Generalized Dynamic Time Warping: Unleashing the Warping Power Hidden in Point-Wise Distances

Abstract—**Domain-specific distance measures preferred by analysts for exploring similarities among time series tend to be "point-to-point" distances. Unfortunately, this point-wise nature limits their ability to perform meaningful comparisons between sequences of different lengths and with mis-alignments. Analysts instead require "elastic" alignment tools such as Dynamic Time Warping (DTW) which enable this kind of flexible comparisons. However, the existing alignment tools are limited in that they do not incorporate most of the suitable distances for specific applications. To address this shortcoming, our work introduces the first conceptual framework called Generalized Dynamic Time Warping (GDTW) that supports alignment (warping) of a large array of domain-specific distances in a uniform manner. While the classic DTW and its prior extensions incorporate the Euclidean Distance, this is the first work to generalize the ubiquitous DTW distance and "extend" its warping capabilities to a rich diversity of popular point-to-point distances. Our GDTW Design Tool enables researchers to incorporate new distances with very little programming effort, thus helping them extend the repository of robust alignment tools. This now represents a valuable resource for the community at large and opens the venue for new research towards improving the state-of-the-art ensemble classifiers. Through extensive evaluation studies on 85 real public domain benchmark datasets, we show empirically that our newly warped distances offer higher classification accuracy than the previously available distances. Lastly, our case study based on heart arrhythmia data contributes to a better understanding of heart conditions and further emphasizes the value of our methodology.**

I. Introduction

A. Motivation and Background

In an era when time series are produced at an unprecedented rate in most application domains, it is crucial to perform various forms of mining that rely on finding relationships based on distances between subsequences both within and across time series. Clearly, there is a broad range of important problems including but not limited to retrieval, classification, and clustering, that rely on such distance-based comparison capabilities. Yet, unfortunately, the selection of a distance[1] deployed to tackle these data mining challenges, if not appropriate for the semantics in a given application domain, can lead to poor or even erroneous results. In this vein, it has been well established that one cannot rely on one single distance function; but instead application domains need a variety of domain-specific distances to solve their given problems [1]. To just name a few examples, compound classification in chemistry selects the most relevant chemical descriptors using *Minkowski* distance [2], *motion detection* applications that index d-dimensional trajectories prefer *Chebyshev* polynomials [3], while *image retrieval* reflect human visual perception tend to utilize *Manhattan* [4], [5] and *Mahalanobis* distances [6].

Unfortunately, all distances mentioned above are only "point-to-point distances". That is, they cannot be used for comparing sequences of different lengths nor those that are not aligned in time. Yet in many real-life scenarios, sequences frequently tend to rather be of *different lengths.* For example, while some patients may be in the intensive care unit for just one afternoon, others may stay there hooked up to various instruments generating medical signals for days or weeks. Even when sequences have the same length, there can be a need to extract subsequences related to certain time periods. For example, a doctor that suspects that a patient experienced an extreme medical condition such as a heart attack may compare his ECG sequence to ECG sequences of longer or shorter lengths of other patients.

Similarly, *misalignment of sequences* meaning that certain patterns may arise at different times in two time series, such as the stock fluctuation of sales volumes by two competitors such as Google and Apple, often occur. They may neither arise at the same time point (i.e., within the first year of their respective public offering) nor within the same time period (while a peak might be reached by Apple more rapidly with the release of a new device, a similar peak might have equally been experienced by Google - but just with a distorted slower effect in time). Or a doctor might want to explore if certain shapes found in the ECG of a patient during a certain medical episode are also found in signals previously recorded for other patients with the same condition - thus aiding in diagnosis.

Performing such data mining tasks clearly requires that the chosen distance can compare sequences that have misalignments, are of different lengths, or both. This critical requirement has been recognized, and thus work to develop so called "elastic" distances that address matching in these more difficult circumstances has been conducted [7], [8], [9], [10]. However, the existing approaches tend to be rigid and specific to one base distance. That is, as we will describe below, there is currently no methodology that enables an application developer to work with the most appropriate point-to-point distance for their domain and particular problem, yet empower this targeted distance to address this robust matching.

[1] From the mathematical point of view, distance measures are defined as a quantitative degree of how far apart two objects are. The smaller the distance between two time series, the more similar they are. In contrast, similarity measures are indicative of the degree of similarity, meaning, the higher the value of the similarity measure the more similar they are considered to be [1]. In general, similarity measures can be expressed in terms of distances. In this paper for simplicity, we will not make the distinction between these two categories and refer to them simply as "distances".

Application developers thus face the major dilemma of having to choose between the most appropriate distance to tackle the problem of supporting flexible sequence matching. There is thus a need to enable analysts to use the distance that is best suited for their specific domain without limiting them to only compare sequences of the same length or without any local shifting. In this work we now provide the first solution to tackle this open problem through a framework that uniformly can extend warping abilities to a wide array of distances, regardless of their mathematical expressions.

B. State-of-the-Art and Its Limitations.

Elastic distances including Dynamic Time Warping [7], [8], the Longest Common Subsequence (LCSS) [9], and Edit Distance with Real Penalty (EDR) [10] enable elastic sequence matching. Particularly, Dynamic Time Warping (DTW) [8], popular for time series data mining allows sequences to be stretched or compressed along the time axis, i.e., a point of one sequence can be matched to one or more points of another sequence. DTW has become increasingly popular due to its expressiveness – being applied to RNA expression data in bioinformatics [11], ECG pattern matching in medicine [12], and aligning biometric data in surveillance systems [13].

Despite its popularity, DTW has been shown to not always be the most appropriate distance for exploring time series because it can produce pathological results through non-intuitive alignments [14]. One reason for this shortcoming stems from the fact that DTW is restricted to using the Euclidean distance as its base distance [8]. This limits its utility for applications that require other distances as we further describe.

As we show in Sec. VIII, countless modifications of the classic DTW have been proposed to either (1) optimize its performance by indexing, caching, and other optimizations [9], [15], [7], [16], [17] or (2) improve the quality of alignments between time series [14], [18], [19]. Closer to our work is [20], which uses sum-based distances in the dynamic programming strategy. However, this method [20] is rather limited, as it only incorporates simple metrics based on sums such as Euclidean or Manhattan – falling short in handling most of the point-to-point distances motivated above.

That is, none of the state-of-art DTW warping methods supports any of the widely popular distances, such as Minkowski or Chebyshev, Sorensen, Cosine, Pearson, Jaccard, and so on. These distances are based on combinations of mathematical operations such as fractions, products, min, and max. We illustrate in this paper that such distances can now all be successfully "warped" using our proposed framework.

