Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures

John Paparrizos
University of Chicago
jopa@uchicago.edu

Chunwei Liu
University of Chicago
chunwei@uchicago.edu

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

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

ABSTRACT

Distance measures are core building blocks in time-series analysis and the subject of active research for decades. Unfortunately, the most detailed experimental study in this area is outdated (over a decade old) and, naturally, does not reflect recent progress. Importantly, this study (i) omitted multiple distance measures, including a classic measure in the time-series literature; (ii) considered only a single time-series normalization method; and (iii) reported only raw classification error rates without statistically validating the findings, resulting in or fueling four misconceptions in the time-series literature. Motivated by the aforementioned drawbacks and our curiosity to shed some light on these misconceptions, we comprehensively evaluate 71 time-series distance measures. Specifically, our study includes (i) 8 normalization methods; (ii) 52 lock-step measures; (iii) 4 sliding measures; (iv) 7 elastic measures; (v) 4 kernel functions; and (vi) 4 embedding measures. We extensively evaluate these measures across 128 time-series datasets using rigorous statistical analysis. Our findings debunk four long-standing misconceptions that significantly alter the landscape of what is known about existing distance measures. With the new foundations in place, we discuss open challenges and promising directions.

ACM Reference Format:
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 (SIGMOD'20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3318464.3389760

1 INTRODUCTION

The understanding of a multitude of natural or human-made processes involves the analysis of observations over time.

Measure Category	Category Cardinality	Scaling Methods	[45]
Lock-step	52	8	4 (1)
Sliding	4	8	✘
Elastic	7	1	5 (1)
Kernel	4	1	✘
Embedding	4	1	✘

Table 1: Summary of our comprehensive experimental evaluation across 128 datasets. Last column shows summary of category cardinality and scaling methods (in parentheses) evaluated previously in [45] across 38 datasets.

The recording of such time-varying measurements leads in an ordered sequence of data points called *time series* or, more generally, *data series*, to include sequences ordered on dimensions other than time [105, 106]. In the last decades, time-series analysis has become increasingly prevalent, affecting virtually all scientific disciplines and their corresponding industries [41], including astronomy [4, 67, 142], biology [13–15, 49], economics [22, 56, 89, 91, 125, 134, 138], energy sciences [6, 9, 94], engineering [70, 97, 121, 139, 150], environmental sciences [58, 61, 64, 65, 99, 124, 148], medicine [36, 113, 123], neuroscience [19, 77], and social sciences [22, 95]. With sensors and devices becoming increasingly networked and with the explosion of Internet-of-Things (IoT) applications, the volume of produced time series is expected to continue to rise [90]. This unprecedented growth and ubiquity of time series generates tremendous interest in the extraction of meaningful knowledge from time series.

The basis for most analytics over time series involves the detection of similarities between time series. The measurement of similarity, through a *distance or similarity measure*, is the most fundamental building block in time-series data mining, fueling tasks such as querying [2, 33, 45, 71, 76, 78, 84, 96, 107, 118, 136, 143], indexing [24, 25, 29, 48, 51, 72, 73, 75, 86, 117, 129, 135, 140, 146, 164], clustering [5, 12, 46, 69, 74, 85, 110, 116, 145, 147, 153, 161], classification [11, 63, 66, 87, 101, 109, 120, 156], motif discovery [17, 31, 86, 102, 157, 158], and anomaly detection [21, 40, 88, 133, 155, 157]. In contrast to other data types where distance measures often process observations independently, for time series, distance measures consider sequences of observations together [48]. This characteristic complicates the definition of distance measures for time series and, therefore, it is desirable to study the factors that determine their effectiveness.

SIGMOD'20, June 14–19, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6735-6/20/06.
https://doi.org/10.1145/3318464.3389760

The difficulty in formalizing accurate distance measures stems from the inability to express precisely the notion of similarity. As humans we easily recognize perceptually similar time series, by ignoring a variety of distortions, such as fluctuations, misalignments, and stretching of observations. However, it is challenging to derive definitions to reflect the similarity for mathematically non-identical time series [50]. Due to that difficulty and the need to handle the variety of distortions that are characteristic of the time series, dozens of distance measures have been proposed in the time-series literature [8, 18, 27, 28, 30, 45, 51, 53, 100, 109, 110, 127, 137, 141].

Despite this abundance of time-series distance measures and their implications in the effectiveness for a multitude of time-series tasks, less attention has been given in their comprehensive experimental validation. Specifically, in the past two decades, only a single comprehensive experimental evaluation has been dedicated to studying the accuracy of 9 influential time-series distance measures over 38 datasets [45]. Unfortunately, this study suffers from three main drawbacks: (i) this study omitted multiple distance measures, including one of the most classic measures in the time-series literature, namely, the cross-correlation measure [20, 112]; (ii) this study considered only a single time-series normalization method; and (iii) this study reported raw classification error rates without performing any rigorous statistical analysis to assess the significance of the findings. Therefore, the analysis is incomplete, and, the findings might not be conclusive. Importantly, this study is now outdated (more than a decade old), and, naturally, it does not reflect recent progress. Considering the previous drawbacks as well as the remarkable interest in time-series analysis during the last decade, we believe it is critical to revisit this subject in more detail.

However, our effort is not only motivated by the necessity to address the aforementioned issues or to extend the previous study with newer datasets and distance measures. Instead, the thorough experimental evaluation of time-series distance measures that we present in this paper is the byproduct of our attempt to challenge four long-standing misconceptions (see $\mathcal{M}1-\mathcal{M}4$ in Section 2) that have appeared in the time-series literature. These misconceptions are concerned with the (i) normalization of time series; (ii) identification of the state-of-the-art distance measure in every category of measures; (iii) performance of the omitted measures against state-of-the-art measures; and (iv) detection of the most powerful category of measures. Such misconceptions originated from several influential papers [2, 18, 51, 59, 135], some of which date back a quarter of a century, and are fueled by recent inconclusive findings [46] as well as successive claims in the literature that we discuss later. Considering how widely cited and impactful these papers are, we believe it is risky not to challenge such persistent misconceptions that might disorientate newcomer researchers and practitioners.

Motivated by the aforementioned issues and our curiosity to shed some light on these misconceptions, we conduct a comprehensive experimental evaluation to validate the effectiveness of 71 time-series distance measures. These distance measures belong to five categories: (i) 52 *lock-step* measures, which compare the ith point of one time series with the ith point of another; (ii) 4 *sliding* measures, which are the sliding versions of lock-step measures when comparing one time series with all shifted versions of the other; (iii) 7 *elastic* measures, which create a non-linear mapping between time series by comparing one-to-many points in order to align or stretch points; (iv) 4 *kernel* measures, which use a function (with lock-step, sliding, or elastic properties) to implicitly map data into a high-dimensional space; and (v) 4 *embedding* measures, which exploit distance or kernel measures indirectly for constructing new representations for time series. In addition, we consider 8 normalization methods for time series, which serve as preprocessing steps. Table 1, summarizes our comprehensive evaluation and compares some statistics against the decade-old influential study [45].

We perform an extensive evaluation of these distance measures across 128 datasets [41] and compare their classification accuracy obtained from one-nearest-neighbor classifiers (1-NN) under both supervised and unsupervised settings. We conduct a rigorous statistical validation of our findings by employing two statistical tests to assess the significance of the differences in classification accuracy when comparing pairs of measures or multiple measures together. In summary, our study identifies (i) normalization methods leading to significant improvements in a number of distance measures; (ii) new lock-step measures that significantly outperform the current state of the art; (iii) an omitted baseline that most highly popular elastic measures do not outperform; and (iv) new elastic and new kernel measures that significantly outperform the current state of the art. These findings debunk the four long-standing misconceptions and, therefore, alter the landscape of what is known about existing measures.

We start with the description of the four misconceptions in the literature (Section 2) and we review the relevant background (Section 3). Then, we present our contributions:

- We explore for the first time 8 normalization methods in conjunction with 56 distance measures (Section 4).
- We study 52 lock-step distance measures (Section 5).
- We investigate 4 classic sliding measures omitted from virtually every previous evaluation (Section 6).
- We validate the accuracy of 7 elastic measures under both supervised and unsupervised settings (Section 7).
- We compare for the first time 4 kernel functions (Section 8) and 4 embedding distance measures (Section 9).
- We present an accuracy-to-runtime analysis (Section 10).

Finally, we conclude with the implications of our work and a discussion of new directions and challenges (Section 11).

2 THE FOUR MISCONCEPTIONS

In this section, we describe four misconceptions that have appeared in the time-series data mining literature.

These misconceptions have originated in part from several influential papers [2, 18, 51, 59, 135]. Subsequently, these misconceptions were fueled by a comprehensive study of time-series distance measures [45] as well as dozens of subsequent papers in the literature trusting its findings. Even though an extension of this study appeared five years later [144], this newer version focused on elaborating on the previous results. Recent studies that have focused on time-series classification [11, 87] performed a statistical analysis of several classifiers, including the distance measures in [45, 144]. Unfortunately, these studies only considered supervised tuning of necessary parameters, which does not reflect the use of distance measures for similarity search [48]. Importantly, some results in [11] contradict other results in [87], which, in turn, validated claims that there is no significant difference between the evaluated elastic measures [45, 144]. Interestingly, the improved accuracy found for some measures was attributed to the evaluation framework used while otherwise it was claimed to be undetectable [11]. Considering such apparent difficulties in providing conclusive evidence for this important subject, it is not surprising that the following misconceptions have persisted for so long.

Before we dive into the details, we emphasize that we do not believe or imply that any of these misconceptions were created on purpose. On the contrary, we believe that they are based on evidence, trends, and resources available at the given point in time. We describe the four misconceptions in the form of answers to questions a newcomer researcher or practitioner would likely identify by studying the literature.

$\mathcal{M}1$**: How to normalize time series?** The consensus is to use the z-score or z-normalization method. Starting with the work of Goldin and Kanellakis [59], a follow-up of the two seminal papers for sequence [2] and subsequence [51] search in time-series databases, that suggested first to normalize the time series to address issues with scaling and translation, z-normalization became the prevalent method to preprocess time series. Despite the proposal of alternative methods the same year [3], the z-normalization was subsequently preferred as the suggested transformations are also applicable to the widely popular Fourier representation [2, 51, 117]. Due to the ubiquity of z-normalization, a valuable resource for time series, the UCR Archive [41], offered until recently the datasets in their z-normalized form. To the best of our knowledge, no study has ever extensively evaluated normalization methods for time series. We review 8 relevant approaches in Section 4 and study their performance in Sections 5 and 6.

$\mathcal{M}2$**: Which lock-step measure to use?** The consensus is to use the Euclidean distance (ED). ED was the method of choice in the first paper for sequence search in time series [2] due to its usefulness in many cases and its applicability over feature vectors. Considering that ED is straightforward to implement, parameter-free, efficient, as well as tightly connected with the Fourier representation and widely supported by indexing mechanisms (in contrast to other L_p-norm variants [160]), there is no surprise about its popularity. Besides, evidence that with increased dataset sizes, the classification error of ED converges to the error of more accurate measures [135], justified its use from virtually all current time-series indexing methods [48]. (Our results in Section 10 suggest that classification error of ED may not always converge to the error of more accurate measures, at least not always with the same speed of convergence.) In Section 5, we evaluate 52 lock-step measures to determine the state of the art.

