distance measures and tuning procedures can lead practitioners to make suboptimal decisions.

$\mathcal{RQ}$ 3: **Do deep learning-based methods, including foundation models, outperform SOTA classical clustering methods?**

Discussion: While the prevailing consensus suggests that deep learning-based methods for time-series clustering surpass classical approaches, reflecting their success in domains such as computer vision [65, 147] and natural language processing [114, 155] the evidence supporting their superiority in time-series clustering is not definitive. Foundation models, such as large pre-trained neural networks and transformers, have introduced new possibilities for time-series clustering. Though successful in other domains, their application to time-series clustering is still largely unexplored and lacks comprehensive evaluation. Studies asserting that deep learning-based techniques surpass traditional algorithms often encounter several methodological challenges [88, 101, 103].

First, the choice of datasets, assessment metrics, and baselines can be arbitrary, lacking comprehensive evaluation and rendering the results difficult to generalize. Additionally, some studies [101] rely on baseline comparisons derived from previous research without actually re-running the original methods, which raises concerns about fairness in these settings. Others use highly tuned training parameters, such as learning rate and batch size, tailored to each dataset. Moreover, the absence of documentation for both the parameters and the tuning procedures impedes the generalizability of the approach and, in some instances, makes it impossible to reproduce the reported findings. This lack of transparency also calls into question the purportedly unsupervised nature of these methods, given the considerable degree of manual intervention involved. There is also a lack of surveys or benchmarks focusing on deep learning-based methods. Previous reviews [72, 77, 121] have predominantly excluded deep learning-based methods, leaving a gap in the comparative understanding of these approaches. The sole benchmark study on deep learning-based time-series clustering [88] reports significant performance improvements attributable to deep learning. However, despite its extensive evaluation, this study introduces too many variables to draw rigorous conclusions and overlooks a substantial portion of classical methods in its baseline comparisons. Moreover, among the classical methods selected as baselines, we identified several implementation bugs. For instance, the tslearn implementation of k-Shape is inconsistent with the original author's implementation and was erroneously applied to multivariate data by using only the first channel.

3 BACKGROUND

In this section, we review the relevant background necessary for our benchmarking study of time-series clustering methods.

Datasets: We conduct our evaluation using the UCR Time Series Archive [40], which is currently the largest publicly available collection of labeled time series datasets. The archive consists of 128 datasets collected from a diverse range of application domains, including biosignals, motion-capture data, image-based data, spectral and audio data, and device and power readings, among others. The datasets contain between 40 and 24,000 time series, with sequence lengths ranging from 15 to 2,844. All datasets are z-normalized, and each time series is associated with a single class label. A small subset of the datasets contains missing values and varying sequence lengths. Following the recommendations of the archive's authors [34], we apply linear interpolation to impute missing values and resample shorter time series.

Statistical Analysis: We used the Wilcoxon signed-rank test [14, 43, 60, 150] with a 99% confidence level to perform analysis on pairwise comparison of results over multiple datasets. To control the family-wise error rate resulting from multiple pairwise comparisons, we adjusted the obtained p-values using the Holm–Bonferroni correction [73]. This reduces the likelihood of false positives while maintaining greater statistical power. We also apply the Friedman test [56] followed by the Nemenyi test [48] with a 90% confidence level to compare results across multiple methods and datasets as pairwise testing is not always adequate since null hypotheses are rejected because of random chance.

Platform: We conduct experiments on a cluster of 3 servers with identical configuration: Dual Intel(R) Xeon(R) Platinum 8168 (96-core with 2-way SMT), 2.70 GHz, 2TB RAM. Each server has an 8 NVIDIA Tesla V100-32GB with Ubuntu 18.04.3 LTS (64-bit) system.

Implementation: We have compiled a Python library containing state-of-the-art time-series clustering approaches evaluated in our study to ensure that all the comparisons are performed under the same framework for a consistent evaluation in terms of both performance and efficiency. For reproducibility purposes, we make all datasets, source codes, and results publicly available at [1].

Experimental Settings: To ensure the robustness of our findings, each algorithm evaluated in this study was executed independently 10 times. The resulting metrics were averaged prior to reporting and subsequent statistical analysis. To facilitate a fair comparison among clustering algorithms, common parameters were consistently set across all methods, unless explicitly stated otherwise. The number of clusters k was set equal to the true number of classes in each dataset; the number of iterations was fixed at 100; and the initialization strategy was specified as ”*random*”.

Evaluation Framework: We evaluate the clustering performance using the following evaluation criteria: Rand Index (RI) [136], Adjusted Rand Index (ARI) [74] and Normalized Mutual Information (NMI) [157]. RI is a popular evaluation criterion in the literature [64, 101, 120, 151, 156, 159]. It ranges from 0 to 1, where 1 indicates perfect clustering and 0 indicates complete disagreement. However, the expected RI of two random clustering results is not constant. ARI assumes a generalized hyper-geometric distribution where ground-truth and predicted clusters are randomly chosen, while the number of clusters and objects remains constant. NMI compares ground-truth to predicted clusters by quantifying mutual information. For the remainder of the evaluation, tabular results for all methods are presented in the following format. The "Better (Adj P Val)" column indicates whether an algorithm significantly outperforms the baseline based on the Wilcoxon test, with statistical significance determined using the Holm–Bonferroni-adjusted p-values. The symbol ✔ signifies that the algorithm exhibits statistically significant improvement relative to the baseline. Conversely, ✘ denotes statistically inferior performance compared to the baseline. The "RI," "ARI," and "NMI" columns display the mean values for the Rand Index, Adjusted Rand Index, and Normalized Mutual Information across 128 datasets. The last three columns show the number of datasets where an algorithm's RI is better (" > "), equal (" = "), or worse (" < ") compared to the baseline. To assess the statistical differences in performance across multiple methods, we apply the Friedman-Nemenyi test to obtain the average rank of methods across all datasets, using the results to generate the critical difference (CD) diagrams. The solid line in the CD diagram indicates the group of methods that show no statistical significance.

Table 2: Pair-wise comparison of scalable partitional clustering algorithms with k-AVG + ED as the baseline.

Clustering Algorithm	Distance Measure	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
k-Shape	SBD	✔ (1.02e-6)	0.7335	0.2610	0.3444	86	4	38
KASBA	MSM	✘ (3.60e-3)	0.7223	0.2487	0.3345	81	0	47
k-DBA	DTW	✘ (1.00e-0)	0.6791	0.2021	0.2776	47	0	81
k-SC	STID	✘ (9.99e-1)	0.6282	0.1788	0.2492	38	0	90
k-AVG	**ED**	-	**0.7160**	**0.2152**	**0.2994**	-	-	-

4 CLASSICAL TIME-SERIES CLUSTERING

In this section, we will discuss time-series clustering methods, categorized into two approaches by data utilization, as shown in Figure 1. The first includes methods that operate on raw data, adapting algorithms with novel distance measures or centroid computations. The second approach involves feature or model based methods that transform raw sequential data into representations suitable for Euclidean space, enabling conventional clustering algorithms. Our evaluation examines five categories within the raw-data based domain: partitional, kernel, density, hierarchical, and distribution-based techniques. Additionally, we investigate three categories within the model and feature based domain: shapelet-based, semi-supervised, and model-based techniques.

4.1 Partitional Clustering

k-AVG [102] and Partition Around Medoids (PAM) [82] are two popular partitional clustering methods. k-AVG minimizes intra-cluster distances by iteratively updating centroids, computed as the average of all points in a cluster, which may not correspond to actual samples. In contrast, PAM selects representative data points—medoids—as cluster centers, ensuring that all centers are real samples. Although PAM offers robustness against outliers and accommodates a variety of distance measures beyond Euclidean, its scalability is limited in comparison to k-AVG, which demonstrates linear scalability with dataset size. Shape-based clustering

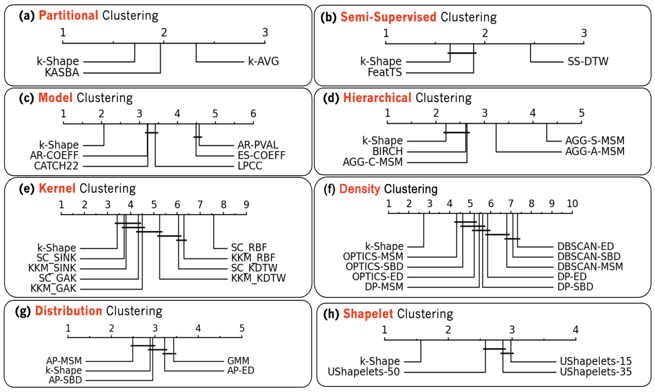


Figure 2: CD diagrams of (a) Partitional, (b) Semi-Supervised, (c) Model, (d) Hierarchical, (e) Kernel, (f) Density, (g) Distribution, and (h) Shapelet clustering algorithms based on their average ranks across datasets. The solid lines indicate groups of methods whose differences are not statistically significant.

algorithms, which utilize distance measures invariant to scaling, translation, and shifting, represent another approach within partitional clustering. Among these, k-Shape [120] emerges as the leading k-AVG-like algorithm due to its scalability and accuracy in effective distance measure and centroid computation. Additionally, k-DBA [134] extends the k-AVG by incorporating DTW as the distance measure and employing DTW Barycenter Averaging (DBA) for centroid update, offering a more representative average of sequential data sets than simple averaging. Similarly, k-SC [154] adapts k-AVG to accommodate a scaling and translation invariant distance measure (STID), further refining the centroid update through the spectral norm of a matrix. KASBA [71] extends k-AVG with MSM distance measure and an alternative elastic k-Means++ centroids. The distance calculation can be pruned using the triangle inequality, and centroids are updated via stochastic subgradient descent for elastic barycenter averaging.

Evaluation of Partitional Clustering: For the first set of experiments, we performed an analysis of several widely recognized *scalable* partitional clustering algorithms. For these comparisons, we selected k-AVG as the baseline method due to its simplicity and widespread usage in clustering benchmarks. The results, presented in Table 2, demonstrate that k-Shape and KASBA surpass the baseline algorithm, k-AVG, in 86 and 81 out of 128 datasets, respectively, whereas, k-DBA and k-SC do so on only 47 and 38 datasets, respectively. Wilcoxon test (as presented in the "Better" column of Table 2) indicates that only k-Shape significantly outperforms k-AVG. Furthermore, the Friedman-Nemenyi test from Figure 2(a) reveals that k-Shape significantly outperforms both KASBA and k-AVG.