C. Our Proposed GDTW Framework

In this work, we overcome the problem of extending warping abilities to diverse point-wise distances by designing a universal alignment tool, called Generalized Dynamic Time Warping. GDTW is flexible enough to use different point-to-point distances in computing the warping path, yet powerful enough to enable time warping. Defining such generalized distance is not enough, it must be complemented by the ability to devise efficient strategies for exploring even large time series datasets. Our work fundamentally changes the classic DTW, while keeping and extending its main purpose, which is robustness to local misalignments in time. We propose a step-by-step methodology that enables a large number of point-wise distances to perform warping and do so efficiently. Its merit stems from (1) the ability to consistently expand the repository of warped distances and (2) the broad range of problems that these newly warped distances can solve, including but not limited to classification, clustering, best match retrieval, and addressing over-warping.

Contributions.

- We introduce GDTW, the first conceptual framework that overcomes the above mentioned open problem and transforms a diversity of point-wise distances into elastic distances by extending warping properties using a uniform approach. (Sec. III-A and III-B).
- We devise a multi-step methodology that empowers analysts to "warp" their desired point-wise distances (Sec. III-C and III-D). This formal transformation based on the GDTW framework and supported by the GDTW Design Tool enables analysts to "warp" new distances without much programming effort (Sec. IV)
- We validate our GDTW framework theoretically and practically by applying it to popular point-wise distances with diverse mathematical characteristics [1], including distances that could not work under the classic DTW algorithm, e.g., Minkowski and Sorensen. This results in a repository of warped distances, which in its own right represents a valuable resource for the community (Sec. V-A, V-B)
- Our extensive experimental study on the 85 datasets from diverse application domains from the benchmark UCR Archive[2] shows the effectiveness of our newly warped distances for a variety of data mining tasks compared to the state-of-the-art DTW (Sec. VI-B, VI-C,VI-D).
- Our study of the Arrhythmia Dataset guided by domain experts, shows the utility of GDTW distances for better interpreting ECG similarity in medical domains. (Sec. VII).

II. Classic Dynamic Time Warping

Suppose we have two time series $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_m)$. To align these sequences using DTW, an $n \times m$ matrix $M(X, Y)$ is constructed, where the $(i, j)^{th}$ element of the matrix is the Euclidean Distance between x_i and y_j, i.e., $w_{i,j} = ED(x_i, y_j)$. Then a *warping* path P is a set of elements that forms a path in the matrix from $(1, 1)$ to (n, m). The t^{th} element of P denoted as $p_t = (i_t, j_t)$ refers to the indices i_t, j_t of (x_{i_t}, y_{j_t}) of this matrix element in the path. Thus a path P is $P = (p_1, p_2, \ldots, p_t, \ldots, p_T)$, where $n \leq T \leq 2n - 1$, $p_1 = (1, 1)$ and $p_T = (n, m)$.

Definition 1: **Warping Path Weight:** Given two time series $X = (x_1, ..., x_n)$ and $Y = (y_1, ..., y_m)$, the **weight of the warping path** P is defined as:

[2] $www.cs.ucr.edu/\ eamonn/time\_series\_data/$

$$w(P) = \sqrt{\sum_{t=1}^{T} w_{i_t,j_t}^2}. \qquad (1)$$

The **DTW distance** then is defined to be the weight of the path with the minimum weight ($\min_P(w(P))$).

A warping path is subject to the following constraints:
1. Boundary condition. $p_1 = (1,1)$ and $p_T = (n,m)$ or the path has to start and end on the opposite corners of the matrix.
2. Continuity condition. The steps in the warping path are restricted to adjacent cells, including diagonally adjacent cells. Using the simplified notations [14], for $p_i = (u,v)$ we have $p_{i-1} = (u',v')$, where $u - u' \leq 1$ and $v - v' \leq 1$.
3. Monotonicity condition. The elements on the path must monotonically progress in one direction, namely $u - u' \geq 0$ and $v - v' \geq 0$ and $(u,v) \neq (u',v')$.
Since there is an exponential number of warping paths satisfying these conditions, finding the minimum weight warping path is prohibitively expensive. Fortunately, the warping path can be efficiently calculated by using dynamic programming [20]. Conceptually, given the matrix M containing pairwise Euclidean distances of all elements in the sequences X and Y, we construct a dynamic programming matrix Γ by filling in the values using the following recursive expression. $\gamma(i,j) = ED^2(x_i,y_j) +$
$min(\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1))$
The current distance $\gamma(i,j)$ in the *cell (i,j)* is the sum of the square of the distance currently found in the cell in the same position in the original matrix M and the minimum of the cumulative distances found in the adjacent cells (diagonal, left and down) in the dynamic programming matrix Γ. Then $DTW^2(X,Y) = \gamma(n,m)$. Further details can be found in [21] and [17].

Fig. 1 illustrates an example of computing the classic DTW warping path for two sequences X and Y as depicted with values in bold font. The leftmost matrix M from the classic DTW algorithm contains the pairwise square ED between the elements of the sequences. The middle matrix Γ showcases the dynamic programming strategy used for computing the path. For example, as indicated on the red dotted arrow, the element 0 in matrix M is summed with the minimum of the three elements (left, down and diagonally-down) in Γ leading to the value 0 in Γ. The gray values indicate that values are calculated "as needed" to find the path efficiently. Lastly, the matrix on the right highlights the resulting path in blue.

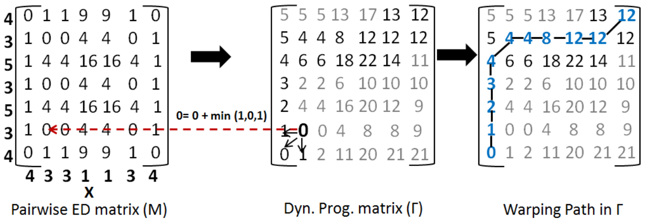


Fig. 1: Computing the warping path with classic DTW

III. Generalized Time Warping

A. Towards a Generalized Distance

We design the GDTW framework to preserve all advantages of DTW, while also supporting the transformation of a wide array of popular point-to-point distances d into their warped counterparts $GDTW_d$. Better yet, unlike previous work, our GDTW approach "empowers" analysts to warp any existing point-to-point distance d of their choice. We offer efficient strategies for computing these warping paths for distances meeting the recursive and symmetry properties described below. Our approach fundamentally changes the algorithm for computing the weight of the warping path by generalizing it to allow the embedding of alternate distances, regardless of their mathematical expressions. This overcomes the limitations of previous approaches as those at best can only "warp" distances based on simple sums.

While our generalized DTW can be applied conceptually to any point-to-point distance, it is important in practice to compute it efficiently. Thus, we change the DP strategy from the classical DTW algorithm and adapt it to work in our generalized context.