$\mathcal{M}3$**: Are elastic better than sliding measures?** The answer is currently unknown. Despite the wide popularity of the cross-correlation measure, also known as sliding Euclidean or dot product distance, in the signal and image processing literature [23], cross-correlation has largely been omitted from distance measure evaluations. We believe two factors are responsible for that. First, cross-correlation was considered in the seminal paper [2] as a typical similarity measure, but ED was preferred instead because (i) cross-correlation reduces to ED; and (ii) for the aforementioned reasons in $\mathcal{M}2$. Second, in the introduction of Dynamic Time Warping (DTW) [18], an elastic measure, as an alternative to ED a year later, no comparison was performed against cross-correlation, an obvious baseline. Subsequently, virtually all research on that subject focused either on lock-step or elastic measures [48, 50], with a few exceptions [83, 110, 128]. Interestingly, cross-correlation was not considered as a baseline method in any of the proposed elastic measures [27, 28, 30, 100, 137, 141], neither in any of the experimental evaluations of distance measures discussed previously [11, 45, 87, 144]. Strangely, cross-correlation was also omitted from many popular surveys [50, 119]. Therefore, it remains unknown if elastic measures outperform sliding measures. We study 4 sliding measures in Section 6 and analyse their performance against elastic measures in Section 7.

$\mathcal{M}4$**: Is *DTW* the best elastic measure?** The general consensus that has emerged is yes. Since the introduction of DTW as a distance measure for time series [18], DTW has inspired the exploration of edit-based distances and it is widely used as the baseline method for this problem [11, 27, 28, 30, 87, 100, 110, 137, 141]. It is not uncommon to identify statements even in the abstracts of papers that 1-NN with DTW is exceptionally difficult to outperform [114–116, 152]. Such statements have been backed over the years by the aforementioned extensive evaluations, which conclude that (i) the accuracy of other elastic measures is very close to that of DTW [45, 144]; (ii) there is no significant difference

in the accuracy of elastic measures [87]; and (iii) that it is "a little embarrassing" that most classifiers do not outperform 1-NN with DTW [11]. Therefore, there is little space to doubt that DTW is the best elastic measure. To study that misconception, we validate 7 elastic measures in Section 7.

To complete the analysis and capture recent progress, we also include kernel measures and embedding measures in our evaluation (Sections 8 and 9). With the detailed presentation of the four misconceptions, we believe we have now convinced the reader that these misconceptions are not based on any personal biases but, instead, have originated naturally along with the evolution of this area. However, it is risky to not challenge their validity, which may result in confusion for newcomer researchers and practitioners and discourage them from tackling problems in that area. Importantly, it is surprising to consider that half a century of scientific progress has not resulted in any significant improvements over ED or the 50-year-old DTW [126].

Next, we review the relevant background required to validate the accuracy of the normalization methods and distance measures. Even though the efficiency of measures is another important factor of their effectiveness, there are many ways to accelerate each measure, ranging from hardware-aware implementations to algorithmic solutions such as the use of indexing or comparison pruning. We refer the reader to an excellent recent study of data-series similarity search [48], which shows the level of detail required to only evaluate ED. Therefore, we leave such detailed study for future work but we present an accuracy-to-runtime analysis in Section 10. e

3 PRELIMINARIES AND BACKGROUND

In this section, we review the necessary background for our experimental evaluation of time-series distance measures.

Terminology and definitions: We consider a time-series dataset as a set of n real-valued vectors $X = [\vec{x}_1, \ldots, \vec{x}_n]^\top \in \mathbb{R}^{n \times m}$, where each time series, $\vec{x}_i \in \mathbb{R}^m$, is an m-dimensional ordered sequence of data points. From this definition, it becomes clear that we consider *univariate* time series of equal length, where each of these points is a scalar. Following the previous evaluations [11, 45, 144], we consider that the sampling rates of all time series are the same and, therefore, we can omit the discrete time stamps. In addition, time series into consideration do not track errors as in the case of *uncertain* time series [39, 40, 159].[1]

Datasets: To conduct our extensive evaluation, we use one of the most valuable public resources in the time-series data mining literature, the UCR Time-Series Archive [41]. This archive contains the largest collection of class-labeled time-series datasets. Currently, the archive consists of 128 datasets and includes time series from sensor readings, image outlines, motion capture, spectrographs, medical signals, electric devices, as well as simulated time series. Each dataset contains from 40 to 24,000 time series, the lengths vary from 15 to 2,844, and each time series is annotated with a single label. The majority of the datasets are already z-normalized and, therefore, we apply the same normalization to all datasets.

The latest version of the archive has deliberately left a small number of datasets containing time series with varying lengths and missing values to reflect the real world. Following the recommendation of the authors of the archive, who performed similar steps to report classification accuracy numbers on the UCR archive website [41], we resample shorter time series to reach the longest time series in each dataset and we fill missing values using linear interpolation. Through these steps, we make the new datasets compatible with previous versions of the archive as well as with all distance measures that we consider in this study [108].

Evaluation framework: Following the previous studies [11, 45], we also employ the 1-NN classifier in our evaluation framework, with important differences. 1-NN classifiers are suitable methods for distance measure evaluation for several reasons [45]. Specifically, 1-NN classifiers: (i) resemble the problem solved in time-series similarity search [48]; (ii) are parameter-free and easy to implement; (iii) dependent on the choice of distance measure; and (iv) provide an easy-to-interpret (classification) accuracy measure, which captures if the query and the nearest neighbor belong to the same class.

A critical step for the effectiveness of classifiers is the splitting of a dataset into training and test sets. Previous studies [11, 45, 144] used the k-cross-validation resampling procedure, which produces k groups of time series, tunes necessary parameters on the $k-1$ groups, and evaluates the distance measures using the group of time series left. Strangely, [45, 144] tuned parameters only on a single group and evaluated the distance measures using the $k-1$ groups, which contradicts the common practice. In [11], the improved accuracy of some measures is attributed to such a resampling procedure, while otherwise, it was claimed to be undetectable. Therefore, to eliminate biases from resampling, we respect the split of training and test sets provided by the UCR archive as well as the class distribution in the datasets (i.e., some datasets contain the same number of time series in each class while other datasets contain imbalanced classes). This decision makes our evaluation framework as close to deterministic as possible and enables reproducibility.

More formally, given a matrix $F = [\vec{f}_1, \ldots, \vec{f}_p]^\top \in \mathbb{R}^{p \times m}$ with the p time series in the training set, a matrix $G = [\vec{g}_1, \ldots, \vec{g}_r]^\top \in \mathbb{R}^{r \times m}$ with the r time series in the test set, and any choice of distance measure, $d(\cdot,\cdot)$, our 1-NN classifier relies on two dissimilarity matrixes to produce the final

[1] Most of the measures we consider can be extended with some effort for uncertain time series or *multivariate* time series where each point represents a vector [10], but we leave such exploration for future work.

Algorithm 1: 1-Nearest-Neighbor (1-NN) Classifier

```
Input  : E is an r-by-p dissimilarity matrix
         GL is a 1-by-r vector with the class labels of time series in G
         FL is a 1-by-p vector with the class labels of time series in F
Output : acc is a scalar storing the classification accuracy
1  function acc = OneNNWITHDM(E, GL, FL):
2      acc = 0
3      for i = 1 to r do
4          best_dist = ∞
5          for j = 1 to p do
6              dist = E(i, j)
7              if dist < best_dist then
8                  class = FL(j)
9                  best_dist = dist
10         if GL(i) == class then
11             acc = acc + 1
12     acc = acc / r
```

classification accuracy. Specifically, matrix $W \in \mathbb{R}^{p \times p}$ contains the dissimilarity values between all pairs of time series in the training set, with $W_{ij} = d(\vec{f}_i, \vec{f}_j) \forall \vec{f}_i, \vec{f}_j \in F$, whereas matrix $E \in \mathbb{R}^{r \times p}$ contains the dissimilarity values between each time series in the test set with each time series in the training set, with $E_{ij} = d(\vec{g}_i, \vec{f}_j) \forall \vec{g}_i \in G, \vec{f}_j \in F$.

Algorithm 1 shows the pseudocode of our 1-NN classifier that evaluates the *test* accuracy given a matrix E (as well as vectors FL and GL containing the class labels of F and G, respectively). By providing as input, a matrix W (as well as two times the vector FL containing the class labels of F), the same algorithm computes the leave-one-out *training* accuracy, which enables parameter tuning. With this setup, we decouple the processes of distance matrix computation, parameter tuning, and distance measure evaluation. Importantly, it facilitates easy distribution of the computation of the dissimilarity matrixes for different parameters and avoids the need to find the appropriate k value to perform k-cross-validation, which is another factor that might have affected the findings in the previous studies [11, 45].

Statistical analysis: To assess the significance of the differences in accuracy, we employ two statistical tests to validate the pairwise comparisons of measures and the comparisons of multiple measures together. Specifically, following the highly influential [42], we use the Wilcoxon test [149] with a 95% confidence level to evaluate pairs of measures over multiple datasets, which is more appropriate than the t-test [122]. As with pairwise tests we cannot reason about multiple measures together and following [42], we also use the Friedman test [54] followed by the post-hoc Nemenyi test [104] to compare multiple measures over multiple datasets and report statistical significant results with 90% confidence level (because these tests require more evidence than Wilcoxon).

Availability of code and results: We implemented the evaluation framework in Matlab, with imported C and Java codes for several distance measures. To ensure the reproducibility of our findings, we make the code available and we provide the results in a website to ease exploration.[2]

Environment: We ran our experiments on 15 identical servers: Dual Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz and 196GB RAM. Each server has 24 physical cores (12 per CPU), which provided us with 360 cores for four months.

Next, we start with the study of normalization methods.

4 TIME-SERIES NORMALIZATIONS

In this section, we review 8 normalization methods usually performed as a preprocessing step before any comparison.

As we discussed earlier, a critical issue when comparing time series is how to handle a number of distortions that are characteristic of the time series. For complex distortions, sophisticated distance measures are required as offering invariances to such distortions is not trivial, which explains the proliferation of distance measures in the literature. However, in several cases, a simple preprocessing step is generally sufficient to eliminate particular distortions, as we see next.

Consider the following two examples [59]: (i) two products with similar sales patterns but different sales volume; and (ii) temperatures of two days starting at different values but exhibiting the exact same pattern. The first is an example of the difference in *scale* between two time series, whereas the second is an example of the difference in *translation*. Despite such differences, in many cases, it is useful to recognize the similarity between time series. Formally, for any constants a (scale) and b (translation), linear transformations in time series of the form $a\vec{x} + b$ should not affect their similarity.

Several methods have been proposed to handle these popular distortions. *Normalization* methods transform the data to become normally distributed, whereas *standardization* methods place different data ranges on a common scale. In the machine-learning literature, feature *scaling* is also used to refer to such methods. In practice, all terms are used interchangeably to refer to some data transformation.