Table 3: Pair-wise comparison of PAM across various popular distance measures using k-Shape as the baseline. For each parameter-dependent elastic measure, the first row in the "Parameters" column indicates the best parameters obtained with supervision [125], while the second row shows the unsupervised parameters.

Similarity Measure	Parameters	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
MSM	LOOCV	✘ (1.00e-0)	0.7235	0.2335	0.3563	47	0	81
	c = 0.5	✘ (1.00e-0)	0.7209	0.2311	0.3532	46	0	82
TWED	LOOCV	✘ (1.00e-0)	0.7225	0.2335	0.3567	48	0	80
	$\lambda, v = 1, 0.0001$	✘ (1.00e-0)	0.7185	0.2195	0.3429	38	0	90
ERP	-	✘ (1.00e-0)	0.7222	0.2299	0.3541	43	0	85
SBD	-	✘ (1.00e-0)	0.7173	0.2180	0.3393	37	0	91
SWALE	LOOCV	✘ (1.00e-0)	0.7089	0.2072	0.3265	44	0	84
	$\epsilon = 0.2$	✘ (1.00e-0)	0.7060	0.1927	0.3101	41	0	87
DTW	LOOCV	✘ (1.00e-0)	0.7117	0.2076	0.3378	40	0	88
	$\delta = 0.1$	✘ (1.00e-0)	0.7087	0.2008	0.3284	44	0	84
EDR	LOOCV	✘ (1.00e-0)	0.7074	0.1919	0.3099	40	0	88
	$\epsilon = 0.1$	✘ (1.00e-0)	0.7034	0.1732	0.2898	36	0	92
LCSS	LOOCV	✘ (1.00e-0)	0.7060	0.1980	0.3156	41	0	87
	$\delta, \epsilon = 5, 0.2$	✘ (1.00e-0)	0.6998	0.1637	0.2855	33	0	95
ED	-	✘ (1.00e-0)	0.7012	0.1752	0.2988	31	0	97
***k*-Shape**	-	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

Consequently, we focus our subsequent analyses on k-Shape, as it is the only method that surpasses the baseline. We have identified that the widely used k-Shape implementation from `tslearn` contains critical bugs. To address these issues, we employed the original implementation provided by the k-Shape authors to generate our results, thereby resolving reproducibility concerns observed in recent literature. Given that k-Shape outperforms all scalable partitional methods, we will adopt k-Shape as the new baseline for subsequent analyses until a new method is found that statistically outperforms k-Shape. Subsequently, we evaluated the efficacy of PAM, employing seven different elastic measures, namely, MSM [141], TWED [105], ERP [30], SWALE [112], DTW [16], EDR [31], and LCSS [5, 146] as well as the most effective lock-step and sliding measures from existing literature, Euclidean (ED) and SBD [120], respectively. We evaluated the performance of parameter-dependent elastic measures in both supervised and unsupervised settings, using parameter values drawn from [125]. Although utilizing supervised parameter selection may confer an inherent advantage, potentially bordering on bias, it was deemed necessary to fully ascertain the capabilities of these measures. From the 'Better' column in Table 3, it is evident that none of these measures, under both supervised and unsupervised settings, statistically outperform k-Shape according to the Wilcoxon test. The observed (1.00e-0) values in Table 3 can be attributed to the adjustment provided by the Holm–Bonferroni correction, which substantially reduces the likelihood of false positives while maintaining greater statistical power. Similarly, the upper portion of Figure 3(a) indicates that elastic measures employed with supervision on parameter tuning have no significant difference in performance compared to k-Shape, as indicated by the Friedman-Nemenyi test. In contrast, the lower section of Figure 3(a) provides compelling evidence of k-Shape's superior performance over these measures in unsupervised settings.

4.2 Kernel-based Clustering

Kernel k-AVG (KKM) and Spectral Clustering (SC) offer distinct advantages for identifying clusters that are non-linearly separable within the original input space. KKM leverages a kernel function to project sequences into a higher-dimensional feature space, thereby facilitating the partitioning of data that becomes linearly separable in this new space [44]. In contrast, SC employs a different approach by computing eigenvectors from the affinity matrix and subsequently using these eigenvectors for clustering the data with k-AVG [113]. Our evaluation of these kernel-based methods incorporates four prominent kernel functions. Initially, we utilize the Radial Basis Function (RBF) [36], which effectively extends the Euclidean measure to a high-dimensional space, making sequences linearly separable. We also explore a sliding kernel, SINK [122], which assesses all possible alignments between two time-series

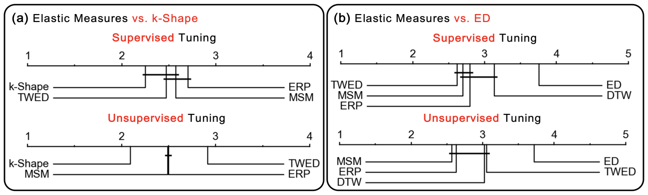


Figure 3: (a) CD diagrams of the top three Elastic measures with both supervised and unsupervised parameter settings [125], with k-Shape clustering (a strong baseline) across 128 datasets. (b) CD diagrams of the top three elastic measures with supervised and unsupervised parameter settings [125], along with DTW and ED, based on their average ranks across datasets. The solid lines indicate groups of methods whose differences are not statistically significant.

Table 4: Pair-wise comparison of Kernel-based clustering algorithms with k-Shape as the baseline.

Clustering Algorithm	Distance Measure	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
Kernel k-AVG	SINK	✘ (1.00e-0)	0.7287	0.2553	0.3461	56	0	72
	GAK	✘ (1.00e-0)	0.7119	0.2237	0.3499	42	0	86
	KDTW	✘ (1.00e-0)	0.6825	0.1020	0.2125	37	0	91
	RBF	✘ (1.00e-0)	0.6643	0.0241	0.1368	21	0	107
SC	SINK	✘ (1.00e-0)	0.7321	0.2661	0.3513	61	0	59
	GAK	✘ (1.00e-0)	0.6871	0.2421	0.3546	49	0	79
	KDTW	✘ (1.00e-0)	0.5681	0.1721	0.2896	30	0	98
	RBF	✘ (1.00e-0)	0.4863	0.0104	0.1182	14	0	114
***k*-Shape**	-	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

sequences, thus providing a nuanced analysis of their similarities. Additionally, we investigate two kernels, GAK [37] and KDTW [105], designed to extend elastic measures, offering a sophisticated means of comparing the dissimilarity of time-series data.

Evaluation of Kernel-based Clustering: From the previous section, we adopted k-Shape as the new baseline for our subsequent analyses, given its strong performance, until we encounter a new method that statistically outperforms it. Now, we focus on evaluating the performance of KKM and SC, using four representative kernel measures: RBF, SINK, GAK, and KDTW. Our findings, presented in Table 4, juxtapose the clustering efficacy of these kernel measures against k-Shape. The comparative analysis reveals that, even under supervised conditions, none of the kernel measures statistically outperform k-Shape, as determined by the Wilcoxon test. Specifically, KKM's performance with SINK, GAK, KDTW, and RBF kernels under supervised settings outperforms k-Shape in 56, 42, 37, and 21 instances, respectively. Similarly, SC's efficacy with the same kernels outperforms k-Shape in 61, 49, 30, and 14 datasets, respectively. Poor performance of the RBF kernel underscores that Euclidean based measures underperform relative to elastic or alignment-based similarities in time-series clustering. As depicted in Figure 2(e), the Friedman-Nemenyi test suggests that none of the kernel-based methods outperform k-Shape.

4.3 Density-based Clustering

DBSCAN [51] is a leading density-based clustering algorithm that identifies densely packed, non-spherical clusters while isolating sparse outliers. It relies on the concept of reachability, wherein clusters are expanded by including neighboring data points within a specified radius. OPTICS [6] extends DBSCAN by producing an ordered traversal of points to detect clusters across regions of varying density; however, both methods require careful selection of parameters such as the neighborhood radius and minimum point count. In contrast, the Density Peaks algorithm [137] obviates explicit parameter specification by selecting cluster centers whose neighbors have lower local density and are maximally distant from points of higher density, then assigning remaining points to their nearest high-density neighbor.

Table 5: Pair-wise comparison of Hierarchical clustering algorithms with k-Shape as the baseline.

Clustering Algorithm	Distance Measure	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
BIRCH	-	✘ (1.00-e0)	0.7123	0.2305	0.3483	47	0	81
AGG-C	MSM	✘ (1.00-e0)	0.7058	0.2415	0.3712	51	0	77
	SBD	✘ (1.00-e0)	0.6828	0.1962	0.3370	35	0	93
	ED	✘ (1.00-e0)	0.6820	0.1705	0.3006	32	0	96
AGG-A	MSM	✘ (1.00-e0)	0.6450	0.2152	0.3515	36	0	92
	SBD	✘ (1.00-e0)	0.6210	0.1577	0.2909	29	1	99
	ED	✘ (1.00-e0)	0.5959	0.1584	0.3010	30	0	98
AGG-S	MSM	✘ (1.00-e0)	0.4421	0.0841	0.1859	21	0	107
	SBD	✘ (1.00-e0)	0.4222	0.0660	0.1638	18	0	110
	ED	✘ (1.00-e0)	0.4222	0.0635	0.1600	18	0	110
***k*-Shape**	-	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

Evaluation of Density-based Clustering: For this set of experiments, we evaluate the performance of various density-based clustering methods. For this evaluation, we incorporated the ED measure alongside two of the most effective sliding and elastic measures, SBD and MSM, identified as top performers in Section 4.1. To ensure a fair comparison: minPts was chosen via grid search over {5, 10, 15}; the ε parameter in DBSCAN and the distance-cutoff parameter in Density Peaks were both determined using knee-point detection. Although density-based approaches are designed to robustly handle outliers, none of these methods consistently outperformed k-Shape on more than 30 datasets, thereby underscoring the superior performance of k-Shape. Moreover, as evidenced by the Friedman-Nemenyi test shown in Figure 2(f), k-Shape significantly outperforms density methods.