B. Fundamentals of GDTW Warping Path

We define the concept of a general warping path and explain how to incorporate new functions in computing it. Given two sequences $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_m)$, with $n \geq m$, we construct an $n \times m$ grid graph G, as a generalization of the matrix Γ from the classic DTW. As shown in Fig. 2, we define a *warping* path P as a sequence of elements that forms a contiguous path from $(1,1)$ to (n,m). By "decoding" this general warping path and extracting the values for x_{i_k} and y_{j_k} at every position on the path, we conceptually construct the following two equal-length vectors: $X_P = (x_{i_1}, x_{i_2}, ..., x_{i_T})$ and $Y_P = (y_{j_1}, y_{j_2}, ..., y_{j_T})$, where some of the x_{i_k} and y_{j_k} are repeated while advancing on the path. Considering an arbitrary distance measure d, the weight of the warping path P is then defined as the distance between X_P and Y_P, which is computed using d. That is, we have $w(P) = d(X_P, Y_P)$.

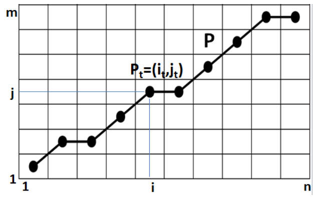


Fig. 2: General Warping Path

Definition 2: The **Generalized Dynamic Time Warping Distance** corresponding to a distance d, denoted by $GDTW_d$, is the weight of the path P with the minimum weight, namely:

$$GDTW_d(X,Y) = \min_P(d(X_P, Y_P)).$$

Theoretically the generalized dynamic time warping distance, as defined in Def. 2, accommodates any distance measure, not just based on sums, but on maximum and minimum, fractions of sums, products, etc. However, as written, it requires us to find all the warping paths first, then determine their weight, and lastly pick the one path with the minimum weight. This is not feasible in practice.

Thus, following the principles of the DTW framework, the key idea is that we must be able to construct the distance function recursively by indicating how to incorporate the n^{th} coordinates in the distance measure based on the previous n-1 coordinates. For this, we introduce a key recursive property that must first be identified and then utilized to define and compute the weight of the warping paths.

Definition 3: The distance measure d in Def. 2 must satisfy the following ***recursive condition***: There exists a 3-variable function $f_d : \mathbf{R}^+ \times \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}^+$ where $\mathbf{R}$ denotes the set of real numbers and $\mathbf{R}^+$ denotes the set of non-negative real numbers with respect to a distance d such that for vectors $X_P = (x_1, x_2, ..., x_n)$ and $Y_P = (y_1, y_2, ..., y_n)$ $(n \geq 2)$, we have:

$$d(X_P, Y_P) = d((x_1, \dots, x_n), (y_1, \dots, y_n)) =$$
$$= f_d \left(d((x_1, \dots, x_{n-1}), (y_1, \dots, y_{n-1})), x_n, y_n \right). \quad (2)$$

The f_d function tells us, given the distance measure on the first $n-1$ coordinates $(x_1, ...x_{n-1}, y_1...y_{n-1})$, how to incorporate the n^{th} coordinates (x_n, y_n). This expression assumes that the distance measure is symmetric in the coordinates. This means swapping the order of two coordinates (x_i, x_j) and (y_i, y_j) respectively in each sequences does not change the distance between sequences.

To illustrate the $GDTW$ with a concrete example, we now re-examine the well-known Euclidean Distance (ED) [22], previously applied in the classic DTW, in our proposed new context. That is, we give the recurrence for ED as per Def. 3.
Euclidean Distance Example: Given the Euclidean distance (ED) between two sequences X and Y defined as

$$d_{ED}(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \quad (3)$$

The **recursive expression** of ED according to Def. 3 is:

$$f_{d_{ED}}(a, x_n, y_n) = \sqrt{a^2 + |x_n - y_n|^2},$$

where a is the value of the function d_{ED} for the first n-1 coordinates. As we show in Sec. V-A, with some mathematical effort all distances in [1] can be expressed in terms of the above recursive property as per Def. 3.

C. Efficient Strategy for Computing the GDTW Warping Path

Let us assume that we have a distance measure d satisfying Eqn (2). Then we propose to *change* the DP strategy designed for the classical DTW (see Sec. II) and adapt it to work with any distance, regardless of its mathematical expression. The classic DP strategy always computes the SUM between the previous value and the min of the three cumulative values of the base distance advancing on the path, therefor it can only work if the base distance is a sum. The new DP strategy directly computes the MIN of the three values incorporating the base distance between the previous value and each of the cumulative values respectively. This new strategy incorporates the core mathematical operation of each base distance, i.e. max for Minkowski or Chebyshev, fraction for Sorensen, etc., while the classic strategy was simply based on sum.

Definition 4: The dynamic programming ***general recursive expression*** for warping a distance d is:

$$\gamma(i, j) = \min \begin{cases} f_d(\gamma(i-1, j-1), x_i, y_j), \\ f_d(\gamma(i-1, j), x_i, y_j), \\ f_d(\gamma(i, j-1), x_i, y_j). \end{cases} \quad (4)$$

with $\gamma(1, 1) = d(x_1, y_1)$.

Definition 5: Using Eqn (4), the "warped" version of a distance d returns a **general dynamic warping distance** defined as:

$$GDTW_d(X, Y) = \gamma(n, m) \quad (5)$$

Given a distance d, we first design the function f_d in Eqn (2). Thereafter, we plug the former into Eqn (4) to derive a DP solution that computes the warped version of distance d.

To illustrate, we apply the above process to our running example of Euclidean distance. Namely, by modifying the general DP expression in Eqn (4), we derive the following **dynamic programming recurrence** for warping the Euclidean distance:

$$\gamma(i, j) = \min \begin{cases} \left(\gamma(i-1, j-1)^2 + |x_i - y_j|^2\right)^{\frac{1}{2}}, \\ \left(\gamma(i-1, j)^2 + |x_i - y_j|^2\right)^{\frac{1}{2}}, \\ \left(\gamma(i, j-1)^2 + |x_i - y_j|^2\right)^{\frac{1}{2}}. \end{cases}$$

We note that the DP recursive expressions derived from (4), when applied to ED, are identical to those in the classic DTW.

D. Proposed GDTW Methodology

In brief, the formal steps of our GDTW methodology for creating a corresponding warping distance $GDTW_d$ for a given distance d are:

1) **Select** a desired distance d as the potential warping candidate. If the distance d satisfies the recursive condition (Sec. III-B), then the following steps provide the strategy to efficiently compute the warping path[3]
2) **Design** the function f_d for the recursive expression in Def. 2 to serve as the weight of the corresponding GDTW warping path (Sec. III-B). This step is crucial to efficiently computing the warping path.
3) **Find** the recursive dynamic programming expression by plugging f_d into Eqn (4) to devise efficiently compute the weight of the path (Sec. III-C).