Z-score normalization: The most popular normalization method in the time-series literature is by far the z-score (see Section 2). Z-score transforms data such that the resulting distribution has zero-mean and unit-variance:

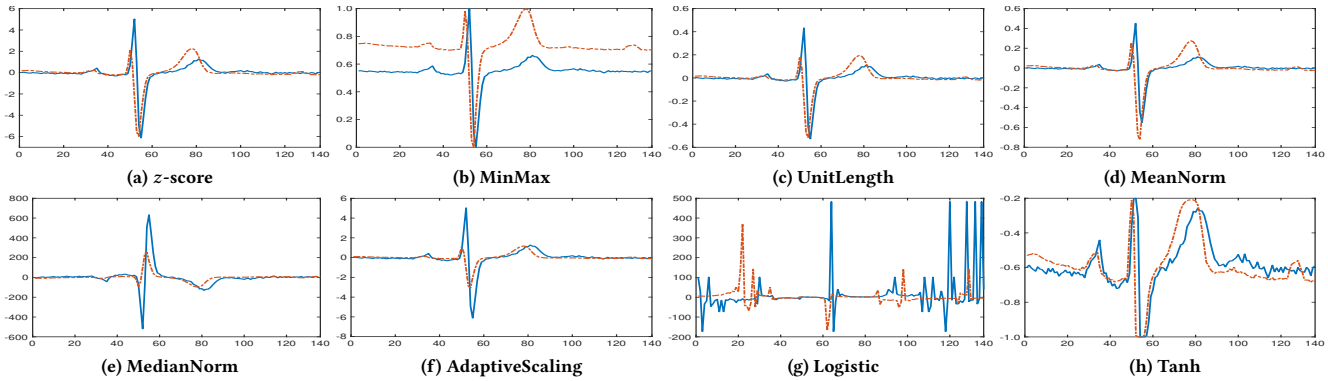
$$\vec{x}' = \frac{\vec{x} - average(\vec{x})}{std(\vec{x})} \tag{1}$$

where $average(\cdot)$ is the mean of $\vec{x}$ and $std(\cdot)$ is its standard deviation. Z-score is also widely used in many machine-learning algorithms, which might explain its popularity [62].

Min-max normalization (MinMax): An alternative approach is to scale time-series values in the [0,1] range:

$$\vec{x}' = \frac{\vec{x} - min(\vec{x})}{max(\vec{x}) - min(\vec{x})} \tag{2}$$

[2] http://benchmarks.timeseries.org



(a) z-score (b) MinMax (c) UnitLength (d) MeanNorm

(e) MedianNorm (f) AdaptiveScaling (g) Logistic (h) Tanh

Figure 1: Example of how each of the 8 normalization methods transforms two time series of the ECGFiveDays dataset [41].

However, many distance measures cannot deal with zero values and, therefore, scaling time series between an arbitrary set of values [a,b] is often preferred:

$$\vec{x}' = a + \frac{\vec{x} - min(\vec{x}) \cdot (b - a)}{max(\vec{x}) - min(\vec{x})} \tag{3}$$

The selection of the range is data-dependent and might require tuning to maximize its effectiveness.

Mean normalization (MeanNorm): Another option to normalize time series is to combine the previous methods:

$$\vec{x}' = \frac{\vec{x} - average(\vec{x})}{max(\vec{x}) - min(\vec{x})} \tag{4}$$

such that numerator is based on z-score and the denominator is based on MinMax normalization.

Median normalization (MedianNorm): Another method is to divide the data points by the median (or mean):

$$\vec{x}' = \frac{\vec{x}}{median(\vec{x})} \tag{5}$$

which is less popular due to numerical issues that may arise.

Unit length normalization (UnitLength): A common way to normalize time series is to scale the data points such that the whole time series has length one:

$$\vec{x}' = \frac{\vec{x}}{||\vec{x}||} \tag{6}$$

where $||\cdot||$ denotes the Euclidean norm.

Adaptive scaling (AdaptiveScaling)) [32, 154]: In contrast to all previous normalization methods, this approach computes the scaling factor between pairs of time series:

$$a = \frac{\vec{x}_i \cdot \vec{x}_j{}^T}{\vec{x}_i \cdot \vec{x}_j{}^T} \tag{7}$$

which is used in each pairwise comparison (e.g., ED($\vec{x}_i$,$a \cdot \vec{x}_j$).

Recently, several activation functions for neural networks gained popularity [81]. We explore two such functions.

Logistic or sigmoid normalization (Logistic)): The logistic function uses the formula below to activate time series:

$$\vec{x}' = \frac{1}{1 + e^{-\vec{x}}} \tag{8}$$

Hyperbolic tangent normalization (Tanh)): This method uses the formula below to activate time-series values:

$$\vec{x}' = \frac{e^{2\vec{x}} - 1}{e^{2\vec{x}} + 1} \tag{9}$$

Figure 1, shows an example of how each one of the previously described normalization methods transforms a pair of time series from the ECGFiveDays dataset [41]. We observe that in some cases, the differences are only visible in the range of values (e.g., z-score vs. UnitLength), but, in others, the visual effect is more distinct (e.g., MinMax, MeanNorm, and AdaptiveScaling). The most unexpected visual effects come from the two non-linear transformations (i.e., Logistic and Tanh). We evaluate the accuracy of these 8 methods along with the 52 lock-step measures in the next section.

5 TIME-SERIES LOCK-STEP DISTANCES

In this section, we study 52 lock-step measures that have been proposed across different scientific disciplines.

Distance measures provide a numerical value to quantify how distant are pairs of objects represented as points, vectors, or matrixes. Due to the difficulty in formalizing the notion of similarity, as well as the need to handle a variety of distortions and applications, hundreds of distance measures have been proposed in the literature. This proliferation of distance measures across different scientific areas has resulted in multi-year efforts to organize this knowledge into dictionaries [43] and encyclopedias [44] of distance measures.

As it is understandable, not all of these measures are applicable to time-series data. Thankfully, different endeavors have already been conducted to identify appropriate measures for a variety of tasks across different fields [55, 162]. An influential study [26] identified 50 lock-step distance measures that we adapt in our evaluation of time-series distance measures. We note that a previous study [57] evaluated a subset of these measures (45) using 1-NN over 42 datasets from the UCR archive and concluded that there is no significant differences between these lock-step distance measures.

Unfortunately, we identified issues with this study. First, several of the evaluated measures are known to be equivalent to each other and, therefore, they should provide identical classification accuracy results. For example, this is the case for the Euclidean distance and the inner product (or Pearson's correlation), which under z-normalization, they should provide the same accuracy numbers. Second, several distance measures were not properly implemented, resulting in using as distance values either the real part of complex numbers or the first value of a normalized vector of the input time series. Therefore, the analysis of these lock-step measures is incomplete, and the findings of the study are inconclusive.

In our study, we have carefully re-implemented all 50 distance measures from [26]. The distance measures belong to 7 different families of measures: (1) 4 measures belong to the L_p Minkowski family; (2) 6 measures belong to the L_1 family; (3) 7 measures belong to the *Intersection* family; (4) 6 measures belong to the *Inner Product* family; (5) 5 measures belong to the *Fidelity* family; (6) 8 measures belong to the L_2 family; and (7) 6 measures belong to the *Entropy* family. Apart from these 42 measures, we also consider the 3 measures that utilize ideas from multiple other measures (*Combinations*) as well as 5 measures proposed in the survey but not reported in the literature (until that point), overall 50 measures.

Besides these measures, we also include two measures that have substantial differences from the previous lock-step measures. Specifically, DISSIM [53] defines the distance as a definite integral of the function of time of the ED in order to take into consideration different sampling rates of time series. This computationally expensive operation can be approximated by a modified version of ED that considers in the distance of the ith points the $i+1$th points, which is a form of a smoothing operation. Finally, the adaptive scaling distance (ASD), embeds internally the AdaptiveScaling normalization described previously with an inner product measure to compare time series under optimal scaling [32, 154]. We exclude from our analysis and leave for future study three recently proposed correlation-aware measures [98] that are more complex in nature than those we consider here.

Evaluation of lock-step measures: In lieu of including several pages with formulas and tables with raw classification accuracy numbers, we created a website to ease the exploration of our results, as noted earlier. For all mathematical formulas, we refer the reader to the previous survey [26]. Below, we only report the summary of raw results and findings from our rigorous statistical analysis. Specifically, we evaluate 52 distance measures and their combinations with 8 normalization methods using our 1-NN classifier over 128 datasets (see Section 3). Table 4 contains all distance measures requiring parameter tuning along with the evaluated parameters. From the lock-step measures, only one measure, the Minkowski distance, requires tuning.

Distance Measure	Scaling Method	Better	Average Accuracy	>	=	<
Minkowski (L_p-norm)	z-score	✔	0.7083	79	13	36
	MinMax	✔	0.7041	70	12	46
	UnitLength	✔	0.7083	79	13	36
	MeanNorm	✔	0.7082	81	10	37
	Tanh	✘	0.6941	60	7	61
Lorentzian	z-score	✔	0.7022	71	8	49
	MinMax	✔	0.7010	66	7	55
	UnitLength	✔	0.7024	76	9	43
	MeanNorm	✔	0.7061	75	9	44
	Tanh	✘	0.6950	63	9	56
Manhattan (L_1-norm)	z-score	✔	0.7017	76	11	41
	MinMax	✔	0.7017	66	11	51
	UnitLength	✔	0.7017	76	11	41
	MeanNorm	✔	0.7051	76	9	43
	Tanh	✘	0.6913	63	11	54
Avg L_1/L_∞	z-score	✔	0.7012	75	10	43
	MinMax	✔	0.7013	68	5	55
	UnitLength	✔	0.7012	75	10	43
	MeanNorm	✔	0.7046	76	9	43
	Tanh	✘	0.6911	60	13	55
DISSIM	z-score	✔	0.7013	78	6	44
	MinMax	✔	0.7016	66	8	54
	UnitLength	✔	0.7013	78	6	44
	MeanNorm	✔	0.7039	73	9	46
	Tanh	✘	0.6917	64	10	54
Jaccard	MinMax	✘	0.6955	66	12	50
	MeanNorm	✔	0.6939	76	19	33
ED (L_2-norm)	MinMax	✘	0.6947	69	13	46
	MeanNorm	✘	0.6896	67	11	50
Emanon4	MinMax	✔	0.7034	72	6	50
Soergel	MinMax	✔	0.7011	73	4	51
Clark	MinMax	✘	0.6986	73	4	51
Topsoe	MinMax	✘	0.6962	71	4	53
Chord	MinMax	✘	0.6934	64	8	56
ASD	MinMax	✘	0.6884	56	13	59
Canberra	MinMax	✘	0.6933	56	4	68
ED	**z-score**	-	**0.6863**	-	-	-

Table 2: Comparison of lock-step distance measures. "Scaling Method" column indicates the underlying time-series normalization. "Better" column denotes that the distance measure outperforms the baseline with statistical significance. "Average Accuracy" column shows the mean accuracy achieved across 128 datasets. The last three columns indicate the number of datasets over which a distance measure is better (">"), equal ("="), or worse ("<") than the baseline.