4.4 Hierarchical Clustering

In our evaluation study, we selected Agglomerative Clustering (AGG) [82] and BIRCH [160] as representative hierarchical clustering methods due to their widespread adoption. AGG employs a bottom-up strategy, initially treating each sequence within the dataset as a cluster. This approach progressively merges clusters based on their similarity, culminating in a unified cluster that encompasses all sequences. To ascertain the proximity between clusters, we employed recognized linkage criteria: single, average, and complete linkage. BIRCH is notable for its scalability and robustness in managing outliers. It constructs a cluster-feature tree that encapsulates the data's essential cluster configurations while optimizing memory usage. Subsequent global clustering leverages summaries derived from the cluster-feature tree, employing agglomerative clustering techniques to ultimately achieve comprehensive clustering.

Evaluation of Hierarchical Clustering: Akin to the evaluation from the previous section, we performed an assessment of AGG methods using the ED, SBD, and MSM measures. Our findings reveal a clear performance hierarchy among the agglomerative clustering strategies. Specifically, complete linkage consistently outperformed average linkage, which in turn showed significant superiority over single linkage across all evaluated distance measures. As summarized in Table 5, even the best-performing hierarchical methods, such as the agglomerative clustering with complete linkage, only

Table 6: Pair-wise comparison of Distribution-based clustering algorithms with *k*-Shape as the baseline.

Clustering Algorithm	Distance Measure	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
AP	MSM	✗ (6.85-e1)	0.7289	0.2204	0.4269	70	0	58
	SBD	✗ (1.00-e0)	0.7284	0.2180	0.4001	66	0	62
	ED	✗ (1.00-e0)	0.7137	0.1662	0.3731	57	0	71
GMM	-	✗ (1.00-e0)	0.7165	0.2193	0.3067	49	0	79
***k*-Shape**	**-**	**-**	**0.7335**	**0.2610**	**0.3444**	-	-	-

outperformed *k*-Shape in 51 out of the 128 datasets. The "Better" column in Table 5 and the corresponding Figure 2(d) provide further statistical evidence from Wilcoxon and Friedman-Nemenyi tests that none of the hierarchical methods outperform *k*-Shape.

4.5 Distribution based Clustering

Affinity Propagation (AP) [55] identifies exemplars within datasets, around which clusters of data points are formed. Furthermore, this process entails treating all data points as potential exemplars and facilitating a message exchange among them until a consensus on the exemplars is reached. Gaussian Mixture Models (GMMs), as discussed in the works of [42, 142], offer a sophisticated framework for distribution-based clustering. Moreover, GMM posits that the data are generated from a finite mixture of Gaussian distributions, each corresponding to a cluster. These distributions, characterized by unknown parameters, are unraveled through the Expectation-Maximization algorithm, enabling the clustering of unlabeled data by estimating the parameters of the Gaussian mixture.

Evaluation of Distribution based Clustering: In line with our previous analysis, we selected the ED, SBD, and MSM measures. Our findings, detailed in Table 6, indicate that AP using MSM, SBD, and ED, as well as GMM, outperforms *k*-Shape clustering in 70, 66, 57, and 49 out of 128 datasets, respectively. However, despite the robust performance of the AP methods, statistical analysis using the Wilcoxon test and the Friedman-Nemenyi test indicates that none of the distribution-based methods decisively outperform *k*-Shape.

Table 7: Pair-wise comparison of Shapelet-based clustering algorithms with *k*-Shape as the baseline. An asterisk (*) indicates that some methods are evaluated on a subset of datasets due to unfeasible runtimes.

Clustering Algorithm	Shapelet Length	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
UShapelet	50%	✗ (1.00e-0)	0.5718	0.1510	0.2385	26	0	102
	35%	✗ (1.00e-0)	0.5227	0.1081	0.2014	24	0	104
	15%	✗ (1.00e-0)	0.4976	0.0885	0.1715	23	0	105
***k*-Shape**	-	-	**0.7335**	**0.2610**	**0.3444**	-	-	-
*LDPS**	-	-	0.6849	0.2835	0.3248	-	-	-
*USLM**	-	-	0.5008	0.1213	0.1532	-	-	-
k*-Shape	-	-	**0.6925**	**0.2943**	**0.3420**	-	-	-

Table 8: Pair-wise comparison of Semi-Supervised clustering algorithms with *k*-Shape as the baseline.

Clustering Algorithm	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
FeatTS	✗ (1.00-e0)	0.7203	0.2823	0.3229	59	0	69
SS-DTW	✗ (1.00-e0)	0.6307	0.1383	0.2427	28	1	99
***k*-Shape**	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

4.6 Shapelet and Semi-Supervised Clustering

Unlike methods that use entire time-series sequences for clustering, U-Shapelets [156] utilize subsequences with pronounced patterns, shapelets, to identify outliers. However, this approach is confined to smaller datasets and does not scale with increasing data size. To mitigate this limitation, we employ a scalable U-Shapelets variant [145], which sustains clustering quality without significant compromise. Dynamic Time Warping (DTW) is renowned for its precision in quantifying similarity between time-series sequences, yet its application is hindered by the high computational cost associated with longer sequences. The Learning DTW Preserving Shapelets (LDPS) [99] framework approximates DTW distance using ED between shapelets, preserving the integrity of the original sequences. Additionally, the Unsupervised Shapelet Learning Model (USLM) [158] presents a shapelet-based clustering algorithm that employs an iterative learning process, leveraging pseudo-labels, spectral analysis, shapelet regularization, and regularized least-squares to derive shapelets and define decision boundaries effectively. In the semi-supervised category, FeatTS [144] leverages graph encoding and community detection to construct a co-occurrence matrix from extracted statistical features. SS-DTW [41] addresses the critical task of selecting an optimal DTW warping-window width, a parameter essential for clustering performance across diverse domains of datasets.

Evaluation of Shapelet-based and Semi-Supervised Clustering: We conducted an evaluation of the UShapelet model using three distinct shapelet lengths: 50%, 35%, and 15%. The results, summarized in Table 7, indicate that the U-Shapelet methods do not surpass the performance of the *k*-Shape algorithm in more than 26 of the 128 datasets. Other shapelet-based clustering techniques, such as USLM and LDPS, were evaluated on a smaller subset of 25 datasets due to prohibitive computational runtimes. Furthermore, both the "Better" column in Table 7 and Figure 2 (h) provide compelling evidence of *k*-Shape's superiority over shapelet-based clustering methods, as demonstrated by Wilcoxon and Friedman-Nemenyi statistical tests. Shapelet-based clustering, although conceptually promising, as it captures localized patterns, did not yield robust results. One possible explanation is that the unsupervised discovery of shapelets is inherently challenging and susceptible to overfitting noise or irrelevant patterns. Within the semi-supervised clustering category, as shown in Table 8, the FeatTS method emerged as the most effective, outperforming *k*-Shape in 59 of the 128 datasets. However, as illustrated in Figure 2(b), while FeatTS significantly outperforms SS-DTW, it demonstrates no substantial performance difference when compared with the *k*-Shape approach. For both categories, all parameters were set as in original implementations.

4.7 Model and Feature based Clustering:

Model-based clustering assumes each sequence in a cluster is generated by a model following a probability distribution. Piccola [135] introduced complete linkage agglomerative clustering on autoregressive coefficients using Euclidean similarity. Kalpakis [81] applied *k*-Medoids clustering to Linear Predictive Coding Cepstra (LPCC), employing Euclidean similarity derived from autoregressive coefficients. This approach, emphasizing cepstral coefficients, distinguishes time series more effectively than coefficients from the Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), or Principal Component Analysis (PCA). The Chi-Square

Table 9: Pair-wise comparison of Model-based clustering algorithms with k-Shape as the baseline.

Clustering Algorithm	Parameters	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
k-AVG	AR - COEFF	✘ (1.00e-0)	0.6885	0.1159	0.1881	32	0	96
	Catch22	✘ (1.00e-0)	0.6870	0.1409	0.2247	29	0	99
	LPCC	✘ (1.00e-0)	0.6851	0.1126	0.1820	33	0	95
	AR - P VAL	✘ (1.00e-0)	0.6502	0.0489	0.1135	17	0	111
	ES - COEFF	✘ (1.00e-0)	0.5839	0.0803	0.1557	26	0	102
k-Shape	-	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

test assesses significant differences between stationary time series. Maharaj [104] proposed agglomerative hierarchical clustering based on Chi-Square p-values. A major limitation of model-based methods is their reliance on assumptions that may not hold, limiting applicability. catch22 [100] selects 22 features from the hctsa suite's 4,791 features, capturing diverse time-series characteristics and providing a feasible solution for time-series signatures.

Evaluation of Model and Feature based Clustering: Techniques such as AR-COEFF, LPCC, AR - P VAL, and ES-COEFF utilize the coefficients from various time-series modeling methods for clustering, whereas Catch22 employs 22 meticulously selected time-series features for clustering purposes. Table 9 reveals that none of these methods outperform k-Shape in more than 33 out of 128 datasets. Similarly, "Better" column in Table 9 and Figure 2 (c) indicate that k-Shape statistically outperforms all model-based time-series clustering methods.

4.8 Addressing $\mathcal{RQ}1$ and $\mathcal{RQ}2$

The results of our comprehensive evaluation of classical time-series clustering methods challenge several claims in the literature. For example, [72] report that k-Shape does not outperform k-AVG, yet Section 4.1 demonstrates this finding arises from unfair parameter settings (as discussed in Section 2). Similarly, [77, 88] reported underwhelming k-Shape performance in the literature, largely due to bugs in the tslearn implementation. However, using the original implementation shows that k-Shape, a decade-old approach, still outperforms all scalable partitional methods, establishing it as a robust baseline. Employing k-Shape as our benchmark, we evaluated other classical methods and found that none significantly surpassed it, thus providing critical insights into $\mathcal{RQ}1$.