IV. GDTW Design Tool for New Distances

Once the appropriate expressions have been designed, the analyst can utilize our GDTW Design Tool to implement their own warped distance with ease simply by plugging in parameters. Our design scheme is based on the template pattern which supports the generalization of the core steps

[3] If the distance d is not already in our repository and it does not satisfy this condition, other strategies can be devised to compute the warping path.

of the GDTW methodology. First, the mathematical expressions defining the distances are abstracted into functions. These functions are encapsulated in classes extended from the **DistanceMetric** interface. The **DistanceMetric** interface requires analysts to implement five methods: `getName()`, `getDescription()`, `init()`, `reduce()`, and `norm()`, described below.

- `getName()` and `getDescription()` enable the analyst to name and describe the chosen distance.
- `init`(): *Cache* returns the initial value of the distance, denoted as a *Cache*.
- `reduce`(*Cache* prev, *data_t* Xi, *data_t* Yi): *Cache* combines the accumulated value aand data points from two time series Xi and Yi (data_t represents the data type of each point) according to the recursive expressive of the distance function f_d in Def. 3 to produce a *Cache* value.
- `norm`(*Cache* total, *TimeSeries* X, *TimeSeries* Y): *data_t* calculates the numeric value of a *Cache*.

When extending the `DistanceMetric` interface, analysts need to define the `Cache`for each distance. Its semantics are expressed as the accumulated value in the distance recursive function (the $f_d(x_1..x_{n-1}), (y_1...y_{n-1}))$ defined in Def. 3). The Cache object of distances based on simple mathematical operations (such as ED, Manhattan or Minkowski) only contains one numeric value. However, for more complex distances, such as Sorensen, we must instantiate several accumulated values in the Cache object, as shown later in our examples. Using this interface we construct the point-wise distance d and its warping variant $GDTW_d$ via the `init()`, `reduce()`, and `norm()` methods. We note here that our DP general expression is the same for all distances. The analyst simply has to design f_d to define `reduce()`, while the remaining work for "warping" d is done automatically.

In summary, our modular system enables analysts to construct point-wise distances using recursive expressions and calculate their warped counterparts in a uniform manner with very little programming effort.

V. Building a Community Resource of Warped Distances

A. *Repository of Warped Distances*

We now show how our proposed GDTW framework can be used for warping diverse distances using our three-step methodology described above. We focus on distances collected in the highly cited survey paper [1] due to its large coverage of popular point-to-point distances. In particular, we show well-known distances such as Manhattan, Minkowski and Sorensen, popular in studying similarity of time series. We note that Minkowski (same mathematical expression as the Chebyshev distance) is based on max and *it could not work using the classic DTW algorithm, which is only valid for sum-based distances.* Based on a fraction of sums, Sorensen distance could not work using the classic DTW either.

Our case study achieves three objectives:

(1) It demonstrates the utility of the GDTW methodology for warping in a consistent manner a rich diversity of distances, including those composed of complex expressions including sum, difference, division, square root, max, min, fractions of sums, products, etc.

(2) Our work constructs a valuable "start-up" repository of off-the-shelf warped important metrics ready to use by anyone.

(3) The availability of these examples will help designers of distances in the future find the needed recurrence expressions.

L_p-distances in general, for p=1 and p=2, leading to Manhattan and respectively Euclidean distances: Given the L_p distance between two time series X and Y defined as:

$$d_{L_p}(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}, \tag{6}$$

the **recursive expression** of the L_p distance is stated as:

$$f_{d_{L_p}}(a, x_n, y_n) = (a^p + |x_n - y_n|^p)^{\frac{1}{p}},$$

where a is the total value of the distance measured for the first n-1 coordinates. The p in the L_p can be plugged in accordingly to model specific L_p norms, as mentioned above. The **dynamic programming** recurrence for warping L_p distances is:

$$\gamma(i,j) = \min \begin{cases} (\gamma(i-1,j-1)^p + |x_i - y_j|^p)^{\frac{1}{p}}, \\ (\gamma(i-1,j)^p + |x_i - y_j|^p)^{\frac{1}{p}}, \\ (\gamma(i,j-1)^p + |x_i - y_j|^p)^{\frac{1}{p}}. \end{cases}$$

Euclidean Distance was reviewed earlier in Sec. III-B (and thus not repeated here), so we show now Manhattan distance.

Manhattan Distance: Given the Manhattan distance d_{MD} between two time series X and Y, defined as

$$d_{MD}(X,Y) = \sum_{i=1}^{n} |x_i - y_i|, \tag{7}$$

its **recursive expression** is:

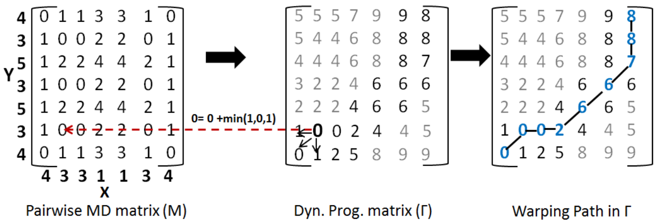
$$f_{d_{MD}}(a, x_n, y_n) = (a + |x_n - y_n|)$$

and the **recursive dynamic programming** is:

$$\gamma(i,j) = \min \begin{cases} (\gamma(i-1,j-1) + |x_i - y_j|), \\ (\gamma(i-1,j) + |x_i - y_j|), \\ (\gamma(i,j-1) + |x_i - y_j|). \end{cases}$$

Similarly to Fig. 1, we give an example for computing the warping path for the same pair of sequences using $GDTW_{MD}$ in Fig. 3. We note here that the resulting path differs than the path found by the classic DTW.

Minkowski Distance: Given the Minkowski distance d_{Mink}



$$\begin{array}{cc|ccccccc} & 4 & 0 & 1 & 1 & 3 & 3 & 1 & 0 \\ & 3 & 1 & 0 & 0 & 2 & 2 & 0 & 1 \\ & 5 & 1 & 2 & 2 & 4 & 4 & 2 & 1 \\ \mathrm{Y} & 3 & 1 & 0 & 0 & 2 & 2 & 0 & 1 \\ & 5 & 1 & 2 & 2 & 4 & 4 & 2 & 1 \\ & 3 & 1 & 0 & 0 & 2 & 2 & 0 & 1 \\ & 4 & 0 & 1 & 1 & 3 & 3 & 1 & 0 \\ \hline & & 4 & 3 & 3 & 1 & 1 & 3 & 4 \\ & & & & & \mathrm{X} & & & \end{array}$$

$$\begin{bmatrix} 5 & 5 & 5 & 7 & 9 & 9 & 8 \\ 5 & 4 & 4 & 6 & 8 & 8 & 8 \\ 4 & 4 & 4 & 6 & 8 & 8 & 7 \\ 3 & 2 & 2 & 4 & 6 & 6 & 6 \\ 2 & 2 & 2 & 4 & 6 & 6 & 5 \\ 1 & 0 & 0 & 2 & 4 & 4 & 5 \\ 0 & 1 & 2 & 5 & 8 & 9 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & 5 & 7 & 9 & 9 & 8 \\ 5 & 4 & 4 & 6 & 8 & 8 & 8 \\ 4 & 4 & 4 & 6 & 8 & 8 & 7 \\ 3 & 2 & 2 & 4 & 6 & 6 & 6 \\ 2 & 2 & 2 & 4 & 6 & 6 & 5 \\ 1 & 0 & 0 & 2 & 4 & 4 & 5 \\ 0 & 1 & 2 & 5 & 8 & 9 & 9 \end{bmatrix}$$