Table 2 reports the performance of lock-step measures against the baseline listed in the last row. Following the convention [52], we also report the average accuracy across datasets but we note that this number is meaningless when not accompanied by rigorous statistical analysis. Specifically,

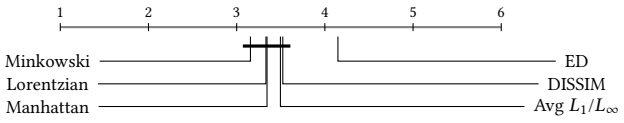


Figure 2: Ranking of lock-step measures under z-score based on the average of their ranks across datasets.

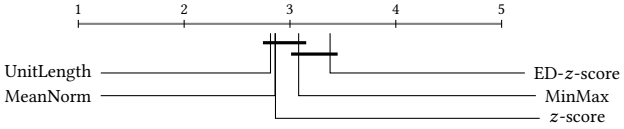


Figure 3: Ranking of normalization methods in combination with the Lorentzian distance based on the average of their ranks across datasets. ED uses z-score normalization.

from all combinations of distance measures and normalization methods ($52 \cdot 8 = 416$ in total), we only report those resulting in an average accuracy higher than the one achieved by ED with z-score, the current state-of-the-art lock-step measure. We omit measures achieving the same or worse average accuracy than ED. We observe 14 measures with some improvement in their average accuracy in contrast to ED and overall 36 combinations with different normalization methods. However, only about half of these combinations result in statistically significant differences according to the Wilcoxon test. (We perform all pairwise comparisons with Wilcoxon to ensure we did not miss any accurate measure.)

In particular, Minkowski distance performs better in terms of average accuracy than all other distance measures, but it is also the only measure that requires tuning, which is not always desirable. Interestingly, several measures of the L_1 family, namely, the Lorentzian (i.e., the natural logarithm of L_1), the Manhattan, and the Avg L_1/L_∞, outperform ED with statistically significant differences. DISSIM, a measure that integrates some form of smoothing of time series, also outperforms ED significantly. In all of these measures, we observe a similar trend: all combinations with z-score, UnitLength, and MeanNorm normalizations lead to significant improvements. However, our extensive experimentation reveals three unknown measures in the time-series literature that achieve statistically significant improvement over ED, namely, the Jaccard distance with MeanNorm, the Emanon4 distance with MinMax, and the Soergel distance with MinMax. Interestingly, all three distance measures do not show improvements under z-score, which shows the importance of considering different normalizations. Among all methods, MeanNorm seems to perform the best. However, the Wilcoxon test suggests no statistically significant differences between any of the 17 combinations that outperform ED with z-score.

To better understand the performance of lock-step measures, we also evaluate the significance of their differences in accuracy when considering several distance measures together, using the Friedman test followed by a post-hoc Nemenyi test. Specifically, we perform two analyses: (i) we evaluate different distance measures under the same normalization; and (ii) we evaluate standalone distance measures under different normalizations; Figure 2 shows the average rank across all datasets of the distance measures, which under z-score normalization, outperformed previously ED. The thick line connects measures that do not perform statistically significantly better. We observe that Lorentzian is ranked first (once we ignore the supervised Minkowski), meaning that it performed best in the majority of the datasets. All 5 measures significantly outperform ED, but we observe no difference between them. Figure 3 evaluates a standalone distance measure, the Lorentzian measure that performed the best previously, with different normalization methods against ED with z-score. We observe that the 3 out of the 4 combinations that were better than ED under the Wilcoxon test remain better under this statistical analysis, and there is no difference between them. We omit similar figures for the other measures as we observe the similar trends.

Debunking $\mathcal{M}1$ and $\mathcal{M}2$: Our evaluation shows clear evidence that normalization methods other than z-score can lead to significant improvements, which debunks $\mathcal{M}1$. Even though for standalone measures, we did not observe significant improvements (e.g., ED with MeanNorm vs. ED with z-score), that does not reject our hypothesis. We note that the majority of the UCR datasets are in their z-normalized form and, therefore, for fairness, we z-normalized all datasets, which may have limited this analysis. Despite that, we identified two new distance measures, unknown until now, that only under MinMax and MeanNorm methods outperform ED with z-score and, importantly, z-score is not suitable for them. Normalizations such as MeanNorm, which combines z-score and MinMax methods, seems to perform the best for several measures. Similarly, our analysis shows that distance measures other than ED can lead to significant improvements, which debunks $\mathcal{M}2$. We identified 7 distance measures that significantly outperform ED. We emphasize that no previous study considered different normalization methods in order to challenge $\mathcal{M}1$, and our findings contradict both previous studies [45, 57], which concluded that there is no significant difference in the accuracy of lock-step measures.

Next, we focus on sliding versions of lock-step measures.

6 TIME-SERIES SLIDING DISTANCES

We study 4 variants of cross-correlation, a measure that has largely been omitted from distance measure evaluations.

Starting with the concurrent introduction of lock-step and elastic measures for the problem of time-series similarity search [2, 18, 51], the vast majority of research focused on these two categories of measures (see $\mathcal{M}3$ in Section 2). Cross-correlation, which is similar to convolution, dates back in the 1700s [47] but received practical popularity only

Distance Measure	Scaling Method	Better	Average Accuracy	>	=	<
NCC_c (SBD)	z-score	✔	0.7309	86	9	33
	MinMax	✔	0.7186	72	8	48
	UnitLength	✔	0.7309	86	9	33
	MeanNorm	✔	0.7309	86	9	33
	MedianNorm	✘	0.7050	63	7	58
	Adaptive	✘	0.7072	72	8	48
	Tanh	✘	0.7077	58	8	62
NCC_b	z-score	✔	0.7309	86	9	33
	UnitLength	✔	0.7309	86	9	33
NCC	z-score	✔	0.7309	86	9	33
	UnitLength	✔	0.7309	86	9	33
Lorentzian	**UnitLength**	-	**0.7024**	-	-	-

Table 3: Comparison of sliding distance measures. "Scaling Method" column indicates the underlying time-series normalization. "Better" column denotes that the distance measure outperforms the baseline with statistical significance. "Average Accuracy" column shows the mean accuracy achieved across 128 datasets. The last three columns indicate the number of datasets over which a distance measure is better (">"), equal ("="), or worse ("<") than the baseline.

after the invention of Fast Fourier Transform (FFT) [34], which dramatically reduced its computational cost. Cross-correlation is one of the most fundamental operations in signal processing [23] and, lately, in deep neural networks [79, 80]. Recently, research focusing on time-series clustering used cross-correlation and achieved state-of-the-art performance for this task [110, 111]. However, this work assumed z-normalized time series and performed evaluations only against ED and DTW. Next, we present cross-correlation following the notation used in [110] in an attempt to establish consistent terminology with the recent literature.

In simple terms, the cross-correlation measure maximizes the correlation (or, equivalently, minimizes the ED [103]) between a time-series $\vec{x}$ and all shifted versions of another time series $\vec{y}$. By shifting (or sliding), we refer to an operation, $\vec{x}_{(s)}$, that rearranges the data points by moving all points by $|s|$ positions to the right, for $s \geq 0$, or left, for $s < 0$. For example, $\vec{x}_{(1)}$ moves all data points by one position to the right and brings the final entry to the first position (or differently pads the empty positions with zeros; both approaches lead to similar measures). When we consider all shifts, $s \in [-m, m]$, where m is the length of both time series (the measure can also operate with unequal lengths), we produce the cross-correlation sequence, $CC_w(\vec{x}, \vec{y})$, with $w \in \{1, 2, \ldots, 2m-1\}$, of length $(2m-1)$, containing the inner product of the two time series in every possible shift.

Unfortunately, the computation of $CC_w(\vec{x}, \vec{y})$ is expensive, $O(m^2)$, but, thankfully, with the use of FFT ($\mathcal{F}(\cdot)$) and its inverse version ($\mathcal{F}^{-1}(\cdot)$), the cost reduces to $O(m \cdot \log(m))$:

$$CC_w(\vec{x}, \vec{y}) = \mathcal{F}^{-1}\{\mathcal{F}(\vec{x}) * \mathcal{F}(\vec{y})\} \tag{10}$$

Having introduced the necessary notation and considering popular normalizations, we can derive the following 4 variants of cross-correlation similarity measures [110]:

$$NCC_q(\vec{x}, \vec{y}) = \begin{cases} \max\left(\frac{CC_w(\vec{x},\vec{y})}{m}\right), & q = \text{“}b\text{”}\ (NCC_b) \\ \max\left(\frac{CC_w(\vec{x},\vec{y})}{m-|w-m|}\right), & q = \text{“}u\text{”}\ (NCC_u) \\ \max\left(\frac{CC_w(\vec{x},\vec{y})}{||\vec{x}||\cdot||\vec{y}||}\right), & q = \text{“}c\text{”}\ (NCC_c) \\ \max(CC_w(\vec{x},\vec{y})), & q = \text{“}\cdot\text{”}\ (NCC) \end{cases} \tag{11}$$

known as the normalized cross-correlation (because it assumes some underlying time-series normalization), NCC, the biased estimator, NCC_b, the unbiased estimator, NCC_u, and the coefficient normalization or SBD [110], NCC_c.

Evaluation of sliding measures: Due to the resemblance of cross-correlation to the sliding version of Pearson's correlation, when time series are z-normalized, the majority of the literature assumes this underlying data normalization [110]. To the best of our knowledge, the performance of cross-correlation as a measure to compare time series under different normalization methods is not well explored. Table 3 reports the performance of the combinations of cross-correlation variants with normalization methods. Specifically, from 32 such combinations (i.e., 4 measures × 8 normalizations), we report only those resulted in an average accuracy higher than the one achieved by Lorentzian (with z-score followed by UnitLength, the last row of the Table 3), the new state-of-the-art lock-step distance measure based on our previous analysis (Section 5). As before, we perform all pairwise comparisons using the Wilcoxon statistical test to ensure we did not miss any accurate combination.

We observe that from all 4 cross-correlation measures, the unbiased estimator, NCC_u, performs worse than all other variants. Specifically, no combination of NCC_u with any of the normalization methods outperforms the Lorentzian distance. In contrast, the remaining 3 variants, NCC, NCC_b, and NCC_c, all include combinations that outperform the baseline. In particular, combinations of NCC and NCC_b with z-score and UnitLength normalizations significantly outperform the baseline. We observe negligible differences between these combinations (i.e., only one dataset is slightly affected). Interestingly, the coefficient normalization variant, NCC_c, outperforms the baseline with 6 combinations of normalization methods. However, only half of these combinations outperform the Lorentzian distance with a statistically significant difference. In all of these measures, we observe a similar trend: all combinations with z-score and UnitLength normalization methods lead to significant improvements. However, for NCC_c, another normalization, namely, MeanNorm, also

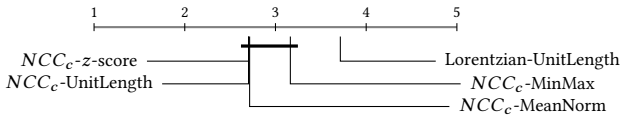


Figure 4: Ranking of different normalization methods for NCC_c based on the average of their ranks across datasets, using Lorentzian with UnitLength as the baseline method.

achieves similar improvement, which is not the case when combined with NCC and NCC_b. Even though NCC, NCC_b, and NCC_c perform similarly in terms of average accuracy, NCC_c appears to be the most robust cross-correlation measure as it leads to improvement over the baseline with more normalization methods than the other variants and for three such combinations the improvement is statistically significant. Among all combinations that outperform the baseline, Wilcoxon suggests no statistically significant differences.