Addressing $\mathcal{RQ}2$, our empirical results provide key insights that help resolve some ambiguities regarding distance measures in time-series clustering. Contrary to [72]'s assertion of the inefficacy of DTW compared to ED, our empirical evaluation (see Figure 3(b)) provides compelling evidence that DTW consistently outperforms ED with statistical significance across both supervised and unsupervised settings. This finding effectively addresses the first part of $\mathcal{RQ}2$. The choice of parameters for parameter-dependent distance measures, such as MSM, TWED, SWALE, DTW, EDR, LCSS, SINK, GAK, KDTW, and RBF, is arbitrary in the literature. From Figure 3(a), we observe that, in the unsupervised setting, k-Shape outperforms the top elastic measure with statistical significance; in the supervised setting, there is no significant difference in performance. Likewise, TWED outranks all other elastic measures in the supervised context, whereas in the unsupervised setting it is outranked by parameter-free measures such as ERP. Similarly, kernel measures perform better in supervised than unsupervised settings. These results underscore the variability of distance-measure efficacy across contexts. It is notable that the right parameter choice significantly influences distance-measure performance. Hence, there is a serious need to explore methods that can determine accurate parameters in an unsupervised fashion. Therefore, there is clear evidence that parameter selection is crucial for parameter-dependent measures to reach fullest potential, thus addressing the second part of $\mathcal{RQ}2$.

Table 10: Pair-wise comparison of Deep Learning-based clustering algorithms with k-Shape as the baseline.

Clustering Algorithm	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
IDEC	✘ (1.00-e0)	0.7159	0.2150	0.2967	46	0	82
DEPICT	✘ (1.00-e0)	0.7111	0.1900	0.2743	42	0	86
SDCN	✘ (1.00-e0)	0.7104	0.2000	0.2884	45	0	83
DEC	✘ (1.00-e0)	0.7090	0.1935	0.2790	43	0	85
DTC	✘ (1.00-e0)	0.7085	0.2123	0.2985	43	1	85
ClusterGAN	✘ (1.00-e0)	0.7082	0.2100	0.2965	41	0	87
VADE	✘ (1.00-e0)	0.7027	0.1734	0.2605	33	0	95
DTCR	✘ (1.00-e0)	0.6832	0.1392	0.2184	28	0	100
SOM-VAE	✘ (1.00-e0)	0.6457	0.0976	0.1804	21	0	107
DCN	✘ (1.00-e0)	0.5716	0.0444	0.1097	15	0	113
k-Shape	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

5 DEEP LEARNING BASED TIME-SERIES CLUSTERING

This section presents our investigation into deep learning-based clustering methods for time series, structured in two parts, as shown in Figure 1. First, we examine the efficacy of deep learning models, including foundation models, for time-series clustering in the literature. Then, we perform a comparative analysis to evaluate the impact of architectural elements and loss function choices on model performance. This approach offers a detailed understanding of how individual components affect overall clustering performance.

5.1 Existing Work

Most deep learning-based time-series clustering algorithms have predominantly adhered to a structured methodology, with minor variations in handling representation vectors. DCN [152] jointly optimizes representation learning via a deep autoencoder and k-Means clustering. DEC [151] iteratively optimizes a Kullback–Leibler divergence objective on an auxiliary target distribution to refine cluster assignments. IDEC [64] extends DEC by jointly optimizing clustering and autoencoder reconstruction losses, preserving local data structure. DTCR [101] integrates the k-Means objective with temporal reconstruction and an auxiliary classification task that distinguishes genuine and synthetic samples, enriching encoder representations with contextual depth. DTC [103] expanded on [64] by incorporating a more sophisticated autoencoder and various temporal similarity metrics to assess the Kullback-Leibler divergence between predicted and target distributions. SOM-VAE [52] introduced a framework employing a gradient-based self-organizing map alongside a Markov model, enabling discrete representation of time series sequences and probabilistic interpretation of temporal transitions. DEPICT [58] extends [64] by applying multi-reconstruction for pretext loss and cross-entropy for clustering loss, whereas SDCN [17] integrates a graph convolutional network to preserve local neighborhood structures within the latent space. VADE [80] extends variational autoencoders by learning distinct distributions for each cluster, and ClusterGAN [57] introduces a GAN-based framework that incorporates a clusterer network to

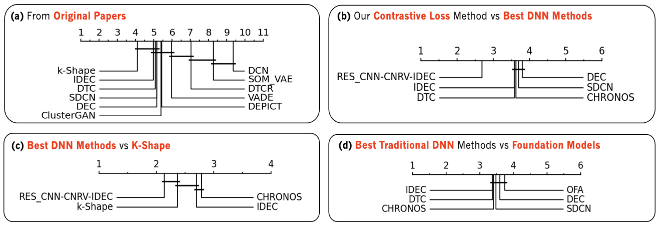


Figure 4: (a) CD diagrams of deep learning models proposed in literature. (b) CD diagrams of the top five deep learning models from literature, along with our proposed contrastive approach. (c) CD diagrams of the best deep learning models in our evaluation, versus *k*-Shape. (d) CD diagrams of the top three deep learning models from literature, along with foundation models. The solid lines indicate groups of methods whose differences are not statistically significant.

generate latent features and a discriminator to differentiate the joint distribution of samples and features.

A new class of deep learning models, termed foundation models, leverages large-scale time-series data to pretrain transformer architectures akin to large language and vision models. CHRONOS [7] is a framework for pretrained probabilistic time series models, tokenizing time-series values through scaling and quantization into a fixed vocabulary, and using transformer-based language models with cross-entropy loss. OFA [161] leverages Frozen Pretrained Transformers where pretrained language and computer vision models are adapted to time series tasks without altering self-attention and feedforward layers. MOMENT [62] segments a time series into fixed-length patches, mapping each to a D-dimensional embedding. During pretraining, patches are randomly masked and replaced by a MASK embedding, aiming to learn patch embeddings that enable accurate time series reconstruction with a lightweight head.

Evaluation of Existing Work: Now, we present a comparative analysis of clustering assessment metrics between deep learning-based clustering methods from the literature and *k*-Shape, a scalable partitional clustering method that has demonstrated superior performance over other classical methods, as detailed in Section 4. From Table 10, we pick five top-performing deep learning models: IDEC, DEPICT, SDCN, DEC, and DTC to compare against *k*-Shape. These models surpass *k*-Shape only on 46, 42, 45, 43, and 43 out of 128 datasets, respectively. Results from the Wilcoxon test in the "Better" column show that all our deep learning-based clustering baselines are significantly worse than *k*-Shape. The findings in Figure 4(a), according to the Friedman–Nemenyi test, none of the deep learning methods reported in the literature outperform *k*-Shape with statistical significance, aligning with our results in the table. A critical risk with foundation models is the potential overlap between pretraining and evaluation data. Our analysis from Table 11 reveals that, although the MOMENT model demonstrates robust performance, its pretraining on the UCR dataset compromises the integrity of the evaluation. Accordingly, MOMENT has been excluded from further consideration. By contrast, neither CHRONOS nor OFA incurs this form of data contamination; nevertheless, they each outperform the *k*-Shape baseline on only 52 and 48 datasets, respectively. Similarly, Wilcoxon test in Table 11 and Friedman-Nemenyi test indicate that none of the foundation models outperform *k*-Shape. Furthermore, as depicted in, Figure 4(d), the Friedman-Nemenyi test comparing foundation models with existing deep learning time-series models show that there is no significant difference in performance between them. Our findings indicate that, despite the advancements in deep learning for time-series clustering, none of these methods substantially surpass traditional models like *k*-Shape across all metrics and statistical tests. Unlike previous studies that report impressive results, our findings suggest a different outcome. Moreover, these studies have often used *k*-Shape implementations from popular libraries like tslearn, known to contain bugs affecting performance. We attribute the relatively low performance of deep learning methods to their predominantly singular structures derived from general-purpose clustering, often lacking adaptation to time series.

Table 11: Pair-wise comparison of Foundation Models for clustering algorithms with *k*-Shape as the baseline. An asterisk (*) denotes Foundation Models that utilized the UCR dataset in training.

Clustering Algorithm	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
MOMENT*	✘ (1.00e-0)	0.7304	0.2551	0.3436	66	1	61
CHRONOS	✘ (1.00e-0)	0.7172	0.2066	0.2925	47	0	81
OFA	✘ (1.00e-0)	0.7103	0.1949	0.2817	38	0	90
***k*-Shape**	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

Table 12: Comparison of Pretext and Clustering Loss combinations.

			Pretext Loss				
			CNRV	MREC	REC	VAE	TRPLT
Clustering Loss	**IDEC**	RI	0.7337	0.7209	0.7187	0.7007	0.6908
		ARI	0.2565	0.2480	0.2331	0.2236	0.1969
		NMI	0.3709	0.3366	0.3194	0.3012	0.2821
	None	RI	0.7393	0.7183	0.7181	0.7174	0.6901
		ARI	0.2702	0.2474	0.2319	0.2367	0.1813
		NMI	0.3832	0.3389	0.3184	0.2579	0.2921

5.2 Comparative Study

Thus far, we have employed standalone deep learning methods as recommended in the literature, following their suggested parameters. However, each method comprises several distinct components and it is hard to discern the individual contributions of loss function and architectural design choices. To address this complexity, we adopted and extended the insightful analysis of [88], categorizing the deep learning-based approaches into three primary components: neural network architecture, pretext loss (including the introduction of a new contrastive loss) and clustering loss.

• **Neural Network Architecture:** We implemented autoencoder models from three popular classes of neural network architectures: **(a) Fully Connected Network (FCN):** We have implemented a Multi-Layer Perceptron (MLP) [138] model. **(b) Recurrent Neural Network (RNN):** We implemented Bi-Directional RNN (BI-RNN) [140], BI-Gated Recurrent Unit (BI-GRU) [101], Dilated-RNN [29], BI-Long Short Term Memory [101], and BI-RNN + Attention (BI-RNN + ATTN) [76] models. **(c) Convolutional Neural Network (CNN):** We implemented Simple CNN (S-CNN) [89], Dilated CNN (D-CNN) [53], and Residual CNN (RES-CNN) [66] models.

• **Pretext Loss:** We utilize five pretext losses from the literature. **(a) Reconstruction Loss (REC)** [68]: It is a loss function for training an autoencoder and is computed by the mean squared error between the input and the reconstructed output. **(b) Multi-Reconstruction Loss (MREC)** [68]: It is an extension of reconstruction loss where the autoencoder network is required to have symmetry between

Table 13: Pair-wise comparison of Deep Learning Architectures.