Fig. 3: Computing the warping path with $GDTW_{MD}$.

between two time series X and Y defined as

$$d_{Mink}(X,Y) = \max_{i=1}^{n} |x_i - y_i|, \tag{8}$$

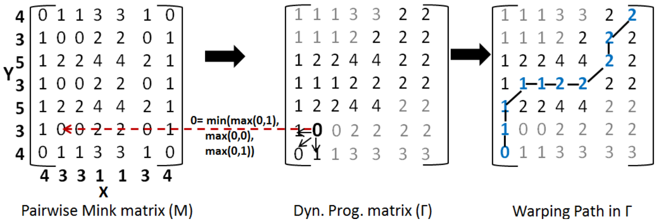
its **recursive expression** is:

$$f_{d_{Mink}}(a, x_n, y_n) = \max(a, |x_n - y_n|). \tag{9}$$

with the **dynamic programming recursive** expressions:

$$\gamma(i,j) = \min \begin{cases} \max(\gamma(i-1,j-1), |x_i - y_j|), \\ \max(\gamma(i-1,j), |x_i - y_j|), \\ \max(\gamma(i,j-1), |x_i - y_j|). \end{cases}$$

Similarly to Fig. 1, we give an example for computing the warping path for the same pair of sequences using $GDTW_{Mink}$ in Fig. 4. We note two differences: the resulting path is different than all previous paths, and the DP strategy unlike DTW uses a completely different expression, based on max not sum. The warped Minkwoski distance could not work using the DP expression of the classic DTW.



$$\begin{array}{c|ccccccc} 4 & 0 & 1 & 1 & 3 & 3 & 1 & 0 \\ 3 & 1 & 0 & 0 & 2 & 2 & 0 & 1 \\ 5 & 1 & 2 & 2 & 4 & 4 & 2 & 1 \\ 3 & 1 & 0 & 0 & 2 & 2 & 0 & 1 \\ 5 & 1 & 2 & 2 & 4 & 4 & 2 & 1 \\ 3 & 1 & 0 & 0 & 2 & 2 & 0 & 1 \\ 4 & 0 & 1 & 1 & 3 & 3 & 1 & 0 \\ \hline & 4 & 3 & 3 & 1 & 1 & 3 & 4 \end{array} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 4 & 4 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 4 & 4 & 2 & 2 \\ 1 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 3 & 3 & 3 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 4 & 4 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 4 & 4 & 2 & 2 \\ 1 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 3 & 3 & 3 & 3 \end{bmatrix}$$

Fig. 4: Computing the warping path with $GDTW_{Mink}$.

The **Sorensen** distance, used in ecology [23], is another example that could not be accommodated by the classic DTW because of its complex form (a fraction of sums).
Given the Sorensen distance d_{sor} between two sequences X and Y defined as:

$$d_{Sor}(X,Y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} |x_i + y_i|} \tag{10}$$

Its **recursive expression** is:

$$f_{d_{Sor}}(\frac{a}{b}, x_n, y_n) = \frac{a + |x_n - y_n|}{b + |x_n + y_n|} = \frac{a'}{b'}. \tag{11}$$

where a and b denote the total value of the differences and respectively the sums of the first n-1 coordinates.
The **dynamic programming recursive expression** is:

$$\gamma(i,j) = \min \begin{cases} \frac{a_1 + |x_i - y_j|}{b_1 + |x_i + y_j|}, \\ \frac{a_2 + |x_i - y_j|}{b_2 + |x_i + y_j|}, \\ \frac{a_3 + |x_i - y_j|}{b_3 + |x_i + y_j|}, \end{cases}$$

where $\gamma(i-1,j-1) = \frac{a_1}{b_1}$, $\gamma(i-1,j) = \frac{a_2}{b_2}$, $\gamma(i,j-1) = \frac{a_3}{b_3}$.

As shown above, it is clear that all distances in [1] can be warped by our GDTW methodology. These distances which are essential for classification, clustering and retrieval problems, are categorized into eight families in [1]. We emphasize that the key to warping these distances is to formulate a recursive expression to fit Def. 3 and then embedding that into Eqn (4). Due to space constraints, we offer other examples, namely warping the **Cosine distance** in our extended repository of warped distances [24]. The **Cosine** distance, which measures the angles between two vectors, corresponds to the normalized Inner Product and also has a complex form that could not work with the DP of the classic DTW. Many other popular distances such as Jaccard, Dice and Pearson, based on similar arithmetic expressions can be warped with using our methodology.

B. GDTW Design Tool for Incorporating New Distances.

We showcase here the use of our Designer Tool on two examples: $GDTW_{Mink}$ (warped Minkowski or Chebyshev), and the more complex warped Sorensen distance. These distances are not based on sums and could not work under any of the previous approaches to generalize DTW. We follow up on our previous examples of "theoretically" warping these distances by explaining how our interface assists in their implementation without much programming effort. We omit the examples of classic DTW and $GDTW_{MD}$, as their implementation is very similar to $GDTW_{Mink}$.

Implementing $GDTW_{Mink}$**.** We show here the implementation of `DistanceMetric` with parameters omitted for brevity:

```
init() { -INF; }
reduce() { max(prev, abs(Xi - Yi)); }
norm() { total; }
```

ChebyshevCache contains only one value x because only one cumulative value is needed in the recursive expression of the distance in Eqn (9).

Implementing $GDTW_{Sor}$. For the Sorensen distance shown in Eqn (10) whose mathematical formula is more complicated , we now fully utilize the capabilities of the Cache. On the right side of this equation there are two separate terms, the cumulative sum of the differences in coordinates and the cumulative sum of the sums of coordinates respectively. The Cache thus must store two values and the function `norm` computes the final distance according to the recursive expression in Eqn (11)).

```
init() { [0,0]; }
reduce()
 { [prev[0] + |Xi-Yi|, prev[1] + |Xi+Yi|]; }
norm() { total[0] / total[1]; }
```

where total[0] and total[1] refer to the distinct terms on the right side of the equation.

In summary, using our extensible interface analysts can construct point-wise distances and their warped counterparts by using our three methods previously described, in a uniform manner with very low programming effort.