In addition to these pairwise comparisons, we also evaluate the significance of the differences when considered all together. Figure 4 shows the average rank across datasets of five combinations of NCC_c with normalization methods (we excluded Tanh normalization as from Table 3 we observe that, despite the increase in average accuracy, the Lorentzian distance still outperforms this combination in more datasets). Similarly to the pairwise analysis, we observe that combinations with z-score, MeanNorm, and UnitLength normalizations lead to significant improvements according to the Friedman test followed by a post-hoc Nemenyi test to assess the significance of the differences in the ranking. Combinations of NCC_c with AdaptiveScaling or MinMax do not achieve significant improvement. We observe that both statistical evaluation approaches lead to similar conclusions. We omit figures for NCC and NCC_b with similar findings.

For completeness, we report another analysis using ED as the baseline instead of the Lorentzian distance (we omit the figure due to space limitation). NCC_c in combination with z-score, UnitLength, and MeanNorm normalization methods outperform ED but, in contrast to Figure 4, now combinations with AdaptiveScaling and MinMax are also significantly better than ED. This analysis confirms our results in Section 5 that the Lorentzian distance (and other L_1 variants) are more powerful than ED. In addition, our analysis indicates that NCC_c outperforms all lock-step measures with all different normalizations, making it a strong baseline method for time-series comparison.

We now turn our focus to elastic measures and, particularly, to their performance against sliding measures.

7 TIME-SERIES ELASTIC MEASURES

In this section, we study 7 elastic measures, a popular category of measures for time-series comparison.

Distance Measure	Parameters
MSM	$c \in \{0.01, 0.1, 1, 10, 100, 0.05, 0.5, 5, 50, 500\}$
DTW	$\delta \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 100\}$
EDR	$\epsilon \in \{0.001, 0.003, 0.005, 0.007, 0.009, 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$
LCSS	$\delta \in \{5, 10\}$ $\epsilon \in \{0.001, 0.003, 0.005, 0.007, 0.009, 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$
TWE	$\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$ $\nu \in \{0.00001, 0.0001, 0.001, 0.01, 0.1, 1\}$
Swale	$\epsilon \in \{0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}, p \in \{5\}, r \in \{1\}$
Minkowski	$p \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1, 1.3, 1.5, 1.7, 1.9, 2, 3, 5, 7, 9, 11, 13, 15, 17, 20\}$
KDTW	$\gamma \in \{2^{-15}, 2^{-14}, 2^{-13}, 2^{-12}, 2^{-11}, 2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}\}$
GAK	$\gamma \in \{0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$
SINK	$\gamma \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$
RBF	$\gamma \in \{2^{-15}, 2^{-14}, 2^{-13}, 2^{-12}, 2^{-11}, 2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}\}$
GRAIL	$\gamma \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$
RWS	$\gamma \in \{10^{-3}, 3 \cdot 10^{-3}, 10^{-2}, 3 \cdot 10^{-2}, 0.1, 0.14, 0.19, 0.28, 0.39, 0.56, 0.79, 1.12, 1.58, 2.23, 3.16, 4.46, 6.30, 8.91, 10, 31.62, 10^2, 3 \cdot 10^2, 10^3\}, D_{max} = 25$
SIDL	$\lambda \in \{0.1, 1, 10\}, r \in \{0.1, 0.25, 0.5\}$

Table 4: Parameter choices for distance measures..

As discussed earlier, sliding measures find a global alignment by sliding one time series against the other. In contrast, elastic measures create a non-linear mapping between time-series data points to support flexible alignment of different regions. Through this mapping, elastic measures permit time series to "stretch" or "shrink" their observations to improve time-series matching. Most elastic measures rely on dynamic programming to find this mapping efficiently by defining recursive formulas over a m-by-m matrix M that contains in each cell the ED (or some other lock-step measure) between every point of one time series against every point of another time series. In general, the goal of different elastic measures in the literature is to employ different strategies to find a *warping path*, $W = \{w_1, \ldots, w_k\}$, with $k \geq m$, a contiguous set of matrix cells that shows the mapping of every point of one time series to one, more, or none of the points of the other time series. To improve the efficiency and the accuracy of elastic measures, it is a common practice to introduce constraints (in the form of parameters) to guide the warping path to visit only a subset of cells in M or to determine above which distance threshold two points should match.

Distance Measure	Parameter Tuning	Better	Average Accuracy	>	=	<
MSM	LOOCCV	✔	0.7628	86	3	39
	$c = 0.5$	✔	0.7627	89	2	37
TWE	LOOCCV	✔	0.7632	85	4	39
	$\lambda = 1, \nu = 0.0001$	✔	0.7622	89	4	35
DTW	LOOC	✔	0.7519	75	16	37
	$\delta = 100$	✘	0.7248	54	10	64
	$\delta = 10$	✘	0.7372	64	6	58
EDR	LOOCCV	✔	0.7485	74	8	46
	$\epsilon = 0.1$	✘	0.7202	62	5	61
Swale	LOOCCV	✔	0.7499	72	8	48
	$\epsilon = 0.2$	✘	0.7229	63	4	61
ERP	-	✔	0.7488	77	5	46
LCSS	LOOCCV	✘	0.7398	66	6	56
	$\delta = 5, \epsilon = 0.2$	✘	0.7160	63	2	63
NCC$_c$	-	-	**0.7309**	-	-	-

Table 5: Comparison of elastic measures against NCC_c. "Parameter Tuning" indicates supervised or unsupervised tuning. "Better" denotes that the measure outperforms the baseline with statistical significance. "Average Accuracy" shows the mean accuracy achieved across 128 datasets. The last three columns indicate the datasets over which a measure is better (">"), equal ("="), or worse ("<") than the baseline.

The first elastic measure, DTW [126, 127], was proposed as a speech recognition tool and, later, it was introduced in the time-series literature as a suitable approach for time-series comparison [18]. DTW finds the warping path that minimizes the distances between all data points. In the original form, DTW is parameter-free, however, many approaches have been proposed to define *bands* (i.e., the shape of the subset cells of matrix M that the warping path is permitted to visit) and the *width or window* (i.e., size) of the bands. We use the Sakoe-Chiba band [127], which is the most frequently used in practice [45], and we tune the window δ using parameters shown in Table 4. For example, a value $\delta = 10$ indicates a window size 10% of the time-series length.

The Longest Common Subsequence (LCSS) distance is another type of elastic measure that was derived from the idea of edit-distances for characters. Specifically, LCSS introduces a parameter ϵ that serves as a threshold to determine when two points of time series should match [7, 141]. Similarly to DTW, LCSS also constrains the warping window by introducing an additional parameter δ [141]. Edit Distance on Real sequence (EDR) distance [28] is another edit-distance-based measure that similarly to LCSS, uses a parameter ϵ to quantify the distance of points as 0 or 1. EDR also introduces penalties for gaps between matched subsequences. Edit Distance with Real Penalty (ERP) distance [27] is a measure that bridges DTW and EDR distance measures by more carefully computing the distance between gaps of time series.

Differently than the previous approaches, the Sequence Weighted Alignment model (Swale) [100] proposes a model to compute the similarity of time series using rewards for matching points and penalties for gaps. Apart from a threshold ϵ parameter, Swale also requires parameters for the reward r and the penalty p. The Move–split–merge (MSM) distance [137] is another elastic measure based on edit-distance but in contrast to DTW, LCSS, and EDR, MSM is a metric. MSM uses a set of operations to replace, insert, or delete values in time series to improve their matching. Finally, Time Warp Edit (TWE) distance [92] is a measure that combines merits from LCSS and DTW. TWE introduces a stiffness parameter ν to control the warping but at the same point it also penalizes matched points (parameter λ).

For each one of these 7 elastic measures, several variants and extensions have been proposed in the literature. For example, Derivative DTW (DDTW) [60] combines raw time series with their first-order differences (derivatives). Complexity Invariant distance (CID) [16] is a weighting scheme to compensate for differences in the complexity of two time series. Finally, Weighted DTW (WDTW) [68] adds a penalty to the warping path of DTW. All of these approaches describe extensions that can potentially be used in combination with all previously described elastic measures. Importantly, each of these extensions often introduces additional parameters that require tuning. To avoid an explosion of evaluated approaches, we do not include such variants in our analysis. An excellent recent study [11] focusing on time-series classification has evaluated several of these approaches (and did not identify significant improvements from their use).

Evaluation of elastic vs. sliding measures: With the introduction of the 7 elastic measures we are now in position to evaluate their performance against sliding measures, an experiment that has been omitted in all previous studies [11, 45]. Table 5 compares the classification accuracy of elastic measures against the accuracy of NCC_c, the state-of-the-art sliding measure based on our previous experiment in Section 6. As we did not observe significant differences from using different normalization methods for NCC_c, for all subsequent experiments we always use z-normalized time series. We also do that so that our results are closely comparable to those reported previously [45, 144]. Table 5 includes two experimental settings, one supervised and one unsupervised. In the supervised setting, the necessary parameters of the elastic measures are tuned on the training set using cross-validation with a leave-one-out-classifier (LOOCCV), as noted in Section 3. In the unsupervised setting, we consult the parameters selected through LOOCCV to identify a set of parameters that perform well on average across all datasets. This step involves several trial-and-error attempts and a post-hoc analysis of the results (i.e., we observe the

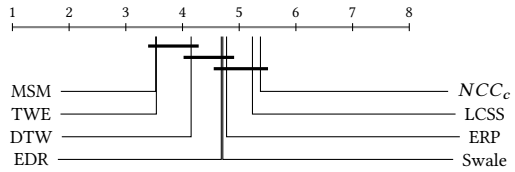


Figure 5: Ranking of elastic and sliding distance measures based on the average of their ranks across datasets, using supervised tuning for their parameters.

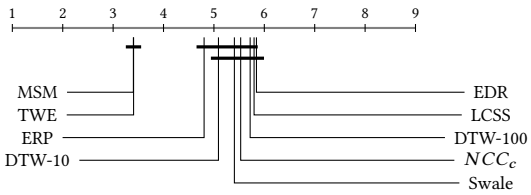


Figure 6: Ranking of elastic and sliding distance measures based on the average of their ranks across datasets, using unsupervised tuning for their parameters.

accuracy on the test sets). Even though this is an unfair advantage against parameter-free approaches, such as NCC_c, we perform this post-hoc analysis to ensure that elastic measures are not misrepresented in such unsupervised scenario and that a domain expert could potentially identify such parameters without supervised tuning. For some measures, such as DTW, that was an easy step, however, for others, we had to perform multiple attempts to identify parameters that achieve a competitive accuracy on average.