Model Architecture	Architecture Type	Clustering Algorithm	RI	ARI	NMI
RES-CNN	**Convolution**	**REC + None**	**0.7201**	**0.2359**	**0.3207**
S-CNN	Convolution	REC + None	0.7122	0.2262	0.3164
MLP	**Fully Connected**	**REC + None**	**0.7102**	**0.2234**	**0.3089**
BI-RNN	**Recurrent**	**REC + None**	**0.7060**	**0.1983**	**0.2877**
BI-GRU	Recurrent	REC + None	0.6959	0.1818	0.2681
BI-GRU + ATTN	Recurrent	REC + None	0.6935	0.1916	0.2763
D-CNN	Convolution	REC + None	0.6845	0.2282	0.3115
D-RNN	Recurrent	REC + None	0.6823	0.1752	0.2526
BI-LSTM	Recurrent	REC + None	0.6687	0.1781	0.2831

Table 14: Pair-wise comparison of Architectural, Pretext Loss, and Clustering Loss combinations, with k-Shape as the baseline.

Clustering Algorithm	Model Architecture	Better (Adj. P Val)	RI	ARI	NMI	>	=	<
CNRV + NONE	RES-CNN	✘ (1.00e-0)	0.7393	0.2702	0.3832	64	0	64
CNRV + IDEC	RES-CNN	✘ (1.00e-0)	0.7337	0.2565	0.3709	55	0	73
MREC + IDEC	RES-CNN	✘ (1.00e-0)	0.7209	0.2480	0.3366	48	0	80
REC + IDEC	RES-CNN	✘ (1.00e-0)	0.7187	0.2331	0.3194	52	0	76
MREC + NONE	RES-CNN	✘ (1.00e-0)	0.7183	0.2474	0.3389	54	0	74
REC + NONE	RES-CNN	✘ (1.00e-0)	0.7181	0.2319	0.3184	53	0	75
CNRV + NONE	FCN	✘ (1.00e-0)	0.7201	0.1994	0.3191	56	0	72
CNRV + IDEC	FCN	✘ (1.00e-0)	0.7198	0.1932	0.3094	54	0	74
MREC + IDEC	FCN	✘ (1.00e-0)	0.7110	0.2227	0.3083	55	0	73
REC + IDEC	FCN	✘ (1.00e-0)	0.7098	0.2218	0.3075	52	0	76
MREC + NONE	FCN	✘ (1.00e-0)	0.7089	0.2205	0.3062	52	0	76
REC + NONE	FCN	✘ (1.00e-0)	0.7102	0.2215	0.3069	55	0	73
CNRV + NONE	BI-RNN	✘ (1.00e-0)	0.7187	0.2026	0.3182	51	0	77
CNRV + IDEC	BI-RNN	✘ (1.00e-0)	0.7147	0.1918	0.3080	46	0	82
MREC + IDEC	BI-RNN	✘ (1.00e-0)	0.6959	0.1830	0.2692	48	0	80
REC + IDEC	BI-RNN	✘ (1.00e-0)	0.6956	0.1825	0.2681	37	0	91
MREC + NONE	BI-RNN	✘ (1.00e-0)	0.6967	0.1846	0.2700	37	0	91
REC + NONE	BI-RNN	✘ (1.00e-0)	0.6951	0.1811	0.2667	39	0	89
***k*-Shape**	-	-	**0.7335**	**0.2610**	**0.3444**	-	-	-

encoder and decoder, and the mean squared error is computed between each layer in the decoder and the corresponding reflected layer in the encoder. **(c) Variational Autoencoder Loss (VAE)** [85]: Unlike regular autoencoder model, variational autoencoder maps input data to a multivariate latent distribution. Encoder and decoder are trained jointly to minimize reconstruction loss and converge expected and observed distributions. **(d) Triplet Loss (TRPLT)** [139]: It is a supervised method that pulls encoder representations from same class closer to the input while pushing away representations from other classes. Triplet loss in clustering utilizes a time-based sampling strategy to generate same and different classes samples in an unsupervised manner. **(e) Contrastive Loss (CNRV)** [33]: It learns representations by maximizing agreement between the input time-series and closest time-series to the input in the dataset by Euclidean (ED) metric.

- **Clustering Loss:** We employ seven popular clustering losses from the literature **(a) DEC** [151]: It improves object assignment confidence to its cluster using KL divergence between the soft assignment distribution and a target distribution from current cluster assignments. **(b) IDEC** [64]: An extension of DEC, combining reconstruction loss and KL divergence, with weight γ set to 0.1. **(c) DEPICT** [58]: It extends the IDEC with multi-reconstruction loss for pretext phase and standard cross-entropy loss for clustering phase. **(d) SDCN** [17]: It trains graph convolutional networks with the encoder to preserve local data relations using DTW distance based KNN-graph. **(e) VADE** [80]: It modifies variational autoencoder loss to better fit the clustering task by learning K (number of clusters) expected and observed distributions for each cluster. **(f) DTCR** [101]: It is a weighted combination of three training objectives where the first component is reconstruction loss, second component is K-Means loss and the final component is an auxiliary classification loss that can identify real and fake time-series data. **(g) ClusterGAN** [57]: It modifies regular GAN by adding an additional clusterer network C along with a discriminator network D and a generator network G. The clusterer network $C : x \longrightarrow \hat{z}$ generates the representation vectors from real input data and the generator network $G : z \longrightarrow \hat{x}$ generates the realistic input data from representation vectors. The discriminator network discriminates to identify if the joint distributions of samples and features $(C(x), x)$ and $(z, G(z))$ belong to generator or clusterer.

Evaluation of Comparative Study: We conduct a comparative analysis to elucidate performance variances among components of deep learning models for time-series clustering. Our evaluation focuses on three elements: architectures, pretext losses, and clustering losses. First, we assess the performance of various architectures using reconstruction loss as the pretext loss without any clustering loss. The findings, summarized in Table 13, indicate that convolution-based architectures, particularly the RES-CNN, surpass others in performance. Among fully-connected and recurrent architectures, the MLP and BI-RNN emerge as top performers, respectively. However, Friedman-Nemenyi test finds no statistically significant differences across architectures. In the subsequent phase, we investigate the effectiveness of various pretext losses in conjunction with IDEC and None clustering losses, employing the RES-CNN architecture. The findings, as detailed in Table 12, identify CNRV, MREC, and REC as the most effective pretext losses for both clustering frameworks. Using Friedman-Nemenyi tests, we gather compelling evidence that CNRV significantly outperforms the other pretext losses, while MREC and REC exhibit similar performance levels with no substantial differences. We select the best architectures, pretext losses, and clustering losses based on prior findings. The RES-CNN, MLP, and BI-RNN are designated as the premier models across convolutional, fully-connected, and recurrent architectural categories, respectively. Among pretext losses, CNRV, MREC, and REC are recognized for their superior performance. For clustering losses, both IDEC and None are identified as the most effective. An examination of these selected components, as depicted in Table 14, unveils intriguing observations. RES-CNN consistently outperforms other architectures across all pretext and clustering loss combinations. In the context of pretext losses, CNRV consistently outperforms alternatives, improving performance irrespective of architecture or clustering loss. Figure 4(b) shows that combining RES-CNN with CNRV and None clustering loss yields statistically significant improvements over other deep learning-based methods from literature. Finally, comparing IDEC and None clustering losses reveals no significant performance difference.

5.3 Addressing $\mathcal{RQ}3$

The perceived advancements within the research community regarding the impact of deep learning on time-series clustering appear somewhat illusory. Many studies that introduce deep learning-based methods for time-series clustering typically adapt general-purpose deep clustering models or extend existing frameworks.

Additionally, there are significant issues with the evaluation frameworks used in these studies. For example, DTCR uses only 36 datasets. Additionally, DTCR uses only a few baselines and relies on baseline results borrowed from various studies, which carries the risk of adopting potential implementation flaws from those earlier works. Similarly, DTC's evaluation is limited to only 13 datasets and relies on the ROC metric, which may not be the most suitable choice for clustering tasks. Additionally, DTC compares against just two baselines and lacks statistical testing, which considerably weakens the reliability of its results. SOM-VAE utilizes image data instead of time-series data. Similar to the other methods, it does not include statistical testing and overlooks several important techniques from the literature in its baseline comparisons. The absence of thorough statistical testing raises questions about whether the proposed advancements are statistically significant when compared against both classical state-of-the-art methods and earlier deep learning methods. Our analysis, as illustrated in Figure 4(a), demonstrates that the performance differences among deep learning-based clustering models are negligible, with none surpassing k-Shape.

A recent comparative study [88] exhaustively evaluates the core components of deep learning-based time-series clustering methods, but we identified several concerns. It introduces too many variables, making it difficult to draw rigorous conclusions, and fails to include foundational models or many classical methods in its baseline comparisons. We also discovered multiple implementation errors among the baselines; for example, the tslearn implementation of the k-Shape algorithm—deviating from the original author's implementation was incorrectly applied. In response, our study aims for a more streamlined evaluation, focusing on the architectural decisions, pretext loss, and clustering loss choices to provide clarity. Our findings, detailed in Figure 4(b), identify contrastive learning based models like RES-CNN architecture combined with CNRV pretext loss and NONE clustering loss as the sole configuration demonstrating superior performance over other deep learning-based methods documented in the literature. We attribute this to its ability to adeptly integrate time-series domain features to optimize contrastive loss. However, from Figure 4(c), we find that even this model does not exhibit a significant performance difference when compared to the k-Shape. Similarly, while foundation models have demonstrated exceptional performance in image and language tasks, they have failed to outperform classical methods in the time-series domain. This discrepancy prompts a critical reevaluation of the application of deep learning techniques to time-series data, suggesting the need for domain-specific adaptations rather than the uncritical replication of architectures designed for image and language domains. Therefore, the experiments demonstrate that deep learning-based time-series clustering methods, including foundation models, as described in existing literature, do not statistically surpass the performance of established classical methods. This insight critically addresses $\mathcal{RQ}3$, shedding light on the actual impact of deep learning innovations in this field.

6 EXPERIMENTAL ANALYSIS

We present a detailed analysis of each clustering method's performance under various conditions, including accuracy-to–runtime trade-offs (Section 6.1), sensitivity to data distribution characteristics (Section 6.2), and scalability on large-scale datasets (Section 6.3).