VI. Experimental Evaluation

A. Experimental Methodology

Our GDTW framework can warp a plethora of distances, but here we aim to show that all warped distances are valuable and can solve specific data mining problems in diverse application domains. Thus, we implement a select subset of new warped distances namely, $GDTW_{Mink}$ (warped Minkowski or Chebyshev), $GDTW_{ED}$ (DTW), and $GDTW_{MD}$ (warped Manhattan). The reason for choosing these is three-fold: (1) they are well known to the research community, (2) we documented in the introduction that their point-wise versions are valuable in diverse domains, yet cannot perform flexible sequence matchings, (3) $GDTW_{Mink}$ was chosen because it is based on max and it cannot work using the classic algorithm for DTW, but it now works under our GDTW framework. Additional experimental results[4] are found at [24].

Data Sets. We use the largest public collection focussed on time-series datasets in particular that we are aware of, the University of CA, Riverside Archive[5] containing 85 benchmark datasets from various domains.

Three Classes of Evaluation:

Experiment 1: Time Series Classification. We evaluate the effectiveness of our newly warped distances for time series classification. For this we apply each distance over the training and test sets of the 85 datasets in the UCR archive, using them as (parameter-free) 1-NN classifiers. We compute the classification accuracy, i.e., number of correctly classified instances over all instances, and the error rate in performing 1-NN classification. Because the 1-NN classifier is deterministic, we only perform this computation once.

Experiment 2: Best Match Retrieval and Clustering. We first find the best match (or the nearest neighbor) for a given sample query sequence first by using a "point-to-point" distance and then by using its "warped" counterpart. We then compare these matches. Our experiment aims to show that: (1) the warped distances tend to return different results compared to their point-wise versions, as expected when the sequences are not aligned in time, and (2) diverse warped distances return often different results, each providing insights that would be missed by the other warped distances. In addition, we demonstrate the impact of using these new distances on an average linkage hierarchical clustering problem.

Experiment 3: Evaluation of Warping Characteristics. Similarly to the well-known Derivative Dynamic Time Warping method [14], which studies the "over-warping" produced by the classic DTW, we compute the *amount of warpings* produced by our GDTW variants for pairs of sequences as:

$$W = (l - average(m, n))/average(m, n) \qquad (12)$$

where $0 \leq W \leq 1$ and m, n are the lengths of the compared sequences. W=0 if the algorithm does not find a warping between two sequences. W increases to a maximum value of 1 as the warping "discovered" by the algorithm increases. Analysts interested in finding similar sequences with fewer warpings can utilize these findings. Similarly to [14], we also measure the *sensitivity of our warped distances to local distortion* by introducing distortions in a controlled fashion into pairs of synthetic sequences.

B. Classification Using 85 Diverse Time Series Benchmarks

Time series classification [10][25][26] is an important problem where a distance is used as a subroutine in the K-Nearest Neighbor (K-NN) algorithm. This simple algorithm has been shown to be surprisingly competitive, by consistently outperforming rival methods such as decision trees, neural networks, Bayesian networks, and Hidden Markov Models [26], [27]. Moreover, time series classification has thus far been one of the few tasks to resist significant progress from "deep learning" [28]. Given this, the choice of "which" distance to use is important. Literally dozens of distances have been proposed (see [9], [28] and the references therein). However, an extensive recent empirical comparison (performing 36 million experiments) has confirmed the excellent performance of classic DTW-based 1-NN (in our case $GDTW_{ED}$), which is only beaten by Ensemble Classifiers [28].

We now evaluate if other warped distances such as $GDTW_{MD}$ or $GDTW_{Mink}$ are even more effective for classifying time series. To test this research question, we performed classification experiments on all 85 datasets from the UCR Archive. The train/test splits were identical. We fixed the warping window to 100% for all experiments, thus any differences can be attributed solely to the effect of changing the base distance. The raw results and the details of the experiments including pairwise comparisons using error-rate binary plots are archived at [24].

For brevity, we present here a compact visual summary displaying a comparison between the results obtained using the classic DTW (our $GDTW_{ED}$), $GDTW_{MD}$ and $GDTW_{Mink}$. Pairwise comparisons of distance measures are often presented as 2D-scatter-plots [25]. We compare three algorithms and present the results as a trivariate plot in Fig. 5. In this plot, the locations of the points do not correspond to the actual error rate, but are *proportional* to them. This ternary

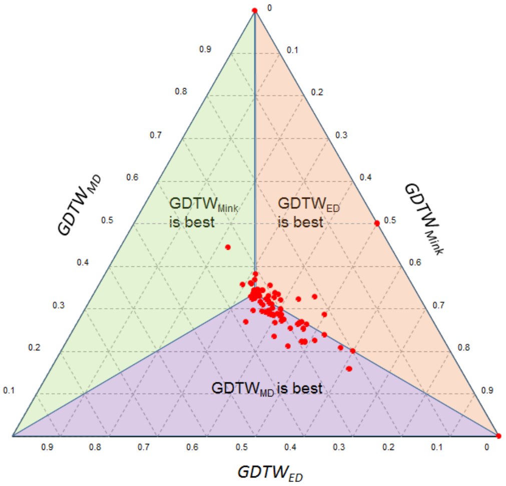


Fig. 5: A trivariate plot comparing $GDTW_{ED}$, $GDTW_{MD}$, and $GDTW_{Mink}$. For points close to the center of the figure, all 3 methods produce similar error rates. For points away from the center, at least one method performs poorly.

[4] Our code will be publicly made available upon acceptance of the paper.

[5] $www.cs.ucr.edu/\ eamonn/time\_series\_data/$

plot shows the error rate distribution for the three distances and marks the areas where each individual distance performs better than the others. The red dots in the light purple area (bottom) indicate the wins for $GDTW_{MD}$ (in 56 cases), while the ones in the pink area (right) show where $GDTW_{ED}$ is better (in 44 cases) and the ones in the green area (left) correspond to $GDTW_{Mink}$ performing better (in 15 cases). The results are surprisingly diverse, with 44 "wins" for $GDTW_{ED}$, 56 "wins" for $GDTW_{MD}$, and 15 "wins" for $GDTW_{Mink}$. There are fewer dots in the green area, indicating that $GDTW_{Mink}$ generally performed poorer than the other two distances. The error rates for $GDTW_{MD}$ and $GDTW_{ED}$ are fairly close, as shown by the high concentration of points in the center.

In conclusion we succeeded at improving on the standard benchmark of 1-NN DTW, simply just by switching to $GDTW_{MD}$. This suggests that usage of our framework to forge new time-warping distances has the potential to lead to other substantial improvements on state-of-the-art time series classification [24].

C. Best Match Retrieval and Clustering Experiments

For each dataset in a subset of datasets from the UCR archive, we randomly select a subsequence and "promote" it to be a query, similarly to [29]. Then we find the best match for this query sample by using three point-wise distances (ED, MD, $Mink$) and their warped counterparts ($GDTW_{ED}$, $GDTW_{MD}$, $GDTW_{Mink}$). We repeat this experiment for 10 random sample sequences. Due to the space constraints we show the details in the additional experimental results [24], while here we offer only a summary analysis. The results vary significantly when using different point-to-point distances and their warped versions, as expected. The point-wise distances can be *at most* as good as their warped versions, especially for sequences that are aligned in time. The $GDTW$ variants often return different results, each providing best matches that would otherwise be missed. Only in 10% of the scenarios did all three variants return the same best match.