From Table 5, we observe that when parameters are selected under supervised settings (lines with LOOCCV tuning) all elastic measures significantly outperform NCC_c with one exception, the LCSS measure, which marginally outperforms NCC_c but the difference is not statistically significant according to Wilcoxon. However, the picture is different for the unsupervised scenario. Specifically, we observe that 4 out of the 7 elastic measures do not outperform NCC_c. Interestingly, LCSS, EDR, and DTW (with $\delta = 100$, which resembles an equivalent parameter-free measure to NCC_c) are slightly worse. MSM, TWE, and ERP on the other side significantly outperform NCC_c in the unsupervised setting as well. Among all elastic measures, ERP is the only parameter-free measure that achieves significantly better accuracy than NCC_c in both supervised and unsupervised settings.

To better understand the performance of elastic measures against NCC_c, in addition to all previous pairwise statistical comparisons, we also evaluate the significance of the differences when considered all together. Specifically, Figure 5 shows the average ranks of the elastic measures in the supervised setting and Figure 6 shows the average ranks in the unsupervised setting. The ranking of measures in Figure 5 contradicts some of the pairwise results observed in Table 5. Specifically, we observe that even under supervised settings, 4 out of the 7 elastic measures, namely, LCSS, ERP, EDR, and Swale, do not achieve significantly better performance than NCC_c. The results for MSM, TWE, and DTW, are consistent in both statistical evaluations. For the unsupervised setting, both statistical evaluation approaches agree to an extent. In particular, Figure 6 shows clearly that MSM and TWE outperform NCC_c. However, the remaining 5 elastic measures perform similarity to NCC_c. Interestingly, as we observed in Table 5, NCC_c slightly outperforms 3 elastic measures.

To validate our findings, we repeat the analysis (we omit figures due to space limitation) and evaluate the significance of the differences when we consider all elastic measures together (i.e., excluding NCC_c). Specifically, we observe that Swale, ERP, EDR, and LCSS do not outperform DTW-10 with statistically significant difference. Interestingly, the supervised LCSS is slightly worse than the unsupervised DTW-10. ERP, which under pairwise evaluation appears to significantly outperform DTW-10, when all measures are considered together, both appear to achieve comparable performance. MSM, TWE, and DTW also perform similarly and all three supervised measures outperform DTW-10. However, under unsupervised settings, we observe that MSM and TWE significantly outperform all other elastic measures.

Debunking $\mathcal{M}3$ and $\mathcal{M}4$: Our comprehensive evaluation shows clear evidence that sliding measures are strong baselines that most elastic measures do not manage to outperform either in supervised or unsupervised settings, which debunks $\mathcal{M}3$. Specifically, from all 5 elastic measures evaluated in the decade-old study [45], namely, LCSS, Swale, EDR, ERP, and DTW, only DTW significantly outperforms cross-correlation under the supervised scenario. In the unsupervised setting, none of the 5 measures outperforms cross-correlation and, interestingly, several of them perform slightly worse. This is a remarkable finding, showing that the simplest type of alignment between time series is very effective and it should have served as a baseline method for elastic measures. Only MSM and TWE, two measures that appeared after [45] show promising results and outperform cross-correlation with statistically significant differences in both supervised and unsupervised settings. Importantly, MSM is the only method that significantly outperforms DTW under supervised settings (according to Wilcoxon) and, under unsupervised settings, both MSM and TWE significantly outperform DTW (with both statistical tests validating this result). Therefore, there is clear evidence that the widely popular DTW is no longer the best elastic distance measure, which debunks $\mathcal{M}4$.

8 TIME-SERIES KERNEL MEASURES

Until now, our analysis focused on three categories of distance measures, namely, lock-step, sliding, and elastic measures, with the goal to provide answers to the four-long

Distance Measure	Parameter Tuning	Better	Average Accuracy	>	=	<
KDTW	LOOCCV	✔	0.7668	89	7	32
	$\gamma = 0.125$	✔	0.7501	85	7	36
GAK	LOOCCV	✔	0.7474	79	9	40
	$\gamma = 0.1$	✔	0.7387	72	10	46
SINK	LOOCCV	✔	0.7469	69	13	46
	$\gamma = 5$	✘	0.7396	58	11	59
RBF	LOOCCV	✪	0.6869	20	29	79
	$\gamma = 2$	✪	0.6613	13	29	86
NCC$_c$	-	-	**0.7309**	-	-	-

Table 6: Comparison of kernel measures. "Better" denotes that the distance measure outperforms the baseline with statistical significance. "Average Accuracy" shows the mean accuracy achieved across 128 datasets. The last three columns indicate the datasets over which a measure is better (">"), equal ("="), or worse ("<") than the baseline.

standing misconceptions that we discussed in Section 2. Recently, kernel functions [130, 131], a different category of similarity measures, have started to receive attention due to their competitive performance [1]. In contrast to all previously described measures, kernel functions must satisfy the positive semi-definiteness property (p.s.d) [132]. The precise definition is out of the scope of this work (we refer the reader to recent papers for a detailed review [1, 109]) but in simple terms, a function is p.s.d. if the similarity matrix, which contains all pairwise similarity values, has positive eigenvalues. This important property results in convex solutions for several learning tasks involving kernels [35]. In this section, we study 4 representative kernel functions and evaluate their performance against sliding and elastic measures.

Specifically, the first kernel we consider is the Radial Basis Function (RBF) [37], a general purpose kernel function that internally exploits ED but maps data into a high-dimensional space where their separation is easier. To capture similarities between the shifted versions of time series, [142] proposed a sliding kernel to consider all possible alignments between time-series. We include a recently proposed variant of this kernel, namely, SINK, that has achieved competitive results to NCC_c and DTW [109]. Finally, we include two elastic kernel functions, the Global Alignment Kernel (GAK) [38] and Dynamic Time Warping Kernel (KDTW) [93].

Evaluation of kernel functions: Having introduced the 4 kernel functions, we are now in position to evaluate their performance against sliding and elastic measures. Table 6 compares the classification accuracy of kernel functions against the accuracy of NCC_c. Table 4 lists the parameters considered in each function. As before, we consider both supervised and unsupervised settings. In the supervised setting, we observe that all kernel functions significantly outperform NCC_c with the exception of RBF, which is significantly worse

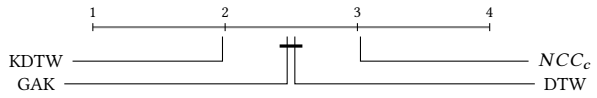


Figure 7: Ranking of kernel measures based on the average of their ranks across datasets, using supervised tuning.

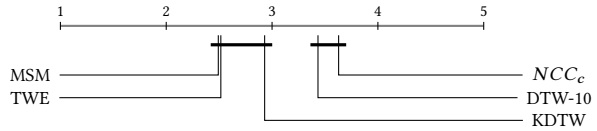


Figure 8: Ranking of kernel measures based on the average of their ranks across datasets, using unsupervised tuning.

(marked with ✪). In the unsupervised settings, KDTW and GAK significantly outperform NCC_c, as before, but SINK achieves comparable performance without outperforming NCC_c. To better understand the performance of KDTW and GAK, which appear to be the strongest kernel functions, we also evaluate the significance of the differences when considered together with all elastic and sliding measures. Figure 7 presents the results for supervised settings and Figure 8 for unsupervised settings. We have omitted elastic measures that based on the earlier analysis did not show competitive results. We observe that GAK achieves comparable performance to DTW under both settings. However, KDTW, significantly outperforms DTW in both unsupervised and superivsed settings. This is in contrast to TWE and MSM measures that were significantly better only under the unsupervised settings. To the best of our knowledge, this is the first time that a kernel function is reported to outperform DTW in both settings. We have also verified this under Wilcoxon.

9 TIME-SERIES EMBEDDING MEASURES

Previously, we studied approaches that directly exploit a kernel function or a distance measure to compare time series. In this section, we study 4 embedding measures, which are alternative approaches that employ a similarity measure only to construct new representations [1]. These representations are similarity-preserving as the comparison of two representations with ED approximates the comparison of the corresponding original time series with the employed similarity measure used to construct the representations.

We consider 4 approaches to construct embedding measures (i.e., ED over learned representations). Specifically, we consider the Generic RepresentAtIon Learning (GRAIL) framework, which employs the SINK kernel [109], the Shift-invariant Dictionary Learning (SIDL) method, which preserves alignment between time series [163], the Similarity Preserving Representation Learning method (SPIRAL), which employs DTW [82], and the Random Warping Series (RWS), which preserves the GAK kernel [151].

Distance Measure	Parameter Tuning	Better	Average Accuracy	>	=	<
GRAIL	LOOCCV	✘	0.7407	56	8	64
RWS	LOOCCV	✪	0.7128	45	3	80
SPIRAL	-	✪	0.6494	26	4	98
SIDL	LOOCCV	✪	0.5759	12	1	115
NCC$_c$	-	-	**0.7309**	-	-	-

Table 7: Comparison of embedding measures. "Better" denotes that the distance measure outperforms the baseline with statistical significance. "Average Accuracy" shows the mean accuracy achieved across 128 datasets. The last three columns indicate the datasets over which a measure is better (">"), equal ("="), or worse ("<") than the baseline.

Evaluation of embedding measures: For all approaches, we follow [109] and tune required parameters using the recommended values from their corresponding papers. We construct representations of same length (100) for fairness. Table 7 presents the results against NCC$_c$. We observe that GRAIL, is the only framework that constructs robust representations that when ED is used for comparison (under the 1-NN settings), it achieves similar performance to NCC$_c$, but without significant difference. All other embedding measures perform significantly worse (marked with ✪) and none of the embedding measures outperform DTW. We note, however, that embedding measures (as well as kernel methods), achieve much higher accuracy under different evaluation frameworks (e.g., with SVM classifiers), as shown in [109]. We leave such extensive analysis for future work.

10 ACCURACY-TO-RUNTIME ANALYSIS

Until now, we performed an extensive evaluation of distance measures based on their accuracy results. However, it is also important to understand the cost associated with each one of these distance measures. In Figure 9, we summarize the accuracy-to-runtime performance of the most prominent measures. The runtime performance includes only inference time (i.e., evaluation on the testing sets). We observe that ED, and all lock-step measures (omitted), are the fastest, but achieve relatively low accuracy (all these measures have $O(m)$ runtime cost). NCC$_c$ and SINK, two methods that rely on the classic cross-correlation measure, provide a good trade-off between runtime and accuracy in comparison to ED (these measures have $O(m \log m)$ runtime cost). We observe that all other elastic or kernel methods require substantially higher runtime cost to achieve comparable accuracy results to NCC_c (these measures have $O(m^2)$ runtime cost). We also observe that embedding measures show promise as they can be both efficient and accurate. We note that for elastic measures, the runtime cost can be substantially improved with the use of lower bounding measures (i.e., efficient measures to prune the expensive pairwise comparisons). To the best

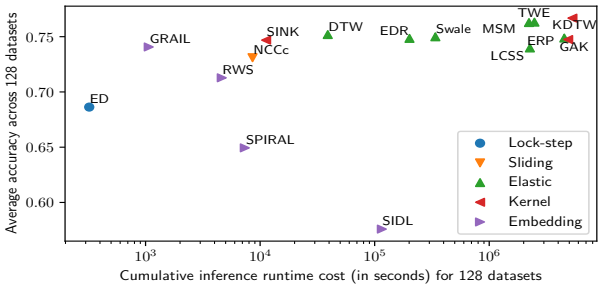


Figure 9: Accuracy-to-runtime comparison.