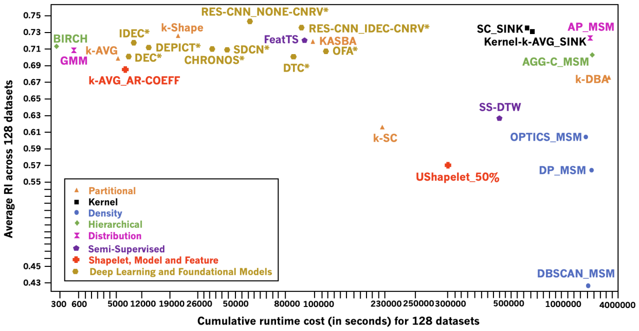


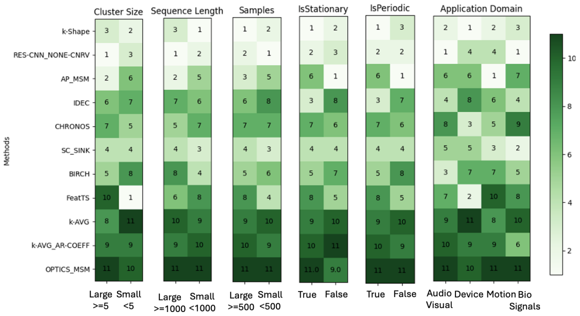
Figure 5: Performance-to-runtime comparison.

6.1 Accuracy-to-Runtime Analysis

We perform an in-depth analysis of the runtime expenses associated with various time-series clustering algorithms, juxtaposed with their performance metrics. Figure 5 presents the runtime costs of key algorithms explored in this study, as reported in the existing literature. We define runtime performance as the cumulative duration required by a method for fitting and inference. Our analysis reveals that BIRCH emerges as the fastest algorithm, demonstrating commendable efficiency while maintaining satisfactory clustering performance. BIRCH's memory-efficient, online-learning optimization for large-scale datasets contrasts sharply with the substantially longer runtimes of other hierarchical methods. Furthermore, we identify k-Shape as the sole traditional algorithm that offers an advantageous balance between runtime efficiency and clustering efficacy. Deep learning based approaches also show potential by leveraging GPU acceleration to achieve speed and accuracy improvements unattainable by CPU only processing. For example, fitting models such as DEC and IDEC on a CPU requires up to five and eight times longer, respectively. Despite these gains, no deep learning method significantly surpasses classical strategies like k-Shape. However, contrastive methods like RES-CNN framework, using CNRV pretext loss and NONE clustering loss, show promising results, albeit with increased runtime. It is worth noting that runtime efficiencies for our contrastive method could be improved through strategic sampling of positive and negative samples.

6.2 Data Distribution Analysis

In our evaluation of the full UCR time-series archive (128 univariate datasets), we partitioned the data by several characteristics: cluster size (small: < 5 vs. large: ≥ 5), number of samples (small: $< 1{,}000$ vs. large: $\geq 1{,}000$), sequence length (small: < 500 vs. large: ≥ 500), stationarity, periodicity, and application domains. For each subset, we computed the rank for each algorithm. Periodicity, are assessed using periodogram analysis [83] and autocorrelation function tests [26], while stationarity is assessed using the augmented Dickey–Fuller test [46] and the KPSS test [87]. Figure 6 illustrates the resulting average-rank performance over 128 datasets. Overall, the k-Shape algorithm and the proposed contrastive learning method consistently achieved the best ranks across nearly all conditions, demonstrating robust performance. However, the analysis also revealed notable deviations: the feature-driven method FeatTS attained the best rank on datasets with fewer than five clusters (reflecting its graph-based encoding of cluster structure).



Methods	Cluster Size: Large >=5	Cluster Size: Small <5	Sequence Length: Large >=1000	Sequence Length: Small <1000	Samples: Large >=500	Samples: Small <500	IsStationary: True	IsStationary: False	IsPeriodic: True	IsPeriodic: False	Application Domain: Audio Visual	Application Domain: Device	Application Domain: Motion	Application Domain: Bio Signals
k-Shape	3	2	3	1	1	2	1	2	1	3	2	1	2	3
RES-CNN_NONE-CNRV	1	3	1	2	2	1	2	3	2	2	1	4	4	1
AP_MSM	2	6	2	5	3	5	6	1	6	1	6	6	1	7
IDEC	6	7	7	6	6	8	3	8	3	7	4	8	6	4
CHRONOS	7	5	5	7	7	7	7	6	7	6	8	3	5	9
SC_SINK	4	4	4	3	4	3	4	4	4	4	5	5	3	2
BIRCH	5	8	8	4	5	6	5	7	5	8	3	7	7	5
FeatTS	10	1	6	8	8	4	8	5	8	5	7	2	10	8
k-AVG	8	11	10	9	9	10	9	10	9	10	9	11	8	10
k-AVG_AR-COEFF	9	9	9	10	10	9	10	11	10	9	10	9	9	6
OPTICS_MSM	11	10	11	11	11	11	11.0	9.0	11	11	11	10	11	11

Figure 6: Average rank of the ten best-performing time-series clustering methods (along with k-AVG baseline), one for each category, evaluated across varying dataset characteristics: cluster size, number of samples, sequence length, stationarity, periodicity, and application domain. A lighter color indicates a superior performance.

6.3 Scalability Analysis

A comprehensive scalability evaluation of the clustering methods was conducted, with results summarized in Figure 7. Experiments systematically varied both the length of individual time series and the number of samples, measuring runtime (in seconds) for each method. Both axes in the figures are plotted on a logarithmic scale to illustrate performance trends across diverse data sizes. Synthetic data generated in the style of the UCR Cylinder-Bell-Funnel (CBF) dataset enabled controlled variation of sequence length and sample count while preserving benchmark characteristics. All experiments were performed on a single CPU core to ensure a consistent, hardware-independent comparison—isolating differences attributable solely to algorithmic complexity and implementation.

This study evaluates the computational scalability of various time-series clustering algorithms as functions of the number of series (N) and sequence length (T). Standard k-AVG and its autoregressive (AR)-enhanced variant both exhibit linear complexity in N and T ($O(NT)$), whereas k-Shape scales near-linearly in N but incurs cubic growth in T ($O(NT^3)$). The hierarchical method BIRCH achieves near $O(N \log N)$ behaviour in N and linear scaling in T through its $O(NT)$ incremental summary updates. In contrast, algorithms relying on pairwise distance measure (MSM) distances, such as AP-MSM and OPTICS-MSM, must compute an $N \times N$ distance matrix with each distance costing $O(T^2)$ via dynamic programming, resulting in an overall $O(N^2T^2)$ distance-computation cost. The spectral clustering variant SC-SINK similarly builds an $N \times N$ affinity matrix at $O(N^2T^2)$ and then solves an $O(N^3)$ eigenproblem, rendering it impractical for moderate to large datasets without algorithmic acceleration. Deep learning based clustering methods methods incur significantly higher runtimes due to iterative gradient-descent optimization and complex architectures, making them prohibitively slow for large N or long T absent hardware acceleration. However, when GPUs are available, models such as CNRV can leverage parallel batch training and vectorized computations to achieve scalability comparable to classical approaches.

Empirical results indicate that k-Shape offers the best balance between clustering accuracy and computational efficiency: although it is slower than k-AVG, k-AVG AR, and BIRCH, our evaluation above shows that k-Shape nonetheless attains superior clustering performance. It occupies a “sweet spot” between fast/low-performance and slow/high-performance methods. This renders k-Shape particularly well suited for large-scale applications with constrained computational resources. Similarly, it is important to note that deep learning–based models can significantly benefit from GPU acceleration due to parallel batch training and vectorized processing. Specifically, CNRV not only demonstrated top-tier clustering performance among deep learning–based methods but also shows strong potential for scalability in large-scale settings.

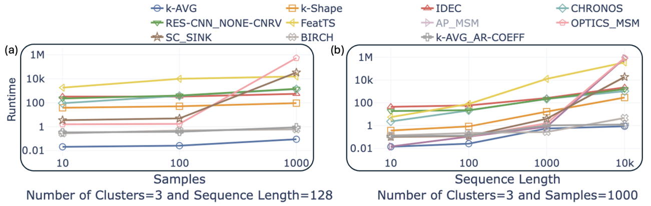


Figure 7: Scalability analysis of the ten best-performing time-series clustering methods (k-AVG as the baseline) for each category. Clustering runtime (seconds) is presented as (a) a function of number of time-series samples N and (b) as a function of sequence length T.

7 CONCLUSIONS AND DISCUSSION

Time-series clustering is a prominent task in time series analysis, yet the literature reveals a substantial gap in systematic, comprehensive evaluations and benchmarking. Despite decades of research, existing benchmarks have significant limitations. We identify these shortcomings and present the most comprehensive analysis to date, evaluating 84 clustering methods across ten distinct classes in data mining, machine learning, and deep learning. Our study yields insights that challenge prevailing assumptions. Although many methods have been proposed, no algorithm consistently outperforms the decade-old baseline k-Shape, suggesting perceived progress may be illusory due to limited evaluation practices. Comprehensive assessments across diverse datasets with rigorous significance testing are imperative to confirm observed improvements genuine rather than artifacts of selective benchmarking. Moreover, reproducibility issues further obscure true performance. Notably, parameter-dependent distance measures, when optimally tuned, demonstrate significant performance gains over untuned counterparts, highlighting the importance of parameter optimization.

Deep learning–based approaches, despite their popularity, have yet to surpass classical methods, often achieving comparable results at substantially higher computational cost. Emerging foundation models offer promise through large-scale pretraining; however, reliance on data overlapping standard evaluation sets introduces contamination and in-distribution bias, so reported gains may not generalize to truly unseen data. Rigorous out-of-distribution testing and strict dataset separation are required to validate genuine advances. In this context, we propose a new time-series distance measure based contrastive learning approach that shows promise and may benefit from refined sampling strategies. By addressing three persistent research questions, our analysis provides insights into design choices that advance time-series clustering and enhances understanding of current techniques. Ultimately, our findings underscore the critical importance and ongoing demand for refined time-series clustering methodologies, calling for further research.

REFERENCES

[1] [n.d.]. *TSClusteringEval is available at www.timeseries.org/tsclusteringeval.* Accessed: July 10, 2025.

[2] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. Time-series clustering–a decade review. *Information Systems* 53 (2015), 16–38.