TABLE I: Percentage scenarios where pairs of distances return the same best match for a sample sequence in the ECG dataset

Pair of distances	*Percent scenarios*
ED and DTW	20
MD and $GDTW_{MD}$	20
$Mink$ and $GDTW_{Mink}$	0
DTW and $GDTW_{Mink}$	10
$GDTW_{MD}$ and $GDTW_{Mink}$	20
DTW and $GDTW_{MD}$	50

Lastly, we offer an experiment that reveals new insights uncovered by our newly warped distances using **hierarchical clustering**. We select five sequences from the ECG dataset. Two of them are randomly chosen samples from class 1 (red/bold), while the other three are randomly selected from class -1 (blue/none-bold). We cluster these sequences using our three $GDTW$ variants. We repeat the experiment five times. As the dendogram in Fig. 6 shows, $GDTW_{MD}$ indeed clusters together sequences from the same class (red) from start, while $GDTW_{ED}$ and $GDTW_{Mink}$ do not. This reaffirms that $GDTW_{MD}$ has a higher accuracy than DTW.

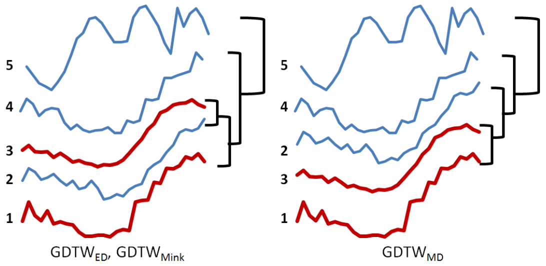


Fig. 6: Average linkage hierarchical clustering

In summary, our newly warped distances can reveal best matches missed by the classic DTW and potentially improve clustering quality.

D. Evaluating Warping Characteristics

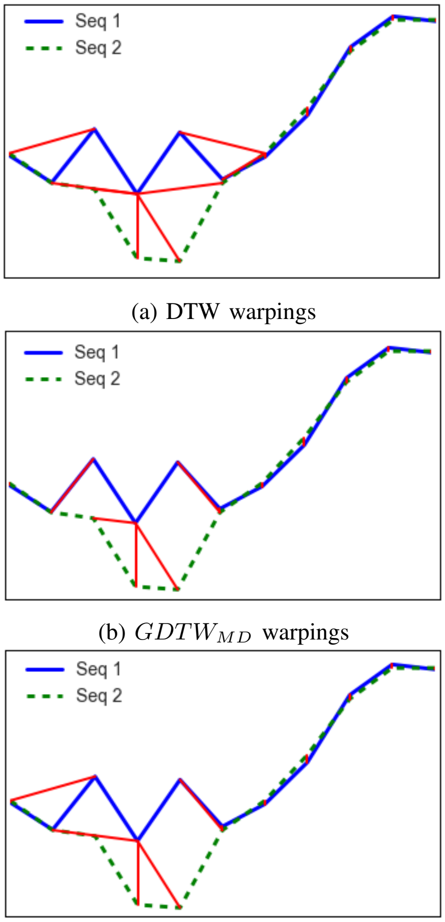
Evaluating Cardinality of Warpings. We randomly select 20 pairs of sequences 10 pairs of the same and 10 pairs of different lengths. We find the matching elements of the sequences by using $GDTW_{ED}$, $GDTW_{MD}$, and $GDTW_{Mink}$. We compute the average amount of warpings for each $GDTW$ variant for various datasets (See Table II). Summarizing our findings (full details and visual displays are available along with additional experimental results) [24], $GDTW_{Mink}$ and $GDTW_{MD}$ discover fewer warpings than the ones created by $GDTW_{ED}$. This indicates that these warped distances avoid singularities or "over-warping" incurred by classic DTW.

TABLE II: Average warpings for sequences of any length

Dataset	*DTW*	$GDTW_{MD}$	$GDTW_{Mink}$
ItalyPower	0.4	0.3	0.23
ECG	0.43	0.34	0.17
Wafer	0.49	0.38	0.33
Face	0.32	0.28	0.11

Evaluating Sensitivity to Local Distortions. We test the ability to find the correct warpings by using different $GDTW$ variants for pairs of sequences for which the "warping" is known. Similarly to [14], we distort the y-axis by adding or subtracting a distortion (Gaussian bump) on randomly chosen anchor points of the sequences. As shown in Fig. 7, we find the warping paths for these modified sequences using our three warped distances. Each distance leads to a warping path different than the other two distances. DTW "over-warps" the distorted sequences, while $GDTW_{MD}$ and $GDTW_{Mink}$ find a shorter warping path. The performance for DTW and $GDTW_{MD}$ tends to degrade even for small distortions of the y-axis, while $GDTW_{Mink}$ maintains a better warping performance.

In summary, our experimental results confirm the utility of our newly warped distances (in particular, $GDTW_{Mink}$) in avoiding singularities or "over-warping" problems incurred by the classic DTW. They also are less sensible to distortion. In other words, they promise to be useful in practice.



(a) DTW warpings

(b) $GDTW_{MD}$ warpings

(c) $GDTW_{Mink}$ warpings

Fig. 7: **Warpings for distorted sequences**. (a), (b), and (c) show the warpings using respectively the classic DTW, $GDTW_{MD}$ and $GDTW_{Mink}$

VII. Studying Heart Arrhythmia using GDTW

In collaboration with expert cardiologists, we explore the MIT-BIH Arrhythmia Database, created by Beth Israel Deaconess Medical Center and MIT, which supports research into arrhythmia analysis and related subjects. The MIT-BIH Arrhythmia Database [30], [31] contains 48 half-hour excerpts of two-channel ambulatory ECG recordings obtained from 47 subjects. 23 recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital. To address the imbalance in the data, the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that otherwise would not be well-represented in a small random sample.

Medical staff studies similarity of ECGs for diagnosing arrhythmia which refers to changes of the normal sequence of electrical impulses. Electrical impulses may cause the heart to beat too fast, too slowly, or erratically. When the heart does not pump blood effectively, the lungs, brain and other organs cannot work properly and may shut down or be damaged.