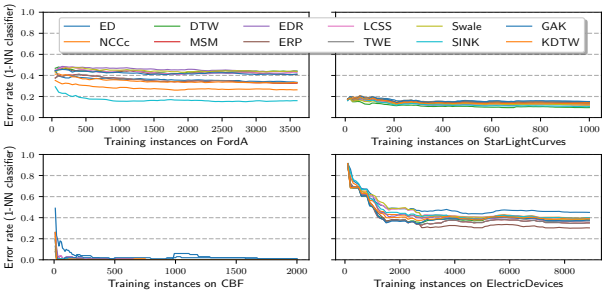


Figure 10: Error rates with increasingly larger datasets.

of our knowledge, for NCC$_c$ and the kernel methods, no lower bounding measures exist and, therefore, we leave such extensive runtime analysis for future work. Finally, Figure 10 suggests that with increasingly larger dataset sizes the classification error of ED may not always converge to the error of more accurate measures, at least not always with the same speed of convergence, which highlights the importance of considering measures other than ED (see Section 2).

11 CONCLUSION

We presented a comprehensive evaluation to validate the performance of 71 distance measures. Our study not only debunked four long-standing misconceptions in the time-series literature but also established new state-of-the-art results for lock-step, sliding, elastic, kernel, and embedding measures. Our findings prepare the ground to facilitate further development of distance measures with implications to virtually every task. With the new knowledge in place, several new challenges open that we hope to sparkle new research directions. For example, identifying more accurate normalizations will result in substantial improvement for many tasks. There is a lack of methods for unsupervised tuning of parameters. Finally, embedding measures might show the most promise considering their runtime-to-accuracy trade-off.

Acknowledgments: We thank the anonymous reviewers whose comments have greatly improved this manuscript. We also thank Themis Palpanas and Eamonn Keogh for useful discussions after acceptance. This research was supported in part by a Google DAPA Research Award, gifts from NetApp, Cisco Systems, and Exelon Utilities, and an NSF Award CCF-1139158. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation.

REFERENCES

[1] Amaia Abanda, Usue Mori, and Jose A Lozano. 2019. A review on distance based time series classification. *Data Mining and Knowledge Discovery* 33, 2 (2019), 378–412.

[2] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. 1993. Efficient Similarity Search In Sequence Databases. In *FODO*. 69–84.

[3] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim. 1995. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceeding of the 21th International Conference on Very Large Data Bases*. Citeseer, 490–501.

[4] Shadab Alam, Franco D Albareti, Carlos Allende Prieto, Friedrich Anders, Scott F Anderson, Timothy Anderton, Brett H Andrews, Eric Armengaud, Éric Aubourg, Stephen Bailey, et al. 2015. The eleventh and twelfth data releases of the Sloan Digital Sky Survey: final data from SDSS-III. *The Astrophysical Journal Supplement Series* 219, 1 (2015), 12.

[5] Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. 2003. Discovering clusters in motion time-series data. In *CVPR*. 375–381.

[6] Francisco Martinez Alvarez, Alicia Troncoso, Jose C Riquelme, and Jesus S Aguilar Ruiz. 2010. Energy time series forecasting based on pattern sequence similarity. *IEEE Transactions on Knowledge and Data Engineering* 23, 8 (2010), 1230–1243.

[7] Henrik André-Jönsson and Dushan Z Badal. 1997. Using signature files for querying time-series data. In *European Symposium on Principles of Data Mining and Knowledge Discovery*. Springer, 211–220.

[8] Johannes Aßfalg, Hans-Peter Kriegel, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. 2006. Similarity search on time series based on threshold queries. In *International Conference on Extending Database Technology*. Springer, 276–294.

[9] 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.

[10] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).

[11] 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.

[12] Anthony J Bagnall and Gareth J Janacek. 2004. Clustering time series from ARMA models with clipped data. In *KDD*. 49–58.

[13] Ziv Bar-Joseph. 2004. Analyzing time series gene expression data. *Bioinformatics* 20, 16 (2004), 2493–2503.

[14] 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.

[15] 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.

[16] 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.

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

[18] Donald J Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series.. In *AAAI Workshop on KDD*. 359–370.

[19] Bharat B Biswal, Maarten Mennes, Xi-Nian Zuo, Suril Gohel, Clare Kelly, Steve M Smith, Christian F Beckmann, Jonathan S Adelstein, Randy L Buckner, Stan Colcombe, et al. 2010. Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4734–4739.

[20] R Bracewell. 1965. Pentagram notation for cross correlation. The Fourier transform and its applications. *New York: McGraw-Hill* 46 (1965), 243.

[21] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.

[22] Peter J Brockwell and Richard A Davis. 2016. *Introduction to time series and forecasting*. springer.

[23] Lisa Gottesfeld Brown. 1992. A survey of image registration techniques. *ACM computing surveys (CSUR)* 24, 4 (1992), 325–376.

[24] Yuhan Cai and Raymond Ng. 2004. Indexing spatio-temporal trajectories with Chebyshev polynomials. In *SIGMOD*. 599–610.

[25] 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.

[26] Sung-Hyuk Cha. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *City* 1, 2 (2007), 1.

[27] Lei Chen and Raymond Ng. 2004. On the marriage of Lp-norms and edit distance. In *VLDB*. 792–803.

[28] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *SIGMOD*. 491–502.

[29] Qiuxia Chen, Lei Chen, Xiang Lian, Yunhao Liu, and Jeffrey Xu Yu. 2007. Indexable PLA for efficient similarity search. In *VLDB*. 435–446.

[30] Yueguo Chen, Mario A Nascimento, Beng Chin Ooi, and Anthony KH Tung. 2007. Spade: On shape-based pattern detection in streaming time series. In *ICDE*. 786–795.

[31] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. 2003. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 493–498.

[32] Kelvin Kam Wing Chu and Man Hon Wong. 1999. Fast time-series searching with scaling and shifting. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. Citeseer, 237–248.

[33] Richard Cole, Dennis Shasha, and Xiaojian Zhao. 2005. Fast window correlations over uncooperative time series. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 743–749.

[34] James W Cooley and John W Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.* 19, 90 (1965), 297–301.

[35] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[36] 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.

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

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

[39] Michele Dallachiesa, Besmira Nushi, Katsiaryna Mirylenka, and Themis Palpanas. 2012. Uncertain time-series similarity: Return to the basics. *Proceedings of the VLDB Endowment* 5, 11 (2012), 1662–1673.

[40] 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.

[41] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

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

[43] Michel-Marie Deza and Elena Deza. 2006. *Dictionary of distances*. Elsevier.

[44] Michel Marie Deza and Elena Deza. 2009. Encyclopedia of distances. In *Encyclopedia of distances*. Springer, 1–583.

[45] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment* 1, 2 (2008), 1542–1552.

[46] 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.

[47] Alejandro Domínguez. 2015. A history of the convolution operation [Retrospectroscope]. *IEEE pulse* 6, 1 (2015), 38–49.

[48] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2018. The lernaean hydra of data series similarity search: An experimental evaluation of the state of the art. *Proceedings of the VLDB Endowment* 12, 2 (2018), 112–127.

[49] Jason Ernst and Ziv Bar-Joseph. 2006. STEM: a tool for the analysis of short time series gene expression data. *BMC bioinformatics* 7, 1 (2006), 191.

[50] Philippe Esling and Carlos Agon. 2012. Time-series data mining. *ACM Computing Surveys (CSUR)* 45, 1 (2012), 12.

[51] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast Subsequence Matching in Time-series Databases. In *SIGMOD*. 419–429.

[52] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research* 15, 1 (2014), 3133–3181.

[53] Elias Frentzos, Kostas Gratsias, and Yannis Theodoridis. 2007. Index-based most similar trajectory search. In *ICDE*. 816–825.

[54] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Amer. Statist. Assoc.* 32 (1937), 675–701.

[55] Daniel G Gavin, W Wyatt Oswald, Eugene R Wahl, and John W Williams. 2003. A statistical approach to evaluating distance metrics and analog assignments for pollen records. *Quaternary Research* 60, 3 (2003), 356–367.

[56] Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, and Rajeev Motwani. 2000. Mining the stock market: Which measure is best. In *Proc. of the 6th ACM SIGKDD*. 487–496.

[57] Rafael Giusti and Gustavo EAPA Batista. 2013. An Empirical Comparison of Dissimilarity Measures for Time Series Classification. In *BRACIS*. 82–88.

[58] Steve Goddard, Sherri K Harms, Stephen E Reichenbach, Tsegaye Tadesse, and William J Waltman. 2003. Geospatial decision support for drought risk management. *Commun. ACM* 46, 1 (2003), 35–37.

[59] Dina Q Goldin and Paris C Kanellakis. 1995. On similarity queries for time-series data: constraint specification and implementation. In *International Conference on Principles and Practice of Constraint Programming*. Springer, 137–153.

[60] Tomasz Górecki and Maciej Łuczak. 2013. Using derivatives in time series classification. *Data Mining and Knowledge Discovery* 26, 2 (2013), 310–331.

[61] 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*. ACM, 379–386.

[62] Joel Grus. 2019. *Data science from scratch: first principles with python*. O'Reilly Media.

[63] 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.

[64] 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.

[65] Rie Honda, Shuai Wang, Tokio Kikuchi, and Osamu Konishi. 2002. Mining of moving objects from time-series images and its application to satellite weather imagery. *Journal of Intelligent Information Systems* 19, 1 (2002), 79–93.

[66] Bing Hu, Yanping Chen, and Eamonn Keogh. 2013. Time Series Classification under More Realistic Assumptions. In *SDM*. 578–586.

[67] 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.

[68] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. 2011. Weighted dynamic time warping for time series classification. *Pattern Recognition* 44, 9 (2011), 2231–2240.

[69] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. 2001. Distance measures for effective clustering of ARIMA time-series. In *ICDM*. 273–280.

[70] Kunio Kashino, Gavin Smith, and Hiroshi Murase. 1999. Time-series active search for quick retrieval of audio and video. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, Vol. 6. IEEE, 2993–2996.

[71] Shrikant Kashyap and Panagiotis Karras. 2011. Scalable knn search on vertically stored time series. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1334–1342.

[72] Eamonn Keogh. 2006. A decade of progress in indexing and mining large time series databases. In *VLDB*. 1268–1268.