[3] Mohammed Ali, Ali Alqahtani, Mark W Jones, and Xianghua Xie. 2019. Clustering and classification for time series data in visual analytics: A survey. *IEEE Access* 7 (2019), 181314–181338.

[4] Ali Alqahtani, Mohammed Ali, Xianghua Xie, and Mark W Jones. 2021. Deep Time-Series Clustering: A Review. *Electronics* 10, 23 (2021), 3001.

[5] Henrik André-Jönsson and Dushan Z Badal. 1997. Using signature files for querying time-series data. In *Principles of Data Mining and Knowledge Discovery: First European Symposium, PKDD'97 Trondheim, Norway, June 24–27, 1997 Proceedings 1*. Springer, 211–220.

[6] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

[7] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815* (2024).

[8] Martin Bach-Andersen, Bo Rømer-Odgaard, and Ole Winther. 2017. Flexible non-linear predictive models for large-scale wind turbine diagnostics. *Wind Energy* 20, 5 (2017), 753–764.

[9] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery* 31, 3 (2017), 606–660.

[10] Anthony J Bagnall and Gareth J Janacek. 2004. Clustering time series from ARMA models with clipped data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 49–58.

[11] Ziv Bar-Joseph, Georg K Gerber, David K Gifford, Tommi S Jaakkola, and Itamar Simon. 2003. Continuous representations of time-series gene expression data. *Journal of Computational Biology* 10, 3-4 (2003), 341–356.

[12] Ziv Bar-Joseph, Anthony Gitter, and Itamar Simon. 2012. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics* 13, 8 (2012), 552–564.

[13] Mohini Bariya, Alexandra von Meier, John Paparrizos, and Michael J Franklin. 2021. k-shapestream: Probabilistic streaming clustering for electric grid events. In *2021 IEEE Madrid PowerTech*. IEEE, 1–6.

[14] Gustavo EAPA Batista, Eamonn J Keogh, Oben Moses Tataw, and Vinicius MA De Souza. 2014. CID: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery* 28, 3 (2014), 634–669.

[15] Nurjahan Begum and Eamonn Keogh. 2014. Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment* 8, 2 (2014), 149–160.

[16] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*. 359–370.

[17] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural deep clustering network. In *Proceedings of the web conference 2020*. 1400–1410.

[18] Paul Boniol, Ashwin K Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S Tsay, Aaron Elmore, Michael J Franklin, and John Paparrizos. 2025. VUS: effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal* 34, 3 (2025), 32.

[19] Paul Boniol, Qinghua Liu, Mingyi Huang, Themis Palpanas, and John Paparrizos. 2024. Dive into Time-Series Anomaly Detection: A Decade Review. *arXiv preprint arXiv:2412.20512* (2024).

[20] Paul Boniol, John Paparrizos, Yuhao Kang, Themis Palpanas, Ruey S Tsay, Aaron J Elmore, and Michael J Franklin. 2022. Theseus: navigating the labyrinth of time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3702–3705.

[21] Paul Boniol, John Paparrizos, and Themis Palpanas. 2023. New Trends in Time Series Anomaly Detection.. In *EDBT*. 847–850.

[22] Paul Boniol, John Paparrizos, and Themis Palpanas. 2024. An interactive dive into time-series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5382–5386.

[23] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND in action: subsequence anomaly detection for streams. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2867–2870.

[24] 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.

[25] Paul Boniol, Emmanouil Sylligardos, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2024. Adecimo: Model selection for time series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5441–5444.

[26] George E. P. Box and Gwilym M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA.

[27] Peter J Brockwell and Richard A Davis. 2002. *Introduction to time series and forecasting*. Springer.

[28] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. 2010. isax 2.0: Indexing and mining one billion time series. In *2010 IEEE International Conference on Data Mining*. IEEE, 58–67.

[29] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. 2017. Dilated recurrent neural networks. *Advances in neural information processing systems* 30 (2017).

[30] Lei Chen and Raymond Ng. 2004. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 792–803.

[31] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 491–502.

[32] Qiuxia Chen, Lei Chen, Xiang Lian, Yunhao Liu, and Jeffrey Xu Yu. 2007. Indexable PLA for efficient similarity search. In *Proceedings of the 33rd international conference on Very large data bases*. 435–446.

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[34] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. 2015. The UCR Time Series Classification Archive. www.cs.ucr.edu/~eamonn/time_series_data/.

[35] Madalena Costa, Ary L Goldberger, and C-K Peng. 2002. Multiscale entropy analysis of complex physiologic time series. *Physical review letters* 89, 6 (2002), 068102.

[36] Nello Cristianini and John Shawe-Taylor. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

[37] Marco Cuturi. 2011. Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 929–936.

[38] Michele Dallachiesa, Themis Palpanas, and Ihab F Ilyas. 2014. Top-k nearest neighbor search in uncertain data series. *Proceedings of the VLDB Endowment* 8, 1 (2014), 13–24.

[39] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. 1998. Rule Discovery from Time Series.. In *KDD*, Vol. 98. 16–22.

[40] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[41] Hoang Anh Dau, Nurjahan Begum, and Eamonn Keogh. 2016. Semi-supervision dramatically improves time series clustering under dynamic time warping. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 999–1008.

[42] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[43] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[44] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 551–556.

[45] Jens E d'Hondt, Haojun Li, Fan Yang, Odysseas Papapetrou, and John Paparrizos. 2025. A Structured Study of Multivariate Time-Series Distance Measures. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–29.

[46] David A. Dickey and Wayne A. Fuller. 1979. Distribution of the estimators for autoregressive time series with a unit root. *J. Amer. Statist. Assoc.* 74, 366a (1979), 427–431.

[47] Rui Ding, Qiang Wang, Yingnong Dang, Qiang Fu, Haidong Zhang, and Dongmei Zhang. 2015. Yading: fast clustering of large-scale time series data. *Proceedings of the VLDB Endowment* 8, 5 (2015), 473–484.

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

[49] Adam Dziedzic*, John Paparrizos* (*equal contribution), Sanjay Krishnan, Aaron Elmore, and Michael Franklin. 2019. Band-limited training and inference for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 1745–1754.

[50] Jens E d'Hondt, Odysseas Papapetrou, and John Paparrizos. 2024. Beyond the Dimensions: A Structured Evaluation of Multivariate Time Series Distance Measures. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 107–112.

[51] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd*, Vol. 96. 226–231.

[52] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. 2018. Som-vae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199* (2018).

[53] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. 2019. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems* 32 (2019).

[54] Pasi Fränti and Sami Sieranoja. 2018. K-means properties on six clustering benchmark datasets. *Applied Intelligence* 48, 12 (2018), 4743–4759.

[55] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

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

[57] Kamran Ghasedi, Xiaoqian Wang, Cheng Deng, and Heng Huang. 2019. Balanced self-paced learning for generative adversarial clustering network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4391–4400.

[58] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. 2017. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*. 5736–5745.

[59] Apostolos Giannoulidis, Anastasios Gounaris, and John Paparrizos. 2025. BURST: Rendering Clustering Techniques Suitable for Evolving Streams. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4054–4063.

[60] Rafael Giusti and Gustavo EAPA Batista. 2013. An empirical comparison of dissimilarity measures for time series classification. In *2013 Brazilian Conference on Intelligent Systems*. IEEE, 82–88.

[61] Rahul Goel, Sandeep Soni, Naman Goyal, John Paparrizos, Hanna Wallach, Fernando Diaz, and Jacob Eisenstein. 2016. The social dynamics of language change in online networks. In *International conference on social informatics*. Springer, 41–57.

[62] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885* (2024).

[63] Aditya Grover, Ashish Kapoor, and Eric Horvitz. 2015. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 379–386.

[64] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved Deep Embedded Clustering with Local Structure Preservation.. In *IJCAI*. 1753–1759.

[65] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. 2016. Deep learning for visual understanding: A review. *Neurocomputing* 187 (2016), 27–48.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[67] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. 2014. Classification of time series by shapelet transformation. *Data mining and knowledge discovery* 28, 4 (2014), 851–881.

[68] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[69] Kaisei Hishida, Chunwei Liu, John Paparrizos, and Aaron Elmore. 2025. Beyond Compression: A Comprehensive Evaluation of Lossless Floating-Point Compression. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4396–4409.

[70] Ove Hoegh-Guldberg, Peter J Mumby, Anthony J Hooten, Robert S Steneck, Paul Greenfield, Edgardo Gomez, C Drew Harvell, Peter F Sale, Alasdair J Edwards, Ken Caldeira, et al. 2007. Coral reefs under rapid climate change and ocean acidification. *science* 318, 5857 (2007), 1737–1742.

[71] Christopher Holder and Anthony Bagnall. 2024. Rock the KASBA: Blazingly Fast and Accurate Time Series Clustering. *arXiv preprint arXiv:2411.17838* (2024).

[72] Chris Holder, Matthew Middlehurst, and Anthony Bagnall. 2022. A Review and Evaluation of Elastic Distance Functions for Time Series Clustering. *arXiv preprint arXiv:2205.15181* (2022).

[73] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6, 2 (1979), 65–70.

[74] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2, 1 (1985), 193–218.

[75] Pablo Huijse, Pablo A Estevez, Pavlos Protopapas, Jose C Principe, and Pablo Zegers. 2014. Computational intelligence challenges and applications on large-scale astronomical time series databases. *IEEE Computational Intelligence Magazine* 9, 3 (2014), 27–39.

[76] Dino Ienco and Ruggero G Pensa. 2019. Deep triplet-driven semi-supervised embedding clustering. In *Discovery Science: 22nd International Conference, DS 2019, Split, Croatia, October 28–30, 2019, Proceedings 22*. Springer, 220–234.

[77] Ali Javed, Byung Suk Lee, and Donna M Rizzo. 2020. A benchmark study on time series clustering. *Machine Learning with Applications* 1 (2020), 100001.

[78] Hao Jiang, Chunwei Liu, Qi Jin, John Paparrizos, and Aaron J Elmore. 2020. PIDS: attribute decomposition for improved compression and query performance in columnar storage. *Proceedings of the VLDB Endowment* 13, 6 (2020), 925–938.

[79] Hao Jiang, Chunwei Liu, John Paparrizos, Andrew A Chien, Jihong Ma, and Aaron J Elmore. 2021. Good to the Last Bit: Data-Driven Encoding with CodecDB. In *Proceedings of the 2021 International Conference on Management of Data*. 843–856.