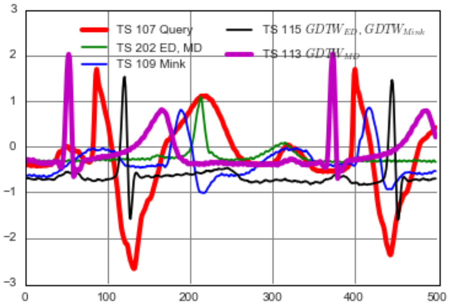


Fig. 8: Case study best match sequences

We use our newly warped distances to explore this database and find the best match for a given ECG shape. For this experiment, we first chose the sample heart rate shape of the record labeled 107. This male patient (age 63) has a complete heart block condition in which the impulse generated in the sinoatrial node in the atrium of the heart does not propagate to the ventricles. We randomly selected 20 records from the dataset, including that of the patient with record 107 and asked our cardiologist collaborators to find the best match for this ECG shape. The cardiologists identified the ECG for the patient with record number 113, as having the closest heart rate, meaning average heart rate in beats per minute. Independent of their findings, we retrieved the best match for this sequence by using ED, MD, M_{ink}, $GDTW_{ED}$, $GDTWMD$, and $GDTW_{Mink}$. As seen in Fig. 8, different distances returned different best matches. Sequence 113 was returned as best match by $GDTW_{MD}$. All other distances returned different matches. In this case $GDTW_{MD}$ found the same best match as the domain expert. We repeated this experiment five times using different sample sequences and arrived to the same conclusion based on comparing the answers provided by the cardiologists with the ones retrieved by our system. Visual samples are archived along with our additional experimental results [24].

In short, in this ECG Arrhythmia use case, the newly warped $GDTW_{MD}$ consistently finds the best match confirmed by experts. A match missed by the classic DTW.

VIII. Related Work

DTW has been popular in a large range of application domains including medicine [12], and spoken word recognition [20]. Despite its ability to compare mis-aligned time series, it can produce pathological results [14]. Many modifications of DTW have been proposed to improve the performance, produce better alignments, and to handle "singularities". Most constrain the warpings, while continuing to use the Euclidean Distance as base distance, as we discuss below.

For **performance improvement**, Keogh and Pazzani [32] introduced PDTW, which applies the classic DTW algorithm to a higher level abstraction of the data (Piece Aggregate Approximation), outperforming DTW with little loss of accuracy. Indexing methods [7], [16], [17] further improved response time in retrieving similar sequences. Some works aim to address singularities and **produce superior alignments,** by modifying the way the warping path is computed. Unfortunately they still keep ED as intrinsic base distance. For example, [33] introduced a variable penalty whenever a non-diagonal step is taken. This reduces the number of non-diagonal moves and improves the alignment of chromatogram signals. WDTW [19] penalizes points with a higher phase difference between a reference point and a testing point to prevent minimum distance distortion caused by outliers. Closer conceptually to our idea, [20] replaces ED with another base distance. However, it is restricted to only incorporating base distances that are based on sums, such as Euclidean or Manhattan. Our work is now a major step forward, as our method is general enough to "warp" any distance, regardless of its mathematical expression. Symmetric DTW [21] addresses slope weighting. In computer graphics, Iterative Motion Warping [18] finds a spatial temporal warping between two instances of motion captured data. In contrast, Derivative DTW [14] produces superior alignments by replacing ED with the square of the difference of the derivatives of the sequences in computing the warping path, thus gaining more information about the shape. Closer conceptually to our framework, this replaces ED with a different base distance. Unlike our work, they stop at using only one derivative based distance, while our methodology incorporates a wide array of base distances.

IX. Conclusion

Our proposed general time warping framework offers the first universal solution for transforming point-wise distances into robust alignment tools, capable of performing flexible sequence matching. Our three-step methodology insures that warped distances can be designed in a consistent manner, establishing a valuable resource for the entire research community. This repository can now include warped versions of popular distances with complex mathematical expressions. While our paper demonstrates improved accuracy of time series classification [10], [25], it opens the avenue for important new research. Studies leveraging variants of $GDTW$ could now further contribute to solving a broad range of problems including but not limited to classification, clustering, and addressing singularities, etc.

References

[1] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, 2007.

[2] E. Karakoc, A. Cherkasov, and S. Sahinalp, "Distance based algorithms for small biomolecule classification and structural similarity search," *Bioinformatics*, 2006.

[3] J. Mason and D. Handscomb, *Chebyshev polynomials*. CRC Press, 2002.

[4] M. Stricker and M. Orengo, "Similarity of color images," in *Symposium on Electronic Imaging: Science and Technology*. International Society for Optics and Photonics, 1995.

[5] M. Swain and D. Ballard, "Color indexing," *International journal of computer vision*, 1991.

[6] J. Smith, "Integrated spatial and feature image systems: Retrieval, analysis and compression," Ph.D. dissertation, Columbia University, 1997.

[7] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, 2005.

[8] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*. Seattle, WA, 1994.

[9] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases*. ACM, 1994, vol. 23, no. 2.

[10] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth international conference on VLDB-Volume 30*. VLDB, 2004.

[11] J. Aach and G. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, 2001.

[12] E. Caiani, A. Porta *et al.*, "Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume," in *Computers in Cardiology 1998*. IEEE, 1998.

[13] D. Gavrila, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, 1999.

[14] E. Keogh and M. Pazzani, "Derivative dynamic time warping." SIAM, 2001.

[15] B. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary lp norms." VLDB, 2000.

[16] M. Vlachos, M. Hadjieleftheriou *et al.*, "Indexing multi-dimensional time-series with support for multiple distance measures," in *Proceedings of the ninth ACM SIGKDD*. ACM, 2003.

[17] T. Rakthanmanon, Campana *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *18th ACM SIGKDD*, 2012.

[18] E. Hsu, K. Pulli, and J. Popovici, "Style translation for human motion," in *ACM Transactions on Graphics (TOG)*. ACM, 2005.

[19] Y. Jeong, M. Jeong, and O. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, 2011.

[20] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 1978.

[21] J. Kruskall and M. Liberman, "The symmetric time warping algorithm: From continuous to discrete. time warps, string edits and macromolecules," 1983.

[22] R. Agrawal, C. Faloutsos, and A. Swami, *Efficient similarity search in sequence databases*. Springer, 1993.

[23] J. Looman and J. Campbell, "Adaptation of sorensen's k (1948) for estimating unit affinities in prairie vegetation," *Ecology*, 1960.

[24] "Additional material," https://github.com/GDTW-Material/GDTWMaterial, 2016, [Online].

[25] Ding, Trajcevski *et al.*, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, 2008.

[26] A. Bagnall, A. Bostrom *et al.*, "The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version," *arXiv preprint arXiv:1602.01711*, 2016.

[27] X. Xi, E. Keogh *et al.*, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 1033–1040.

[28] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.

[29] A. Fu, E. Keogh *et al.*, "Scaling and time warping in time series querying," *The VLDB Journal*, 2008.

[30] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[31] A. Goldberger, L. Amaral, Glass *et al.*, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[32] E. Keogh and M. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of the sixth ACM SIGKDD*. ACM, 2000.

[33] D. Clifford, G. Stone *et al.*, "Alignment using variable penalty dynamic time warping," *Analytical chemistry*, 2009.