[73] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. In *SIGMOD*. 151–162.

[74] Eamonn Keogh and Jessica Lin. 2005. Clustering of time-series subsequences is meaningless: Implications for previous and future research. *Knowledge and Information Systems* 8, 2 (2005), 154–177.

[75] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7, 3 (2005), 358–386.

[76] Chan Kin-pong and Fu Ada. 1999. Efficient Time Series Matching by Wavelets. In *ICDE*. 126–133.

[77] S Knieling, J Niediek, E Kutter, J Bostroem, CE Elger, and F Mormann. 2017. An online adaptive screening procedure for selective neuronal responses. *Journal of neuroscience methods* 291 (2017), 36–42.

[78] Flip Korn, H. V. Jagadish, and Christos Faloutsos. 1997. Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In *SIGMOD*. 289–300.

[79] Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[81] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer, 9–48.

[82] Qi Lei, Jinfeng Yi, Roman Vaculin, Lingfei Wu, and Inderjit S Dhillon. 2017. Similarity preserving representation learning for time series analysis. *arXiv preprint arXiv:1702.03584* (2017).

[83] Chung-Sheng Li, Philip S. Yu, and Vittorio Castelli. 1996. Hierarchyscan: A hierarchical similarity search algorithm for databases of long sequences. In *ICDE*. IEEE, 546–553.

[84] Xiang Lian, Lei Chen, Jeffrey Xu Yu, Guoren Wang, and Ge Yu. 2007. Similarity match over high speed time-series streams. In *ICDE*. 1086–1095.

[85] Jessica Lin, Michail Vlachos, Eamonn Keogh, and Dimitrios Gunopulos. 2004. Iterative incremental clustering of time series. In *EDBT*. 106–122.

[86] 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.

[87] Jason Lines and Anthony Bagnall. 2015. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery* 29, 3 (2015), 565–592.

[88] 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.

[89] Helmut Lütkepohl, Markus Krätzig, and Peter CB Phillips. 2004. *Applied time series econometrics*. Cambridge university press.

[90] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P Sheth. 2017. Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks* (2017).

[91] Rosario N Mantegna. 1999. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems* 11, 1 (1999), 193–197.

[92] 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.

[93] Pierre-François Marteau and Sylvie Gibet. 2014. On recursive edit distance kernels with application to time series classification. *IEEE transactions on neural networks and learning systems* 26, 6 (2014), 1121–1133.

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

[95] Richard McCleary, Richard A Hay, Erroll E Meidinger, and David McDowall. 1980. *Applied time series analysis for the social sciences*. Sage Publications Beverly Hills, CA.

[96] Vasileios Megalooikonomou, Qiang Wang, Guo Li, and Christos Faloutsos. 2005. A multiresolution symbolic representation of time series. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*. IEEE, 668–679.

[97] Katsiaryna Mirylenka, Vassilis Christophides, Themis Palpanas, Ioannis Pefkianakis, and Martin May. 2016. Characterizing home device usage from wireless traffic time series.

[98] Katsiaryna Mirylenka, Michele Dallachiesa, and Themis Palpanas. 2017. Data series similarity using correlation-aware measures. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. 1–12.

[99] A Morales-Esteban, Francisco Martínez-Álvarez, A Troncoso, JL Justo, and Cristina Rubio-Escudero. 2010. Pattern recognition to forecast seismic time series. *Expert Systems with Applications* 37, 12 (2010), 8333–8342.

[100] Michael D Morse and Jignesh M Patel. 2007. An efficient and accurate method for evaluating time series similarity. In *SIGMOD*. 569–580.

[101] Abdullah Mueen, Eamonn Keogh, and Neal Young. 2011. Logical-shapelets: An expressive primitive for time series classification. In *KDD*. 1154–1162.

[102] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. 2009. Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 473–484.

[103] Abdullah Mueen, Yan Zhu, Michael Yeh, Kaveh Kamgar, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. 2017. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

[104] Peter Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph.D. Dissertation. Princeton University.

[105] Themis Palpanas. 2015. Data series management: the road to big sequence analytics. *ACM SIGMOD Record* 44, 2 (2015), 47–52.

[106] Themis Palpanas. 2016. Big sequence management: A glimpse of the past, the present, and the future. In *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, 63–80.

[107] Panagiotis Papapetrou, Vassilis Athitsos, Michalis Potamias, George Kollios, and Dimitrios Gunopulos. 2011. Embedding-based subsequence matching in time-series databases. *TODS* 36, 3 (2011), 17.

[108] John Paparrizos. 2019. 2018 UCR Time-Series Archive: Backward Compatibility, Missing Values, and Varying Lengths. https://github.com/johnpaparrizos/UCRArchiveFixes.

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

[110] 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*. ACM, 1855–1870.

[111] John Paparrizos and Luis Gravano. 2017. Fast and Accurate Time-Series Clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 8.

[112] Athanasios Papoulis. 1962. *The Fourier integral and its applications*. McGraw-Hill.

[113] 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.

[114] François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. 2014. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE international conference on data mining*. IEEE, 470–479.

[115] François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. 2016. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowledge and Information Systems* 47, 1 (2016), 1–26.

[116] 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.

[117] Davood Rafiei and Alberto Mendelzon. 1997. Similarity-based queries for time series data. In *ACM SIGMOD Record*, Vol. 26. ACM, 13–25.

[118] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*. 262–270.

[119] Chotirat Ann Ralanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. 2005. Mining time series data. In *Data mining and knowledge discovery handbook*. Springer, 1069–1103.

[120] Chotirat Ann Ratanamahatana and Eamonn Keogh. 2004. Making time-series classification more accurate using learned constraints. In *SDM*. 11–22.

[121] Usman Raza, Alessandro Camerra, Amy L Murphy, Themis Palpanas, and Gian Pietro Picco. 2015. Practical data prediction for real-world wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 8 (2015), 2231–2244.

[122] John Rice. 2006. *Mathematical statistics and data analysis*. Cengage Learning.

[123] Joshua S Richman and J Randall Moorman. 2000. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology* 278, 6 (2000), H2039–H2049.

[124] Kexin Rong, Clara E Yoon, Karianne J Bergen, Hashem Elezabi, Peter Bailis, Philip Levis, and Gregory C Beroza. 2018. Locality-sensitive hashing for earthquake detection: A case study of scaling data-driven science. *Proceedings of the VLDB Endowment* 11, 11 (2018), 1674–1687.

[125] Eduardo J Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. 2012. Correlating financial time series with micro-blogging activity. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 513–522.

[126] Hiroaki Sakoe and Seibi Chiba. 1971. A dynamic programming approach to continuous speech recognition. In *ICA*. 65–69.

[127] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.

[128] Yasushi Sakurai, Spiros Papadimitriou, and Christos Faloutsos. 2005. Braid: Stream mining through group lag correlations. In *SIGMOD*. ACM, 599–610.

[129] Patrick Schäfer and Mikael Högqvist. 2012. SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *Proceedings of the 15th International Conference on Extending Database Technology*. ACM, 516–527.

[130] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1997. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*. Springer, 583–588.

[131] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation* 10, 5 (1998), 1299–1319.

[132] Bernhard Schölkopf and Alexander J Smola. 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[133] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2015. Time series anomaly discovery with grammar-based compression.. In *Edbt*. 481–492.

[134] Dennis Shasha. 1999. Tuning time series queries in finance: Case studies and recommendations. *IEEE Data Eng. Bull.* 22, 2 (1999), 40–46.

[135] Jin Shieh and Eamonn Keogh. 2008. i SAX: indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 623–631.

[136] Yutao Shou, Nikos Mamoulis, and David Cheung. 2005. Fast and exact warping of time series using adaptive segmental approximations. *Machine Learning* 58, 2-3 (2005), 231–267.

[137] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. 2013. The move-split-merge metric for time series. *TKDE* 25, 6 (2013), 1425–1438.

[138] Ruey S Tsay. 2014. Financial Time Series. *Wiley StatsRef: Statistics Reference Online* (2014), 1–23.

[139] Kuniaki Uehara and Mitsuomi Shimada. 2002. Extraction of primitive motion and discovery of association rules from human motion data. In *Progress in Discovery Science*. Springer, 338–348.

[140] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2006. Indexing multidimensional time-series. *The VLDB Journal* 15, 1 (2006), 1–20.

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

[142] 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.

[143] Hao Wang, Yilun Cai, Yin Yang, Shiming Zhang, and Nikos Mamoulis. 2014. Durable Queries over Historical Time Series. *TKDE* 26, 3 (2014), 595–607.

[144] 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* (2013), 1–35.

[145] Xiaozhe Wang, Kate Smith, and Rob Hyndman. 2006. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery* 13, 3 (2006), 335–364.

[146] Yang Wang, Peng Wang, Jian Pei, Wei Wang, and Sheng Huang. 2013. A data-adaptive and dynamic segmentation index for whole matching on time series. *Proceedings of the VLDB Endowment* 6, 10 (2013), 793–804.

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

[148] Peter J Webster, Greg J Holland, Judith A Curry, and H-R Chang. 2005. Changes in tropical cyclone number, duration, and intensity in a warming environment. *Science* 309, 5742 (2005), 1844–1846.

[149] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* (1945), 80–83.

[150] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[151] Lingfei Wu, Ian En-Hsu Yen, Jinfeng Yi, Fangli Xu, Qi Lei, and Michael Witbrock. 2018. Random Warping Series: A Random Features Method for Time-Series Embedding. In *AISTATS*. 793–802.

[152] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. 2006. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 1033–1040.

[153] Yimin Xiong and Dit-Yan Yeung. 2002. Mixtures of ARMA models for model-based time series clustering. In *ICDM*. 717–720.

[154] Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *WSDM*. 177–186.

[155] Dragomir Yankov, Eamonn Keogh, and Umaa Rebbapragada. 2008. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. *Knowledge and Information Systems* 17, 2 (2008), 241–262.

[156] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 947–956.

[157] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins

for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1317–1322.

[158] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Zachary Zimmerman, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2018. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery* 32, 1 (2018), 83–123.

[159] Mi-Yen Yeh, Kun-Lung Wu, Philip S Yu, and Ming-Syan Chen. 2009. PROUD: a probabilistic approach to processing similarity queries over uncertain data streams. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, 684–695.

[160] Byoung-Kee Yi and Christos Faloutsos. 2000. Fast time sequence indexing for arbitrary Lp norms. VLDB.

[161] Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. 2012. Clustering Time Series Using Unsupervised-Shapelets. In *ICDM*. 785–794.

[162] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. 2006. *Similarity search: the metric space approach*. Vol. 32. Springer Science & Business Media.

[163] Guoqing Zheng, Yiming Yang, and Jaime Carbonell. 2016. Efficient shift-invariant dictionary learning. In *SIGKDD*. ACM, 2095–2104.

[164] Kostas Zoumpatianos, Stratos Idreos, and Themis Palpanas. 2016. ADS: the adaptive data series index. *The VLDB Journal—The International Journal on Very Large Data Bases* 25, 6 (2016), 843–866.