[80] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2016. Variational deep embedding: A generative approach to clustering. *CoRR, abs/1611.05148* 1 (2016).

[81] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. 2001. Distance measures for effective clustering of ARIMA time-series. In *Proceedings 2001 IEEE international conference on data mining*. IEEE, 273–280.

[82] Leonard Kaufman and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.

[83] Steven M. Kay and S. L. Marple. 1981. Spectrum analysis—A modern perspective. *Proc. IEEE* 69, 11 (1981), 1380–1419.

[84] Eamonn Keogh and Shruti Kasetty. 2003. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery* 7, 4 (2003), 349–371.

[85] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[86] Sanjay Krishnan, Aaron J Elmore, Michael Franklin, John Paparrizos, Zechao Shang, Adam Dziedzic, and Rui Liu. 2019. Artificial intelligence in resource-constrained and shared environments. *ACM SIGOPS Operating Systems Review* 53, 1 (2019), 1–6.

[87] Denis Kwiatkowski, Peter C. B. Phillips, Peter Schmidt, and Yongcheol Shin. 1992. Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics* 54, 1–3 (1992), 159–178.

[88] Baptiste Lafabregue, Jonathan Weber, Pierre Gançarski, and Germain Forestier. 2021. End-to-end deep representation learning for time series clustering: a comparative study. *Data Mining and Knowledge Discovery* (2021), 1–53.

[89] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

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

[91] Michele Linardi and Themis Palpanas. 2018. Scalable, variable-length similarity search in data series: The ULISSE approach. *Proceedings of the VLDB Endowment* 11, 13 (2018), 2236–2248.

[92] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2586–2598.

[93] Chunwei Liu, John Paparrizos, and Aaron J Elmore. 2024. AdaEdge: A Dynamic Compression Selection Framework for Resource Constrained Devices. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1506–1519.

[94] 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.

[95] 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.

[96] Qinghua Liu, Seunghak Lee, and John Paparrizos. 2025. TSB-AutoAD: Towards Automated Solutions for Time-Series Anomaly Detection. *PVLDB* 18, 11 (2025), 4364–4379.

[97] 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*.

[98] Shinan Liu, Tarun Mangla, Ted Shaowang, Jinjin Zhao, John Paparrizos, Sanjay Krishnan, and Nick Feamster. 2023. 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 (2023), 1–26.

[99] Arnaud Lods, Simon Malinowski, Romain Tavenard, and Laurent Amsaleg. 2017. Learning DTW-preserving shapelets. In *International Symposium on Intelligent Data Analysis*. Springer, 198–209.

[100] Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. 2019. catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery* 33, 6 (2019), 1821–1852.

[101] Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. 2019. Learning representations for time series clustering. *Advances in neural information processing systems* 32 (2019), 3781–3791.

[102] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[103] Naveen Sai Madiraju. 2018. *Deep temporal clustering: Fully unsupervised learning of time-domain features*. Ph.D. Dissertation. Arizona State University.

[104] Elizabeth Ann Maharaj. 2000. Cluster of time series. *Journal of Classification* 17, 2 (2000), 297–314.

[105] Pierre-François Marteau. 2008. Time warp edit distance with stiffness adjustment for time series matching. *IEEE transactions on pattern analysis and machine*

intelligence 31, 2 (2008), 306–318.

[106] G Martin, A Dragomir, I Piotr, and M Rajeev. 2000. Mining the stock market which measure is best. In *proceedings of ACM SIGKDD Int. Conference On Knowledge Discovery and Data Mining [C]*. 487–496.

[107] Francisco Martínez-Álvarez, Alicia Troncoso, Gualberto Asencio-Cortés, and José C Riquelme. 2015. A survey on data mining techniques applied to electricity-related time series forecasting. *Energies* 8, 11 (2015), 13162–13193.

[108] Kathy McKeown, Hal Daume III, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R Fleischmann, et al. 2016. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology* 67, 11 (2016), 2684–2696.

[109] Errol E Meidinger. 1980. *Applied time series analysis for the social sciences*. Sage Publications.

[110] Katsiaryna Mirylenka, Vassilis Christophides, Themis Palpanas, Ioannis Pefkianakis, and Martin May. 2016. Characterizing home device usage from wireless traffic time series. In *19th International Conference on Extending Database Technology (EDBT)*.

[111] Takaki Mori and Kuniaki Uehara. 2001. Extraction of primitive motion and discovery of association rules from motion data. In *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No. 01TH8591)*. IEEE, 200–206.

[112] Michael D Morse and Jignesh M Patel. 2007. An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 569–580.

[113] Andrew Y Ng, Michael I Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*. 849–856.

[114] Daniel W Otter, Julian R Medina, and Jugal K Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems* 32, 2 (2020), 604–624.

[115] Ioannis Paparrizos. 2018. *Fast, scalable, and accurate algorithms for time-series analysis*. Ph.D. Dissertation. Columbia University.

[116] John Paparrizos, Paul Boniol, Qinghua Liu, and Themis Palpanas. 2025. Advances in Time-Series Anomaly Detection: Algorithms, Benchmarks, and Evaluation Measures. In *SIGKDD*.

[117] 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.

[118] John Paparrizos, Ikraduya Edian, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2022. Fast Adaptive Similarity Search through Variance-Aware Quantization. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2969–2983.

[119] John Paparrizos and Michael J Franklin. 2019. Grail: efficient time-series representation learning. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1762–1777.

[120] John Paparrizos and Luis Gravano. 2015. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 1855–1870.

[121] John Paparrizos and Luis Gravano. 2017. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 1–49.

[122] 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.

[123] John Paparrizos, Haojun Li, Fan Yang, Kaize Wu, Jens E d'Hondt, and Odysseas Papapetrou. 2024. A Survey on Time-Series Distance Measures. *arXiv preprint arXiv:2412.20574* (2024).

[124] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikraduya Edian, Aaron J Elmore, Michael J Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices.. In *CIDR*.

[125] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2020. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.

[126] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2023. Querying Time-Series Data: A Comprehensive Comparison of Distance Measures. *Data Engineering* (2023), 69.

[127] John Paparrizos and Sai Prasanna Teja Reddy. 2023. Odyssey: An Engine Enabling the Time-Series Clustering Journey. *Proceedings of the VLDB Endowment* 16, 12 (2023), 4066–4069.

[128] John Paparrizos and Sai Prasanna Teja Reddy. 2025. Time-Series Clustering: A Comprehensive Study of Data Mining, Machine Learning, and Deep Learning Methods. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4380–4395.

[129] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Detecting devastating diseases in search logs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 559–568.

[130] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Screening for pancreatic adenocarcinoma using signals from web search logs: Feasibility study and results. *Journal of Oncology Practice* 12, 8 (2016), 737–744.

[131] John Paparrizos, Kaize Wu, Aaron Elmore, Christos Faloutsos, and Michael J Franklin. 2023. Accelerating Similarity Search for Elastic Measures: A Study and New Generalization of Lower Bounding Distances. *Proceedings of the VLDB Endowment* 16, 8 (2023), 2019–2032.

[132] John Paparrizos, Fan Yang, and Haojun Li. 2024. Bridging the Gap: A Decade Review of Time-Series Clustering Methods. *arXiv preprint arXiv:2412.20582* (2024).

[133] C-K Peng, Shlomo Havlin, H Eugene Stanley, and Ary L Goldberger. 1995. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos: an interdisciplinary journal of nonlinear science* 5, 1 (1995), 82–87.

[134] François Petitjean, Alain Ketterlin, and Pierre Gançarski. 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition* 44, 3 (2011), 678–693.

[135] Domenico Piccolo. 1990. A distance measure for classifying ARIMA models. *Journal of time series analysis* 11, 2 (1990), 153–164.

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

[137] Alex Rodriguez and Alessandro Laio. 2014. Clustering by fast search and find of density peaks. *science* 344, 6191 (2014), 1492–1496.

[138] Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65, 6 (1958), 386.

[139] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[140] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[141] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. 2012. The move-split-merge metric for time series. *IEEE transactions on Knowledge and Data Engineering* 25, 6 (2012), 1425–1438.

[142] Markus Svensén and Christopher M Bishop. 2007. Pattern recognition and machine learning.

[143] 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.

[144] Donato Tiano, Angela Bonifati, and Raymond Ng. 2021. FeatTS: Feature-based Time Series Clustering. In *Proceedings of the 2021 International Conference on Management of Data*. 2784–2788.

[145] Liudmila Ulanova, Nurjahan Begum, and Eamonn Keogh. 2015. Scalable clustering of time series with u-shapelets. In *Proceedings of the 2015 SIAM international conference on data mining*. SIAM, 900–908.

[146] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings 18th international conference on data engineering*. IEEE, 673–684.

[147] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. 2018. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience* 2018 (2018).

[148] Gabriel Wachman, Roni Khardon, Pavlos Protopapas, and Charles R Alcock. 2009. Kernels for periodic time series arising in astronomy. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 489–505.

[149] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery* 26, 2 (2013), 275–309.

[150] Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics*. Springer, 196–202.

[151] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. PMLR, 478–487.

[152] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*. PMLR, 3861–3870.

[153] Fan Yang and John Paparrizos. 2025. SPARTAN: Data-Adaptive Symbolic Time-Series Approximation. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–30.

[154] Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 177–186.

[155] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine* 13, 3 (2018), 55–75.

[156] Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. 2012. Clustering time series using unsupervised-shapelets. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 785–794.

[157] Hui Zhang, Tu Bao Ho, Yang Zhang, and M-S Lin. 2006. Unsupervised feature extraction for time series clustering using orthogonal wavelet transform. *Informatica* 30, 3 (2006).

[158] Qin Zhang, Jia Wu, Hong Yang, Yingjie Tian, and Chengqi Zhang. 2016. Unsupervised Feature Learning from Time Series.. In *IJCAI*. New York, USA, 2322–2328.

[159] Qin Zhang, Jia Wu, Peng Zhang, Guodong Long, and Chengqi Zhang. 2018. Salient subsequence learning for time series clustering. *IEEE transactions on pattern analysis and machine intelligence* 41, 9 (2018), 2193–2207.

[160] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. *ACM sigmod record* 25, 2 (1996), 103–114.

[161] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.